12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Fault Detection via Autoencoder Latent Space Differences Between Reference Model and the Plant Operation ⋆

Enrique Luna Villagómez * **Hamidreza Mahyar** **
Vladimir Mahalec ***

* *McMaster University, Hamilton, ON L8S 4LS, Canada (e-mail: lunavile@mcmaster.ca)*
** *McMaster University, Hamilton, ON L8S 4LS, Canada (e-mail: mahyarh@mcmaster.ca)*
*** *McMaster University, Hamilton, ON L8S 4LS, Canada (e-mail: mahalec@mcmaster.ca)*

Abstract: Abnormal plant operations are caused by disturbances, process measurement faults, or malfunctioning equipment. Steady-state or dynamic models of the process units are widely available. Since continuous process plants operate under closed-loop control and available plant data often covers a narrow operating window, the process model can generate normal operating data over a wider window to train an autoencoder to represent that data. For deployment in real-time, the plant model accepts process inputs from the plant and calculates outputs; one instance of the autoencoder accepts data from the plant, and the other accepts data from the model. The occurrence of a process fault leads to differences in the latent space variables of the two instances of the autoencoder, which enables fault detection. Compared to a traditional PCA-based fault detection framework, an autoencoder-based framework can model nonlinear processes, which is not possible by using PCA or dynamic PCA.

Keywords: fault detection via autoencoder latent space, latent space differences between the plant and the reference model, reference model-based fault detection.

1. INTRODUCTION

Fault detection (FD) based on plant data has become preferred over the last two decades due to the large amount of available process data that is captured by real-time databases (Sun et al., 2020). Data-driven models of normal operating conditions (NOCs) that are the basis for such methods must properly represent the entire range of plant operations. Due to the widely spread implementation of model-based control, continuous process plants typically operate in a narrow region representing the desired state. Consequently, data-driven models of NOCs are only valid in narrow regions, which limits the applicability of data-driven FD methods.

There is also an abundance of first principles models developed for process design or real-time monitoring and optimization. Even though these models do not match the plants perfectly, they represent well the behavior of the process unit as it moves from one mode of operation to another, i.e., first principles models capture the nonlinear characteristics of the processes while typically exhibiting minor errors in predicting the properties of the products (e.g., for given operating conditions, the molar fraction of the product impurities predicted by the model could be 0.15 instead of 0.1 in the plant). In the plant, control loops maintain process variables at a desired value (e.g., 0.1 molar fraction of impurities); if one specifies in the model the same target value (molar fraction of impurities to be 0.1), the model will adjust some manipulated variable to achieve the desired output. This typically results in a mismatch between the model and the plant. Despite the mismatch, the model can be used to generate (somewhat inaccurate) NOC data that can be used to develop a reduced space model of the plant that can detect the process faults in real-time, as introduced by this work.

⋆ We are deeply grateful for the financial support provided by the Ontario Research Fund and the McMaster Advanced Control Consortium (MACC).

2. REFERENCE MODEL BASED FAULT DETECTION

2.1 Fault Detection Architecture

A fault detection algorithm should be accurate regardless of the plant operating in one or more modes, even if the plant behaves nonlinearly. Since the transition from one mode to another takes place via setpoint changes, the algorithm should not generate false positives if there are setpoint changes. We assume that the reference model exhibits the same behavior as the plant; in that case, as the plant transitions from one mode to another, so will the reference model. This ensures that the performance of the proposed fault detection algorithm should not be impacted by the changes in the plant operating conditions.

If there are no faults occurring in the plant, the differences between the plant and the model should be zero or constant if there is a plant-model mismatch. In reality, there are always mismatches between the plant and the model, resulting in variables measured in the plant being different from those computed by the model. Direct comparison of process variables between the model and the plant becomes cumbersome, and detection of faults based on such a comparison, like FD via parity-equations (Isermann, 2005), becomes nontrivial since the differences may be due to the model mismatch or due to the faults or both.

