ANN-Based Adaptive NMPC for Uranium Extraction-Scrubbing Operation in Spent Nuclear Fuel Treatment Process*

Duc-Tri Vo[1], Ionela Prodan[1], Laurent Lefèvre[1], Vincent Vanel[2], Sylvain Costenoble[2], Binh Dinh[2]

Abstract— This study focuses on the challenges in optimizing the control of uranium extraction and scrubbing within the PUREX process. The objective is to maintain the system at a target solvent saturation level while adhering to operational constraints, managing disturbances, and adjusting to changes in set points. The French Alternative Energies and Atomic Energy Commission (CEA) developed a specialized simulator called PAREX to model liquid-liquid extraction in the PUREX process. However, the underlying mathematical model is complicated, consisting of nonlinear, high-dimensional differential-algebraic equations (DAEs). Consequently, direct application of optimal control methods would result in computationally intensive, large-scale nonlinear programming problems. To overcome this, we propose training a neural network to forecast process outputs based on historical measurements. This network architecture incorporates long short-term memory (LSTM), linear regression, and logistic regression networks, effectively reducing the number of state variables and simplifying the optimization problem. Furthermore, we formulate and solve nonlinear model predictive control (NMPC) and moving horizon estimation (MHE) problems using the Particle Swarm Optimization (PSO) algorithm. Simulation results demonstrate the effectiveness of this adaptive optimal control approach in meeting the control objectives, suggesting its potential for real-world implementation.

I. INTRODUCTION

A. Motivation

The PUREX process, an acronym for "Plutonium, Uranium, Reduction, EXtraction," was developed to recover uranium and plutonium from spent nuclear fuels, which is composed of 95% uranium, 1% plutonium, and 4% high radioactive toxic waste (the ultimate waste). This process offers a high-purity U-Pu recovery and recycling, reducing the ultimate waste volume and thus contributing to sustainable nuclear energy development. The overall control objective is quickly driving the process to achieve a desired solvent saturation level, guarantee constraints, handle the disturbances, and set point variations.

PAREX is a simulation program developed by the French Alternative Energies and Atomic Energy Commission (CEA). It can simulate liquid-liquid extraction operations within the PUREX process. As reported in [1], PAREX is currently used in the nuclear fuel reprocessing industry for process optimization, troubleshooting, and safety analysis.

PAREX offers valuable insights into process dynamics and enables the applicability of model-based control approaches.

This work continues the studies of developing the adaptive Nonlinear Model Predictive Control (NMPC) for the uranium extraction-scrubbing operation in the PUREX process ([2]) and ([3]). We aim to exploit the benefits of the qualified PAREX simulator in the control scheme to satisfy the control objectives and constraints introduced above. However, it requires high-level security controls when developing an ANN replicate of PAREX since PAREX and its data are strictly protected. Therefore, in this first study, we propose a mathematical model that captures the main dynamics of the process, then use it to illustrate and study the developed control strategy in multiple simulations. Note that the proposed algorithm can be generalized to PAREX without any limitation.

In our previous studies ([2] and [3]), a high dimensional process model with 128 states was employed. However, note that from a practical viewpoint, only two state variables have critical roles in the control problem. Therefore, if we can reduce the number of variables in the process model, we can reduce the complexity of the control problem, which is the motivation of this paper.

Our main idea is to develop an artificial neural network (ANN) to predict the essential state variables based on available measurements. Then, the ANN is embedded as a predictor in the Nonlinear Model Predictive Controller (NMPC) scheme and as an estimator in the Moving Horizon Estimator (MHE) strategy. Furthermore, integrating NMPC and MHE allows us to have an adaptive control scheme in which any unmeasured disturbances can be estimated and updated to the controller. To solve the NMPC and MHE optimization problems, we use the enhanced Particle Swarm Optimization (PSO) developed in our previous work ([3]).

The Long Short-term Memory (LSTM) neural network, which was first proposed by [4], is a common choice for time series prediction applications. Therefore, it represents a good candidate method for approximating system dynamics, allowing the application of model-based control techniques such as NMPC. The applicability of LSTM within NMPC was comprehensively discussed by [5]. Note that our proposed ANN architecture is based on the LSTM and linear and logistic regression networks. As will be discussed later in the paper, the ANN is designed based on the particularities of the control problem.

B. Contributions and Paper Organization

This paper presents an ANN-based adaptive control strategy specifically designed for the uranium extraction-scrubbing operation in the PUREX process. The ANN architecture incorporates LSTM, linear, and logistic regression networks. In this control strategy, the ANN functions as a predictor and estimator, with NMPC and MHE solved using the enhanced PSO algorithm proposed in [3]. Briefly, we:

- propose a dynamical model that captures the primary attributes of the uranium extraction-scrubbing operation in the PUREX process. In this paper, this model is used to develop and study the ANN-based adaptive MPC algorithm since the data sourced from PAREX is confidentially protected;
- develop an ANN that can predict the process outputs based on previous measurements. This ANN allows making predictions directly from the available measurements, which has significant advantages compared to the original state space approach since, in this process, not all the state values can be measured online;
- propose an adaptive control strategy that optimally drive the system to work at a desired solvent saturation level while guaranteeing constraints satisfaction (e.g., uranium concentration in the fission product, equipment limits) and adapting to parameters uncertainties (i.e., variation in the fresh solvent flow rate);
- study the efficiency of the proposed control strategy through simulations.

Notations: Vectors are denoted by bold lowercase letters and matrices are denoted by capital letters. $\mathbf{I}$, $\mathbf{0}$ denotes identity and zero matrices of appropriate dimensions. $\mathbf{x}_n$ denotes the nth element of $\mathbf{x}$. $\mathbf{x}_{n:m}$ denotes a vector slice from the nth to m^{th} (included) elements of $\mathbf{x}$. $\|\mathbf{x}\|_{\mathbf{Q}} := \mathbf{x}^T \mathbf{Q} \mathbf{x}$. $\mathbb{N}_{m:n} := \{i \in \mathbb{N} | m \leq i < n\}$.

II. URANIUM EXTRACTION-SCRUBBING OPERATION

A. Mathematical Model

Fig. 1 shows the uranium-scrubbing operation's principle consisting of 16 cascaded mixer settlers. In this work, we propose a model that captures the main dynamics of the qualified model in PAREX. This model is structured as a system of nonlinear, stiff, high-dimensional differential-algebraic equations (DAE). It adheres to mass balances and assumptions such as constant element densities, immiscible aqueous and organic phases, perfect mixing, and interfacial mass transfer derived from the double film theory. The main difference is that it uses a simplified distribution modulus, while PAREX's qualified modulus is derived from practical data. Notations for system parameters are detailed in Fig. 1 and Tab. I while equation (5) describes the primary extraction mechanism.

