Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Integrating Dynamic Neural Network Models with Principal Component Analysis for Model Predictive Control ⋆

Hesam Hassanpour, Brandon Corbett, Prashant Mhaskar *

* *Department of Chemical Engineering, McMaster University, Hamilton, ON L8S 4L7, Canada (e-mail: mhaskar@mcmaster.ca)*

Abstract: This work addresses the problem of identifying models using process data with possibly correlated manipulated variables for model predictive control (MPC) design. The key idea is to use principal component analysis (PCA) to reject the redundancy in the input space and utilize scores to build the dynamic model of the system using recurrent neural networks (RNN). The identified PCA-RNN model is then used in the MPC optimization problem, calculating the optimal scores. The control actions are computed using the loadings of the PCA model. The efficacy of the proposed approach is evaluated using a chemical reactor example. The results are compared with a base-case scenario where the data is directly used to build a dynamic neural network model and used as part of a model predictive control implementation. The simulation results show the superiority of the proposed integrated PCA-RNN models for model predictive control.

Keywords: Principal component analysis, Recurrent neural networks, Model identification, Model predictive control.

1. INTRODUCTION

Automation systems play an increasingly significant role in process operations. Given the challenges of nonlinearity, multi-variable interactions, and process constraints, the design of controllers to handle these challenges is a critical task. Advanced process control strategies like model predictive control (MPC) have been receiving significant attention due to their ability to deal with process constraints and economic objectives. Addressing the problem of system identification, to build a reliable and accurate process model as part of an MPC implementation, is critical to the success of MPC implementations. Given the increased availability of process data, and the challenges associated with developing and maintaining a first principles model, there is significant interest in using data-driven approaches for identifying dynamic models.

The ability of a model to predict the future behavior of a system can be strongly influenced by types of datasets and the conditions under which they are generated. One approach to generate data is perturbing the process inputs using pseudo-random binary sequence (PRBS) signals which are generally uncorrelated (see e.g., Shariff et al. (2013)). Unstable open-loop systems and economic concerns have motivated the use of closed-loop identification techniques (see e.g., Forssell and Ljung (1999) and Qin and Ljung (2003)). Several approaches address the problem of closed-loop identification using data obtained without excitation of the reference signals for processes with sufficiently large time delay or adequately small sampling time (see e.g., Shardt and Huang (2011)). In terms of fast processes, other works consider using step changes to excite the reference signals for closed-loop identification (see e.g., Shardt et al. (2015)). In essence, these techniques rely on the inadequacy of the closed-loop controller to tease out the model dynamics from closed-loop data.

Machine learning (ML) techniques have received significant attention for modeling and control of nonlinear systems. Several approaches consider using subspace identification methods for modeling process dynamics to be employed in model predictive control (see e.g., Corbett and Mhaskar (2016) and Kheradmandi and Mhaskar (2018)). Latent variable-based techniques are also considered to develop latent variable MPC algorithms (see e.g., Flores-Cerrillo and MacGregor (2005) and Golshan et al. (2010)). Artificial neural networks (ANN) are recently being reconsidered to model and control nonlinear systems (see Nikravesh et al. (2000), Kittisupakorn et al. (2009), Wu et al. (2019a), and Wu et al. (2019b))). However, because of the large number of parameters, a significant amount of data is required sometimes to achieve a reliable model, and the problem of reliably avoiding overfitting remains a challenge, especially in the context of closed-loop implementations. One approach to avoid overfitting is to possibly pre-process the data to remove any co-linearity present in inputs in the training data, or alternatively, to ensure that the co-linearity is maintained in the MPC operation to prevent the problem of the process moving significantly away from the training data set.