In order to use process models and avoid the above difficulties, we introduce a fault detection architecture where variables from the model (which reproduces plant trends but has mismatches to the plant) and variables from the plant are processed in real-time by the two instances of the same autoencoder. The occurrence of faults is detected via differences in latent variables between these two instances of the autoencoder.

Fig. 1 depicts the proposed algorithm. The reference model of a given process unit generates steady-state NOC data over a wide feasible operating range, which includes all anticipated modes of operation if the process can operate in several modes. These data are then used to develop an unsupervised reduced dimensionality model of the process (URDM), e.g., PCA, KPCA, or autoencoders.

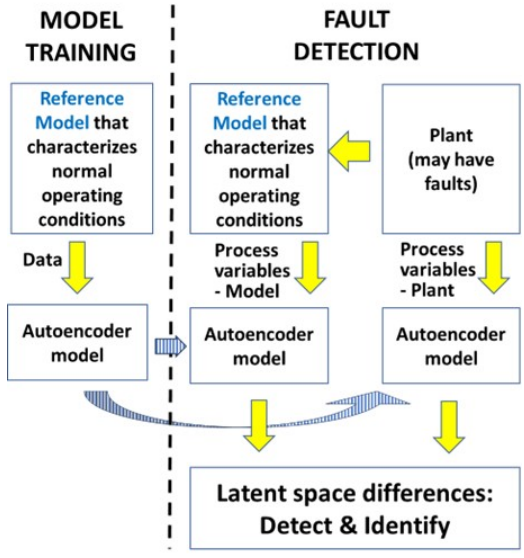

Fig. 1. Fault detection via differences in the autoencoder latent spaces resulting from the model and plant inputs.

In exploring autoassociative networks, Kramer (1992) demonstrates the efficacy of leveraging steady-state relationships among variables for fault detection in processes that are not in steady-state. Consequently, when the reference model has a dynamic component, utilizing only steady-state data for the development of the URDM becomes particularly pertinent. This work uses an autoencoder to represent the plant reduced model since it can deal with linear and nonlinear processes (Bourlard and Kabil, 2022).

2.2 Autoencoders

An autoencoder is an unsupervised learning model based on artificial neural networks (ANNs) that takes as its inputs all process variables that describe the state of the process and its inputs and outputs. It consists of (i) an encoder, which projects the input data into a reduced dimensionality space (latent space), and (ii) a decoder which reconstructs the autoencoder input data. Each part (encoder and decoder) can have multiple layers, as needed, to represent the process behavior. Hence, an autoencoder reproduces the training data set (distribution of the variables); it does not establish the functional dependency between the process variables.

In order to determine the autoencoder parameters, the model is trained using backpropagation to minimize the squared error loss:

$$L_s = \|x - f(g(x))\|_2^2, \tag{1}$$

where g and f stand for the encoder and decoder ANNs, respectively.

Current implementations of the autoencoder architecture include several regularization terms in the loss function to avoid overfitting or to shape the latent space in a particular manner (Bao et al., 2020). In this paper, in alignment with Cacciarelli and Kulahci (2022), we adapted the autoencoder loss function to ensure the generation of pseudo-uncorrelated features in the latent space:

$$L_s = \|x - f(g(x))\|_2^2 + \lambda \|zz^T - I\|_F^2, \tag{2}$$

where z is the latent representation of x.

3. TEST BED

3.1 CSTR benchmark

The CSTR example studied by Yoon and MacGregor (2001) has been selected as a benchmark to evaluate the performance of the proposed fault detection approach. The reactor (see Fig. 2) has two manipulated variables (the flow of coolant and the flow of reactant) and two controlled variables (temperature and concentration in the reactor).