TABLE I

SYSTEM PARAMETERS.

Notation	Description
A, O	Aqueous and organic flow rates (L/h).
V, W	Aqueous and organic volumes (L).
K_U, K_H	Equilibrium constants for U and H.
k_U, k_H	Mass transfer coefficients for U, H.
a, o	Related to aqueous and organic phase.
M, D	Related to mixer and settler.
$[\cdot]$	Concentration (M).
n	Related to stage n of the process.
in	Related to inputs to stage n.
i	Related to concentrations at the interface.
$*$	Related to concentrations at equilibrium.

1) Interface mass transfer: Applying the double film theory for uranium interfacial mass transfer, as illustrated in Fig. 2, we have:

$$k_U^{\text{aq}} \left([U]_n^{\text{aM}} - [U]_{\text{i},n}^{\text{aM}} \right) = k_U^{\text{eq}} \left([U]_{\text{i},n}^{\text{aM}} - [U]_{*,n}^{\text{aM}} \right), \quad (1a)$$

$$k_U^{\text{aq}} \left([U]_n^{\text{aM}} - [U]_{\text{i},n}^{\text{aM}} \right) = k_U^{\text{og}} \left([U]_{\text{i},n}^{\text{oM}} - [U]_n^{\text{oM}} \right). \quad (1b)$$

In this work, we assume that the transfer resistance can be neglectable, which can be expressed by using high mass transfer coefficients:

$$k_U^{\text{aq}} = k_U^{\text{eq}} = k_U^{\text{og}} \gg 10^{-4} \text{ m/s}, \quad (2)$$

hence, (1) can be rewritten as:

$$[U]_{\text{i},n}^{\text{aM}} = 0.5 \left([U]_n^{\text{aM}} + [U]_{*,n}^{\text{aM}} \right), \quad (3a)$$

$$[U]_{\text{i},n}^{\text{oM}} = 0.5[U]_n^{\text{aM}} + [U]_n^{\text{oM}} - 0.5[U]_{*,n}^{\text{aM}}. \quad (3b)$$

Similar relationships for nitric acid concentrations can be obtained by replacing U with H in equations (1)-(3):

$$[H]_{\text{i},n}^{\text{aM}} = 0.5 \left([H]_n^{\text{aM}} + [H]_{*,n}^{\text{aM}} \right), \quad (4a)$$

$$[H]_{\text{i},n}^{\text{oM}} = 0.5[H]_n^{\text{aM}} + [H]_n^{\text{oM}} - 0.5[H]_{*,n}^{\text{aM}}. \quad (4b)$$

2) Thermodynamic equilibrium: The primary extraction mechanism is given by:

$$\text{UO}_2^{2+} + 2\text{NO}_3^- + 2\text{TBP} \overset{K_U}{\rightleftharpoons} \text{UO}_2 \left(\text{NO}_3 \right)_2 \cdot \text{TBP}, \quad (5a)$$

$$\text{H}^+ + \text{NO}_3^- + \text{TBP} \overset{K_H}{\rightleftharpoons} \text{HNO}_3 \cdot \text{TBP}. \quad (5b)$$

At thermodynamic equilibrium condition, we have

$$[U]_{\text{i},n}^{\text{oM}} = K_U[U]_{*,n}^{\text{aM}}[NO_3]_{*,n}^{\text{aM}\,2}[TBP]_{\text{freei},n}^{\text{oM}\,2}, \quad (6a)$$

$$[H]_{\text{i},n}^{\text{oM}} = K_H[H]_{*,n}^{\text{aM}}[NO_3]_{*,n}^{\text{aM}}[TBP]_{\text{freei},n}^{\text{oM}}, \quad (6b)$$

where

$$[NO_3]_{*,n}^{\text{aM}} = 2[U]_{*,n}^{\text{aM}} + [H]_{*,n}^{\text{aM}}, \quad (7a)$$

$$[TBP]_{\text{tot}\,n}^{\text{oM}} = [TBP]_{\text{freei},n}^{\text{oM}} + 2[U]_{\text{i},n}^{\text{oM}} + [H]_{\text{i},n}^{\text{oM}}, \quad (7b)$$

which leads to

$$[TBP]_{\text{freei},n}^{\text{oM}} = \frac{2c_n}{b_n + \sqrt{b_n^2 + 4a_n c_n}}, \quad (8a)$$
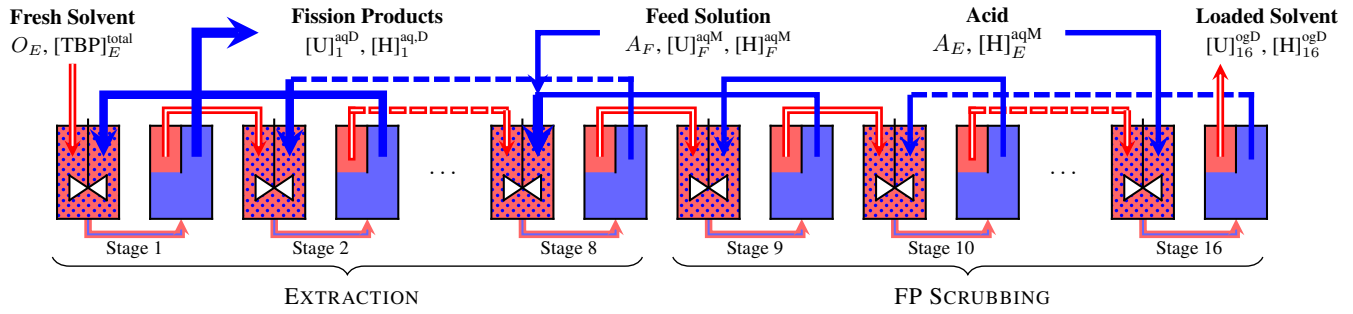
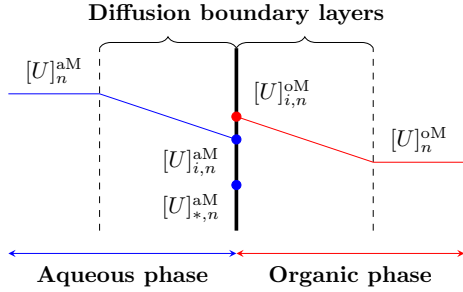Fig. 1. Uranium extraction-scrubbing operations using mixers-settlers ([2]).

