Model-Plant Mismatch Detection with Support Vector Machines ⋆

Qiugang Lu * **R. Bhushan Gopaluni** * **Michael G. Forbes** ** **Philip D. Loewen** * **Johan U. Backström** ** **Guy A. Dumont** *

* *University of British Columbia, Vancouver, Canada, V6T 1Z4, (qglu@chbe.ubc.ca;bhushan.gopaluni@ubc.ca;loew@math.ubc.ca;guyd@ece.ubc.ca).*
** *Honeywell Process Solutions, North Vancouver, Canada, V7J 3S4, (michael.forbes@honeywell.com;johan.backstrom@honeywell.com)*

Abstract: We propose 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 discriminating the MPM from noise model change. Another advantage of this approach is that it is applicable to routine operating data that may lack any external excitations. Theoretical derivations on the closed-loop identification method are provided in this paper, showing that it can furnish 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. We build an SVM model based on process and noise model estimates from training data to predict the occurrence of MPM in the testing data. An example on paper machine control is provided to verify the proposed MPM detection framework.

Keywords: Monitoring and performance assessment, Support vector machine, Model-plant mismatch, Closed-loop identification, Paper machine control

1. INTRODUCTION

Model-plant mismatch (MPM) is the main source of control performance degradation in industrial model-predictive controllers (MPCs). It 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 Wang and Baldea (2015). Significant MPM can lead to suboptimal control decisions, production loss, or even closed-loop instability, in which case a process model re-identification procedure is necessary. Thus from the perspectives of MPC safety, maintenance and operational improvement, it is imperative to examine the MPC performance, particularly the quality of the employed process model Botelho et al. (2016).

Interest in MPM has grown significantly over recent years, focused primarily on evaluation of MPM via model residual analysis Sun et al. (2013) or (partial) correlation analysis Badwe et al. (2009); Yousefi et al. (2015); Tsai et al. (2015). More recent advances include investigating the output autocovariance function Wang and Baldea (2015), using plant-model ratio and examining the nominal sensitivity function Yerramilli and Tangirala (2016). Despite these achievements, the MPM detection problem is not yet completely solved since most current methods exhibit certain restrictions such as inseparability of MPM from noise model change (Harrison and Qin (2009)) and relying on perturbations to the closed-loop system (Lu et al. (2017)). An MPM detection scheme must not be sensitive to a change in noise model for it may result in increased false alarms, and it should not require external perturbations as they will disturb the normal operation of industrial processes. An ideal solution to this problem will be applicable to routine operating data with no external excitation or setpoint changes and will be robust to noise model changes.

In this work, we propose a novel MPM detection approach that addresses the above two challenges. Our idea is inspired by the historical data based benchmarks used in controller performance monitoring Li et al. (2003). 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 are 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 with no MPM) and a "testing" stage. The training data serves as a benchmark, against which we evaluate the presence of MPM in the testing data. Specifically, we propose a novel method, based on closed-loop identification and support vector machine (SVM) classification, that can monitor MPM and noise change independently and thus can directly discriminate MPM from noise model change. The most striking benefit of our method is its suitability for situations where external excitations may not exist.

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

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

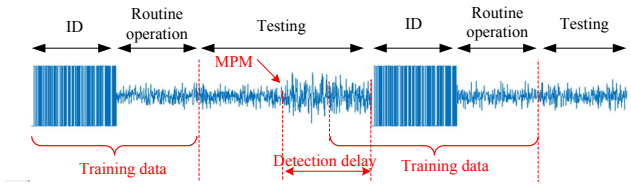

Fig. 1. Illustration of the training and testing data.

on introducing the MPM detection approach. The proposed method is based on a novel closed-loop identification algorithm which is capable of providing consistent estimates for the process model with routine operating data albeit with large variance. The inevitable variance associated with process model estimates impedes us from directly comparing our results with nominal models to identify the mismatch. In other words, discrepancies between model estimates and true plant cannot always be blamed on the MPM and they may simply be an artifact of the variance in the estimates. Thus we have to form a reasonable uncertainty bound around the estimated process model 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 current model. However, comparing the high-dimensional FIR coefficient vectors is non-trivial and it forms the motivation for using SVMs.

