Support Vector Machine Approach for Model-Plant Mismatch Detection

Qiugang Lu[a], Michael G. Forbes[b], Philip D. Loewen[c],
Johan U. Backström[b], Guy A. Dumont[d], R. Bhushan Gopaluni[a,1]

[a] *Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4*
[b] *Honeywell Process Solutions, North Vancouver, BC, Canada, V7J 3S4*
[c] *Department of Mathematics, University of British Columbia, Vancouver, BC, Canada, V6T 1Z3*
[d] *Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z3*

Abstract

We develop a model-plant mismatch (MPM) detection strategy based on a novel closed-loop identification approach and one-class support vector machine (SVM) learning technique. With this scheme we can monitor MPM and noise model change separately, thus separating the MPM from noise model changes. Another advantage of this approach is that it is applicable to routine operating data that may lack any external excitation signals. Theoretical analysis of the proposed closed-loop identification is provided in this paper, showing that it can give a consistent parameter estimate for the process model even in the case where *a priori* knowledge about the true noise model structure is not available. A set of normal operation data with satisfactory performance is collected as the training data. We build SVM models based on process and noise model estimates from training data to predict the occurrence of MPM in the test data. The proposed technique can be applied to both single-input-single-output (SISO) and multi-input-multi-output (MIMO) systems. Two examples from paper machine control are provided to verify the effectiveness of the proposed MPM detection framework.

Keywords: Monitoring and performance assessment; support vector machine; model-plant mismatch; closed-loop identification; paper machine control

1. Introduction

Process monitoring provides a quantitative way of measuring the performance of various model-based controllers (e.g., model-predictive control, MPC) in the industry. For typical control loops, their performance may deteriorate for various reasons such as model-plant mismatch (poor quality of process models), changes in disturbance characteristics, improper tuning of controllers and so on [1]. Among these factors the model-plant mismatch (MPM) is vital since significant MPM often leads to suboptimal control decisions, production loss, or even closed-loop instability. System re-identification will be required in the presence of severe MPM to produce an updated model and deploy it to controllers to account for the changes of process characteristics. As system identification is usually costly and time-consuming, identification experiment is not triggered for other causes of performance loss than the MPM. Therefore, there is an increasing demand for automated and

[1] Corresponding author. Telephone: +1-604-827-5668; email: bhushan.gopaluni@ubc.ca

highly-reliable schemes to directly detect the presence of significant mismatches and distinguish MPM from the other causes with minimal interruptions of routine operations [2].

MPM often arises from the susceptibility of industrial processes to changes of operating conditions over time, such as a drift to the chemical process caused by catalyst deactivation [3]. Traditionally, the MPM detection problem is considered as a component in the diagnostic part of the process monitoring hierarchy [4]. Thus there exist excessive links between MPM detection and performance monitoring. Research interest towards this direction has focused primarily on evaluation of MPC performance with various benchmarks such as minimum variance control (MVC) benchmark [5], linear quadratic Gaussian (LQG) curve, and MPC-specific benchmarks. However, these performance assessment techniques are not specifically established for the detection of MPM, and thus they are not able to separate MPM from the other causes of performance degradation. On the other hand, attentions on developing effective methods to directly monitor the occurrence of MPM have grown significantly over the last few decades. A correlation-based method is proposed in [6] which relies on examining the significance of cross-correlations between setpoint and model error to separate modeling errors from other causes. Multivariate chi-squared test on the output and prediction error is proposed in [7] to diagnose the presence of MPM. A hypothesis test method is proposed in [8] to distinguish the effects of changes in process and disturbance on the control performance drop. As pointed on in [4], during this early stage of research on MPM detection, most efforts concentrated on the performance monitoring and little endeavors are reported to separate the MPM from other causes. Another category of methods on MPM detection is to construct statistics to detect the abrupt changes in process parameters. A variety of statistical tools such as time series modeling, principal component analysis (PCA), generalized likelihood ratio test and several new statistics are presented in [9, 10, 11, 12]. The issue of deciding when to perform the model re-identification, based on PCA and Akaike information criterion, is investigated in [13]. For that method, periodic injection of external excitation signals is required to perform simple tests to detect the model degradation. More recently, attempts on directly identifying the presence or even the model of MPM have become the mainstream in the literature. A method based on partial correlations is illustrated in [14] to address the correlations among manipulated variables (MVs). The correlation between each MV and each model residual, after decorrelating the MVs, is employed to examine the MPM. In [15], an important method is proposed to discriminate the noise covariance mismatch and process model mismatches. Both mismatches affect the Kalman filter design, leading to non-white innovations, and they differ in the order of innovation sequences. The impacts of MPM on the sensitivity function is thoroughly explored in [16] and a resultant performance index is put forward for the diagnosis.

More recently, interests on MPM detection push the frontiers into realistic industrial contexts with closed-loop routine operating data. Several approaches are developed based on model residual analysis [17], output autocovariance function [3], plant-model ratio [18] and the nominal sensitivity function [18]. Despite these achievements, the MPM detection problem is not yet completely solved. The remaining key issues on this topic can be summarized into two categories: the separability between MPM and noise model change [17, 2], and the usage of external excitations [16, 14]. First, most MPCs perform predictions based solely on a process model or together with a presumed simple noise model, e.g., a step or a random-walk disturbance. Despite that changes in true noise model may cause drifts in the variance of process variables, the resultant degradation in control performance shall not be attributed to the MPC. In other words, operators in the industry are more concerned with the quality drop in the underlying process model instead of that in the noise model. An ideal MPM detection approach shall not mistake the changes in the noise model as MPM and be robust to such factors [2]. Most approaches built on variance-based metrics are susceptible to this issue, such as the minimum-variance benchmark [5]. Second, most present approaches to directly identify the MPM depend on certain external excitations, such as dither signals or setpoint changes. Nevertheless, external excitations bring additional perturbations to the system and inevitably cause profit loss to the industry. Although it is possible to identify the MPM during the setpoint change (e.g. grade changes for paper machines), we prefer to monitor MPM during the routine operation stage, given that most of the time the system operates at steady-state that may lack any type of external excitations.

In this work, we propose a novel MPM detection approach that addresses the above two challenges: *separating the MPM from noise model changes and detecting MPM passively without intrusively introducing external excitations to the system.* Our idea is inspired by the historical data based benchmarks used in controller performance monitoring [19]. In those methods, to assess the control performance, process variable metrics under actual data are compared with those under a set of historical data which were collected during a period with satisfactory operation. Similarly, in our method, we partition routine operating data into a "training" stage (that we believe is generated without MPM) and a "testing" stage. The training data serves as a benchmark, against which we evaluate the presence of MPM in the test data. Specifically, we propose a novel method, based on closed-loop identification and one-class support vector machine (SVM) classification, that can monitor MPM and noise change independently and thus can directly separate MPM from noise model change. Moreover, our method is applicable even when external excitations do not exist. A preliminary version of this method has been published in [20] and in this work, we elaborate the theoretical verifications and also extend the results in [20] to MIMO systems. Note that our method is applicable to any model-based controller, although our simulation examples are exclusively from paper-making processes with MPC.

This paper is outlined as follows. In Section 2, we elucidate the framework of the proposed MPM detection approach. Section 3 is devoted to developing a new closed-loop identification method that can provide consistent parameter estimates for the plant model. In Section 4, we elaborate detailed procedures on training an SVM model based on the training data and implementing it in the test data to examine the MPM and noise model change. A single-input-single-output (SISO) example and a multi-input-multi-output (MIMO) example from machine-direction (MD) processes of paper machines is provided to verify the proposed approach in Section 5. We comment on the possible limitations of the proposed approach in practical appliations in Section 6, followed by conclusions in Section 7.

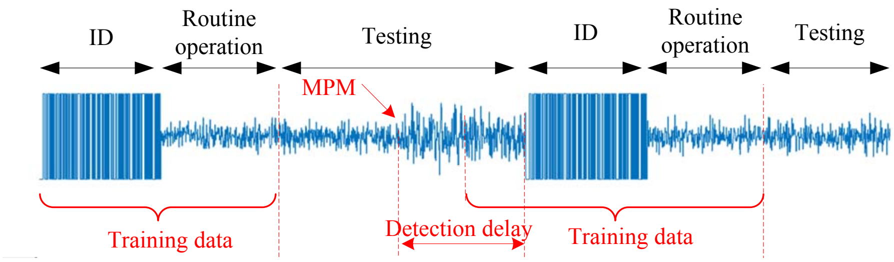

Figure 1: Illustration of the training and test data.