Fig. 2. Profile of aqueous uranium concentration in the mixer, as derived with the two-film theory ([6]).
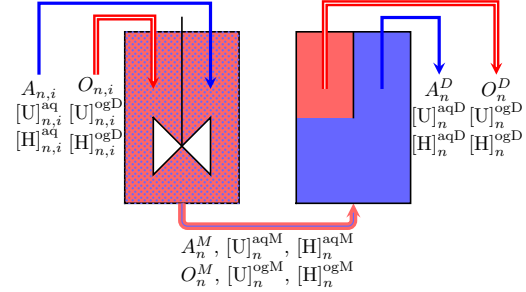

Fig. 3. Mixer-settler model [2].

where

$$a_n = 2K_U[U]_{*,n}^{\text{aM}} \left(2[U]_{*,n}^{\text{aM}} + [H]_{*,n}^{\text{aM}}\right)^2, \tag{8b}$$

$$b_n = 1 + K_H[H]_{*,n}^{\text{aM}} \left(2[U]_{*,n}^{\text{aM}} + [H]_{*,n}^{\text{aM}}\right), \tag{8c}$$

$$c_n = [TBP]_{\text{tot } n}^{\text{oM}}. \tag{8d}$$

By substituting (3b), (4b), (7), and (8) to (6), and assume that $[TBP]_{\text{tot } n}^{\text{oM}} = [TBP]_{\text{tot } E}^{\text{oM}}$ for all stages, we obtain algebraic equations of uranium and acid concentrations, denoted by g_n^U, g_n^H, respectively:

$$g_n^U \left([U]_n^{\text{aM}}, [U]_n^{\text{oM}}, [H]_n^{\text{aM}}, [H]_n^{\text{oM}}, [U]_{*,n}^{\text{aM}}, [H]_{*,n}^{\text{aM}}\right) = 0, \tag{9a}$$

$$g_n^H \left([U]_n^{\text{aM}}, [U]_n^{\text{oM}}, [H]_n^{\text{aM}}, [H]_n^{\text{oM}}, [U]_{*,n}^{\text{aM}}, [H]_{*,n}^{\text{aM}}\right) = 0. \tag{9b}$$

3) Flowrate and volume model: Consider the mixer-settler model showed in Fig. 3, in the absence of monophasic reactions (only extraction phenomenon). The input flowrates and concentrations to the mixer of stage n are:

$$A_{n,i}^M = A_{n+1}^D, \qquad O_{n,i}^M = O_{n-1}^D, \tag{10a}$$

$$[U]_{n,\text{in}}^{\text{aM}} = [U]_{n+1}^{\text{aD}}, \qquad [U]_{n,\text{in}}^{\text{oM}} = [U]_{n-1}^{\text{oD}}, \tag{10b}$$

$$[H]_{n,\text{in}}^{\text{aM}} = [H]_{n+1}^{\text{aD}}, \qquad [H]_{n,\text{in}}^{\text{oM}} = [H]_{n-1}^{\text{oD}}. \tag{10c}$$

Note that at the stages $n \in \{1, 8, 16\}$, we have:

$$O_{1,\text{in}}^M = O_E, \ A_{8,i}^M = A_9^D + A_F, \ A_{16,\text{in}}^M = A_E, \tag{11a}$$

$$[U]_{1,\text{in}}^{\text{oM}} = [H]_{n,\text{in}}^{\text{oM}}, \ [U]_{16,\text{in}}^{\text{aM}} = 0, \ [H]_{16,\text{in}}^{\text{aM}} = [H]_E^{\text{aq}}, \tag{11b}$$

$$[U]_{8,\text{in}}^{\text{aM}} = \left(A_F[U]_F^{\text{aq}} + A_9^D[U]_9^{\text{aD}}\right)/(A_F + A_9^D), \tag{11c}$$

$$[H]_{8,\text{in}}^{\text{aM}} = \left(A_F[H]_F^{\text{aq}} + A_9^D[H]_9^{\text{aD}}\right)/(A_F + A_9^D). \tag{11d}$$

Applying total mass balances to the mixer tank, we have:

$$\dot{V}_n^M = A_{n,i}^M - \frac{A_n^M + O_n^M}{V_n^M + W_n^M} \cdot V_n^M, \tag{12a}$$

$$\dot{W}_n^M = O_{n,i}^M - \frac{A_n^M + O_n^M}{V_n^M + W_n^M} \cdot W_n^M. \tag{12b}$$

The perfect mixing assumption mean $\dot{V}_n^M = \dot{W}_n^M = 0$, hence:

$$A_{n,i}^M W_n^M = O_{n,i}^M V_n^M. \tag{13}$$

In addition, assume that there is no leakages in the system and all mixer-settlers have the same volume, we have:

$$A_n^M = A_n^D, \quad O_n^M = O_n^D, \tag{14a}$$

$$V_1^M + W_1^M = V_2^M + W_2^M = \cdots = V_{16}^M + W_{16}^M. \tag{14b}$$

4) Mass balances of U and H: Applying the mass balances to uranium in the aqueous and organic phases in the mixer tanks, we have:

$$[\dot{U}]_n^{\text{aM}} = \left(A_{n,i}[U]_{n,i}^{\text{aq}} - A_n^M[U]_n^{\text{aM}} - \Phi_n^U\right)/V_n^M, \tag{15a}$$

$$[\dot{U}]_n^{\text{oM}} = \left(O_{n,i}[U]_{n,i}^{\text{og}} - O_n^M[U]_n^{\text{oM}} + \Phi_n^U\right)/W_n^M, \tag{15b}$$

$$\Phi_n^U = \frac{6k_U V_n^M}{d} \left([U]_n^{\text{aM}} - [U]_{\text{i},n}^{\text{aM}}\right), \tag{15c}$$

where Φ_n^U denotes the uranium interfacial mass transfer term, and by substituting (3a) to (15c), we have

$$\Phi_n^U = \frac{3k_U V_n^M}{d} \left([U]_n^{\text{aM}} - [U]_{*,n}^{\text{aM}}\right). \tag{15d}$$

Regarding the settler tanks, the mass balances equations for uranium are given as:

$$[\dot{U}]_n^{\text{aD}} = \left(A_n^M [U]_n^{\text{aM}} - A_n^D [U]_n^{\text{aD}} \right) / V_n^D, \tag{16a}$$

$$[\dot{U}]_n^{\text{oD}} = \left(O_n^M [U]_n^{\text{oM}} - O_n^D [U]_n^{\text{oD}} \right) / W_n^D. \tag{16b}$$

Mass balances equations for H^+ can be obtained by replacing U by H in (15)-(16).