Motivated by above considerations, the present manuscript addresses the problem of handling co-linearity in the input space by using a combination of PCA and RNN to model

⋆ Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the McMaster Advanced Control Consortium (MACC) is gratefully acknowledged.

the process. The PCA-RNN model is then embedded in MPC framework to ensure that the control moves computed by the MPC remain in the same space as the training data set. The rest of the manuscript is organized as follows: In Section 2, a brief description of the simulation testbed is presented. Then, existing approaches for RNN-based model identification, principal component analysis, and RNN-based MPC design are reviewed, and implemented on the motivating example to illustrate the specific problem being addressed. In Section 3, the proposed approach for model identification and MPC design is presented. In Section 4, the efficacy of proposed approach is illustrated by application to the motivating example. Finally, concluding remarks are presented in Section 5.

2. PRELIMINARIES

In this section, a Continuous Stirred Tank Reactor (CSTR) example is first reviewed to provide the motivation for proposed research. Afterwards, a brief review of RNN-based modeling and PCA is presented. Finally, a representative RNN-based MPC is formulated and implemented on the motivating example.

2.1 Motivating Example: CSTR

Consider the continuous-stirred tank reactor (CSTR) example (Du and Mhaskar (2014)), in which an irreversible elementary exothermic reaction (A $\rightarrow$ B) occurs. The mathematical model of the CSTR example takes the following form:

$$\begin{aligned}\dot{C_A} =& \frac{F}{V}(C_{A0} - C_A) - k_0 e^{-E/RT_R} C_A \\ \dot{T_R} =& \frac{F}{V}(T_0 - T_R) + \frac{(-\Delta H)}{\rho c_p} k_0 e^{-E/RT_R} C_A - \frac{UA}{\rho c_p V}(T_R - T_c) \\ \dot{T_c} =& \frac{F_c}{V_c}(T_{cf} - T_c) + \frac{UA}{\rho_c c_{pc} V_c}(T_R - T_c)\end{aligned} \tag{1}$$

where C_A, T_R, and T_c are concentration of reactant A, the temperature in the reactor, and the temperature in the cooling jacket, respectively. F represents the feed flow rate to the reactor with temperature T_0 and concentration C_{A0} and F_c is the cooling stream flow rate to the jacket with temperature T_{cf}. The values of process parameters and the steady-state values of the variables are listed in Table 1.

Table 1. Parameter and steady-state values for the CSTR example

$V = 100$ L	$c_{pc} = 4.2$ J/(g K)
$k_0 = 7.2 \times 10^{10}$ min^{-1}	$C_{A0} = 1$ mol/L
$E/R = 8750$ K	$T_0 = 350$ K
$\Delta H = -3 \times 10^3$ J/mol	$C_{A_s} = 0.86$ mol/L
$\rho = 1000$ g/L	$T_{R_s} = 304.6$ K
$c_p = 0.239$ J/(g K)	$T_{c_s} = 301.3$ K
$UA = 5 \times 10^4$ J/(min K)	$F_s = 14.6$ L/min
$V_c = 20$ L	$F_{c_s} = 4.7$ L/min
$\rho_c = 1000$ g/L	$T_{cf_s} = 293$ K

The CSTR example is used to represent one instance of a control structure where data may be available from process operation with more than one manipulated input being used to control the same output. Thus, in this example, the reactor and cooling jacket temperatures are controlled by manipulating feed flow rate to the reactor (F), the cooling stream flow rate to the jacket (F_c), and the cooling stream inlet temperature (T_{cf}) by using PI controllers. The feed flow rate (F) is manipulated to control the reactor temperature (T_R) in the first control loop and the cooling stream flow rate and its inlet temperature (F_c and T_{cf}) are manipulated in the second control loop to maintain the jacket temperature (T_c). Consider PI controller defined by equation (2), the values of the tuning parameters (proportional and integral gains) for two PI controllers are listed in Table 2.

$$U(t) = K\left(K_c E(t) + \frac{K_c}{T_I}\int_0^t E(t')dt'\right) \tag{2}$$

It should be noted that the vector $K = [k_1 \;\; k_2]^T$ in equation (2) is considered to apply different degrees of the effectiveness on the manipulated variables in the second control loop, where two manipulated inputs are considered to control one controlled variable, otherwise for the first loop, where one manipulated input is considered to control one controlled variable, this value should be set to one. The values of k_1 and k_2 are used as -2 and 1, respectively for manipulated inputs (F_c) and (T_{cf}) in the second loop; thus $K = [-2 \; 1]^T$.