2. The MPM Detection Idea

We mention first that the noise model change detection follows the same course as the MPM detection in our method. So in subsequent sections, our attention is mainly on introducing the MPM detection approach. The proposed method is based on a novel closed-loop identification algorithm which is capable of providing unbiased estimates for process models with routine operating data albeit with large variance. The inevitable variance (which may be large due to the low signal-to-noise ratio) associated with process model estimates impedes us from directly comparing our results with nominal models to identify the mismatch. In other words, discrepancies between the model estimates and the true plant can not always be attributed to the MPM and they may simply be an artifact of the variance in the estimates. Therefore, directly comparing the estimated parameters with those of the nominal model used by the controller may cause misleading detection results. We have to form a reasonable uncertainty bound around estimated process models due to the variance of parameter estimates. Models outside this uncertainty range are regarded as mismatched models. Such an uncertainty range can be naturally captured by using the SVM technique. Note that to synthesize all possible mismatch situations (e.g., gain mismatch and time constant mismatch) with an overall metric, we would represent process model estimates in finite impulse response (FIR) forms. With a high-order FIR model we can capture the process dynamics of any order. Now the problem of detecting MPM can be reduced to that of checking if the estimated FIR coefficients are "equal to" the FIR coefficients of the nominal model. However, comparing the high-dimensional FIR coefficient vectors is non-trivial and it forms the motivation for using SVMs. *In what follows we define "normal" models as the ones that fall inside the uncertainty region around the nominal model due to the variance of parameter estimates from routine operating data. The models falling outside the normal uncertainty region are considered having mismatch with respect to the nominal model. With such definitions, we can distinguish whether the discrepancy of an estimated model (using routine data) from nominal models is caused by parameter identification error or by the MPM.*

Fig. 1 demonstrates the proposed approach to detect the MPM. The industrial data are split into training data and test data. The training data are collected during a time interval in which the MPM is absent, e.g., the period during or right after the identification stage, as shown in the above figure. Notice that our algorithm

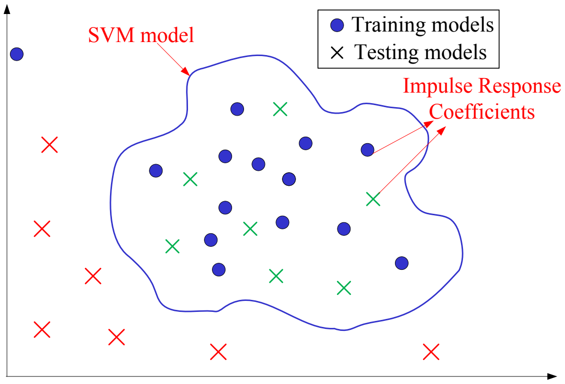

Figure 2: Illustration of the MPM detection idea. Here the training and testing models refer to the process model estimates from training and test data sets.

is in the form of moving windows and the size of training data can be properly selected according to the window size. For each window, we apply the closed-loop identification to attain an estimate of the process model. With the set of estimated process models from moving windows in the training data, a one-class SVM is trained which can be interpreted as an appropriate bound encompassing this set (see Fig. 2). Any model inside this boundary is considered as normal, indicating the absence of MPM. *Note that Fig. 2 is only a graphical illustration of the one-class SVM model characterizing the normal cluster of estimated models. The irregular decision boundary of the SVM depicts the highly nonlinear model generated from SVM combined with kernels. It does not represent any actually trained one-class SVM model.*. For the test data, a similar moving window is performed and each process model estimate obtained is examined by the SVM model to predict whether it is located inside the boundary. If so, the SVM returns a positive score value implying no MPM and otherwise, it returns a negative value to indicate the MPM in the current moving window. *To be cautious in triggering an identification experiment, we define an indicator computing the portion of negative scores over the last period of time. If the portion exceeds a (conservative) threshold, the MPM alarm will be raised since it indicates that extensive moving windows have been identified to have MPM symptoms.* Note that the entire training and testing operations are carried out with routine operating data free of external excitations. In the following sections we focus on the closed-loop identification as well as the SVM training and testing.

3. Closed-Loop Identification

From the explanations above we can see that closed-loop identification plays a fundamental role in determining the performance of our proposed MPM detection algorithm. The last few decades have witnessed a considerable emergence of closed-loop identification methods, such as direct identification, indirect identification, joint input-output methods, projection methods and so on. Nevertheless, none of these available methods are suitable for the particular problem of mismatch detection. The latter three methods require the inclusion of external dither signal and/or linear feedback controller. The direct identification method often leads to biased estimates of the process model if the noise model structure is not sufficiently specified. In this work, we propose a novel closed-loop identification approach, which aims at resolving the bias issue and fits into the proposed mismatch detection context. It is proved that the proposed method can give consistent estimates for parameters in the process model, provided that some mild technical assumptions are satisfied.

Consider the following SISO Box-Jenkins plant

$$y(t) = G_0(q)u(t) + H_0(q)e(t), \tag{1}$$

where $G_0(q)$ and $H_0(q)$ are the true plant and noise model, respectively. We suppose that $H_0(q)$ is monic, stable and inversely stable. The plant $G_0(q)$ is assumed to be stable and contain at least one sample delay. q is the unit-shift operator. $e(t)$ is assumed to be zero-mean Gaussian white noise with constant variance σ^2. $y(t)$ and $u(t)$ represent the output and input signals, respectively. It is assumed that all relevant signals are quasi-stationary. In the closed-loop system with MPC, the feedback controller often displays nonlinear dynamics (if the constraints are active), which are expressed as

$$u(t) = k(t, u^{t-1}, y^t), \tag{2}$$

where $u^{t-1} = \{u(1), \ldots, u(t-1)\}$ and $y^t = \{y(1), \ldots, y(t)\}$. Before identifying the above models it is important to mention that for the case with external excitations, a persistently exciting dither signal can always guarantee that the closed-loop data are informative, regardless of the controller orders. However, without external excitations, for linear controller to achieve the informativity requirement, the following relationship must be satisfied for Box-Jenkins models [21]:

$$\max(n_x - n_f, n_y - n_b) \geq n_d + \min(n_x, n_f), \tag{3}$$

where n_x and n_y are numerator and denominator orders of the linear controller, respectively. n_b and n_f stand for the orders of the process model numerator and denominator polynomials, respectively. n_d denotes the order of the numerator polynomial in the noise model. One observation from (3) is that more complex controllers often imply richer information in the closed-loop data [22]. Additionally, if the controller is nonlinear, as is the case of MPC (2), the closed-loop data are generally sufficiently exciting for relevant system identifications [23]. Another benefit of a nonlinear controller is that it can prevent the identification algorithm from returning an estimate of controller inverse. Moreover, we assume that the prior knowledge of time-delay is approximately known and is specified to the identification algorithm. In this paper, the following assumption is posed.

Assumption 1. The feedback controller (2), e.g., MPC, is complex enough to ensure the closed-loop identifiability for the selected model structures.

The bias in direct closed-loop identification arises from mis-specification of the selected noise model structure. In order to circumvent this limitation, we can represent the noise model with an FIR structure

[24]. Specifically, let us re-write the above true process model into an equivalent form

$$\frac{1}{H_0}y(t) = \frac{1}{H_0}G_0(q)u(t) + e(t), \tag{4}$$

which can be easily represented by a high-order ARX model, with

$$A_0(q)y(t) = B_0(q)u(t) + e(t), \tag{5}$$

where $A_0(q) = \frac{1}{H_0(q)}$, $B_0(q) = A_0(q)G_0(q)$, and

$$\begin{aligned} A_0(q) &= 1 + a_1^0 q^{-1} + \ldots + a_n^0 q^{-n} + \ldots, \\ B_0(q) &= b_0^0 q^{-d} + b_1^0 q^{-d-1} + \ldots + b_n^0 q^{-d-n} + \ldots, \end{aligned}$$

where d is the true time-delay. Our proposed closed-loop identification consists of two steps: high-order ARX model estimation from closed-loop data, and output-error (OE) identification with filtered input and output data, where the filter is chosen as the estimated $A_0(q)$ polynomial. Thus the proposed method is termed as ARX-OE method.