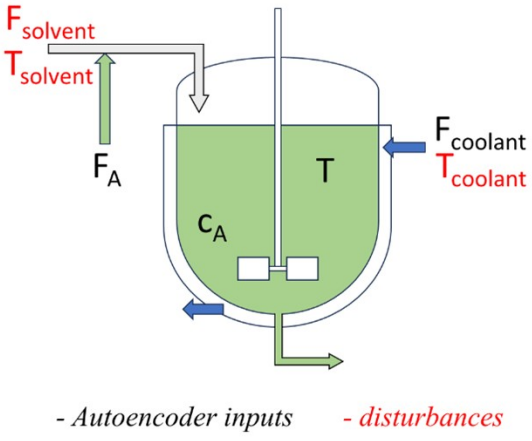

Fig. 2. CSTR benchmark.

The nonlinearity of the process has been assessed through the Pearson correlation coefficient. Given that many pairs in the correlation matrix exhibit relatively low correlations, the selection of an autoencoder as the URDM seems appropriate, as shown in Table 1.

Table 2 presents the parameters and normal operating conditions of this reactor. Disturbances in the process arise from the solvent's flow and temperature, and the coolant's temperature, which are affected by measurement noise and the tuning of flow controllers. Large changes in the solvent flow (F_S) have also been tested to evaluate FD performance when the process moves to a new operating region.

Training and testing data have been generated by changing the manipulated variables over a wide range and capturing the steady-state values. The autoencoder inputs are four variables: flow of reactant (F_A), flow of the coolant (F_C), reactor temperature (T), and concentration of A in the reactor (C_A).

Table 1. Correlation matrix of manipulated and controlled variables in the CSTR

	F_A	F_C	C_A	T	Mean	SD
F_A	1.00				0.20	0.09
F_C	0.00	1.00			14.98	5.44
C_A	0.97	0.18	1.00		1.30	0.44
T	0.82	-0.50	0.68	1.00	97.98	2.79

Table 2. CSTR Normal operating conditions

Variable	Description	Value
T_C	Coolant temperature	91.85 °C
T_f	Feed temperature	96.85 °C
C_{AA}	A's conc. in solute	19.1 kmol/m^3
C_{AS}	A's conc. in solvent	0.1 kmol/m^3
F_S	Solvent flow	0.9 m^3/min
F_A	Pure A flow	0.1 m^3/min
F_C	Coolant flow	0.9 m^3/min
C_A	A's conc. in reactor	0.8 kmol/m^3
T	Reactor's temp.	95.1 °C
α	Catalyst activity	1.0
ϕ	Fouling coefficient	1.0

3.2 Autoencoder architecture

The autoencoder structure (number of layers, number of nodes, activation function, and other hyperparameters) has been optimized using Optuna. This open-source hyperparameter optimization framework uses the Tree-Structure Parzen Estimator (TPE) (Bergstra et al., 2011) to perform the hyperparameter tuning. The number of nodes in each layer of the autoencoder used in this work is shown in Fig. 3.

4. FAULT DETECTION - SLOWLY INCREASING FAULTS

Slowly increasing faults in plant operation may be, e.g., reduction of heat transfer rates due to fouling or catalyst deactivation due to decay. Such faults have been simulated as ramps corresponding to the fault increasing from zero to a target value over a given period.

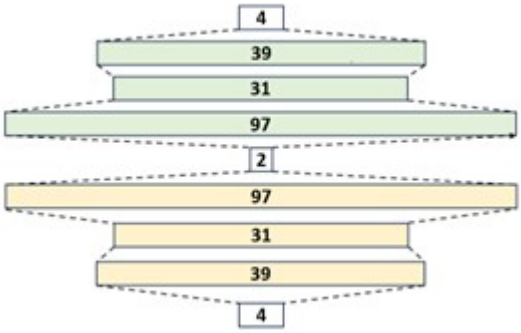

Fig. 3. Autoencoder layers.

