A Deep-Learning-Based Observer for State Estimation of Direct Contact Membrane Distillation System Modeled by Differential Algebraic Equations

Yubin Wang[1], Yasmine Marani[1] and Taous Meriem Laleg Kirati[1,2]

***Abstract*— Due to its high rejection rate and low energy consumption, Direct Contact Membrane Distillation (DCMD) technology is drawing more attention for seawater desalination, to meet the urgent and growing demands for freshwater. State estimation in DCMD system, which is modeled by nonlinear Differential Algebraic Equations (DAE) is crucial for controller design and system's monitoring. In this paper, a novel learning-based observer is proposed for state estimation of the DCMD system. The method consists of an encoder and decoder structure. The encoder allows to transform the DAE system into a linear ODE modulo an output injection in the latent space and the decoder helps in recovering the state estimate from the latent state. First, a brief description of the DCMD system and its DAE model are recalled. Then, the method is presented and illustrated. Explanations on how the learning structures are constructed and trained are provided. Finally, numerical simulations are conducted to illustrate the effectiveness of the proposed learning-based observer design.**

I. INTRODUCTION

With the background of freshwater sources are getting exhausted and seawater desalination is a promising solution to freshwater scarcity issue, Membrane distillation (MD) is placed as a high potential desalination technique due to its impressive solutes' rejection factor approximating 100% [1] and emerging sustainable potential, combining both thermal and membrane-based separation techniques. Better than other desalination processes, MD is operated with a lower hydrostatic pressure and lower temperatures [1], making this energy-friendly technique a promising and sustainable method for water desalination.

However, MD suffers from some limitations, in particular, low production rate. Several advanced control techniques are designed to overcome issues for MD, especially for DCMD configuration. For instance, a Lyapunov-based control was proposed for DCMD modeled by Partial Differential Equations (PDE) in [3], an Extremum Seeking Control was developed in [4], Model Predictive Control (MPC) paradigms were designed and analyzed in [5] and [6]. However, most of the proposed advanced strategies are based on the assumption that the states of DCMD are known, accessible and fully measurable. Unfortunately, it is actually challenging to measure most of the DCMD states, temperatures and heat transfer rates especially, due to the nature of the DCMD system. Hence, several methods for observing the DCMD state have been proposed for instance in [3], [4] and [7] in which the PDE model has been used in the design of the state observers. To reduce the computational complexity, observer structures based on the nonlinear Differential Algebraic Equations (DAE) have been proposed in [8] and an advanced computationally efficient version using Linear Matrix Inequality (LMI) was designed in [9]. However, the convergence of such observers depends on restrictive assumptions on the non-linearities.

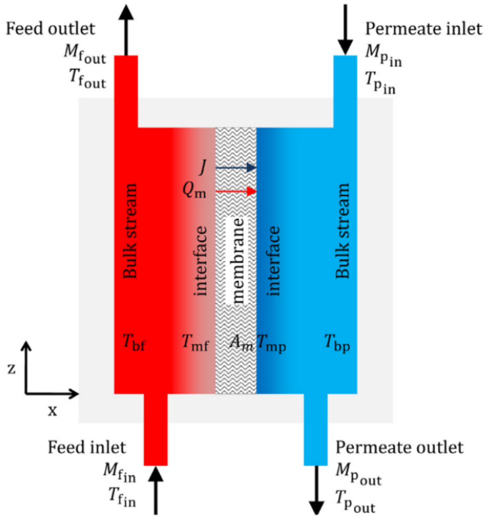


Fig. 1. DCMD module configuration [2]

Different from the above observers, observers dealing with less restrictive assumptions on the non-linearities have been proposed where the main idea consists in identifying a corresponding transformation, which simplifies the nonlinear dynamics to linear or canonical forms [10], [11]. Moreover, a Kazantzis-Kravaris-Luenberger (KKL) observer has been developed with the idea of using a transformation of nonlinear systems into approximately linear or canonical forms modulo an output injection in [12] and is extended to the discrete-time form in [13]. However, despite the fact that the existence of such a transformation has been already established, the analytical and numerical solution is usually not easy to compute. Therefore, with the help of deep neural networks structure, a computationally efficient algorithm has been proposed to compute the transformation through supervised

[1]Computer, Electrical and Mathematical Science and Engineering Division (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia [2]Taous Meriem Laleg-Kirati is also affiliated to the National Institute for Research in Digital Science and Technology, Paris-Saclay, France. yubin.wang@kaust.edu.sa, yasmine.marani@kaust.edu.sa, taousmeriem.laleg@kaust.edu.sa

learning in [14], and unsupervised-learning in [15]. Although the learning-based KKL observer offers a good alternative, [14] [15], the performance for real applications still remains uncertain since the existence of the unique transformation used in KKL observer would be lost due to the non-linearity and control input.