Table 2. Tuning parameters for PI controllers

$K_{c_1} = 2.60$	$K_{c_2} = 0.86$
$T_{I_1} = 0.66$	$T_{I_2} = 1.41$

Instances of process data are generated by simulating the CSTR example in MATLAB. Specifically, the data for training and validation is generated by applying set-point changes in T_R, T_c for PI controllers. The input-output training data is shown in Fig. 1 and Fig. 2. Note that the manipulated inputs are represented by the deviation variables, $\bar{F} = F - F_s$, $\bar{F}_c = F_c - F_{c_s}$, and $\bar{T}_{cf} = T_{cf} - T_{cf_s}$, and the outputs are represented by the deviation variables, $\bar{T}_R = T_R - T_{R_s}$ and $\bar{T}_c = T_c - T_{c_s}$.

2.2 Artificial neural networks for system identification

As mentioned earlier, artificial neural networks (ANN) have recently received significant attention in the area of dynamical modeling of the processes. A particular type of ANN, designed to model a dynamic system, is referred to as recurrent neural network (RNN). Unlike feedforward neural networks, the existence of delayed feedback loops in the RNN structures give the model memory/the ability to model dynamic behavior. Different types of the RNN such as nonlinear autoregressive network with exogenous inputs (NARX), layer recurrent network (LRN), long short-term memory (LSTM) have been developed. It this work, the NARX networks are utilized. The NARX model can be mathematically expressed as follows:

$$\begin{aligned}y(\tau + T_s) = f[&y(\tau), y(\tau - T_s), ..., y(\tau - L_y T_s), \\ &u(\tau), u(\tau - T_s), ..., u(\tau - L_u T_s)]\end{aligned} \tag{3}$$

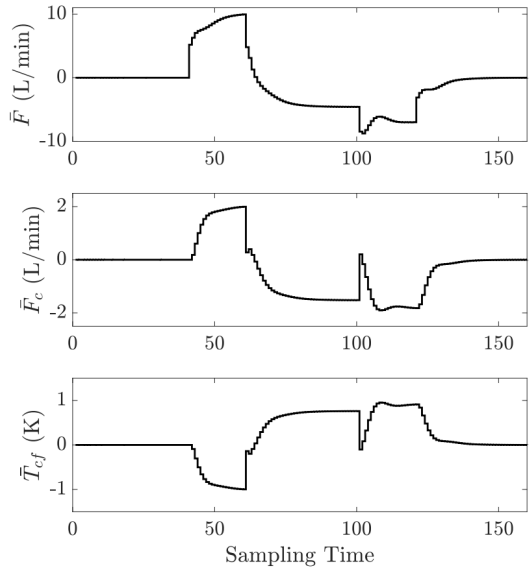

Fig. 1. Model training data: manipulated variables.

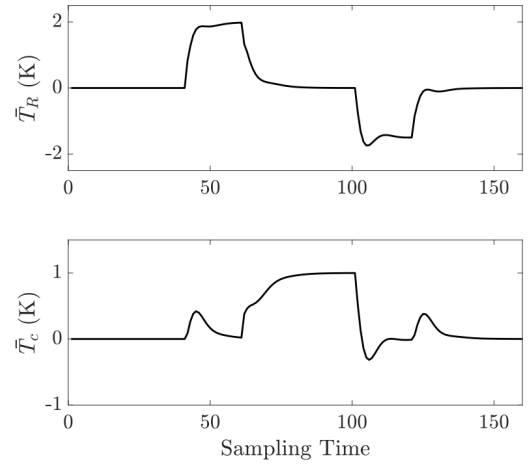

Fig. 2. Model training data: measured outputs.

where y $\in \mathbb{R}^{n_y}$ and u $\in \mathbb{R}^{n_u}$ represent the vector of measured outputs and inputs of the system, respectively. τ, L_y, and L_u represent time, the number of lagged outputs, and the number of lagged inputs required for prediction. As can be noticed, the next values of dependent variables are regressed on the past and current values of dependent variables and independent (exogenous) variables by NARX models. A multilayer perceptron (MLP), a class of feedforward artificial neural network, with the embedded memory can be employed to approximate the nonlinear function 'f' in equation (3). The resulting model is referred to as a NARX recurrent dynamic neural network (see e.g., Narendra and Parthasarathy (1990)). In the present manuscript, we use the above RNN structure to illustrate the key idea of the proposed approach.