5) State space representation: From (9), (15), and (16), the process dynamical model can be represented as a system of differential-algebraic equations as follow:

$$\dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}, \mathbf{x}^{\text{alg}}, u, q), \tag{17a}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{x}^{\text{alg}}, u, q), \tag{17b}$$

where

- $\mathbf{x} \in \mathbb{R}^{128}$ is the vector of system states, i.e., uranium and acid concentrations:

$$
\begin{aligned}
\mathbf{x}_{1:16} &= [[U]_1^{\text{aM}} \quad [U]_2^{\text{aM}} \quad \cdots \quad [U]_{16}^{\text{aM}}]^T, \\
\mathbf{x}_{17:32} &= [[U]_1^{\text{oM}} \quad [U]_2^{\text{oM}} \quad \cdots \quad [U]_{16}^{\text{oM}}]^T, \\
\mathbf{x}_{33:48} &= [[U]_1^{\text{aD}} \quad [U]_2^{\text{aD}} \quad \cdots \quad [U]_{16}^{\text{aD}}]^T, \\
\mathbf{x}_{49:64} &= [[U]_1^{\text{oD}} \quad [U]_2^{\text{oD}} \quad \cdots \quad [U]_{16}^{\text{oD}}]^T, \\
\mathbf{x}_{65:80} &= [[H]_1^{\text{aM}} \quad [U]_2^{\text{aM}} \quad \cdots \quad [U]_{16}^{\text{aM}}]^T, \\
\mathbf{x}_{81:96} &= [[H]_1^{\text{oM}} \quad [U]_2^{\text{aM}} \quad \cdots \quad [U]_{16}^{\text{aM}}]^T, \\
\mathbf{x}_{97:112} &= [[H]_1^{\text{aD}} \quad [U]_2^{\text{aM}} \quad \cdots \quad [U]_{16}^{\text{aM}}]^T, \\
\mathbf{x}_{113:128} &= [[H]_1^{\text{oD}} \quad [U]_2^{\text{aM}} \quad \cdots \quad [U]_{16}^{\text{aM}}]^T;
\end{aligned}
$$

- $\mathbf{x}^{\text{alg}} \in \mathbb{R}^{32}$ is the vector of intermediate algebraic variables, i.e., uranium and acid concentrations:

$$
\begin{aligned}
\mathbf{x}_{1:16}^{\text{alg}} &= [[U]_{*,1}^{\text{aM}} \quad [U]_{*,2}^{\text{aM}} \quad \cdots \quad [U]_{*,16}^{\text{aM}}]^T, \\
\mathbf{x}_{17:32}^{\text{alg}} &= [[H]_{*,1}^{\text{aM}} \quad [H]_{*,2}^{\text{aM}} \quad \cdots \quad [H]_{*,16}^{\text{aM}}]^T;
\end{aligned}
$$

- $u = A_F$ is the manipulated variable (the feed solution flowrate);
- $q = O_E$ is the disturbance variable (the fresh solvent flowrate);
- $\mathbf{f} \in \mathbb{R}^{128}$ is the vector of mass balance equations;
- $\mathbf{g} \in \mathbb{R}^{32}$ is the vector of algebraic equations;
- $y := [U]_9^{\text{aD}} = \mathbf{x}_{41}$ is the controlled variable (the uranium concentration at the outlet of the scrubbing stage);
- $z := [U]_1^{\text{aD}} = \mathbf{x}_{33}$ is the constrained variable (the uranium concentration in the fission products).

Several factors drive this choice of the controlled variable: i) not all the system states can be measured online; ii) these measurements usually have high cost; iii) this particular variable's value is compatible with our sensor's measurement range; and iv) analysis on the process dynamics show that it is a good indicator for the solvent saturation level. For convenience, we denote the discrete state space model as:

$$\mathbf{x}(k+1) = \mathbf{f}_d \left(\mathbf{x}(k), \mathbf{x}^{\text{alg}}(k), u(k), q(k) \right), \tag{18a}$$

$$0 = \mathbf{g} \left(\mathbf{x}(k), \mathbf{x}^{\text{alg}}(k), u(k), q(k) \right), \tag{18b}$$

where $\mathbf{x}(k) := \mathbf{x}(kT)$, and T is the control sampling time of the overall system.

B. Process Dynamics Analysis and Problem Statement

1) Solvent Saturation: An essential objective of the control system is to ensure that the system operates at a desired solvent saturation, which can be indicated by the aqueous uranium concentration at stage 9's settler. In general, a higher saturation level offers a higher decontamination towards fission products. However, there is a critical situation in which the solvent saturation is at maximum, which is indicated as A_F^{s2} in Fig. 4. As solvent saturation increases, the profile aqueous uranium concentration in the settlers is shifted to the left, as illustrated in Fig. 5.

The over-saturation situation is usually unwanted since it risks losing uranium to fission products, which reduces our extraction efficiency (since it is necessary to recover as much uranium as possible). Therefore, in nominal operation conditions, keeping the system operating in the under-saturated condition is preferable. This allows a safety margin towards the over-saturated region. Note that operating under saturation conditions can be profitable in some cases, as it allows for the quick removal of other elements. Therefore, the control system must also adapt to variations in the set point and guarantee that not much uranium leaks into the fission products.
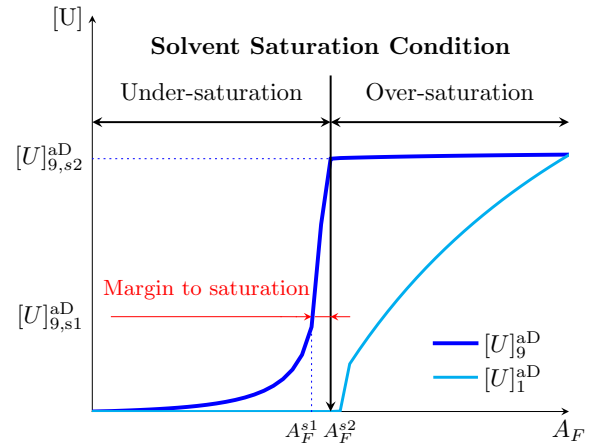