Inspired by [15], this paper introduces a generalized deep-learning-based observer design to estimate the states of the DCMD system modeled by a set of nonlinear DAEs. An auto-encoder-decoder model is built and trained with unsupervised-learning to identify a pair of suitable transformations, which can map the process states into latent space where the new dynamics are linear, guaranteeing the overall convergence of the proposed learning-based observer. Data has been generated using the DAE model in [8]. Numerical simulations are conducted to demonstrate the effectiveness of the proposed data-driven estimation method for the DCMD process. The contributions and novelties are mainly focused on: (i) a novel purely deep-learning-based observer is designed and a corresponding deep-learning implementation is proposed, (ii) to the best of the author knowledge, it's the first time to utilize deep-learning techniques to estimate states of the DCMD system modeled by DAEs, (iii) the presented methodology is generalized and suitable for non-autonomous and high-dimensional systems with constant control input, removing complex limitations on control input and assumptions in the system dynamics.

The rest of paper will be presented as follows: Section II serves as a brief description of the operation principles and the model of DCMD. In section III, a learning-based observer is designed and an auto-encoder-decoder is developed to estimate the process states of DCMD. To illustrate the effectiveness of the proposed implementation, numerical simulations are conducted to evaluate the observer's performance in Section. IV. Finally, conclusions are presented in Section. V.

II. Direct Contact Membrane Distillation (DCMD) System

A. DCMD Configuration Description

The DCMD configuration is illustrated in Figure 1. A hot feed solution (seawater) and cold permeate solution are maintained in direct contact with a hydrophobic membrane which ensures that only pure water vapor passes to the cold permeate solution.

The hot feed water enters the DCMD module at a temperature $T_{f_{in}}$ with an inlet mass flow rate $M_{f_{in}}$. In the permeate side, cold freshwater enters at temperature $T_{p_{in}}$. The water temperature decreases from bulk feed stream T_{bf} to membrane-feed interface T_{mf}. On the other hand, the permeate water temperature increases from the bulk permeate stream T_{bp} to the membrane-permeate interface T_{mp}. This difference of temperature along the membrane induces a gradient of pressure yielding to the water vapor molecules to travel through the membrane pores and condense at the permeate side of the DCMD module. The flux generated is expressed as follows [16]:

$$J = B_m(P_{mf} - P_{mp}) \tag{1}$$

where B_m is the membrane mass transfer parameter, P_{mf} and P_{mp} represent the saturated vapor pressure on the membrane interface from the feed and permeate side, respectively. They can be calculated by Antoine equation with the respective temperature at the membrane interface [16]:

$$P[T] = e^{(23.1964 - \frac{3816.44}{T+227.02})} \tag{2}$$

Overall, the DCMD process is driven by two transport mechanisms: Heat transfer and Mass transfer. In fact, the trans-membrane temperature difference generates a simultaneous mass transfer and heat transfer: the evaporation of water at the membrane feed interface, the transport of water vapor molecules through the pores of the membrane, and water vapor condensation at the cold interface.

The combination of these two transfer mechanisms is used to develop the model in [8] based on thermal-electrical analogy and a lumped capacitance method. By subdividing the DCMD into N coupled cells, the authors were able to assume that the temperature distribution in the feed and permeate bulk streams were uniform to obtain a mono-dimensional distribution. Afterwards, they derived an equivalent electrical network of the DCMD system, and obtained the following Differential Algebraic Equations (DAE):

$$\begin{cases}
\dfrac{dQ_{f_n}}{dt} = \dfrac{1}{L_f^n} T_{bf_{n-1}} - \dfrac{R_{fz}^n}{L_f^n} Q_{f_n} - \dfrac{1}{L_f^n} T_{bf_n} \\
\dfrac{dT_{\mathrm{bf}_n}}{dt} = \dfrac{1}{C_{\mathrm{bf}}} Q_{f_n} - \dfrac{1}{C_{\mathrm{bf}}} \left(\dfrac{1}{R_{\mathrm{f}}} + J_n A_m c_p \right) T_{\mathrm{bf}_n} \\
\qquad - \dfrac{1}{C_{\mathrm{bf}}} Q_{f_{n+1}} + \dfrac{1}{C_{\mathrm{bf}} R_{\mathrm{f}}} T_{\mathrm{mf}_n} \\
\dfrac{dQ_{p_n}}{dt} = \dfrac{1}{L_p^n} T_{bp_{n-1}} - \dfrac{R_{pz}^n}{L_p^n} Q_{p_n} - \dfrac{1}{L_p^n} T_{bp_n} \\
\dfrac{dT_{\mathrm{bp}_n}}{dt} = \dfrac{1}{C_{\mathrm{bp}}} Q_{p_n} - \dfrac{1}{C_{\mathrm{bp}} R_p} T_{\mathrm{bp}_n} - \dfrac{1}{C_{\mathrm{bp}}} Q_{p_{n+1}} \\
\qquad + \dfrac{1}{C_{\mathrm{bp}}} \left(\dfrac{1}{R_{\mathrm{p}}} + J_n A_m c_p \right) T_{\mathrm{mp}_n} \\
0 = \left(\dfrac{1}{R_{\mathrm{f}}} + J_n A_m c_p \right) T_{\mathrm{bf}_n} - \dfrac{1}{R_{\mathrm{f}}} T_{\mathrm{mf}_n} \\
\qquad - \left(\dfrac{1}{R_{\mathrm{p}}} + J_n A_m c_p \right) T_{\mathrm{mp}_n} + \dfrac{1}{R_{\mathrm{p}}} T_{\mathrm{bp}_n} \\
0 = \left(\dfrac{1}{R_m} + \dfrac{1}{R_p} + J_n A_m c_p \right) T_{mp_n} - \dfrac{1}{R_p} T_{bp_n} \\
\qquad - J_n A_m H_v[T_{mf_n}] - \dfrac{1}{R_m} T_{mf_n} \\
\\
0 = T_{f_{out}} - T_{\mathrm{p_{in}}} - R_{\mathrm{f}_{term}} Q_{f_{N+1}} \\
0 = T_{\mathrm{p_{out}}} - T_{\mathrm{f_{in}}} + R_{\mathrm{p}_{term}} Q_{\mathrm{p}_1}
\end{cases} \tag{3}$$