3.1. High-order ARX model

For open-loop stable systems, coefficients of $A_0(q)$ are vanishing for growing n, and the same holds for $B_0(q)$. Thus we parameterize the above infinite-order ARX (5) by using a finite-order ARX model

$$A(q,\eta_n)y(t) = B(q,\eta_n)u(t) + \varepsilon(t), \quad \eta_n \in \Omega_{\eta_n}, \tag{6}$$

where Ω_{η_n} is a compact set,

$$A(q,\eta_n) = 1 + \sum_{k=1}^{n} a_k q^{-k}, \quad B(q,\eta_n) = \sum_{k=0}^{n} b_k q^{-d-k},$$

and we denote

$$\begin{aligned} \eta_n &= [a_1 \ \ldots \ a_n \ b_0 \ b_1 \ \ldots \ b_n]^T, \\ \eta_0 &= [a_1^0 \ \ldots \ a_n^0 \ \ldots \ b_0^0 \ b_1^0 \ \ldots \ b_n^0 \ \ldots]^T. \end{aligned}$$

It is further assumed that $A(q,\eta_n)$ and $B(q,\eta_n)$ are uniformly stable, $\forall \eta_n \in \Omega_{\eta_n}$. Note that for notational simplicity we choose $A(q,\eta_n)$ and $B(q,\eta_n)$ to be of the same order. To reduce the large parameter covariance resulting from high orders we add a "regularization" trick as shown in [25] when identifying (6). For small $\delta > 0$,

$$R^n_{reg}(N) = \begin{cases} R^n(N), & \text{if } \|R^n(N)^{-1}\|_2 < 2/\delta \\ R^n(N) + \frac{\delta}{2}I, & \text{otherwise} \end{cases}$$

where $R^n(N) = \frac{1}{N}\sum_{t=n+1}^{N} \varphi^n(t)\varphi^n(t)^T$ with the regressor $\varphi^n(t) = [-y(t) \ u(t) \ \ldots \ -y(t-n) \ u(t-n)]^T$. N is the sample size. It has been shown in that paper that asymptotically, the first and second order properties of the parameter estimate will not depend on the regularization term. For simplicity of theoretical derivations we assume that the order n of $A(q,\theta_n)$ in (6) increases to infinity as the sample size $N \to \infty$, but n is smaller compared with N. Then we show that asymptotically, the estimate $A(e^{j\omega},\hat{\eta}_n)$ coincides with $A_0(e^{j\omega})$. First, the following assumption is supposed to hold.

Assumption 2. For the selected high-order ARX model structure (6), it holds that

$$n \to \infty,\ n^{3+\delta}/N \to 0,\ \text{as } N \to \infty, \tag{7}$$

where $\delta > 0$ is some constant.

Assumption 2 essentially describes the condition that the model order n in (6) increases to infinity as $N \to \infty$ but with a slower increase speed than N. With this assumption, we further have the following theorem regarding the estimation performance of the high-order ARX model.

Theorem 1. *Suppose that Assumptions 1—2 hold. We have*

$$\sup_{\omega} |A(e^{j\omega},\hat{\eta}_n) - A_0(e^{j\omega})| \to 0, \quad \textit{w.p.1, as } N \to \infty, \tag{8}$$

where

$$\hat{\eta}_n = [\hat{a}_1 \ \ldots \ \hat{a}_n \ \hat{b}_1 \ \ldots \ \hat{b}_n]^T,$$

represents the estimate of parameter vector η_n *in* (6) *from the prediction error method.*

Proof. The proof follows directly from Lemma 2.1 in [26] and is thus omitted here.

Remark 1. Theorem 1 states that allowing the order of the high-order ARX model (6) to increase to infinity but with a much slower increase rate than N, the estimate $A(e^{j\omega},\hat{\eta}_n)$ converges uniformly to the true polynomial $A_0(e^{j\omega})$ with probability one. It essentially depicts the behavior of the high-order ARX model under a large sample. Theorem 1 provides a theoretical basis for the subsequent OE identification with filtered input and output since the estimated filter $A(e^{j\omega},\hat{\eta}_n)$ approximates the true filter $A_0(e^{j\omega})$ sufficiently well. Note that $n \to \infty$ is only an assumption for theoretical derivations, and indeed we can specify a (high-order) finite n based on *a priori* information regarding the noise model when implementing this identification algorithm.

Remark 2. Note that the high-order approximation (6) of the original Box-Jenkins model (1) imposes more stringent on the informativeness of closed-loop data, thereby the condition from 1 may not suffice for closed-loop identifiability. However, the regularization employed above can guarantee the numerical stability and asymptotic properties in estimating the parameters of high-order ARX model.

3.2. OE identification with filtered input-output data

In the second step, we perform an OE model identification on the filtered input and output signals. Here the filter is chosen as the estimated $A(q,\hat{\eta}_n)$ from the first step. The filtered input and output signals are $y(t,\hat{\eta}_n) = A(q,\hat{\eta}_n)y(t)$, $u(t,\hat{\eta}_n) = A(q,\hat{\eta}_n)u(t)$. To identify $G_0(q)$, we need to fit a model as follows

$$y(t,\hat{\eta}_n) = G(q,\rho)u(t,\hat{\eta}_n) + \varepsilon(t), \tag{9}$$

where an FIR structure in $G(q,\rho)$ can be used in general. However, in typical industrial processes we have some prior knowledge to correctly parameterize the plant model. For example, in the MD process of a paper machine, it is widely accepted that a first-order-plus-time-delay model is sufficient to capture the system dynamics. In that case, one can select $G(q,\rho)$ to have the same structure as the true model. The parameter ρ is estimated via minimizing the following criterion:

$$\hat{\rho}_N = \arg\min_{\rho\in\Omega_\rho} \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\varepsilon^2(t,\rho,\hat{\eta}_n), \tag{10}$$

where Ω_ρ is a compact set for parameter ρ and contains the true model, and the prediction error $\varepsilon(t,\rho,\hat{\eta}_n)$ is defined as

$$\varepsilon(t,\rho,\hat{\eta}_n) = [G_0(q) - G(q,\rho)]u(t,\hat{\eta}_n) + \frac{A(q,\hat{\eta}_n)}{A_0(q)}e(t).$$

We have the following theorem regarding the parameter estimate $\hat{\rho}_N$.

Theorem 2. *Consider the true Box-Jenkins model for the plant* (1) *as well as the equivalent high-order ARX form* (5)*. Assume that conditions in Theorem 1 hold and that the plant model is correctly parameterized. Then the parameter estimate $\hat{\rho}_N$ from the prediction error criterion* (10) *is consistent, i.e., we have*

$$\hat{\rho}_N \to \rho_0, \quad \textit{w.p.1, as } N\to\infty, \tag{11}$$

where ρ_0 is the true parameter value of G_0. Moreover, the estimated parameter value $\hat{\rho}_N$ is Gaussian distributed with mean value ρ_0.

Proof. See Appendix Appendix A.

Remark 3. Despite the premise on the correct parameterization of $G_0(q)$ in Theorem 2, it is not supposed to be a restrictive limitation on the proposed ARX-OE method. As shown in the following section, the SVM in MPM detection is trained and tested on the FIR form of $G(q,\hat{\rho}_N)$. Thus using an FIR model in the OE identification step, if *a priori* information about $G_0(q)$ is not accessible, is suggested in such case to eliminate the bias.

Remark 4. Compared with existing closed-loop identification methods, the proposed approach requires no information about the controller and thus is applicable to closed-loop data with both linear and nonlinear controllers. Moreover, it does not rely on external excitations. However, it is also suitable for cases where external excitations are present. In other words, this identification algorithm can be performed for both routine operating data and experimental data. Note that the explicit expression for the variance of $\hat{\rho}_N$ is non-trivial since the variance of parameter estimates in the first step will affect that of parameter estimates in the second step. It is thus recommended to use a set of training data from which we can obtain a collection of process model estimates as an approximation to the variance of transfer function estimates.