4.1 Fault detection when the model matches the plant

Fig. 4 shows the latent space responses of the plant and the model when there is no mismatch between the model and the plant. At time = 1,000 min, fouling starts taking place in the reactor. The fouling increases steadily (ramp) until time = 3,000 min. The latent space variables Z_1 and Z_2 corresponding to the plant and the model have the same value during the normal operation. Once the fault occurs, the plant latent variables decrease while the model latent variables remain unchanged. Fig. 4 also shows the differences

$$\delta_i = Z_i^m - Z_i^p, \tag{3}$$

where Z_i^m and Z_i^p represent the i^{th} latent variables of the reference model and plant, respectively. It can be seen that these differences equal zero as long as the operation is normal. Once a fault occurs, the differences are no longer equal to zero.

The occurrence of a fault manifests itself in the latent space differences departing from zero. The differences remain different from zero as long as the fault is present. If a plot of the differences is made available to a plant operator, the operator will be able to detect visually an occurrence of a fault and track its progress, which in turn will lead to improvements in the human/machine interfaces for fault detection.

If there is a significant level of noise, the latent space variables can be filtered so that the visual recognition of the trends is easier. Fig. 5 shows data from Fig. 4 after applying a simple moving average filter whose window size equals 40 observations.

4.2 Fault detection when the model does not match the plant

Process models built for the design or monitoring of plant operations are never completely accurate due to model parameters being estimated from engineering knowledge or plant data. For instance, a heat transfer coefficient varies with the flow of the cooling media through a reactor jacket or because there may be some fouling on the heat transfer wall. Fig. 6 shows an example where the heat transfer coefficient in the plant is 10% higher than in the model. At time = 1,000 min, fouling starts taking place in the reactor. The fouling increases steadily (ramp) until time = 3,000 min.

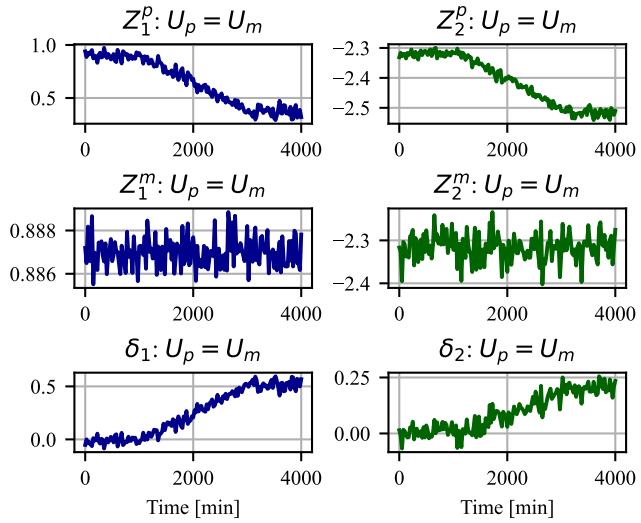

Fig. 4. Perfect model: The faults cause latent space differences to depart from zero.

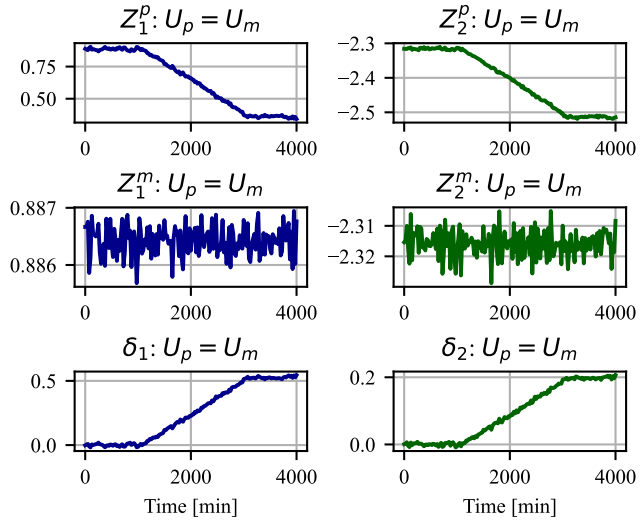

Fig. 5. Filtering of latent space variables leads to fault occurrences being easily noticeable via changes in their differences.