The states equations are in the form of a nonlinear descriptor system:

$$E\dot{x} = F_c(x) + B_u u,$$
$$y = \begin{bmatrix} T_{f_{out}} \\ T_{p_{out}} \\ G(x) \end{bmatrix}. \tag{4}$$

where the state vector $x \in \mathbb{R}^{(6\mathrm{N}+4)}$ is given by:

$$x = [Q_{f_1}, T_{bf_1}, \ldots, T_{bf_N}, Q_{f_{N+1}}, Q_{p_1}, T_{bp_1}, \ldots, T_{bp_N}, Q_{p_{N+1}},$$
$$T_{f_{out}}, T_{p_{out}}, T_{mf_1}, \ldots, T_{mf_N}, T_{mp_1}, \ldots, T_{mp_N}]^T,$$

TABLE I

NOMENCLATURE

Symbol	**Description**	**Unit**
A_m	Differential cell membrane area	m^2
B_m	Membrane transport coefficient	$\mathrm{Kg/m^2sPa}$
C	Thermal capacitance	J/°C
c_p	Specific heat of water	J/Kg°C
G	Production rate	kg/h
H_v	latent heat of vaporization	J/Kg
J	Mass flux	$\mathrm{Kg/m^2\ s}$
L	Thermal coupling inductor	
N	Total number of cells in the DCMD	
Q	Heat transfer rate	W
R	Thermal resistance	°C or K
T	Temperature	°C or K
Subscripts		
b	Bulk	
f	Feed	
in	Inlet	
m	membrane interface	
n	cell index	
out	Outlet	
p	Permeate	
$term$	Terminal cell	
z	coupling	

E is a singular matrix whose rank is smaller than $6N+4$, $F(x)$ is a non-linear function of the states. The input of the system $u = [T_{f_{in}}, T_{p_{in}}]^T$ has two components as illustrated in Figure. 1: the feed inlet temperature $T_{f_{in}}$ and permeate inlet temperature $T_{p_{in}}$. The output of the process y consists of the feed outlet temperature $T_{f_{out}}$, the permeate outlet temperature $T_{p_{out}}$, and the production rate $G = \sum_{n=1}^{N} J_n(x)A_m$. For more details about the system equations and the matrices form, we refer the reader to [8]. For numerical simplicity, we will consider in what follows that the DCMD process contains one cell only, where $N = 1$.

B. Problem Statement

The aim of the present paper is to develop an observer for state estimation in a DCMD system. The proposed strategy consists in identifying a nonlinear mapping and a pseudo-inverse form of the mapping that transforms the nonlinear DAE system into a linear ODE system for which there exists an observer. In what follows, we will rely on unsupervised learning techniques to identify the desired mappings. Giving the discrete nature of deep learning algorithms, the DCMD system (4) is discretized using Euler backward discretization:

$$\begin{cases} Ex_{k+1} = F(x_k, u_k) \\ y_k = \begin{bmatrix} T_{f_{out}} \\ T_{p_{out}} \\ G(x_k) \end{bmatrix} \end{cases} \tag{5}$$

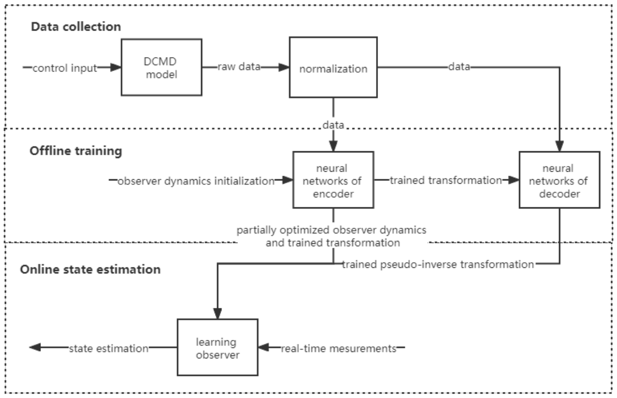

Data collection
control input
DCMD model
raw data
normalization
data
data
Offline training
observer dynamics initialization
neural networks of encoder
trained transformation
neural networks of decoder
partially optimized observer dynamics and trained transformation
Online state estimation
trained pseudo-inverse transformation
state estimation
learning observer
real-time mesurements

Fig. 2. Schematic of a learning-based state estimation methodology for DCMD system

where state is given by:
$x = [Q_{\mathrm{f_1}}, T_{\mathrm{bf}}, Q_{\mathrm{f_2}}, Q_{\mathrm{p_1}}, T_{\mathrm{bp}}, Q_{\mathrm{p_2}}, T_{\mathrm{fout}}, T_{\mathrm{pout}}, T_{\mathrm{mf}}, T_{\mathrm{mp}}]^T \in \mathbb{R}^{10}$, F is a nonlinear vector field resulting from the discretization. $k \in \{1 \ldots K\}$ is the step time, and K represents the number of samples.