The RNN model is thus employed to identify a model for the chemical reactor example. To this end, the closed-loop training data, generated considering the set-point changes in the process, is utilized to train the RNN model. The data samples are normalized and then fed to recurrent neural network (NARX) in MATLAB Machine Learning and Deep Learning toolbox. One hidden layer with 5 neurons, using hyperbolic tangent activation function ($tanh(x) = 2/(1 + e^{-2x}) - 1$), and a delay of 1 is selected. Linear activation function is considered for the output layers and Bayesian regularization backpropagation algorithm, updating the bias and weight values based on Levenberg-Marquardt optimization, is chosen to determine these values in the model. The capability of the RNN model to capture the system dynamics and predict the process outputs is examined in the validation stage. The output profiles (dash-dotted lines) are shown in Fig. 3. As can be seen, the RNN model is able to predict the future behavior of the process with reasonable accuracy.

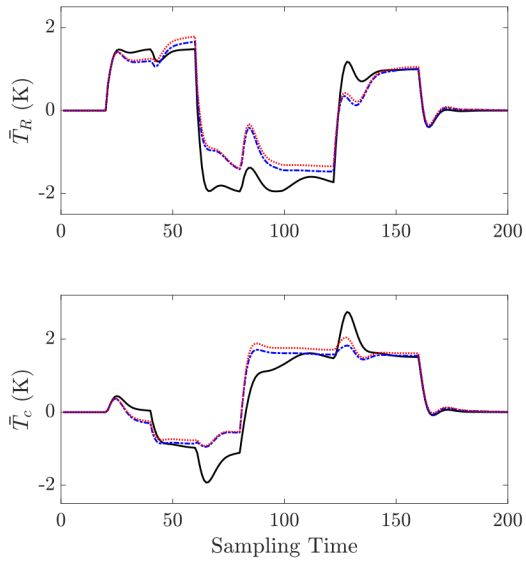

Fig. 3. Model validation results: validation data (solid lines), prediction obtained based on the RNN model (dash-dotted lines), prediction obtained based on the PCA-RNN model (dotted lines).

2.3 Review of principal component analysis

In this section, we review some of the fundamental concepts behind one of the most popular latent variable methods, principal component analysis (PCA). The objective of PCA is to transform a set of observations of possibly correlated variables into a new space (latent space) in which the values of new variables (principal components) are uncorrelated. The mathematical representation of PCA method can be stated as follows in equation (4):

$$X = TP^T + E \tag{4}$$

where X is a matrix with columns and rows containing the process variables and their observations, respectively. T is a score matrix representing the new mutually uncorrelated variables (principal components) by columns and projected values of the observations represented by rows, and P is a loading matrix, containing orthogonal loading vectors, determines the orientation of the latent space. The principal components are ordered so that the largest amount of variation in the data is described by the first and the last one captures the least variation in the data.

In order to apply PCA algorithm, the data matrix should be mean-centered and scaled such that all variables have zero mean and unit variance. Then, orthogonalization techniques such as singular value decomposition (SVD) or nonlinear iterative partial least squares (NIPALS) can be utilized to compute the principal components. Cross validation techniques (see e.g., Wold (1978) and Eastment and Krzanowski (1982)) can be applied to determine the number of proper principal components.