The latent space variables exhibit a pattern similar to the case when the model matches the plant perfectly. The magnitude of the differences is somewhat smaller, which can be discerned by comparing the mean values in Fig. 5 and Fig. 7.

In order to test fault detection capabilities under different errors in model parameters, several tests have been carried out at different levels of mismatch between the plant and the model. For example, when the actual heat transfer coefficient in the plant is larger (smaller) than the heat transfer coefficient in the model, the differences (as defined by Eq. 3) have a negative (positive) bias under the normal operating conditions as shown in Fig. 8.

Fig. 9 shows latent space differences at $U_p = U_m$, $U_p = 1.1U_m$ and $U_p = 0.85U_m$ after filtering the latent variables and adjusting the bias so that the differences are zero under NOCs.

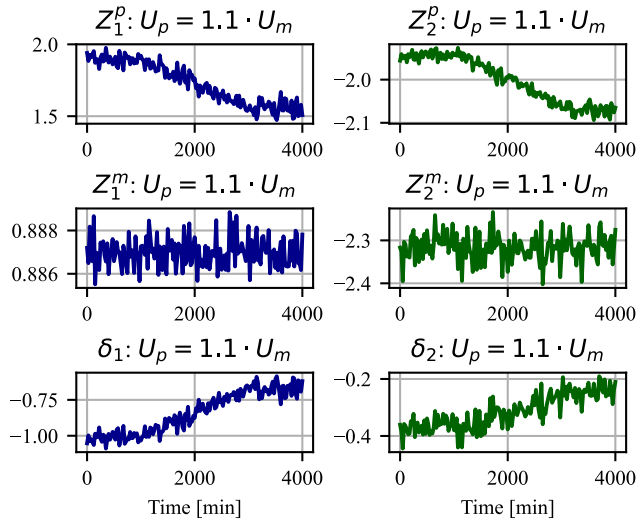

Fig. 6. Imperfect model: under NOC, the latent space differences are not zero; the faults change the magnitude of the differences.

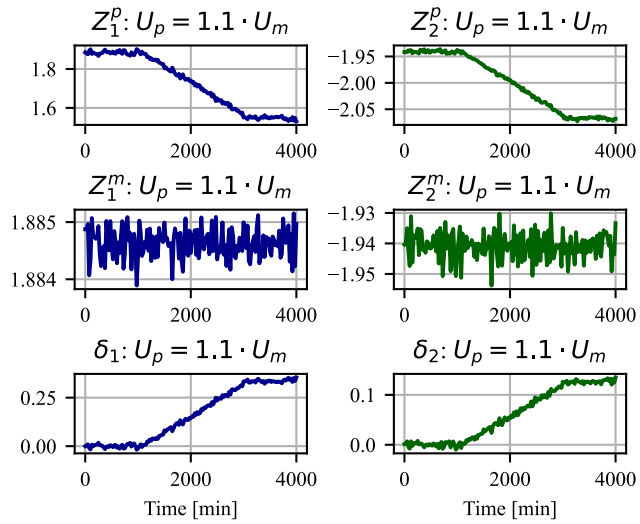

Fig. 7. Imperfect model: adjusting the bias so that under NOCs the differences are zero.

Notice that when U_p is greater than U_m, then the change of difference in the first latent variable δ_1 is greater than the change of difference in the second latent variable δ_2, while opposite is true if U_p is less than U_m. This opens a possibility that the behavior of the differences may offer insight into errors in the model parameters.

5. FAULT DETECTION – STEP CHANGE FAULTS

Examples of step change faults include e.g. an instrument failure or a catalyst poisoning. Presented here are the results of the reactor temperature sensor step change in its reading by 0.5 °C.

Fig. 10 shows the latent space variables and their differences for $U_p = 1.1U_m$. Fig. 11 illustrates the latent spaces differences at $U_p = U_m$), $U_p = 1.1U_m$) and $U_p = 0.85U_m$. The fault occurs at time = 1,000 min. These differences, filtered and bias adjusted so that under NOCs they are zero, are depicted in Fig. 12.