Fig. 4. Steady state relationship of feed solution flow rates and uranium concentrations. Note that $[U]_{9,s1}^{\text{aD}} \approx 37.5\%[U]_{9,s2}^{\text{aD}}$.
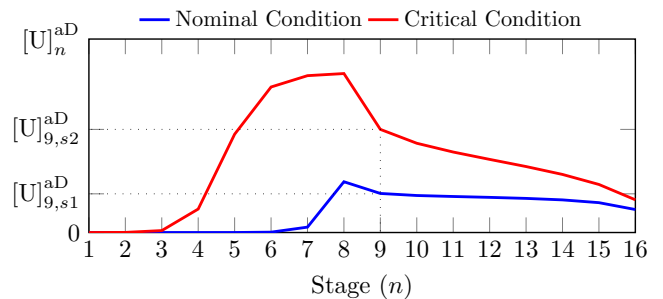

Fig. 5. Profiles of aqueous uranium concentration in nominal and critical cases.

2) Problem Statement: Considering the state space representation (17), the goal is to design an optimal control scheme that can regulate $y = [U]_9^{\text{aD}}$ to the desired set point $y_{\text{set}} = [U]_{9,\text{set}}^{\text{aD}}$ while accounting for unknown disturbances. Additionally, various constraints must be guaranteed during operation, such as aqueous uranium concentration in the extraction raffinates $[U]_1^{\text{aD}}$ (19a), overshoots (denoted by OS) (19b), control input magnitudes A_F (19c), and rates of control inputs (19d). The bounding values $A_F^{\min}$, $A_F^{\max}$, and $\Delta_{A_F}^{\max}$ are defined based on the equipment limits, the flow sheet. In addition, since the constraints (19a)-(19c) have vital impacts on the overall control performance, we consider them as hard constraints, and if necessary, we allow (19d) to be relaxed to guarantee the others:

$$[U]_1^{\text{aD}} \le [U]_{1,\text{tol}}^{\text{aD}}, \tag{19a}$$

$$\text{OS} = y/y_{\text{set}} - 1 \le \text{OS}^{\max}, \tag{19b}$$

$$A_F^{\min} \le A_F \le A_F^{\max}, \tag{19c}$$

$$|A_F(k+1) - A_F(k)| \le \Delta_{A_F}^{\max}. \tag{19d}$$

Furthermore, we accept a steady state error of about 5% of the set-point, i.e.,

$$e = |y - y_{\text{set}}| \le 5\% y_{\text{set}}. \tag{20}$$

III. ANN-BASED ADAPTIVE NMPC

This section presents the development of our proposed ANN-based adaptive NMPC to stabilize the process output y at a desired value y_{set}. The overall control system architecture is depicted in Fig. 6, where we first read measurements from the process and estimate the disturbance variable $\hat{q}(k)$. This action allows the controller to have the best updates of the system parameters for further predictions. Within the controller block, we first compute the desired control input u_{set} by solving the steady state condition:

$$\mathbf{0} = \mathbf{f}_c(\mathbf{x}, \mathbf{x}^{\text{alg}}, u, \hat{q}), \tag{21a}$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{x}^{\text{alg}}, u, \hat{q}), \tag{21b}$$

$$y = \mathbf{x}_{41} = y_{\text{set}}. \tag{21c}$$

The optimal control input is then computed by the NMPC, which uses an ANN as the predictor. Note that if we keep the control input as constant, i.e., $u = u_{\text{set}}$, and the condition (20) holds, we switch off the NMPC and apply directly u_{set} to the system. This strategy helps to reduce computation costs, especially when we are near the steady state.
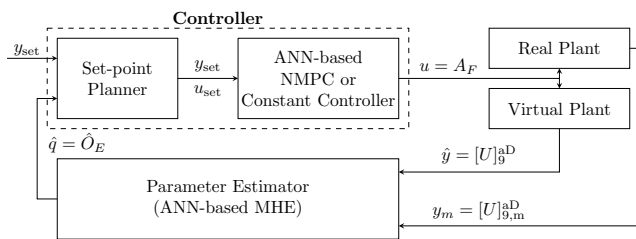

Fig. 6.   Proposed neural network architecture.

A. ANN architecture

Regarding the control problem stated in Sec. II-B.2, to implement the NMPC, two important outputs need to be predicted: $y := [U]_9^{\text{aD}}$ and $z := [U]_1^{\text{aD}}$, which are the aqueous uranium concentrations in the settlers of stage 9 and 1, respectively. In addition, since z is involved only in the constraint (19a), it is not necessary to predict precisely the value of z. Instead, we can use a binary variable $\bar{z} \in \{0, 1\}$, whose value is one if (19a) holds and is zero otherwise. Furthermore, regarding the state space representation (17), since y is the only online measurement that is available in the system, we aim to predict y and $\bar{z}$ based on the history of y, $u := A_F$, and $p := O_E$:

$$y(k+1) = f_y(\boldsymbol{\theta}(k)), \tag{22a}$$

$$\bar{z}(k+1) = f_{\bar{z}}(\boldsymbol{\theta}(k)), \tag{22b}$$

where the input vector $\boldsymbol{\theta}(k)$ is defined as:

$$\boldsymbol{\theta}(k) := \begin{bmatrix} \{y(i)\}_{i=k-N}^k \\ \{u(i)\}_{i=k-N}^k \\ \{q(i)\}_{i=k-N}^k \end{bmatrix}. \tag{22c}$$

The length of vector $\boldsymbol{\theta}(k)$ is $n_\theta = 3N$. Recall that A_F and O_E are the feed solution and fresh solvent flowrates, respectively.