2.4 RNN based model predictive control (MPC)

A representative RNN-based MPC formulation is described below:

$$\begin{aligned} \min_{u_k,\ldots,u_{k+P}} \quad & \sum_{j=1}^{P} \|\tilde{y}_{k+j} - y^{SP}_{k+j}\|^2_{Q_{y_r}} + \|u_{k+j} - u_{k+j-1}\|^2_{R_{du}} \\ s.t. \quad & RNN\ based\ Predictive\ Model, \\ & u_{min} \leq u \leq u_{max} \end{aligned} \tag{5}$$

where P represents the prediction horizon, y^{SP}_{k+j} is the desired set-point values for the outputs, and $\tilde{y}_{k+j}$ is the prediction of output at time $(k+j)T_s$ and $T_s = \Delta\tau$ is the sampling time. Q_{y_r} and R_{du} are penalty matrices corresponding to the output deviations from the set-points and the rate of change in the inputs. It should be noted that the RNN-based predictive model computes multi-step ahead predictions. In this direction, the initial values of inputs and outputs (lagged data) are fed to the model and then the values of the outputs are calculated in one step ahead. Afterwards, the predicted outputs, along with candidate future input values are fed to the model to predict the outputs two steps ahead. Subsequently, this procedure can perform recursively to predict the outputs multi steps ahead up to the prediction horizon.

Next, the above representative RNN-based MPC is implemented on the motivating example, using the RNN model, identified in the earlier section. Specifically, the RNN-based identified model of the process, along with the constraints on the manipulated inputs, where $F \in [0 - F_s \;\; 60 - F_s]$, $F_c \in [0 - F_{c_s} \;\; 10 - F_{c_s}]$, and $T_{cf} \in [289 - T_{cf_s} \;\; 298 - T_{cf_s}]$, are included in the RNN-based MPC represented by equation (5). The optimization problem for the RNN-based MPC is solved using *fmincon* solver in MATLAB and the parameters are chosen as follows: sampling time $T_s = 30$ s, prediction horizon $P = 6$, $Q_{y_r} = [1000\ 0;\ 0\ 1000]$, and $R_{du} = [1\ 0\ 0;\ 0\ 1\ 0;\ 0\ 0\ 1]$. Fig. 4 and Fig. 5 show the output and input profiles obtained by implementing the RNN-based MPC, respectively. As can be seen, this approach fails to track the set-point value for jacket temperature, where two correlated inputs (F_c) and (T_{cf}) are manipulated to achieve the desired set-point.

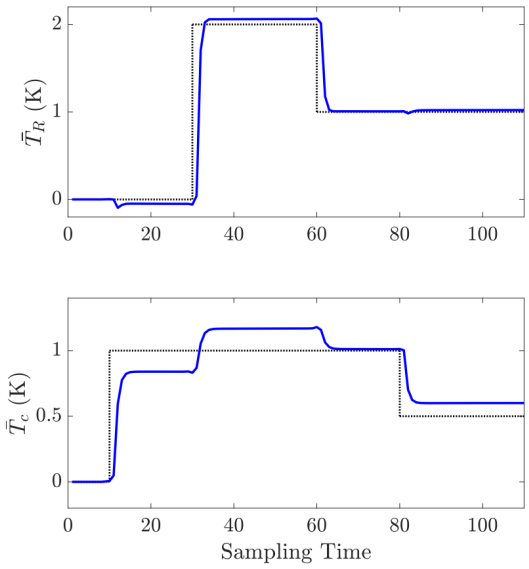

Fig. 4. The trajectories for the reactor and jacket temperatures obtained from the RNN-based MPC (solid lines) and set-points (dotted lines).

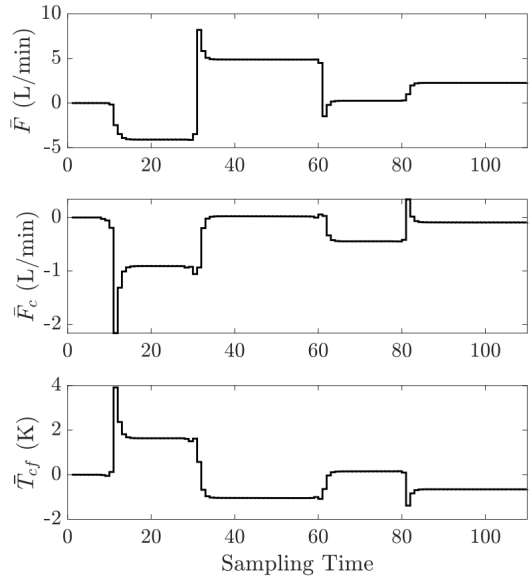

Fig. 5. Manipulated inputs computed using the RNN-based MPC.