Fig. 1 demonstrates the proposed approach to detect the MPM. The industrial data is split into training data and testing data. The training data is 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 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. For the testing 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, the MPM alarm is not raised until we gather a large number of negative scores. 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.

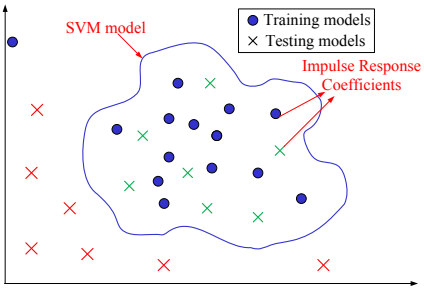

Fig. 2. Illustration of the MPM detection idea. Here the training and testing models refer to the process model estimates from training and testing data sets.

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 can provide a consistent estimate for the process model under weak conditions.

Consider the following single-input-single-output (SISO) Box-Jenkins plant

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

where G_0 and H_0 are the true plant and noise model, respectively. We suppose that H_0 is monic, stable and inversely stable. The plant $G_0(q)$ is assumed to 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 and varying), 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 $y^t = \{y(1), \ldots, y(t)\}$. Note that for the case with external excitations, a persistently exciting dither signal can always guarantee the informativeness of closed-loop data, regardless of the controller orders. However, without external excitations, for a linear controller to achieve the informativity requirement, the following relationship must be satisfied for Box-Jenkins models Gevers et al. (2009):

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

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. n_d denotes the order of the numerator polynomial in the noise model. One observation from (3) is

that more complex controllers and larger time-delays often imply richer information in the closed-loop data Gevers et al. (2009). Additionally, if the controller is nonlinear, as is the case of MPC (2), the closed-loop data is generally sufficiently exciting for relevant system identifications Ljung (1999). 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 true time-delay is known *a priori* and is specified to the identification algorithm.

The bias in direct closed-loop identification arises from mis-specification of the selected noise model structure. In order to circumvent this limitation, we propose to specify the noise model with an FIR structure. Specifically, we re-write the true process model in 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 using 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 $A_0(q) = 1 + a_1^0 q^{-1} + \ldots + a_n^0 q^{-n} + \ldots$, $B_0(q) = b_0^0 q^{-d} + b_1^0 q^{-d-1} + \ldots + b_n^0 q^{-d-n} + \ldots$, and d is the true time-delay. Our proposed closed-loop identification consists of two steps: high-order ARX modeling 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.

3.1 High-Order ARX Model

For open-loop stable systems, the coefficients of $A_0(q)$ tend to zero as n grows, and the same holds for $B_0(q)$. Thus we parameterize the above infinite-order ARX (5) by using a high-order ARX model

$$A(q,\eta_n)y(t) = B(q,\eta_n)u(t) + e(t), \tag{6}$$

where

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

and we denote $\eta_n = [a_1 \ \ldots \ a_n \ b_1 \ \ldots \ b_n]^T$, $\eta_0 = [a_1^0 \ \ldots \ a_n^0 \ \ldots \ b_1^0 \ \ldots \ b_n^0 \ \ldots]^T$. Note that for notational simplicity we choose $A(q,\eta_n)$ and $B(q,\eta_n)$ to be of the same order. We comment that to reduce the large parameter covariance resulting from high orders we add a regularization term. This technique is particularly useful for such purpose. For simplicity of theoretical derivations we assume that the order n of $A(q,\theta_n)$ in (6) increases to infinity as the number of samples $N \to \infty$ but n is smaller compared with N. Then we show that asymptotically, the estimate $A(e^{j\omega},\hat{\eta}_n)$ coincides with $A_0(e^{j\omega})$. Specifically, we have the following theorem regarding the estimation performance of the high-order ARX model.

Theorem 1. Suppose that the following relationship holds for the selected high-order ARX model structure (6):

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

where $\delta > 0$ is some constant. Then we have

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

where $\hat{\eta}_n = [\hat{a}_1 \ \ldots \ \hat{a}_n \ \hat{b}_1 \ \ldots \ \hat{b}_n]^T$ represents the estimate of η_n in (6) by the least-squares method.