The ANN architecture is depicted in Fig. 7, in which f_y is a linear regression model cascaded with an LSTM, and $f_{\bar{z}}$ is a binary classification model:

$$f_y(\boldsymbol{\theta}(k)) := \mathbf{A}\boldsymbol{\theta}(k) + f_{\text{LSTM}}(\boldsymbol{\theta}(k)), \tag{23a}$$

$$f_{\bar{z}}(\boldsymbol{\theta}(k)) := f_{\text{LG}}(\boldsymbol{\theta}(k)). \tag{23b}$$
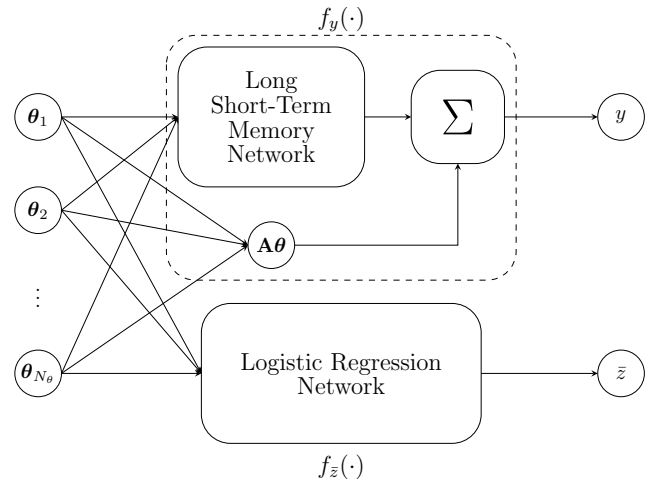

Fig. 7.   Proposed neural network architecture.

The idea behind f_y is to first predict y using a linear regression model, and then compensate for the linear prediction errors using the LSTM network. The training procedure for f_y is described as follows:

1) Run multiple simulations with the virtual plant, collect measurement histories, and construct the input-output sets $\mathcal{X}, \mathcal{Y}_1$:

$$\mathcal{X} := \left\{ \boldsymbol{\theta}^{(i)} \right\}_{i=0}^{N_{\text{sam}}}, \quad \mathcal{Y}_1 := \left\{ y^{(i)} \right\}_{i=0}^{N_{\text{sam}}}, \quad (24)$$

where N_{sam} is the number of data samples, $\boldsymbol{\theta}^{(i)}, y^{(i)}$ denote the ith input-output sample, respectively;

2) Train the linear regression model with $\mathcal{X}$. In other words, find $\mathbf{A}$ in (23a);

3) For each sample i, compute the linear regression error $e_1^{(i)}$ and construct the linear regression error set $\mathcal{E}_1$:

$$e_1^{(i)} := y^{(i)} - \mathbf{A}\boldsymbol{\theta}^{(i)}, \quad \mathcal{E}_1 := \left\{ e_1^{(i)} \right\}_{i=0}^{N_{\text{sam}}}; \quad (25)$$

4) Train the LSTM model with $\mathcal{X}$ and $\mathcal{E}_1$.

In addition, the logistic regression network $f_{\bar{z}}$ can be trained using the input set $\mathcal{X}$ and the output set $\mathcal{Y}_2$:

$$\mathcal{Y}_2 := \left\{ \bar{z}^{(i)} \right\}_{i=0}^{N_{\text{sam}}}. \quad (26)$$

B. Adaptive NMPC design

To allow NMPC to have the ability to adapt to disturbances, which are unknown variations in the feed solution flow rate ($q = O_E$), we develop a Moving Horizon Estimator to estimate this variable. This parameter estimator plays an essential role in the control strategy, i.e., it always allows the controller to be updated.

1) Parameter Estimator: At time step k, denote $\hat{q}(j|k), \hat{y}(j|k)$ the estimated values of q and y at time step j. Suppose that we want to estimate N_e latest values of q, which are denoted by $\{\hat{q}(i|k)\}_{i=k-N_e}^{k-1}$, and assume that $N + N_e$ latest measurements $\{y_m(i)\}_{i=k-N-N_e+1}^{k}$ are available. Then, the best estimates of $\{\hat{q}(i|k)\}_{i=k-N_e}^{k-1}$ can be obtained by solving the optimization problem:

$$\min_{\{\hat{q}(j|k)\}_{j=k-N_e}^{k-1}} \sum_{j=k-N_e+1}^{k} \lambda^{k-j} \left(\hat{y}(j|k) - y_m(j)\right)^2, \quad (27a)$$

such that $\forall j \in \mathbb{N}_{k-N_e:k-1}$:

$$\hat{y}(j+1|k) = f_y\left(\hat{\boldsymbol{\theta}}(j|k)\right), \quad (27b)$$

$$\hat{\boldsymbol{\theta}}(j|k) := \begin{bmatrix} \{\hat{y}(i|k)\}_{j-N}^{j} \\ \{u(i)\}_{j-N}^{j} \\ \{\hat{q}(i|k)\}_{j-N}^{j} \end{bmatrix}, \quad (27c)$$

$$\hat{\boldsymbol{\theta}}(k-N_e|k) := \begin{bmatrix} \{y_m(i)\}_{k-N_e-N}^{k-N_e} \\ \{u(i)\}_{k-N_e-N}^{k-N_e} \\ \{\hat{q}(i)\}_{k-N_e-N}^{k-N_e} \end{bmatrix}, \quad (27d)$$

where $\lambda \in (0,1)$ is the forgetting factor.

Remarks: The ANN-based MHE formulation developed above has N_e decision variable. If the MHE is formulated based on the discrete mathematical model (18), the optimization problem will be:

$$\min_{\hat{\mathbf{x}}(k-N_e), \{\hat{q}(j|k)\}_{j=k-N_e}^{k-1}} \sum_{j=k-N_e+1}^{k} \lambda^{k-j} \left(\hat{y}(j|k) - y_m(j)\right)^2. \quad (28)$$

subject to process dynamics (18). In other words, we will have to estimate the initial condition $\hat{\mathbf{x}}(k - N_e)$, which increases the number of decision variables by $128 \gg N_e$. Therefore, the optimization problem (III-B.1) requires a significant computation cost. An alternative approach was proposed in [3] where we keep track of the error history $\hat{y} - y_m$ to decide which should be the initial condition $\hat{\mathbf{x}}(k - N_e)$ for (III-B.1). Therefore, the ANN allows us to make predictions directly based on the measurements and reduce the optimization problem's complexity, hence enabling practical implementation.