3. PROPOSED APPROACH FOR MODEL IDENTIFICATION AND CONTROL

3.1 PCA-RNN-based model identification

In the proposed PCA-RNN-based approach, PCA is utilized to eliminate the correlations existing in the input space. Then, scores (which are by definition uncorrelated) are used to develop the model for the system using any appropriate machine learning-based techniques. The motivation for using PCA in this procedure is to remove the co-linearity in the input space. Recall that co-linearity in the input space means that it is impossible to identify the unique impact of each input variable. As a result, any data-driven model that uses the raw inputs as manipulated variables without considering the co-linearity would end up assigning arbitrary effects to the inputs. By contrast, the scores calculated in a PCA model are inherently independent and therefore represent directions of variation for which sufficient information is available to identify the unique impact of this reduced dimensional input space on

output variables. Recognizing this, the NARX model in equation (3) can be considered as follows:

$$y(\tau + T_s) = f[y(\tau), y(\tau - T_s), ..., y(\tau - L_y T_s), \quad (6)$$
$$t(\tau), t(\tau - T_s), ..., t(\tau - L_t T_s)]$$

where y $\in \mathbb{R}^{n_y}$ and and t $\in \mathbb{R}^{n_t}$ represent the vector of measured outputs and scores, respectively, and the current time is indicated by τ. The number of lagged outputs and scores required for prediction are represented by L_y and L_t, respectively.

The recurrent neural network (NARX), in Machine Learning and Deep Learning toolbox, is employed to identify the function 'f' in the equation (6) to describe the CSTR example dynamic introduced in Section 2. To this end, the data samples are normalized and then the PCA is applied to the input data, shown in Fig. 1, in the first step. Not surprisingly, the *R-squared*, value suggests using two principal components for model identification. In the second step, the obtained scores, along with the outputs are utilized to train the RNN model. One hidden layer with 4 neurons and a delay of 1 is considered. The hyperbolic tangent activation function and linear transfer function are employed in hidden and output layers, respectively. The Bayesian regularization backpropagation algorithm is chosen in training stage. Fig. 3 (dotted lines) shows the prediction results obtained by implementing this approach. It can be observed the PCA-RNN model has nearly the same capability as the RNN model to capture the process dynamics. This is due to the fact that for the collected data (both for training and validation) the same correlation between the two inputs (F_c and T_{cf}) exists. It should be noted that the objective of this work is not to show the performance improvement of the PCA-RNN model over the RNN model for the same training and validation data, to readily enable the use of the resultant model in MPC to ultimately compare the capability of the RNN-based MPC with PCA-RNN-based MPC to handle the problem of set-point tracking.

3.2 PCA-RNN-based model predictive control (MPC)

In this section, the objective is to use the PCA-RNN-based model to design model predictive control for motivating example introduced in Section 2. Considering the equation (6), the PCA-RNN-based technique is proposed to determine function 'f' to capture the process dynamics in the previous section. The PCA-RNN model is used in MPC formulation given by equation (7):

$$\begin{aligned} \min_{t_k,...,t_{k+P}} \quad & \sum_{j=1}^{P} \|\tilde{y}_{k+j} - y^{SP}_{k+j}\|^2_{Q_{y_p}} + \|t_{k+j} - t_{k+j-1}\|^2_{R_{dt}} \\ s.t. \quad & PCA - RNN\ based\ Predictive\ Model, \\ & t_{min} \le t \le t_{max} \end{aligned} \quad (7)$$

where Q_{y_p} and R_{dt} are penalty matrices corresponding to the output deviations from the set-points and the rate of change in the scores, and the rest of the parameters are as defined earlier.