The goal of the rest of the paper is to estimate the next state x_{k+1} with the knowledge of state x_k and recent output y_k.

III. Learning-Based Observer Design and Deep Learning Implementation

A. Overview of methodology

The overview of the whole process of fully learning-based state estimation methodology is presented in Fig. 2. The proposed scheme contains the following functional block: (i) data collection, (ii) offline training and (iii) online state estimation.

In the data collection section, the raw data is generated for a given control input. After normalization, the data is delivered to the offline training section to feed the neural networks of the encoder and decoder with initialized observer dynamics. The outputs of the offline training block are corresponding transformation trained by encoder, pseudo-inverse transformation trained by decoder and partial optimized observer dynamics. With the deployment of outputs obtained from offline training, online state estimation is conducted at the lower block.

B. Deep-Learning-Based Observer Design

With the preliminaries that the nonlinear system complexity in the original coordinates x can be simplified and analyzed as simply as the linear one in latent space (z coordinates) [15] [17], we introduce a nonlinear mapping:

$$\phi : \mathbb{R}^d \to \mathbb{R}^{d'} \tag{6}$$

where $d' > d$, which means the nonlinear mapping ϕ maps the original d-dimensional dataset to a higher d'-dimensional feature space. Here we set the dimension of latent state $d_z = 15$, where $d_z > d_x = 10$.

Then a generalized learning-based observer using the corresponding mappings is designed in the z-coordinates as follows:

$$\begin{aligned} z_k &= \phi\left(x_k\right) \\ z_{k+1} &= A z_k + B y_k \\ \hat{x}_{k+1} &= \phi^{-1}\left(z_{k+1}\right). \end{aligned} \tag{7}$$

where $x_k \in \mathcal{X}$ is the state, $z_k \in \mathcal{Z}$ is the state representation in the latent space, y_k is the output, step time $k \in \{1, \ldots, K\}$, ϕ^{-1} the pseudo inverse of ϕ is also a nonlinear mapping, (A, B) are controllable pair with A is Hurwitz.

Inspired by [15], we take the recommended fixed matrix

$$\begin{aligned} A = \operatorname{diag}([&1-0.5T_s, 1-1T_s, 1-1.5T_s, \ldots, \\ &1-0.5T_s, 1-1T_s, 1-1.5T_s]) \end{aligned} \tag{8}$$

where discretization sampling time $T_s = 0.01$ to balance the convergence speed and estimation error. Since $z_k = \phi(x_k)$ is not part of the data set D, it cannot be scaled which can have a serious impact on the neural network training. It has been proven in [15] that scaling z_k can be achieved by scaling dynamically the matrix B during exploration phase, by taking

$$\tilde{B} = \operatorname{diag}\left(\left[b_1, b_2, \ldots\right]\right) B. \tag{9}$$

where $b_1, b_2, \cdots \in \mathbb{R}$. Here we use the inverse of the standard deviation of latent state z to scale matrix B of the observer. This step can guarantee a relative larger exploration in latent space when training encoder with unsupervised learning. Here we make the exploration phase training encoder and decoder, $epoch^{ex}$ and $epoch'^{ex}$ last 50 epochs.

The above learning-based observer is designed to achieve state estimation without complicated gain parameters and other feedback terms. It just needs two collaborative nonlinear mappings ϕ and ϕ^{-1} instead.

C. Auto-Encoder-Decoder

Different from previous works [14] [15], we slack the restrictions on candidate selection of nonlinear mapping ϕ and propose a generalized data-driven observer design. With the power of deep learning techniques, we build an auto-encoder-decoder model and train neural networks to fit the suitable candidates of ϕ and ϕ^{-1} which can allow the proposed observer to work in latent space according to equation (7). The whole process of the auto-encoder-decoder scheme is described as in Fig. 3.

The neural networks of the encoder and decoder are constructed using Classic Multi-Layer Perceptrons with two hidden layers. Each layer has a width of 3000 neurons followed by a hyperbolic tangent activation function $tanh$. The two neural networks are visualized by Netron [18] as in Fig. 4.