2) NMPC Design: At time step k, denote $\{\hat{u}(i|k)\}_{i=k}^{k+N_p-1}$ the sequence of predicted control inputs from time step k to $k + N_p - 1$. Here, N_p denotes the prediction horizon length. In addition, assume that over the prediction horizon, the disturbance variable q is constant, and its value equals $\hat{q}(k-1) := \hat{q}(k-1|k)$, computed by the parameter estimator. Next, by denoting $\hat{y}(j|k)$ the predicted value of y at time step j, the optimal predicted control inputs are obtained by solving the following optimization problem:

$$\min_{\{u(i|k)\}_{i=k}^{k+N_p-1}} \sum_{j=k+1}^{k+N_p} w_p \left[\hat{u}(j|k) - y_{\text{set}}\right]^2 + w_q \tilde{y}^2\left(N_p|k\right)$$

$$+ \sum_{j=k}^{k+N_p-1} w_r \left[\hat{u}(j|k) - u_{\text{set}}\right]^2$$

$$+ \sum_{j=k}^{k+N_p-1} w_s \left[\hat{u}(j|k) - \hat{u}(j-1|k)\right]^2 \quad (29a)$$

subject to process dynamics, i.e. $\forall j \in \mathbb{N}_{k:k+N_p-1}$:

$$\hat{y}(j+1|k) = f_y\left(\hat{\boldsymbol{\theta}}(j|k)\right), \quad (29b)$$

$$\bar{z}(j+1|k) = f_{\bar{z}}\left(\hat{\boldsymbol{\theta}}(j|k)\right), \quad (29c)$$

$$\hat{\boldsymbol{\theta}}(j|k) := \begin{bmatrix} \{\hat{y}(i|k)\}_{k}^{j} \\ \{\hat{u}(i)\}_{k}^{j} \\ \{\hat{q}(i|k)\}_{k}^{j} \end{bmatrix}; \quad (29d)$$

constraints (19), i.e. $\forall j \in \mathbb{N}_{k:k+N_p-1}$:

$$\bar{z}(j|k) = 1, \quad (29e)$$

$$\hat{y}(j|k) \leq y_{\text{set}}\left(1 + OS^{\text{max}}\right), \quad (29f)$$

$$A_F^{\text{min}} \leq \hat{u}(j|k) \leq A_F^{\text{max}}, \quad (29g)$$

$$-\Delta_{A_F}^{\text{max}} \leq \hat{u}(j|k) - \hat{u}(j-1|k) \leq \Delta_{A_F}^{\text{max}}; \quad (29h)$$

and initial conditions:

$$\hat{u}(k-1|k) := u(k-1), \quad (29i)$$

$$\hat{y}(j|k) := y_m(j) \; \forall j \in \mathbb{N}_{k-N:k-1}, \quad (29j)$$

$$\hat{q}(i|k) := \hat{q}(k-1) \; \forall j \in \mathbb{N}_{k:k+N_p-1}. \quad (29k)$$

Note that w_p, w_q, w_r, w_s are weighting parameters chosen appropriately.

The optimization problems in MHE and NMPC can be solved using the extended PSO algorithm method developed

in [3]. The main extension lies in handling the constraints (19). Briefly, whenever a candidate solution (a particle) breaks the constraints, we re-initialize it until all constraints are guaranteed.

IV. CASE STUDIES

This section presents studies of the ANN-based Adaptive NMPC developed in Sec. III. The ANN was trained with *Google Colab Pro+* resources and simulations were done on a laptop with *Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz* and 8GB RAM. For simulation of the real and virtual plant (cf. Fig. 6), we use the CasADi toolbox [7] with IDAS from SUNDIALS ([8], [9]) as the DAEs solver. In addition, the ANNs are trained using scikit-learn ([10]) and Tensorflow ([11]). The NMPC and MHE weighting coefficients are chosen heuristically. We will study 3 scenarios: i) the start-up period, ii) the switching to the critical case, and iii) the perturbed case. Simulation parameters are given in Tab. II.

TABLE II
SIMULATION PARAMETERS.

Parameter	Value	Parameter	Value
T	0.5 hours	N_e	3 steps
OS^{max}	20%	λ	0.9
ε	5%	N_p	3 steps
N_{sam}	4E6 samples	$w_p = w_q$	$1/y_{set}$
Train-Validation-Test	98%-1%-1%	$w_r = w_s$	$1/u_{set}$

A. Data generation and ANN training

For the simulations in this paper, we use the mathematical model developed in Sec. II-A to generate data and train the ANN. Firstly, we made several transient simulations with different values of u and q. Then, we round all floating values to the six decimal places. For $N = 2$, the training data has about four million records. Since the data set is big, we divide the train, validation, and test sets with a ratio of 98%-1%-1%. The LSTM and logistic regression networks parameters and training results are given in Tab. III.

TABLE III
ANNs TRAINING INFORMATION.

	LSTM	Logistic Regession
Hidden layers	2	2
Units per layer	10	50
Learning rate	1E-03	1E-03
Batch size	1024	1024
Epochs	13	5
Final training loss	4.47E-4 (MAE)	0.21 (Cross Entropy)
Final validation metric	4.47E-4 (MAE)	0.9 (Accuracy)

B. Start up period

The system's initial condition is described as follows: We assume that uranium is only sent to the system since $t = 0$. In other words, before $t = 0$, only H and TBP are in the process, with system parameters equal to their nominal

values. Furthermore, we assume that at $t = 0$, the system is in a steady state (without uranium).

Simulation results of the start-up period are shown in Fig. 8. It can be seen that the proposed control strategy can effectively stabilize the system at the desired set point while guaranteeing all the constraints (19). Compared to the open loop controller, the controlled system significantly improves the settling time, up to 5 times faster. Furthermore, the switching condition (20) is active since $t = 8.5$h, which allows to turn off NMPC for a lower computation cost.
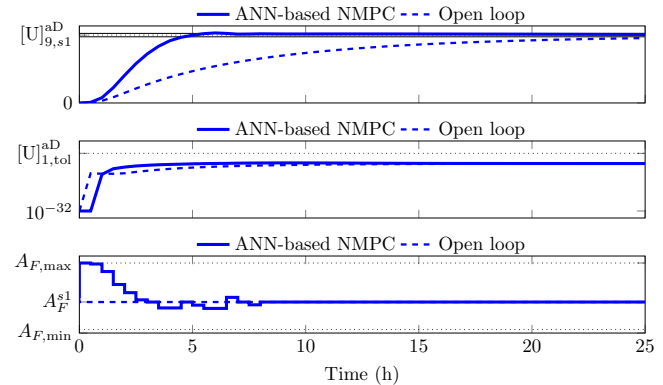

Fig. 8. Simulation results of the start up period.