Contrary to the RNN-based MPC, the optimal values of scores are calculated in the PCA-RNN-based MPC approach. Applying PCA can be thought as a transformation that enables us to design the controller in the uncorrelated space of scores. Subsequently, the values of inputs can be calculated by transforming scores into input space using loading vectors.

4. SIMULATION RESULTS: APPLICATION TO THE MOTIVATING EXAMPLE

The PCA-RNN-based MPC is designed for the chemical reactor example in this section. To this end, the PCA-RNN model, identified in Section 3, is employed in the equation (7) to compute the optimal scores. To this purpose, the following parameters are considered: sampling time T_s = 30 s, prediction horizon $P = 6$, $Q_{y_p} = [1000\ 0; 0\ 1000]$, and $R_{dt} = [1\ 0; 0\ 1]$. Fig. 6 and Fig. 7 respectively show the trajectories for the outputs and the control actions obtained by transforming the values of scores.

Contrary to the RNN-based approach, the PCA-RNN-based MPC can handle the problem of set-point tracking for jacket temperature, where two correlated inputs (F_c) and (T_{cf}) are manipulated to achieve the desired set-point. The reason why the RNN-based MPC approach fails can be explained by analyzing the optimal values of inputs computed using two methods. As shown in Fig. 5 and Fig. 7, the values of the correlated inputs (F_c) and (T_{cf}) obtained from the RNN-based MPC algorithm are different from those that are calculated using the PCA-RNN-based MPC method. It can be concluded the identified RNN model in the RNN-based MPC fails to predict the values of the future outputs because the correlations between the control actions, computed by the RNN-based MPC, are different from those that used for model identification resulting in poor prediction of the RNN model and consequently control failure.

Remark 1. Note that another remedy for the RNN-based MPC design is to consider additional constraints which force the inputs to maintain the same correlations as those that exist in input data samples during model training and identification. This can be performed by applying the principal component analysis on the input space in the constraint of the RNN-based MPC and enforcing the values of squared prediction error (SPE) to not breach some certain thresholds. This remains the subject of future work.

5. CONCLUSION

In this study, a novel MPC design, using the combination of principal component analysis and recurrent neural network to model the process dynamics, is proposed to handle the problem of set-point tracking for complex processes. Principal component analysis is performed to reject the redundancy in the input space and recurrent neural network is employed to capture process dynamics. The identified model is then used in the MPC formulation, computing the optimal scores. The result of proposed approach for the chemical reactor example is illustrated and compared with the RNN-based MPC approach. The results show the effectiveness of the proposed approach to track the set-point trajectories, while the RNN-based approach fails to

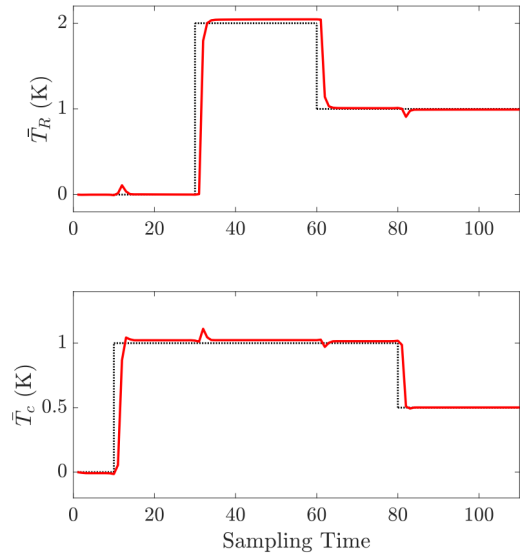

Fig. 6. The trajectories for the reactor and jacket temperatures obtained from the PCA-RNN-based MPC (solid lines) and set-point (dotted lines).