3.3. Closed-loop identification for MIMO systems

The proposed ARX-OE method can be easily extended to MIMO systems. For a low-dimensional MIMO system, each output channel is considered as a closed-loop multi-input-single-output (MISO) system. Similar to SISO identification, in the first step, a high-order MISO ARX model is estimated as

$$A_i(q, \eta_i^a) y_i(t) = \sum_{j=1}^{i} B_{i,j}(q, \eta_{i,j}^b) u_j(t) + \varepsilon_i(t), \quad i = 1, \ldots, m, \tag{12}$$

where m is the number of output channels, and

$$A_i(q, \eta_i^a) = 1 + a_{i,1} q^{-1} + \ldots + a_{i,n} q^{-n}, \quad B_{i,j}(q, \eta_{i,j}^b) = 1 + b_{i,j,1} q^{-1} + \ldots + b_{i,j,n} q^{-n},$$

where $A_i(q, \eta_i^a)$ is the inverse of noise model of the i-th output channel. Define $\hat{A}_i(q)$, $\hat{B}_{i,j}(q)$ as the respective estimates of $A_i(q, \eta_i^a)$ and $B_{i,j}(q, \eta_{i,j}^b)$. We then apply the MISO OE identification after filtering the input and output signal using the obtained noise model estimate. We perform the following identification

$$y_i^f(t) = \sum_{j=1}^{i} G_{i,j}(q, \rho_{i,j}) u_j^f(t) + \varepsilon_i(t), \quad i = 1, \ldots, m,$$

where

$$y_i^f(t) = \hat{A}_i(q) y_i(t), \quad u_j^f(t) = \hat{A}_i(q) u_j(t).$$

A similar statement on the consistency of parameter estimates for $\rho_{i,j}$ can be obtained and is omitted here.

4. MPM Detection

In this section, we will illustrate the procedures of performing MPM detection with SVM classifications. Note that we would only consider that for SISO systems since the corresponding technique follows a similar line for MIMO systems.

4.1. One-class learning support vector machines

As a convention, the SVM is developed particularly for the binary classification problem. It provides a hyperplane that not only separates two classes of data, but also guarantees that it has the maximum distance to either class. The benefit of such a hyperplane is its robustness to outliers, thus considerably reducing false classifications. *For the specific MPM detection problem, we use the set of process models from training data as a reference group representing the behaviors of "no mismatch" process model cluster. The reason is that in practice, normal operation data without MPM are easy to collect from which the cluster of normal models can be directly formed. However, the other group of data is ordinarily not accessible since abnormal situations may occur in a variety of ways such as various parametric mismatches, irregular disturbances and so on. The data under different mismatch scenarios may demonstrate different patterns. Therefore, it is difficult to find a representative abnormal data set as the other class. Besides, the operating data containing MPM are rare in practice since MPM often leads to degraded control performance and thus loss of profit to* *the industry.* Thus the MPM detection is a one-class learning problem, which is also known as the "novelty detection problem".

The one-class learning SVM is depicted in the feature space, i.e., a space that the data are mapped into. Consider the set of training data samples

$$x_1, x_2, \ldots, x_l \in \mathcal{X} \subset R^m, \tag{13}$$

where l is the number of training data and $\mathcal{X}$ is a subset (called input space) of R^m. Prior to one-class SVM training it is necessary to map the data through $\Phi : \mathcal{X} \mapsto F$ into a (higher-dimensional) feature space F. The kernel function $\kappa(x, y)$ is such that the inner product in the feature space can be evaluated in the input space as

$$\kappa(x, y) =< \Phi(x), \Phi(y) >, \quad \forall x, y \in \mathcal{X}. \tag{14}$$

A well-known kernel function that will be used hereafter is the Gaussian kernel

$$\kappa(x, y) = e^{-\|x-y\|^2/c}, \tag{15}$$

where c is a parameter that is used to tune the shape of the Gaussian kernel function. It should be pointed out that with Gaussian kernel function all data points in the feature space are located in the same orthant since $\kappa(x, y) > 0, \ \forall x, y \in \mathcal{X}$. Thus it is possible to find a hyperplane to separate the origin from the training data in the feature space with maximized margin. With this idea the one-class SVM training problem is formulated as [27]

$$\min_{w,\xi,b} \quad \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i=1}^{l} \xi_i - b \tag{16}$$

$$s.t. \quad w^T\Phi(x_i) \geq b - \xi_i, \ \xi_i \geq 0, \tag{17}$$

where w and b represent the slope and offset of the hyperplane in the feature space. The term $v \in (0, 1]$ is a parameter tuning the upper bound of the fraction of outliers and lower bound of the fraction of support vectors. ξ is a slack variable allowing for local violations of the hard boundary determined by the hyperplane. Solving the optimization problem (16)-(17) can be converted into solving the following dual problem,

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\alpha_j\kappa(x_i, x_j) \tag{18}$$

$$s.t. \quad 0 \leq \alpha_i \leq \frac{1}{vl}, \quad \sum_{i=1}^{l} \alpha_i = 1, \tag{19}$$

where α_i are the dual variables. It is obvious that while the primal problem is formulated in the feature space, its dual problem can be solved in the input space by resorting to the kernel function. Thus we can avoid the intense computation arising from large dimensions of the feature space. Efficient algorithms are available in the literature to solve this dual problem. In the simulations presented in Section 5, we used sequential minimal optimization (SMQ) [27] to solve the dual problem above. A feature associated with the solution $\hat{\alpha}$ of dual problem is its sparsity, with most optimal dual variables $\hat{\alpha}_i$ valued at zero. Data points corresponding to nonzero optimal dual variables are known as support vectors and it is revealed that the optimal w and b (denoted as $\hat{w}$ and $\hat{b}$, respectively) are completely determined by those nonzero optimal dual variables. Furthermore, with kernel function, the decision (or score value) function is also represented in the input space, instead of in the high-dimensional feature space, by the following,

$$h(x) = \sum_{i=1}^{l} \hat{\alpha}_i \kappa(x_i, x) - \hat{b} \tag{20}$$

where x is a test example. Note that the sum in (20) typically involves $n_\alpha << l$ nonzero terms, where n_α is the number of nonzero dual variables. This allows for efficient evaluation. For a given test example x, the value $|h(x)|$ represents the distance of x to the separating hyperplane. If $h(x) > 0$, it means that x can be classified into the initial class. Otherwise x does not belong to that class. We note that the incorporation of kernel functions significantly expands the flexibility of SVM in constructing separating boundaries, enabling it to generate a nonlinear classifier in the input space.

A critical issue in applying the one-class SVM training strategy to MPM detection is the limited amount of training data available in the industrial processes. Taking the paper machine as an example, grade changes (setpoint changes) often take place on a daily basis and thus training data has to be collected after each grade change to represent the current operation condition before carrying out MPM detection. Consequently, only a few process model estimates from training data are available to build an SVM model. In order to overcome this issue we use a resampling technique to enlarge the cluster of no mismatch models estimated from training data before performing the SVM training.