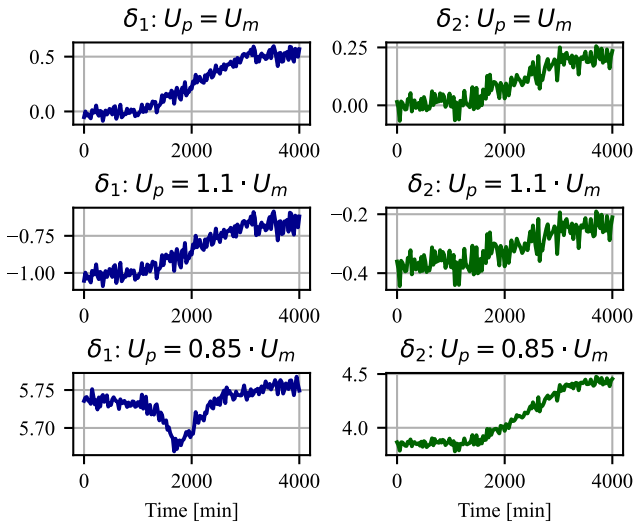

Fig. 8. Imperfect model: heat transfer fouling; latent space differences at different levels of heat transfer coefficient mismatch

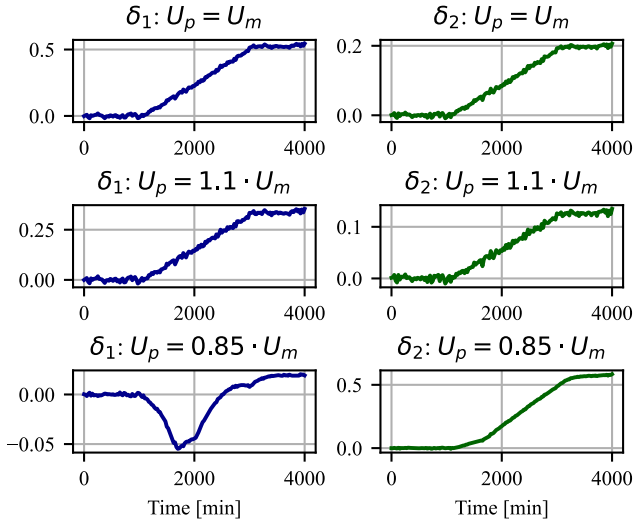

Fig. 9. Imperfect model: heat transfer fouling; latent space differences after filtering and bias adjustments.

Analogously to the case when fault was due to the heat transfer fouling, when U_p is greater than U_m, the change of difference δ_1 in the first latent variable is greater than the change of difference in the second latent variable δ_2, while opposite is true if U_p is less than U_m.

6. NUMERICAL DETECTION OF THE FAULT OCCURANCE

In order to simplify fault detection, we calculate a single anomaly score using the squared Euclidean distance, which encapsulates the latent space differences between the plant and its reference model:

$$\eta(j) = \sum_i (Z_i^m(j) - Z_i^p(j))^2, \tag{4}$$

where $Z_i^m(j)$ and $Z_i^p(j)$ represent the i^{th} latent variables of the reference model and plant at the j^{th} observation, respectively.

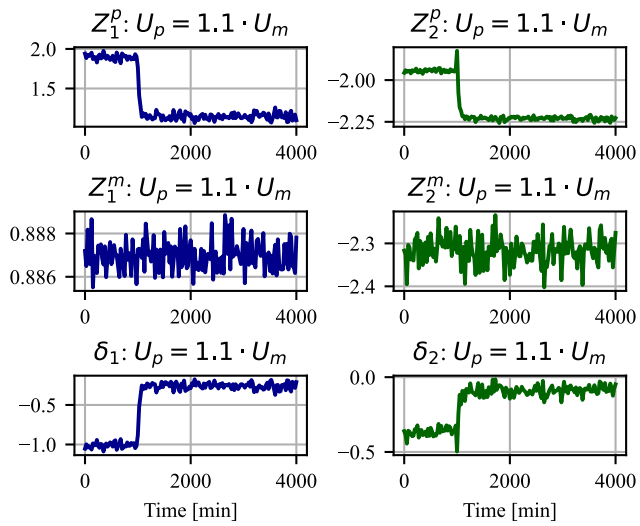

Fig. 10. Imperfect model: temperature sensor failure.