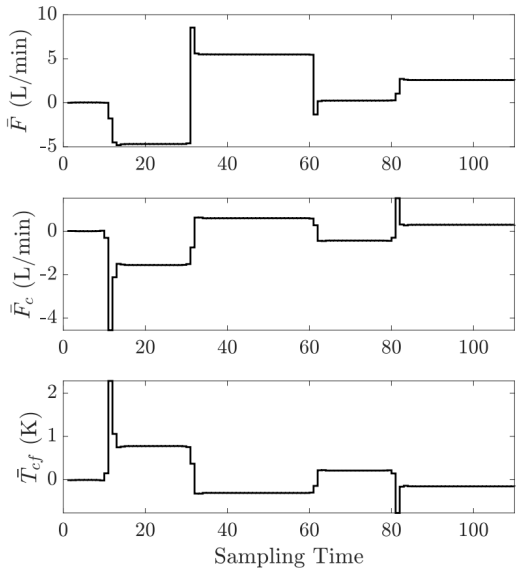

Fig. 7. Manipulated inputs computed using the PCA-RNN-based MPC.

reach the set-points. This is achieved because the optimal values of inputs, calculated by the RNN-based MPC, are obtained without considering the specific correlation existing in the input space during model identification, and consequently results in poor prediction of the RNN model in the RNN-based MPC.

ACKNOWLEDGEMENTS

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the McMaster Advanced Control Consortium (MACC) is gratefully acknowledged.

REFERENCES

Corbett, B. and Mhaskar, P. (2016). Subspace identification for data-driven modeling and quality control of batch processes. *AIChE Journal*, 62(5), 1581–1601.

Du, M. and Mhaskar, P. (2014). Isolation and handling of sensor faults in nonlinear systems. *Automatica*, 50, 1066–1074.

Eastment, H.T. and Krzanowski, W.J. (1982). Cross-validatory choice of the number of components from a principal component analysis. *Technometrics*, 24(1), 73–77.

Flores-Cerrillo, J. and MacGregor, J.F. (2005). Latent variable MPC for trajectory tracking in batch processes. *Journal of process control*, 15(6), 651–663.

Forssell, U. and Ljung, L. (1999). Closed-loop identification revisited. *Automatica*, 35(7), 1215–1241.

Golshan, M., MacGregor, J.F., Bruwer, M.J., and Mhaskar, P. (2010). Latent Variable Model Predictive Control (LV-MPC) for trajectory tracking in batch processes. *Journal of process control*, 20(4), 538–550.

Kheradmandi, M. and Mhaskar, P. (2018). Data driven economic model predictive control. *Mathematics*, 6(4), 51.

Kittisupakorn, P., Thitiyasook, P., Hussain, M.A., and Daosud, W. (2009). Neural network based model predictive control for a steel pickling process. *Journal of Process Control*, 19(4), 579–590.

Narendra, K.S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, 1(1), 4–27.

Nikravesh, M., Farell, A.E., and Stanford, T.G. (2000). Control of nonisothermal CSTR with time varying parameters via dynamic neural network control (DNNC). *Chemical Engineering Journal*, 76(1), 1–16.

Qin, S.J. and Ljung, L. (2003). Closed-loop subspace identification with innovation estimation. *IFAC Proceedings Volumes*, 36(16), 861–866.

Shardt, Y.A. and Huang, B. (2011). Closed-loop identification with routine operating data: Effect of time delay and sampling time. *Journal of Process Control*, 21(7), 997–1010.

Shardt, Y.A., Huang, B., and Ding, S.X. (2015). Minimal required excitation for closed-loop identification: Some implications for data-driven, system identification. *Journal of Process Control*, 27, 22–35.

Shariff, H.M., Rahiman, M.H.F., and Tajjudin, M. (2013). Nonlinear system identification: Comparison between PRBS and Random Gaussian perturbation on steam distillation pilot plant. *In 2013 IEEE 3rd International Conference on System Engineering and Technology*, 269–274.

Wold, S. (1978). Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4), 397–405.

Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019a). Machine learning-based predictive control of nonlinear processes. Part I: Theory. *AIChE Journal*, 65, e16729.

Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019b). Machine-learning-based predictive control of nonlinear processes. Part II: Computational implementation. *AIChE Journal*, 65, e16734.