4.2. Resampling

The main principle we adopt here is to fit a probability density function (PDF) to each impulse response (IR) coefficient of the estimated process model. Then a large number of samples can be generated by sampling randomly from the estimated density function. More specifically, denote the FIR form of the estimated process model $G(q, \hat{\rho}_N)$ in the k-th moving window as

$$G(q, \hat{\rho}_N^k) = \hat{g}_0^k q^{-d} + \hat{g}_1^k q^{-d-1} + \ldots + \hat{g}_m^k q^{-d-m}, \tag{21}$$

where m is the pre-specified number of FIR coefficients. $k = 1, 2, \ldots, N_k$, are indices of moving windows in the training data. It is straightforward that FIR coefficients $\hat{g}_i^k, i = 0, \ldots, m$, are Gaussian distributed, given that $\hat{\rho}_N^k$ has Gaussian distribution (cf. Theorem 2). For each coefficient $\hat{g}_i^k$, several estimated values are attained from moving windows in the training data. Then we can construct rough estimators for the mean and variance of each IR coefficient

$$\begin{aligned}
\hat{\mu}_i &= \mu(\hat{g}_i^1, \hat{g}_i^2, \ldots, \hat{g}_i^{N_k}),\ i = 0, \ldots, m, \\
\hat{\sigma}_i^2 &= \sigma(\hat{g}_i^1, \hat{g}_i^2, \ldots, \hat{g}_i^{N_k}),\ i = 0, \ldots, m,
\end{aligned}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are some functions. One choice of these two functions is sample mean and sample variance. Due to the limited amount of training data (N_k normally is small), the estimated PDF for each FIR coefficient is much more conservative than the true PDF. Thus we use a parameter α to tune the width of the PDF to avoid this problem. If we have plenty of training data, α shall be small. Otherwise, α shall be large. The rationale behind these guidelines is that more training data may give us a more precise and reliable estimate of the PDF and vice versa. The next step is to use the resampling idea to randomly generate a large number of samples for each FIR coefficient subject to the corresponding estimated PDF. Then a one-class SVM model can be developed from these enhanced samples for the initial cluster of "good" process models.

4.3. MPM detection with SVM

With the trained one-class SVM, we first need to estimate FIR coefficients of the process model identified from each moving window in the test data, following the same procedures as previous sections. For estimated FIR coefficients, we apply the SVM model to predict whether they belong to the initial cluster. If so, the SVM returns a positive score value indicating that the current testing window does not display any sign of mismatch. Otherwise the SVM returns a negative score to signify the mismatch. *However, to be cautious to start an identification experiment, we define an indicator showing the portion of negative scores over the last few moving windows. A threshold is selected and the MPM alarm is raised if the portion of negative scores exceeds this threshold.* Specifically, define I_t as the sign of SVM scores for the moving window at time instant t

$$I_t = sign(h(x_t)), \tag{22}$$

with x_t being the FIR coefficient vector of the plant model estimate for the window data at time t. Denote $T_t = \{t - n_T, \dots, t-1, t\}$, where n_T is a detection interval, i.e., the number of previous moving windows under inspection to determine the existence of MPM. We further define an MPM indicator

$$s = \frac{|I_-|}{n_T}, \tag{23}$$

where $I_- := \{I_i = -1 : i \in T_t\}$ and $|I_-|$ is the number of elements in the set $|I_-|$. The users can specify a threshold s_T for the MPM indicator to raise an MPM alarm. We suggest a conservative s_T (e.g., $s_T = 0.95$) to be circumspect in raising the MPM alarm.

Remark 5. Note that the MPM detection method presented above can also be applied to the noise model estimate $A(q, \hat{\eta}_n)$ from (6) to find the noise mismatch. In this manner we can monitor the process and noise models separately to distinguish MPM from noise model change. Note that while the controllers considered in this paper are assumed to be tuned based solely on the plant model, in cases where controller tuning also depends on noise model (such as minimum variance control and some MPCs), detection of a noise model change can also be used to trigger an identification experiment. Moreover, if *a priori* information about the true process model structure is not available, we can specify it with an FIR structure to acquire consistent model estimates and the subsequent mismatch detection scheme is still applicable.

Table 1: Parameters setup for the SISO example

Parameters	Values	Note
W_{size}	2 hours	Moving window size
W_{step}	5 min	Moving window step size
n	9	Order of ARX model
T_{train}	3 hours	Duration of training stage
n_T	2 hours	Mismatch inspection interval
s_T	0.95	Mismatch threshold
α	1.5	Tuning the width of estimated PDF

5. Examples

In this section we demonstrate the MPM detection algorithm through a SISO example and an MIMO example from the MD process of paper machines.

5.1. SISO example

For the SISO example, the controlled variable (CV) is dry weight and the MV is stockflow. The sampling interval is set as 5 seconds. After discretization we obtain the true plant model

$$G_0 = \frac{0.1003}{1 - 0.9048q^{-1}} q^{-17}.$$

In the simulation of MD process the true noise model is selected as

$$H_0 = \frac{1 - 0.3q^{-1}}{1 + 0.6q^{-1}}.$$

We use an MPC as the MD controller. To reflect the reality of a paper machine's operating condition we set the standard deviation of noise to be $\sigma = 0.05$. The entire simulation lasts 15 hours without any setpoint change. During this simulation, initially there is no MPM. After 7 hours we change the true noise model into

$$H_0 = \frac{1 + 0.3q^{-1}}{1 - 0.6q^{-1}},$$

to create a noise mismatch. Furthermore, we double the plant gain to introduce an MPM after 11 hours. *Thus the first 5040 samples in this data set are normal and the rest are abnormal containing either noise change or MPM.* The objective is to examine whether the proposed MPM detection algorithm is able to detect the process and noise model changes separately, with the collected routine operating data. Configurations of parameters relevant to the MPM detection algorithm are summarized in Table 1. The tuning parameters for the SISO MD MPC are illustrated in Table 2. Note that we use the same window size and step size for both training and testing stages.

Fig. 3 depicts the simulated CV and MV profiles after subtracting their respective means. Note that the first vertical red dash-dotted line indicates the time instant at which the noise change is introduced to the process. The second vertical line shows the time when we create an MPM. In plotting this graph we have removed the mean from the profiles. It is obvious that both the noise change and MPM bring significant variations to the profiles which are unfavored since the control objective is to keep the CV profile as flat as possible. However, it is stressed that poor control performance from merely noise change should not trigger an identification experiment. We use the first 3 hours of data (2160 samples) as training data and the rest as test data. *For both SISO and MIMO examples in this section, we used Gaussian kernel with* $c = 1$ *as the chosen kernel and SMO algorithm to solve the optimization problem. The* ν *parameter is set to 0.5 for both examples. During the training stage, we perform 10-fold cross-validation on the training data to determine the outlier rate, which is a parameter controlling the amount of allowable outliers in the normal class. For this example, the selected outlier rate is 0.05, meaning that about 5% of the training data can be considered as outliers. The validation accuracy of the SVM for the plant model is 0.9498.*

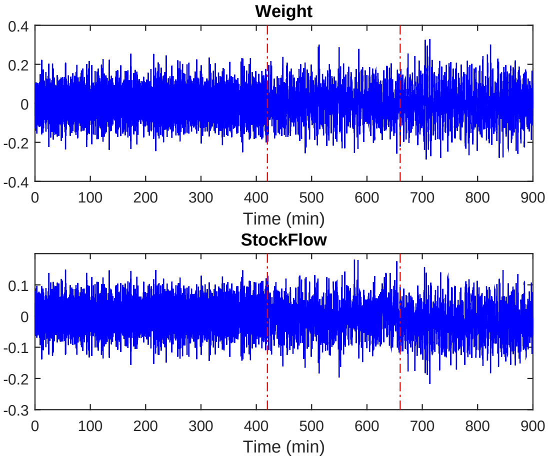

Figure 3: Simulated input and output profiles after subtracting their respective means

Fig. 4 demonstrates the detection results for both noise model change and MPM. Specifically, the first and third figures display the predicted SVM scores (cf. (20)) for noise and process model estimates, respectively. *Note that these scores can be interpreted as the signed distance of the underlying data from the separating boundary provided by the trained SVM. The sign of the scores indicates the classification result with positive value implying the normal case and negative scores implying the occurrence of MPM or noise change. We assign zero score to the one-class SVM predictions if the SVM is under training, e.g., the first 3 hours in Fig. 4.* Clearly, the SVM scores drop to negative values after the corresponding noise model change occurs. The second and fourth plots track the mismatch indicator values s in (23) for both noise and process models. The red dash-dotted line highlights the specified threshold to raise an alarm. Ideally, a system identification experiment is triggered once the s value of process model exceeds the threshold s_T. However, in this example we neglect the identification part since it is beyond the scope of this paper. From Figure 4 it is clear that noise change at the 7-th hour does not affect the prediction of MPM. Thus we can conclude that the proposed