D. Data

1) Raw Data Generation: The large training dataset $D = \{x_k, x_{k+1}, y_k\}$ and validation data set $D' = \{x_k, x_{k+1}, y_k\}$ include the state x in step time k, state in step time $k+1$ and output y in step time k. Assuming the states are available in data collection, the training and validation data set can be obtained offline. In this work, the data sets are generated synthetically from the model in equation (5), where similar to the previous work in [19], a sequence of states x and outputs y is generated for fixed inputs $u = [T_{\mathrm{fin}}, T_{\mathrm{pin}}]^T$. Hence, the training data set D and validation data set D'can be enlarged easily to the size of 3×10^5 and 5×10^4 respectively, by feeding the system with various constant values of the input u.

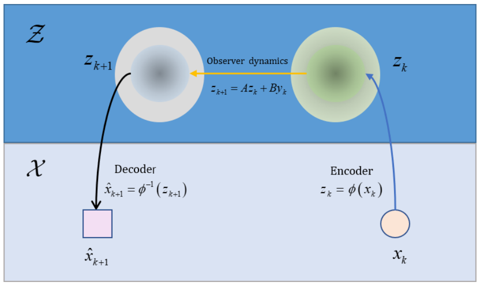
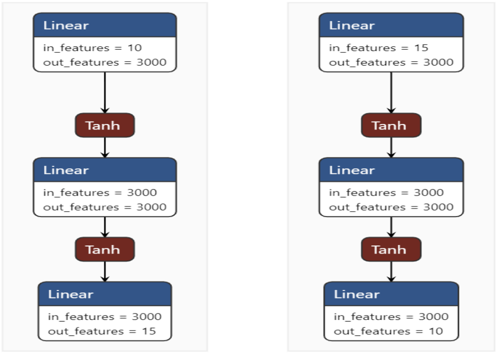


Fig. 3. Auto-encoder-decoder model structure: state x_k is mapped into latent state z_k by encoder, latent state z_{k+1} is obtained by enforcing observer dynamics according to $z_{k+1} = Az_k + By_k$ and state estimation $\hat{x}_{k+1}$ is recovered by decoder.



(a) Neural networks of encoder (b) Neural networks of decoder

Fig. 4. Both neural networks of encoder and decoder are constructed by Classic Multi-layer Perceptrons (Dense Neural Networks) with two hidden layers whose width includes 3000 neurons.

2) Data Normalization: The state x of the DCMD system contains two classes of attributes: temperature and heat transfer rate. The temperate class has low values (less than 100 ^{o}C). However, the heat transfer class takes extremely high values (between 10^3 and 10^4). The huge value difference causes important issues: the gradient of loss function during training will be hard to compute or even be lost.

And we find that the standard data normalization methods are not working anymore for the DCMD system because the sampled data is not subject to any kind of regular distributions, in addition to the huge value difference in x. In this case, the log function is leveraged to scale raw data, decreasing the difference between state variable values for data normalization:

$$\begin{aligned} x^{*} &= \log_{10}(x)/\log_{10}(x_{\max}) \\ y^{*} &= \log_{10}(y)/\log_{10}(y_{\max}) \end{aligned} \tag{10}$$

where x^{*}, y^{*}, x_{max}, y_{max} represent the normalized state, normalized output, state maximum, output maximum, respectively.

This step is extremely essential for learning-based state estimation implementation for DCMD systems, because the smoother gradient calculated by normalized data accelerates the training speed and increases the estimation accuracy.

Algorithm 1: Training Auto-Encoder-Decoder

```
Data: training dataset $D = \{x_k, x_{k+1}, y_k\}$
  and validation dataset $D^{'}$.
Result: Encoder $\phi$ and decoder $\phi^{-1}$ in best
        validation loss, partially optimized
        controllable pair $(A, B^{*})$
Set the matrix $A$ and $B$ of the learning observer in
  (7);
Initialize the parameters $\Theta$ of encoder and decoder
  randomly;
Data Normalization: $D \leftarrow D^{*}, D' \leftarrow D'^{*}$;
for $epoch \leftarrow 1$ to $epoch^{max}$ do
    if $epoch < epoch^{ex}$ then
        Compute $z_k$ for each $x_k \in D$;
        for $i \leftarrow 1$ to $d_z$ do
            Set $b_i$ to the inverse of the standard
              deviation of $\{z_{i,k}\}$
        end
        $B \leftarrow \text{diag}([b_1, b_2, \ldots])\, B$
    end
    Compute $L_{enc}$ following (11) with training data
      $D$;
    Update $\Theta$ of encoder by using Adam optimizer to
      minimize $L_{enc}$;
    Compute validation encoder loss $L'_{enc}$ with
      validation data $D'$;
end
Pick up the best encoder $\phi$, which can obtain the
  minimum of validation encoder loss $L'_{enc}*$;
for $epoch' \leftarrow 1$ to $epoch'^{max}$ do
    Compute $L_{dec}$ following (12) with training data
      $D$;
    Update $\Theta$ of decoder by using Adam optimizer to
      minimize $L_{dec}$;
    Compute validation decoder loss $L'_{dec}$ with
      validation data $D'$;
end
Pick up the best decoder $\phi^{-1}$ which can obtain the
  minimum of validation decoder loss $L'_{dec}*$;
```

E. Training

- The encoder to identify ϕ is trained by optimizing the following L_2 loss function:

$$\mathcal{L}_{enc} = \|\phi(x_{k+1}) - (A\phi(x_k) + By_k)\| \tag{11}$$

where the norm $\|\cdot\|$ is the mean-squared error.