Proof. The proof follows directly from Lemma 2.1 in Zhu and Hjalmarsson (2016) and is thus omitted here. ■

Remark 2. Theorem 1 states that allowing the order of 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. Theorem 1 provides a theoretical basis for the subsequent OE identification with filtered input and output. Note that $n \to \infty$ is merely an assumption for theoretical derivations, and indeed we can specify a (high-order) finite n based on prior information about the noise model when implementing this identification algorithm.

3.2 OE Identification with Filtered Input-Output Data

Replacing $H_0(q)$ in (4) with the estimated noise model inverse $A(e^{j\omega},\hat{\eta}_n)$ from (8), we arrive at the equation

$$y(t,\hat{\eta}_n) = G_0(q)u(t,\hat{\eta}_n) + e(t),$$

where $y(t,\eta)$ represents the signal filtered through $A(q,\eta)$ and $u(t,\eta)$ is defined similarly. To identify $G_0(q)$ we can impose an FIR structure on $G_0(q)$ if no *a priori* information is available. However, in typical industrial processes we have some knowledge to correctly parameterize the plant model. For example, in the machine-direction (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. With this in mind, we perform an OE identification

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

where $G(q,\rho)$ is a correctly selected plant model structure. The parameter ρ is estimated via minimizing the following criterion:

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

where Ω_ρ is a compact set. 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 3. Consider the true Box-Jenkins model for the plant (1) as well as the equivalent high-order ARX form (5). Assume that the 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

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

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. (Outline) Define $\varepsilon(t,\rho) := [G_0 - G(q,\rho)]u(t) + \frac{1}{A(q,\eta_0)}e(t)$, $V_N(\rho,\hat{\eta}_N) := \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}[A(q,\hat{\eta}_N)\varepsilon(t,\rho)]^2$ and $V_N(\rho,\eta_0) := \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}[A(q,\eta_0)\varepsilon(t,\rho)]^2$. Due to Theorem 2B.1 in Ljung (1999) and Theorem 1,

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

where $\overline{V}(\rho,\eta_0) := \overline{E}\frac{1}{2}[A(q,\eta_0)\varepsilon(t,\rho)]^2$. 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 is 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 Ljung (1999) and is thus omitted here. ■

Remark 4. Despite the premise on the correct parameterization of $G_0(q)$ in Theorem 3, it is not supposed to be a restrictive limitation on the proposed identification 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.

Compared with existing closed-loop identification methods, the proposed approach has the following merits:

- It requires no information about the controller and thus is applicable to closed-loop data with both linear and nonlinear controllers;
- It does not rely on external excitations. However, it is also suitable for cases with external excitations;
- A consistent estimate of the process model can be obtained even if *a priori* information about the noise model is not available.

Note that the explicit expression of variance of $\hat{\rho}_N$ is non-trivial. 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.

4. MPM DETECTION

4.1 One-Class Learning Support Vector Machines

As a convention, the SVM is developed particularly for the binary classification problem. The method provides a hyperplane that not only separates two classes of data, but also guarantees that it has the maximum distance to either class. 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. 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. 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 the data is mapped into. Consider the set of training data samples $\{x_1, x_2, \ldots, x_l\}$, $x_i \in \mathcal{X} \subset R^m, \forall i = 1, \ldots, l$, 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

$$\kappa(x,y) =< \Phi(x), \Phi(y) >, \quad \forall x, y \in \mathcal{X}. \tag{12}$$

A well-known kernel function that will be used hereafter is the Gaussian kernel $\kappa(x,y) = e^{-\|x-y\|^2/c}$, where c is a parameter used to tune the shape of the Gaussian kernel function. Note 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 Schölkopf et al. (2001)

$$\min_{w,\xi,b} \ \frac{1}{2}\|w\|^2 - b + \frac{1}{vl}\sum\nolimits_{i=1}^{l} \xi_i \tag{13}$$

$$s.t.\ w^T\Phi(x_i) \geq b - \xi_i,\ \xi_i \geq 0, \tag{14}$$

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 (13)-(14) can be converted into solving the following dual problem,

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\sum\nolimits_{i,j=1}^{l} \alpha_i\alpha_j\kappa(x_i,x_j) \tag{15}$$

$$s.t.\ 0 \leq \alpha_i \leq \frac{1}{vl}, \quad \sum\nolimits_{i=1}^{l} \alpha_i = 1. \tag{16}$$

Although the primal problem is formulated in the feature space, its dual problem can be resolved 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 (e.g., sequential minimal optimization) are available in the literature to solve this dual problem. A feature associated with the solution $\hat{\boldsymbol{\alpha}}$ 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,

$$h(x) = \sum\nolimits_{i=1}^{l} \hat{\alpha}_i\kappa(x_i,x) - \hat{b} \tag{17}$$

where x is a test example. Note that the sum in (17) 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 introduction of kernel functions significantly expands the flexibility of SVM in constructing separating boundaries, enabling it to generate a nonlinear classifier in the input space.