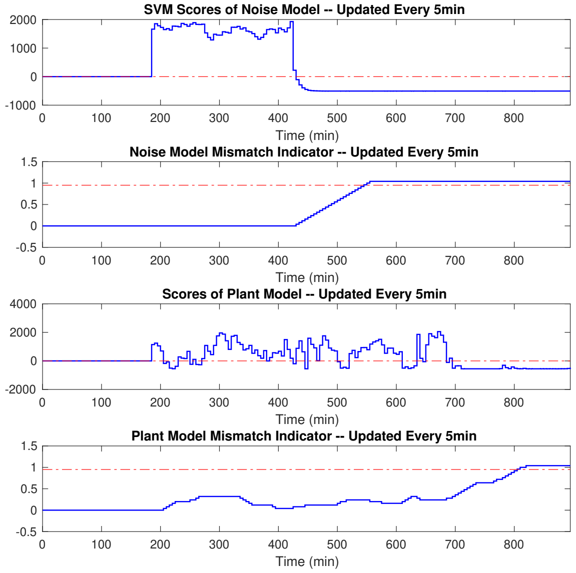

Figure 4: MPM and noise change detection results

Table 2: Tuning parameters of the SISO MD MPC

Tuning parameters	Values	Tuning parameters	Values
Actuator movement weight	0.01	Actuator deviation weight	0
CV target value	23.75 $lbs/3000ft^2$	MV target value	131 gpm
Prediction horizon	40	Control horizon	4
CV target weight	1	MV target weight	1
CV min-max weight	1	MV min-max weight	1
Bounds of MV	[0 3000]	Limit of MV movement	0.1667

305 mismatch detection scheme is capable of monitoring the noise model change and MPM separately and thus is able to discriminate MPM from noise model change with only routine operating data.

5.2. MIMO example

MIMO MD processes consider the interactions among various types of actuators and measured paper properties. A typical MIMO MD process is usually modeled by lower-triangular transfer matrices due to the unique physical coupling relationships between MVs and CVs. In this example, three CVs are considered, namely press dry weight, press moisture and reel moisture. Correspondingly, the three MVs are stockflow,

Table 3: Parameters setup for the MIMO example

Parameters	Values	Note
W_{size}	2 hours	Moving window size
W_{step}	5 min	Moving window step size
n_a	18	Order of each $A_i(q)$
n_b	10	Order of each $B_{i,j}(q)$
T_{train}	4 hours	Duration of training stage
n_T	2 hours	Mismatch inspection interval
s_T	0.95	Mismatch threshold
α	1.5	Tuning the width of estimated PDF

steam 4 and steam 3. The (discrete) true process and noise models are shown to be

$$G_0(q) = \begin{bmatrix} \frac{0.0763q^{-15}}{1-0.0723q^{-1}} & 0 & 0 \\ \frac{0.0305q^{-19}}{1-0.1030q^{-1}} & \frac{-0.0026q^{-23}}{1-0.0183q^{-1}} & 0 \\ \frac{0.0239q^{-9}}{1-0.0318^{-1}} & \frac{-0.0056q^{-7}}{1-0.0234q^{-1}} & \frac{-0.0101q^{-19}}{1-0.1823q^{-1}} \end{bmatrix}, \qquad H_0(q) = \begin{bmatrix} \frac{1-0.5q^{-1}}{1+0.5q^{-1}} & 0 & 0 \\ 0 & \frac{1-0.3q^{-1}}{1+0.6q^{-1}} & 0 \\ 0 & 0 & \frac{1+0.2q^{-1}}{1+0.7q^{-1}} \end{bmatrix}.$$

An observation is that the gain for plant $G_{2,2}(q)$ is small, and this poses significant challenge for the corresponding system identification. The resultant poor signal-to-noise ratio leads to large variance associated with the gain parameter estimate. The entire simulation lasts 1725 minutes with 20700 samples and there is no setpoint change throughout this simulation. A set of different tuning parameters are employed in this example to enhance the performance of the proposed MPM detection method. Lists of tuning parameters for the MPM detection method and for the MD MPC are shown in Table 3 and Table 4, respectively. *From the 10-fold cross-validation, the selected outlier rate in training these one-class SVM models is 0.2.*

In this simulation, the closed-loop system first operates without noise change or MPM. The noise model change was introduced after 575 minutes and the noise model $H_{2,2}(q)$ was altered to $\frac{1+0.3q^{-1}}{1-0.6q^{-1}}$. A gain mismatch was introduced after 1000 minutes on $G_{3,1}(q)$ by doubling the true plant gain while keeping the gain in the model unchanged. After 1250 minutes, a gain mismatch was introduced to $G_{1,1}(q)$ by doubling the gain of true plant. *This data set contains the first 6900 samples as normal data and the rest as abnormal data.* The simulated input-output profiles are shown in Fig. 5—6. Notice that the red vertical lines in Fig. 5 from top to bottom gives the time instants at which the double gain mismatch of $G_{1,1}(q)$, the noise model change, and the double gain mismatch of $G_{3,1}(q)$ are created. These changes in the process characteristics can hardly be detected by inspecting the input and output variances. To verify the proposed MPM detection algorithm, we apply it to this MIMO system and monitor the occurrence of noise change and MPM for each transfer function with the parameters in Table 3.

We use the first 4 hours of data (2880 samples) as training data. The detection results obtained from the proposed method are illustrated in Fig. 7. Note that the sign of these scores indicates whether there is a MPM in the underlying moving window. *Similar to the previous SISO example, these scores show the signed*

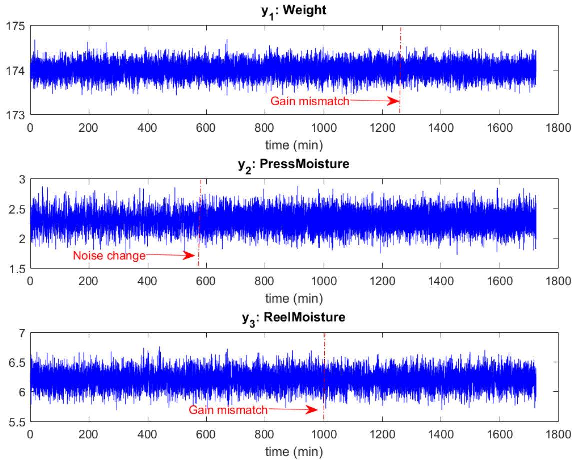

Figure 5: Output profiles for the MIMO MD process

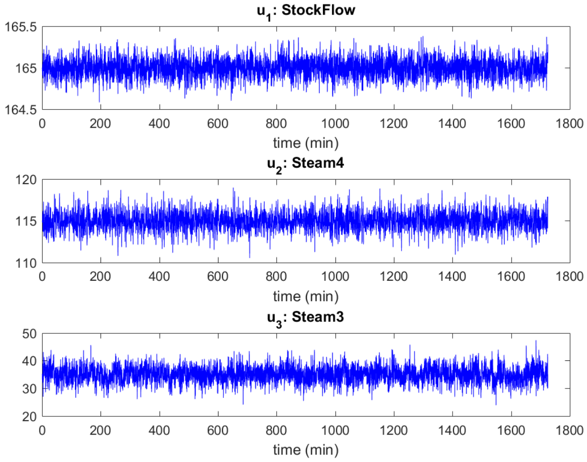

Figure 6: Input profiles for the MIMO MD process

distance of the corresponding data point with respect to the SVM separating boundary. The signs indicate the corresponding classification result: positive values imply the normal case, negative values indicate MPM or noise change, and zero values show that the SVM is still under training . A large number of consecutive negative scores is a sign of MPM. From these plots one can see that the scores for $G_{1,1}(q)$ and $G_{3,1}(q)$ degrade to negative values after the model-plant mismatches were created. Moreover, the monitoring on the other transfer functions is not affected (no false positives for the other models) by the mismatches in $G_{1,1}(q)$ and $G_{3,1}(q)$. This is significant for those processes that require or admit performing closed-loop identification experiments for individual models instead of an MIMO plant-wide identification. In addition, as pointed out in previous sections, our method can also monitor the noise model and thus can provide more enriched information for the diagnosis. To this end, we demonstrate the SVM scores for three noises models in Fig. 8. It is apparent that scores for $H_{2,2}(q)$ drop to be negative rapidly after the noise model change was created, which underscores the effectiveness of the proposed method in discovering this noise model change. It is stressed that this noise change does not trigger any MPM alarms and this further verifies the reliability of our method in discriminating the MPM from noise model changes.