- Having the trained encoder with the minimum of encoder validation loss ${L_{enc}}'^{*}$, we train the decoder to find desired ϕ^{-1} via optimizing the following loss function:

$$\mathcal{L}_{dec} = \left\| x_k - \phi^{-1}(\phi(x_k)) \right\| \tag{12}$$

The whole training process is conducted according to Algorithm 1.

Deep learning details are as the standard training implementations: framework is based on PyTorch [20], batch size is set to 100, Adam optimizer with learning rate initialized to 0.001 is used, learning rate decays every 50 epochs by a factor of 4. The model is trained on a workstation equipped with a AMD Ryzen 9 5950X CPU and a NVIDIA GeForce RTX 3080Ti GPU. The training lasts for 160 epochs monitored by Weights and Biases [21] and requires around 2 hours to converge.

IV. Numerical Simulations

The characteristics of the membrane used in the simulations are presented in Table II.

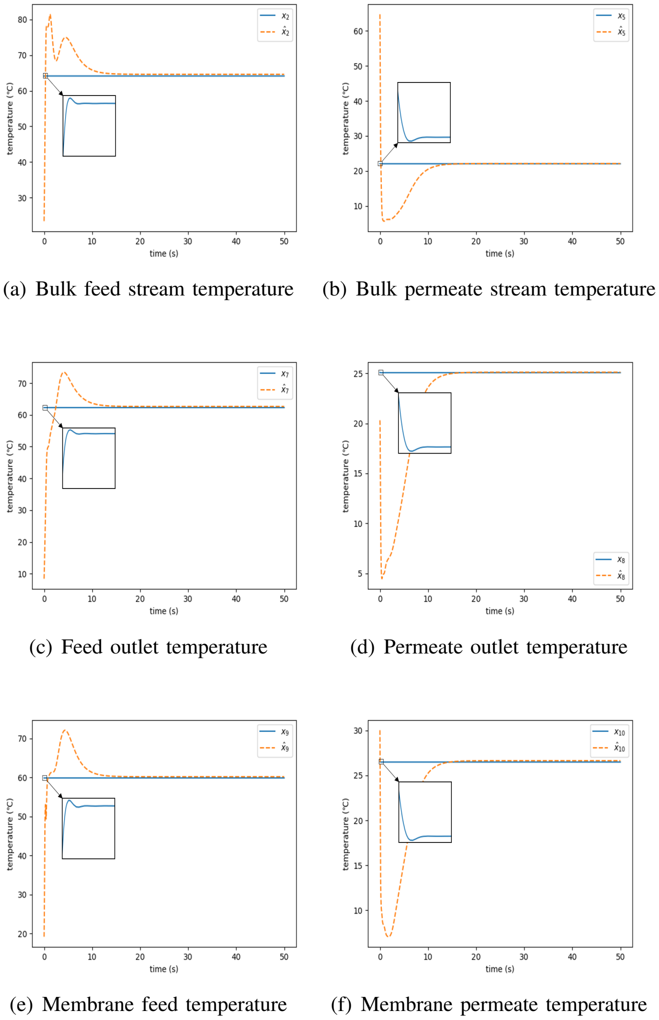
TABLE II
MEMBRANE PROPERTIES

Property	Value
Membrane's length	0.50 m
Membrane's width	0.05 mm
Membrane's thickness	150 μm
Pore size	0.26 μm
Porosity	77%
Tortuosity	1.35 μm
Thermal conductivity	0.2 W /(m K)
Mass transfer coefficient	5.14×10^{-7} kg/(m^2·s·Pa)

We select the model with optimal validation loss as the best model to evaluate the observer's performance.

A. Observer Evaluation

The evaluation duration lasts 5000 time step and each time step is 0.01 seconds. We arbitrarily exploit the constant control input $u = [T_{f_{in}}, T_{p_{in}}]^T$, where the range of temperatures in feed side $T_{f_{in}}$ is from 63°C to 67°C and permeate side temperature $T_{f_{out}}$ is from 18°C to 22°C, to enforce the system's dynamics and the DCMD system will converge to the equilibrium point in short time (about 1.5s). The initial conditions of state estimation can be selected arbitrarily, we set all initial conditions of heat transfer rate to 0 and temperatures to arbitrary integers. We monitor the estimation convergence of observer along time and record state estimation value at final time when $t = 50s$ to calculate convergence accuracy.