Remark 5. Compared with other types of classifiers (e.g. k-means), the advantage of one-class SVM for outlier detection is its suitability for high-dimensional data. More importantly, incorporating kernel functions to one-class SVM dramatically extends its flexibility by allowing for nonlinear separating boundaries for the benchmarking cluster. However, other types of enhanced one-class SVMs (e.g., support vector data description, SVDD) can also be implemented for the MPM detection presented in

this paper. We choose the one-class SVM as the tool to demonstrate our method.

4.2 Resampling

A critical issue for MPM detection with one-class SVM is the limited amount of training data. To address this problem, we propose to fit probability density functions (PDF) to estimated impulse response coefficients. Then a large number of samples can be generated by sampling randomly from the estimated density functions. More specifically, denote the FIR form of the estimated process model $G(q, \hat{\rho}_N)$ in the k-th moving window as

$$G(q, \hat{\rho}_N^k) = \hat{g}_0^k q^{-d} + \hat{g}_1^k q^{-d-1} + \ldots + \hat{g}_m^k q^{-d-m}, \quad (18)$$

where m is a pre-specified number. $k = 1, 2, \ldots, N_k$, 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 a Gaussian distribution (cf. Theorem 3). 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

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

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 more conservative than the true one. Thus we use a parameter α to tune the width of the PDF to avoid this problem. If we have limited training data, α shall be large and vice versa. We then generate a large number of samples of FIR coefficients according to the estimated PDFs. 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

Before MPM detection, we first need to estimate FIR coefficients of the process model identified from each moving window in the testing 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, the MPM alarm is not triggered until we have accumulated a sufficiently large number of mismatch reports. Specifically, define I_t as the sign of score value for time instant t

$$I_t = sign(h(x_t)), \quad (19)$$

with x_t being the FIR coefficient vector of the plant model estimate for the window data at time t. Denote $T_t = \{t - n_T, \ldots, t-1, t\}$ 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

$$s = |I_-|/n_T, \quad (20)$$

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 6. 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. Thus we can monitor process and noise models separately to distinguish MPM from noise model change. We comment that while the controllers considered 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.

5. EXAMPLE

In this section we demonstrate the MPM detection algorithm through a SISO example in the MD process of paper machines. The controlled variable (CV) is dry weight and the manipulated variable (MV) is stockflow. The sampling interval is set as 5 seconds. After discretization we obtain the true plant model $G_0 = 0.1003q^{-17}/(1 - 0.9048q^{-1})$. In the simulation of MD process the true noise model is selected as $H_0 = (1 - 0.3q^{-1})/(1 + 0.6q^{-1})$. 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 mismatch between the true plant and the process model employed in MPC. After 7 hours we change the true noise model into $H_0 = (1 + 0.3q^{-1})/(1 - 0.6q^{-1})$ to create a noise mismatch. Furthermore, we double the plant gain to introduce an MPM after 11 hours. 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. We summarize configurations of parameters relevant to the MPM detection algorithm in Table 1. 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 trends. Note that the first vertical red dash-dotted line indicates the time when 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 data. One can see that both noise change and MPM bring significant variations to the trends which are unfavored since the control objective is to keep the CV trend 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 as training data and the rest as testing data.