6. Limitations

The proposed method aims to address the challenging problem of detecting MPM with routine operating data and separating MPM from noise model changes. For the proposed method, closed-loop identification is essential and failing to meet the identifiability conditions, for example, due to an overly simple controller or a small time-delay, may lead to degraded performance in detecting MPM. Moreover, when the gain of the true plant is small, often implying poor signal-to-noise ratio, the uncertainty on the estimated gain parameter will easily mask the gain mismatch and thus it becomes difficult to distinguish between the mismatch and the uncertainty of parameter estimate. In addition, the determination of scaling parameter α that tunes the

Table 4: Tuning parameters of the MIMO MD MPC

Tuning parameters	Values	Tuning parameters	Values
Actuator movement weight	[0.01 0.01 0.01]	Actuator deviation weight	[0 0 0]
CVs target weight	[1 1 1]	MVs target weight	[1 1 1]
CVs min-max weight	[1 1 1]	MVs min-max weight	[1 1 1]
Actuator movement weight	[0.01 0.01 0.01]	Actuator deviation weight	[0 0 0]
Target value for CV1	174 $lbs/3000ft^2$	Target value for MV1	165 gpm
Target value for CV2	2.3 %	Target value for MV2	115 psi
Target value for CV3	6.2 %	Target value for MV3	35 psi
Prediction horizon	20	Control horizon	2
Upper bounds of MVs	[250 180 180]	Lower bounds of MVs	[0 0 0]
Limits of CV movement	[0.1 0.1 0.1]	Limits of MV movement	[0.2 1 1]

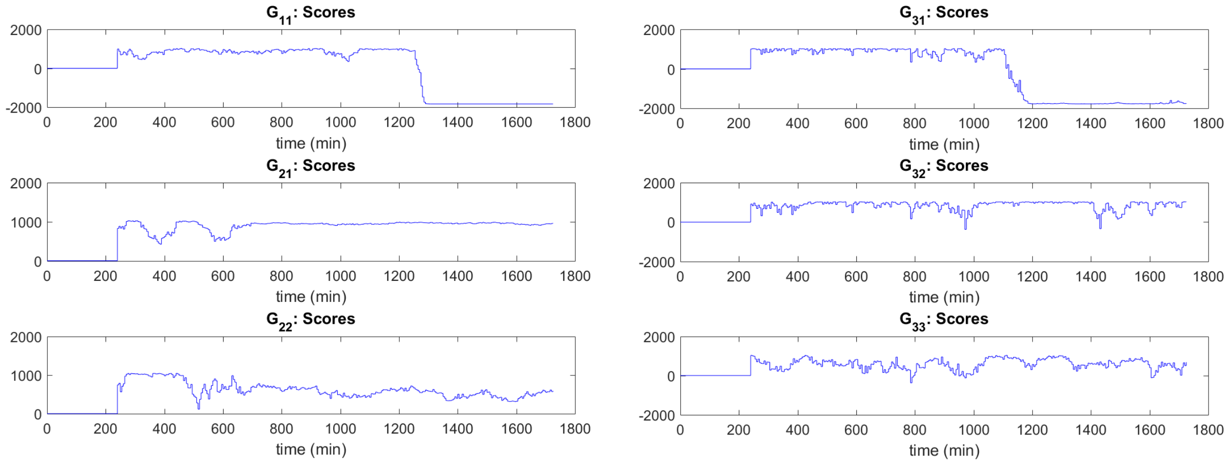

Figure 7: SVM scores for each transfer function of the MIMO example

width of estimated PDF requires insights into the trade-off between the sensitivity and robustness of the one-class SVM classifier. *In addition, this work uses one-class classifier and thus traditional performance metrics for binary classification such as precision, recall and F-score are not applicable to assess the trained classifier. We consider this as one of our future research topics, especially given that attention on developing performance metrics for one-class classifiers is growing rapidly in the literature.*

7. Conclusion

This paper presents a novel MPM detection algorithm that can separate MPM from noise model changes, relying only on routine operating data. To this end, we proposed a new closed-loop ARX-OE method that can give consistent parameter estimates for the process model without the need for any *a priori* information

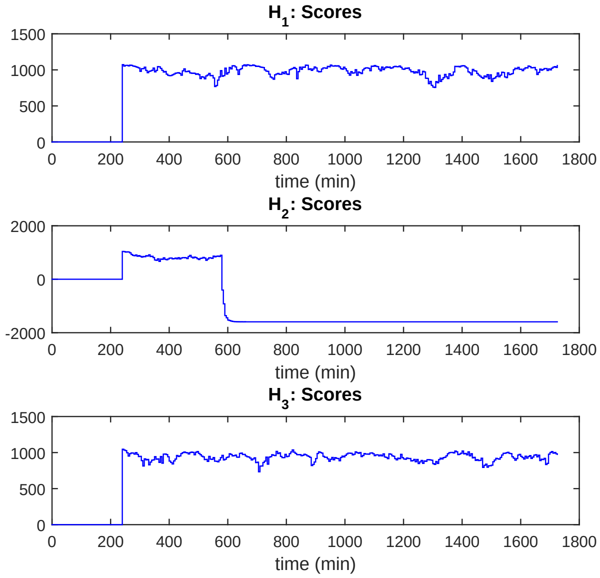

Figure 8: SVM scores for the noise models

about the noise model. This identification algorithms can also be extended to MIMO systems. For the MPM detection problem, we split it into a training stage and a testing stage. In the training stage, one-class SVM models can be developed based on the process model estimates obtained from routine identification with the ARX-OE algorithm. The trained SVM models are then used to monitor the model estimates from test data in order to detect the occurence of MPM. With the same procedures we can train SVM models for noise models to detect noise model changes. This technique is tailored well enough to meet the industrial demands on MPM monitoring. A SISO example and an MIMO example from MD processes of paper machines are presented to illustrate the effectiveness of the proposed methods.

Acknowledgment

The work was supported by Honeywell Process Solutions, Natural Science and Engineering Research Council (NSERC), and Vanier Canada Graduate Scholarships.

Appendix A. Proof of Theorem 2

Define

$$\varepsilon(t,\rho) := [G_0 - G(q,\rho)]u(t) + \frac{1}{A(q,\eta_0)}e(t), \tag{A.1}$$

and

$$\begin{aligned} V_N(\rho,\hat{\eta}_N) &:= \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}[A(q,\hat{\eta}_N)\varepsilon(t,\rho)]^2, \\ V_N(\rho,\eta_0) &:= \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}[A(q,\eta_0)\varepsilon(t,\rho)]^2. \end{aligned}$$

Notice that with the presence of feedback, one can consider $u(t)$ to be generated by passing the reference signal $r(t)$ and external noise $e(t)$ through a set of uniformly stable filters. Moreover, from Assumptions 1 and that $A(q,\eta_0)$ is stable and inversely stable, the conditions in Theorem 2B.1 in [23] are satisfied. Therefore, we have

$$\sup_{\rho\in\Omega_\rho}\left\|\frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\left[A(q,\eta_0)\varepsilon(t,\rho)\right]^2-\overline{V}(\rho,\eta_0)\right\|\to 0, \quad \text{w.p.1, as } N\to\infty, \tag{A.2}$$

where

$$\overline{V}(\rho,\eta_0)=\overline{E}\frac{1}{2}\left[A(q,\eta_0)\varepsilon(t,\rho)\right]^2.$$

A similar conclusion can be drawn for $V_N(\rho,\hat{\eta}_N)$ as well.