(a) Bulk feed stream temperature

(b) Bulk permeate stream temperature

(c) Feed outlet temperature

(d) Permeate outlet temperature

(e) Membrane feed temperature

(f) Membrane permeate temperature

Fig. 5. Temperatures estimation with the proposed Deep-Learning-Based Observer and their real values

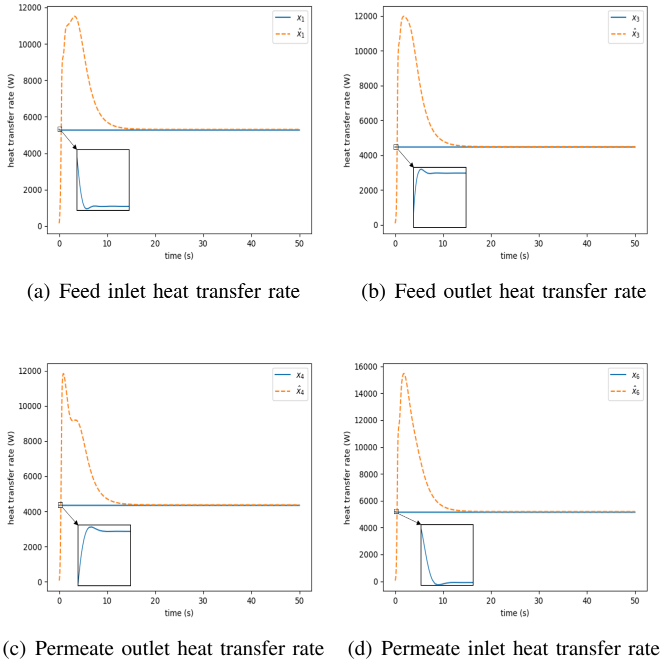
B. Observer's Performance Analysis

The convergence of temperature estimations and heat transfer rate estimations are recorded in Fig 5, Fig 6.

As shown in Fig 5 and Fig 6, all state estimations (except permeate outlet temperature estimation) rise or drop rapidly to go beyond the real value temporarily and then start converging to the real value in finite time. After about 15s, all states estimates converge to the real values with minor steady state error, and we consider that the state estimation for DCMD system is completed.

To take a step further to prove the accuracy of the learning observer implemented in this paper, the relative estimation error $\delta = \left|\dfrac{x - \hat{x}}{x}\right|$ is monitored during whole evaluation process. Results are displayed in Fig 7.

The figure shows that there exists a phase usually called ”oscillation phase” where big peaks appear resulting in high estimation errors. One can notice that the biggest peaks are associated with heat transfer rates. This can be explained by the fact that this type of attributes (heat transfer rate) takes values of the order of 10^3, and starting the learning observer with a zero initial condition is effort-costing for the proposed observer to reach these high values in short time. After the mentioned phase (approximately $5s$), all relative estimation error curves start decreasing rapidly to zero. The convergence is achieved after $15s$.



(a) Feed inlet heat transfer rate

(b) Feed outlet heat transfer rate

(c) Permeate outlet heat transfer rate

(d) Permeate inlet heat transfer rate

Fig. 6. Real heat transfer rates values and their estimation with the proposed Deep-Learning-Based Observer

Table III shows the real states values, their estimations using the learning observer, and the relative estimation error recorded at 50s. The temperature and heat transfer rate estimations converge to the real value with small steady errors. In fact these errors are acceptable, as all the relative estimation errors are less than 1% .

TABLE III

TEMPERATURE AND HEAT TRANSFER RATE ESTIMATION AT 50S

state	real value	estimation	relative error
T_{bf}	64.14	64.60	0.7%
T_{bp}	22.03	22.09	0.3%
T_{fout}	62.29	62.64	0.6%
T_{pout}	25.08	25.12	0.2%
T_{mf}	59.85	60.17	0.5%
T_{mp}	26.48	26.63	0.6%
Q_{fin}	5269	5307	0.7%
Q_{fout}	4467	4486	0.4%
Q_{pin}	4343	4371	0.7%
Q_{pout}	5145	5191	0.9%

From the simulation results, we can conclude that the generalized learning-based observer is working for DCMD systems represented by a DAE structure to estimate the states under constant inputs, if the encoder and decoder are well-trained to identify a suitable collaborative pair of nonlinear

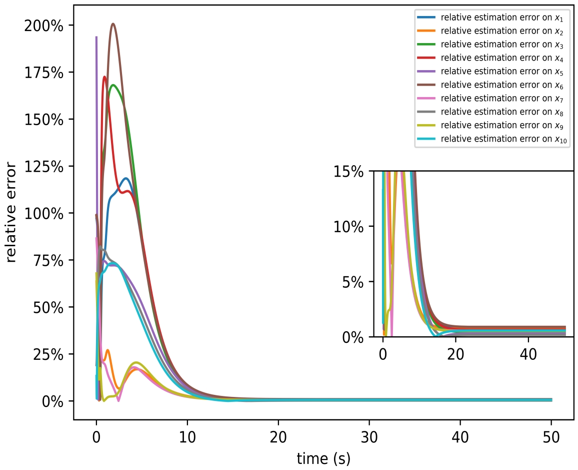


Fig. 7. Relative estimation error monitored during the whole process of proposed learning-based observer evaluation

mappings ϕ and ϕ^{-1}, which can make the observer working in latent space.