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. (17)) for noise and process model estimates, respectively. Clearly, the SVM scores drop to negative values after the cor-

Table 1. Parameters setup of the MPM detection algorithm

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

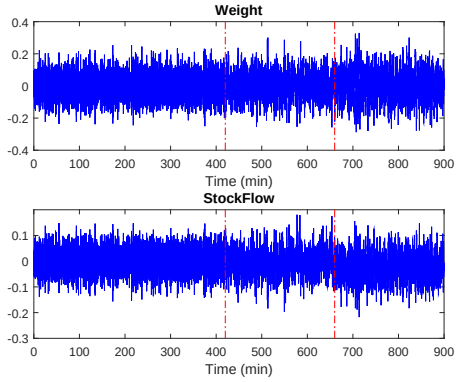

Fig. 3. Simulated input and output data

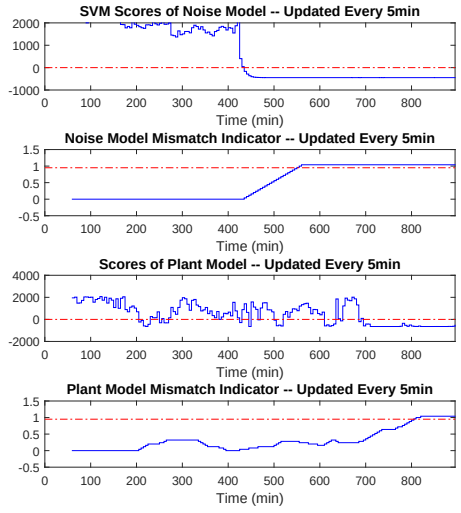

Fig. 4. MPM and noise change detection results

responding noise model change occurs. The second and fourth plots track the mismatch indicator values s in (20) 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. In this example we neglect the identification part. From Figure 4 it is clear that noise change at the 7-th hour does not affect the prediction of MPM. Thus we may conclude that the proposed 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.

6. 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 identification method that can give consistent process model estimates without the need for prior knowledge about the true noise model structure. We split the mismatch detection problem into training and testing stages, and in the training stage a one-class SVM model is developed based on process model estimates. The trained SVM model is then used to detect the MPM in the testing data. With the same procedures we can train another SVM model for noise models to detect noise model changes. This technique is tailored well enough to meet the industrial demand on MPM monitoring. An example on paper machines is presented to illustrate the effectiveness of the proposed method.

REFERENCES

Badwe, A.S., Gudi, R.D., Patwardhan, R.S., Shah, S.L., and Patwardhan, S.C. (2009). Detection of model-plant mismatch in MPC applications. *J. Process Contr.*, 19(8), 1305–1313.

Botelho, V., Trierweiler, J.O., Farenzena, M., and Duraiski, R. (2016). Perspectives and challenges in performance assessment of model predictive control. *Can. J Chem. Eng.*, 94(7), 1225–1241.

Gevers, M., Bazanella, A.S., Bombois, X., and Miskovic, L. (2009). Identification and the information matrix: how to get just sufficiently rich? *IEEE T Auto. Contr.*, 54(12), 2828–2840.

Harrison, C.A. and Qin, S.J. (2009). Discriminating between disturbance and process model mismatch in model predictive control. *J. Process Contr.*, 19(10), 1610–1616.

Li, Q., Whiteley, J., and Rhinehart, R. (2003). A relative performance monitor for process controllers. *Int. J. Adapt. Contrl.*, 17(7-9), 685–708.

Ljung, L. (1999). *System Identification: Theory for the User*. PTR Prentice Hall, Upper Saddle River, NJ.

Lu, Q., Forbes, M.G., Gopaluni, R.B., Loewen, P.D., Backström, J.U., and Dumont, G.A. (2017). Performance assessment of cross-directional control for paper machines. *IEEE Contrl. Syst. Technol.*, 25(1), 208–221.

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

Sun, Z., Qin, J., Singhal, A., and Megan, L. (2013). Performance monitoring of model-predictive controllers via model residual assessment. *J. Process Contr.*, 23(4), 473–482.

Tsai, Y., Gopaluni, R.B., Marshman, D.J., and Chmelyk, T. (2015). A novel algorithm for model-plant mismatch detection for model predictive controllers. In *Proceedings of International Symposium on Advanced Control of Chemical Processes (ADCHEM), Whistler, Canada*, 747 753.

Wang, S. and Baldea, M. (2015). Autocovariance-based MPC model mismatch estimation for SISO systems. In *2015 54th IEEE CDC*, 3032–3037.

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

Yousefi, M., Gopaluni, R.B., Loewen, P.D., Forbes, M.G., Dumont, G.A., and Backstrom, J.U. (2015). Impact of model plant mismatch on performance of control systems: An application to paper machine control. *Control Engineering Practice*, 43, 59–68.

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