To determine the upper control limit for the anomaly score, we use Kernel Density Estimation (KDE) to estimate the distribution of the score under normal operation, and then we set the upper control limit as the 0.9999 quantile of the estimated density. In plant operations where process noise is prevalent, outliers not necessarily related to faulty operation can be labeled as one. Using this statistical approach to establish the anomaly score control limits can lead to such mislabeling. Therefore, it is a common practice to raise the fault alarm once there are N consecutive anomaly scores above the control limit. In this study, we define this tolerance as fifteen consecutive observations, which correspond to 7.5 minutes.

To evaluate the efficacy of the anomaly score, we computed a ratio named ψ, which is defined as follows:

$$\psi_i = |Z_i^p(t_d) - \mu_i^p)|/3\sigma_i^p. \tag{5}$$

In eq. 5, $Z_i^p(t_d)$ represents the value of the i^{th} latent variable of the plant at the detection time t_d. The symbols μ_i^p and σ_i^p denote, respectively, the mean and standard deviation of the i^{th} latent variable of the plant under normal operating conditions. This ratio assesses the effectiveness of the Euclidean-based anomaly score (η) in detecting anomalous behavior compared to tracking individual variable differences. A ratio value below 1.0 indicates that the fault was detected before either of the first two latent variables (Z_1 and Z_1) deviated beyond three standard deviations (3σ) from their normal operation distribution. Table 3 displays this metric for various cases, alongside their respective fault detection times. Sensor bias fault

Table 3. Fault detection times and ψ_i ratios corresponding to the cases addressed in this study.

Case	**Fault**	t_d **[min]**	ψ_1	ψ_2
No mismatch	fouling	102.5	0.3959	0.1302
No mismatch	sensor bias	8.5	3.6286	3.5323
$U_p = 1.1 \cdot U_m$	fouling	150.5	0.4317	0.5363
$U_p = 1.1 \cdot U_m$	sensor bias	9.0	4.1459	3.1403
$U_p = 0.85 \cdot U_m$	fouling	442.5	7.6475	4.3712
$U_p = 0.85 \cdot U_m$	sensor bias	12.5	30.5024	10.5415

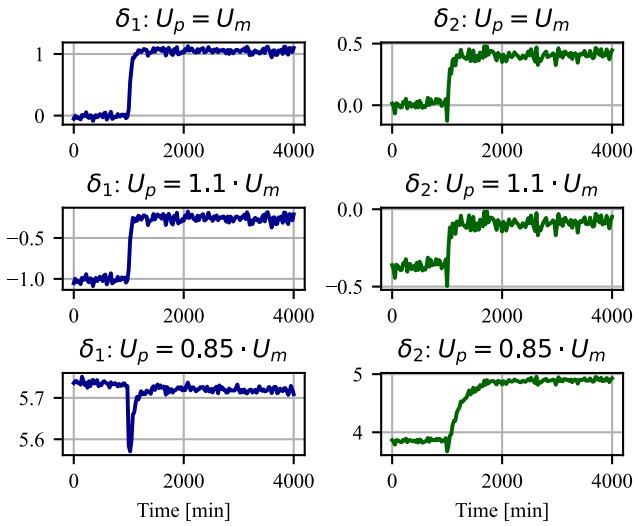

Fig. 11. Imperfect model: temperature sensor failure; latent space differences at different levels of heat transfer coefficient mismatch.