V. CONCLUSIONS

In this paper, a generalized learning-based observer is designed to estimate the states of the DCMD system via two collaborative nonlinear mappings that transform the original system modeled by a DAE into a latent space, where it can be represented by linear ODEs. A data-driven auto-encoder-decoder model is constructed to identify the two nonlinear mappings. Numerical simulations show that the proposed learning observer can estimate the states of the DCMD process with negligible steady-state relative estimation errors. The future work will focus on extending the proposed learning-based observer for DAE systems with time-varying control input, as well as a hardware implementation in order to experimentally validate the proposed observer. Additionally, optimizing the matrix A to avoid possible poor dynamic state capture is an open topic that would be interesting to tackle.

ACKNOWLEDGMENT

Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST) with the Base Research Fund (BAS/1/1627-01-01).

REFERENCES

[1] Zhong Sheng Tai, Mohd Haiqal Abd Aziz, Mohd Hafiz Dzarfan Othman, Ahmad Fauzi Ismail, Mukhlis A. Rahman, and Juhana Jaafar. Chapter 8 - an overview of membrane distillation. In Ahmad Fauzi Ismail, Mukhlis A. Rahman, Mohd Hafiz Dzarfan Othman, and Takeshi Matsuura, editors, *Membrane Separation Principles and Applications*, Handbooks in Separation Science, pages 251–281. Elsevier, 2019.

[2] Ayman M. Karam and Taous Meriem Laleg-Kirati. Membrane fouling modeling and detection in direct contact membrane distillation. *Journal of Process Control*, 81:190–196, 2019.

[3] Fadi Eleiwi and Taous Meriem Laleg-Kirati. Nonlinear observer-based lyapunov boundary control of distributed heat transfer mechanisms for membrane distillation plant. *Journal of Process Control*, 47:78–86, 2016.

[4] Fadi Eleiwi and Taous Meriem Laleg-Kirati. Observer-based perturbation extremum seeking control with input constraints for direct-contact membrane distillation process. *International Journal of Control*, 91(6):1363–1375, 2018.

[5] Juan D Gil, Lidia Roca, Alba Ruiz-Aguirre, Guillermo Zaragoza, and Manuel Berenguel. Optimal operation of a solar membrane distillation pilot plant via nonlinear model predictive control. *Computers & Chemical Engineering*, 109:151–165, 2018.

[6] Xingang Guo, Fahad Albalawi, and Meriem Laleg. Model predictive control paradigms for direct contact membrane desalination modeled by differential algebraic equations. In *2019 American Control Conference (ACC)*, pages 5595–5601. IEEE, 2019.

[7] Mohamed Ghattassi, Jean-Claude Vivalda, and Taous Meriem Laleg-Kirati. State observer design for direct contact membrane distillation parabolic systems. In *2018 Annual American Control Conference (ACC)*, pages 2563–2568. IEEE, 2018.

[8] Ayman Karam and Taous-Meriem Laleg-Kirati. Electrical equivalent thermal network for direct contact membrane distillation modeling and analysis. *Journal of Process Control*, 47, 11 2016.

[9] Xingang Guo, Fahad Albalawi, Ibrahima N'Doye, and Taous-Meriem Laleg-Kirati. State estimation in direct contact membrane distillation based desalination using nonlinear observer ∗∗research reported in this publication has been supported by the king abdullah university of science and technology(kaust). *IFAC-PapersOnLine*, 52(23):61–66, 2019. 1st IFAC Workshop on Control Methods for Water Resource Systems CMWRS 2019.

[10] Claudia Califano, Salvatore Monaco, and Dorothée Normand-Cyrot. Canonical observer forms for multi-output systems up to coordinate and output transformations in discrete time. *Automatica*, 45(11):2483–2490, 2009.

[11] G Besançon, H Hammouri, and S Benamor. State equivalence of discrete-time nonlinear control systems to state affine form up to input/output injection. *Systems & control letters*, 33(1):1–10, 1998.

[12] Pauline Bernard and Vincent Andrieu. Luenberger observers for nonautonomous nonlinear systems. *IEEE Transactions on Automatic Control*, 64(1):270–281, 2018.

[13] Nikolaos Kazantzis and Costas Kravaris. Discrete-time nonlinear observer design using functional equations. *Systems & Control Letters*, 42(2):81–94, 2001.

[14] Louise da C Ramos, Florent Di Meglio, Valéry Morgenthaler, Luís F Figueira da Silva, and Pauline Bernard. Numerical design of luenberger observers for nonlinear systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5435–5442. IEEE, 2020.

[15] Johan Peralez and Madiha Nadri. Deep learning-based luenberger observer design for discrete-time nonlinear systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4370–4375. IEEE, 2021.

[16] A.G. Fane and Rong Wang. *Membrane Distillation: Now and Future*, pages 373–424. 01 2014.

[17] Lucas Brivadis, Vincent Andrieu, and Ulysse Serres. Luenberger observers for discrete-time nonlinear systems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3435–3440, 2019.

[18] Lutz Roeder. Netron: Visualizer for neural network, deep learning and machine learning models. Software available from https://netron.app/.

[19] Yasmine Marani, Tania Camelia Touati, Messaoud Chakir, and Taous Meriem Laleg-Kirati. A nonlinear adaptive resilient observer for fouling detection and localization in direct contact membrane distillation systems. In *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1004–1010. IEEE, 2021.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[21] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.