Notice that the loss function can be decomposed as

$$\begin{aligned} V_N(\rho,\hat{\eta}_N) &= \frac{1}{N}\sum_{t=1}^{N}\left\{\frac{1}{2}\left[A(q,\eta_0)\varepsilon(t,\rho)\right]^2+\frac{1}{2}\left[(A(q,\hat{\eta}_N)-A(q,\eta_0))\varepsilon(t,\rho)\right]\right. \\ &\qquad \left.\cdot\left[A(q,\eta_0)\varepsilon(t,\rho)\right]+\frac{1}{2}\left[(A(q,\hat{\eta}_N)-A(q,\eta_0))\varepsilon(t,\rho)\right]\left[A(q,\hat{\eta}_N)\varepsilon(t,\rho)\right]\right\}. \end{aligned}$$

Since $\varepsilon(t,\rho)$ is bounded uniformly, from (A.2) and Theorem 1, it follows that

$$\sup_{\rho\in\Omega_\rho}\left\|\frac{1}{N}\sum_{t=1}^{N}[(A(q,\hat{\eta}_N)-A(q,\eta_0))\,\varepsilon(t,\rho)][A(q,\eta_0)\varepsilon(t,\rho)]\right\|\to 0, \quad \text{w.p.1, as } N\to\infty,$$

$$\sup_{\rho\in\Omega_\rho}\left\|\frac{1}{N}\sum_{t=1}^{N}[(A(q,\hat{\eta}_N)-A(q,\eta_0))\,\varepsilon(t,\rho)][A(q,\hat{\eta}_N)\varepsilon(t,\rho)]\right\|\to 0, \quad \text{w.p.1, as } N\to\infty.$$

Applying the triangular inequality yields,

$$\begin{aligned} \sup_{\rho\in\Omega_\rho}\left\|V_N(\rho,\hat{\eta}_N)-\overline{V}(\rho,\eta_0)\right\| &\le \sup_{\rho\in\Omega_\rho}\left\|\frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\left[A(q,\eta_0)\varepsilon(t,\rho)\right]^2-\overline{E}\frac{1}{2}\left[A(q,\eta_0)\varepsilon(t,\rho)\right]^2\right\| \\ &\quad+\sup_{\rho\in\Omega_\rho}\left\|\frac{1}{2N}\sum_{t=1}^{N}[(A(q,\hat{\eta}_N)-A(q,\eta_0))\,\varepsilon(t,\rho)][A(q,\eta_0)\varepsilon(t,\rho)]\right\| \\ &\quad+\sup_{\rho\in\Omega_\rho}\left\|\frac{1}{2N}\sum_{t=1}^{N}[(A(q,\hat{\eta}_N)-A(q,\eta_0))\,\varepsilon(t,\rho)][A(q,\hat{\eta}_N)\varepsilon(t,\rho)]\right\| \\ &\to 0, \quad \text{w.p.1, as } N\to\infty. \end{aligned}$$

That is,

$$V_N(\rho,\hat{\eta}_N)\to\overline{V}(\rho,\eta_0), \quad \text{w.p.1, as } N\to\infty,$$

Applying Parseval's theorem yields

$$\overline{V}(\rho,\eta_0)=\frac{1}{4\pi}\int_{-\pi}^{\pi}|G_0(e^{j\omega})-G(e^{j\omega},\rho)|^2\frac{\Phi_u(\omega)}{|H_0(e^{j\omega})|^2}d\omega+\sigma^2.$$

Therefore, if $G(q, \rho)$ is correctly parameterized and closed-loop data are informative enough for selected model structures, we can conclude that (11) is valid. The proof for the Gaussian distribution of $\hat{\eta}_N$ follows the lines of Theorem 9.1 in [23] and is thus omitted here.

References

[1] M. Jelali, Control performance management in industrial automation: assessment, diagnosis and improvement of control loop performance, Springer Science & Business Media, 2012.

[2] V. Botelho, J. O. Trierweiler, M. Farenzena, R. Duraiski, Perspectives and challenges in performance assessment of model predictive control, The Canadian Journal of Chemical Engineering 94 (7) (2016) 1225–1241.

[3] S. Wang, M. Baldea, Autocovariance-based MPC model mismatch estimation for SISO systems, in: 2015 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 3032–3037.

[4] R. S. Patwardhan, S. L. Shah, Issues in performance diagnostics of model-based controllers, Journal of Process Control 12 (3) (2002) 413–427.

[5] T. J. Harris, Assessment of control loop performance, The Canadian Journal of Chemical Engineering 67 (5) (1989) 856–861.

[6] N. Stanfelj, T. E. Marlin, J. F. MacGregor, Monitoring and diagnosing process control performance: the single-loop case, Industrial & Engineering Chemistry Research 32 (2) (1993) 301–314.

[7] P. Kesavan, J. H. Lee, Diagnostic tools for multivariable model-based control systems, Industrial & engineering chemistry research 36 (7) (1997) 2725–2738.

[8] F. Gustafsson, S. F. Graebe, Closed-loop performance monitoring in the presence of system changes and disturbances, Automatica 34 (11) (1998) 1311–1326.

[9] B. Huang, Multivariable model validation in the presence of time-variant disturbance dynamics, Chemical engineering science 55 (20) (2000) 4583–4595.

[10] M. Basseville, On-board component fault detection and isolation using the statistical local approach, Automatica 34 (11) (1998) 1391–1415.

[11] B. Huang, A. Malhotra, E. C. Tamayo, Model predictive control relevant identification and validation, Chemical Engineering Science 58 (11) (2003) 2389–2401.

[12] Q. Zhang, S.-Y. Li, Performance monitoring and diagnosis of multivariable model predictive control using statistical analysis, Chinese Journal of Chemical Engineering 14 (2) (2006) 207–215.

[13] J. S. Conner, D. E. Seborg, Assessing the need for process re-identification, Industrial & engineering chemistry research 44 (8) (2005) 2767–2775.

[14] A. S. Badwe, R. D. Gudi, R. S. Patwardhan, S. L. Shah, S. C. Patwardhan, Detection of model-plant mismatch in mpc applications, Journal of Process Control 19 (8) (2009) 1305–1313.

[15] C. A. Harrison, S. J. Qin, Discriminating between disturbance and process model mismatch in model predictive control, Journal of Process Control 19 (10) (2009) 1610–1616.

[16] A. S. Badwe, R. S. Patwardhan, S. L. Shah, S. C. Patwardhan, R. D. Gudi, Quantifying the impact of model-plant mismatch on controller performance, Journal of Process Control 20 (4) (2010) 408–425.

[17] Z. Sun, S. J. Qin, A. Singhal, L. Megan, Performance monitoring of model-predictive controllers via model residual assessment, Journal of Process Control 23 (4) (2013) 473–482.

[18] S. Yerramilli, A. K. Tangirala, Detection and diagnosis of model-plant mismatch in MIMO systems using plant-model ratio, IFAC-PapersOnLine 49 (1) (2016) 266–271.

[19] Q. Li, J. Whiteley, R. Rhinehart, A relative performance monitor for process controllers, International Journal of Adaptive Control and Signal Processing 17 (7-9) (2003) 685–708.

[20] Q. Lu, R. B. Gopaluni, M. G. Forbes, P. D. Loewen, J. U. Backström, G. A. Dumont, Model-plant mismatch detection with support vector machines, in: IFAC 2017 World Congress, Toulouse, France, Vol. 50, 2017, pp. 7993–7998.

[21] M. Gevers, A. S. Bazanella, X. Bombois, et al., Identification and the information matrix: How to get just sufficiently rich?, IEEE Transactions on Automatic Control 54 (12) (2009) 2828–2840.

[22] Y. A. Shardt, B. Huang, Closed-loop identification condition for armax models using routine operating data, Automatica 47 (7) (2011) 1534–1537.

[23] L. Ljung, System identification: Theory for the User, Springer, 1999.

[24] A. S. Badwe, S. C. Patwardhan, R. D. Gudi, Closed-loop identification using direct approach and high order ARX/GOBF-ARX models, Journal of Process Control 21 (7) (2011) 1056–1071.

[25] L. Ljung, B. Wahlberg, Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra, Advances in Applied Probability 24 (02) (1992) 412–440.

[26] Y. Zhu, H. Hjalmarsson, The Box–Jenkins Steiglitz–McBride algorithm, Automatica 65 (2016) 170–182.

[27] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (7) (2001) 1443–1471.