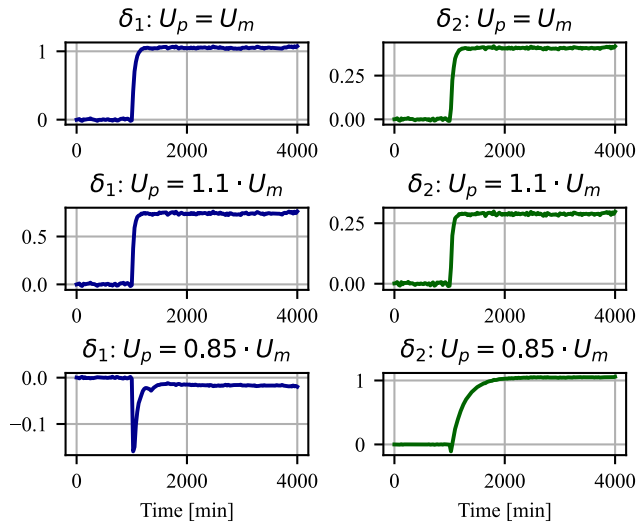

Fig. 12. Imperfect model: temperature sensor failure; latent space differences, filtered and bias adjusted, at different levels of heat transfer coefficient mismatch.

(step change) exhibits high values of ψ_i since the fault is declared only if 15 consecutive readings are above the tolerance, during which period the step fault consequences become very pronounced.

Results in Table 3 show that even though the fault detection times increase if there is a significant plant-model mismatch, the proposed fault detection methodology performs well under plant-model mismatch.

7. CONCLUSIONS

The fault detection architecture presented in this work leverages process models developed by using commercial modelling and simulation software. The parallel configuration (simulation model-plant) can accommodate known events, e.g. changes to the setpoints of the reactor temperature or concentration, since such changes will occur in the plant and in the reference model, thereby ensuring minimal discrepancies between the reference model predictions and the plant observations.

The method is particularly effective for slowly increasing faults (e.g. fouling); if there is no plant-model mismatch, the detection occurs at low values of the signal/noise ratio (eq. 5). In scenarios involving abrupt (step) faults, the proposed method performance remains consistently high across different levels of mismatch. In addition, it is shown that even if the plant model has significant mismatches relative to the plant, an autoencoder model trained on data generated from an imperfect model can detect the fault-induced latent space differences between the model and the plant.

The patterns of the latent space differences indicate that they can be used to determine the nature of the parameter mismatch between the model and the plant. This hypothesis needs to be further explored and verified, which is a part of our current research.

Since the methodology performs well under plant-model mismatch, there is a possibility that one can develop autoencoder representations for typical equipment types (e.g. two product distillation tower, countercurrent heat exchanger, plug flow reactor, CSTR, etc.) and use them for fault detection via the proposed architecture. This is also the subject of our current research.

REFERENCES

Bao, X., Lucas, J., Sachdeva, S., and Grosse, R.B. (2020). Regularized linear autoencoders recover the principal components, eventually. *Advances in Neural Information Processing Systems*.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

Bourlard, H. and Kabil, S.H. (2022). Autoencoders reloaded. *Biological Cybernetics*, 116, 389–406.

Cacciarelli, D. and Kulahci, M. (2022). A novel fault detection and diagnosis approach based on orthogonal autoencoders. *Computers Chemical Engineering*, 163, 107853.

Isermann, R. (2005). Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control*, 29, 71–85.

Kramer, M. (1992). Autoassociative neural networks. *Computers Chemical Engineering*, 16, 313–328.

Sun, W., Paiva, A.R., Xu, P., Sundaram, A., and Braatz, R.D. (2020). Fault detection and identification using bayesian recurrent neural networks. *Computers Chemical Engineering*, 141, 106991.

Yoon, S. and MacGregor, J.F. (2001). Fault diagnosis with multivariate statistical models part i: using steady state fault signatures. *Journal of Process Control*, 11, 387–400.