C. Critical case

To study the ability of the proposed control strategy to adapt to variations in the set point value, we continue the simulation in Fig. 8 and assume that y_{set} increases to the critical value $[U]_{9,s2}^{aD}$ (cf. Fig. 4) at $t = 25$h and decrease to its nominal value $[U]_{9,s1}^{aD}$ since $t = 100$h. The simulation result of this case is depicted in Fig. 9. It can be seen that the controlled system can rapidly track the set point values while guaranteeing all the constraints. Furthermore, as expected, the uranium edge is shifted to the left when the solvent saturation increases, as depicted in Fig. 10.
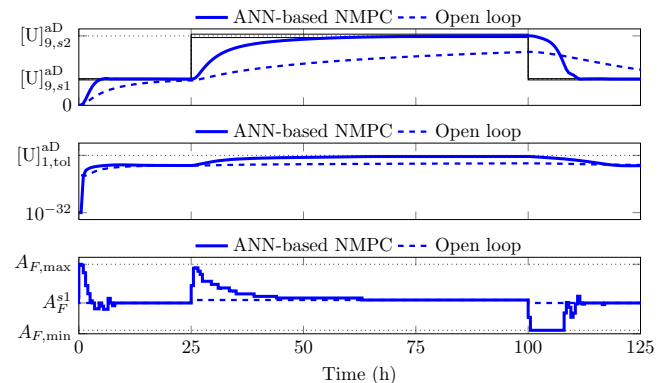

Fig. 9. Simulation results when set point varies, i.e. switching to critical condition.

D. Perturbed case

To study the disturbance rejection ability of the proposed control strategy, we analyze the case in which there are
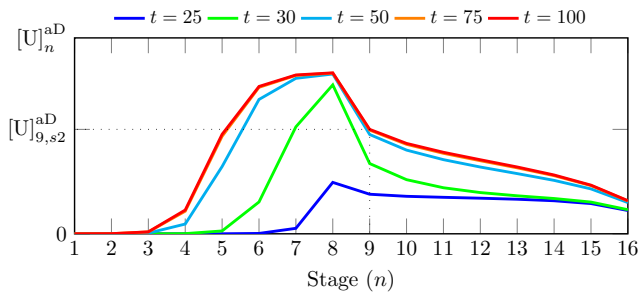
Fig. 10. Profile of aqueous uranium concentration in settlers at certain time steps of the simulation in Fig. 9.

variations in the fresh solvent flow rate O_E. The simulation result for this case is shown in Fig. 11. It can be seen that the developed ANN-based MHE is capable of estimating the unknown disturbance, hence allowing the controller to be always updated with the process. Consequently, the controller can effectively adapt to unknown disturbances.

Additionally, we notice that there is always a delay of around 0.5h for the ANN-based MHE to make a successful estimation. This phenomenon can be explained by the slow dynamics of the process and the nature of our estimation algorithm. Regarding the MHE formulation (27), the disturbance variable q is estimated based on the output errors between the actual plant and its digital twin $\hat{y} - y_m$. However, since the process dynamics are slow, it takes time for the effects of the disturbance to be reflected in the output, i.e., the error $\hat{y} - y_m$ becomes sufficiently large. As a consequence, there are delays in parameter estimation. Furthermore, there are mismatches between the NMPC prediction model and the actual plant within these periods. Therefore, we noticed some high overshoots at certain time steps: 20, 40, and 60h. However, they are acceptable.
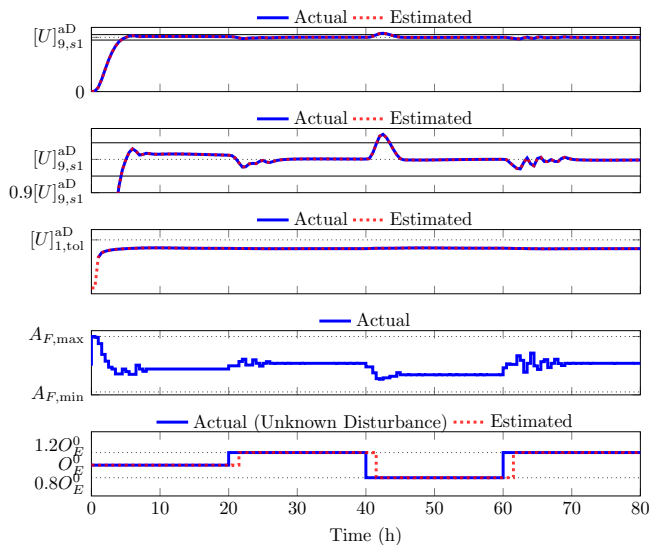

Fig. 11. Simulation results of the perturbed case. Note that the first two subplots both show the behavior of $[U]_9^{aD}$ over time, but at different scales to give a better examination of this variable.

V. CONCLUSIONS

This paper presented a neural network-based adaptive control strategy for the uranium extraction-scrubbing operation in the PUREX process. The artificial neural network (ANN), which included the long-short term memory network, linear regression, and logistic regression networks, played an essential role in the control strategy. Specifically, it was the predictor in the Nonlinear Model Predictive Control (NMPC) and the estimator in the Moving Horizon Estimation (MHE). The MHE and NMPC optimization problems were solved by an enhanced Particle Swarm Optimization (PSO) algorithm, proposed in [3]. The ANN helped reduce the complexity of the MHE optimization problem, thus allowing online implementation ability. Multiple simulation case studies showed that the developed control strategy was a candidate solution for the control problem: it could stabilize the system at a desired set-point while guaranteeing all the constraints, even in critical conditions or under unknown disturbances. Future developments included studies on other uncertainties handling, online learning, stability, robustness, and experimental implementation. Thorough comparisons with and without ANN were to be carried out to assess the benefits of introducing ANN.

REFERENCES

[1] J. Bisson, B. Dinh, P. Huron, and C. Huel, "PAREX, A Numerical Code in the Service of La Hague Plant Operations," *Procedia Chemistry*, vol. 21, pp. 117–124, May 2016.

[2] D.-T. Vo, I. Prodan, L. Lefèvre, V. Vanel, S. Costenoble, and B. Dinh, "Nonlinear model predictive control for uranium extraction-scrubbing operation in spent nuclear fuel treatment process," in *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics*. SCITEPRESS - Science and Technology Publications, 2023.

[3] ——, "Pso-based adaptive nmpc for uranium extraction-scrubbing operation in spent nuclear fuel treatment process," 2023, submitted to The 12th IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM 2024). [Online]. Available: doi.org/10.48550/arXiv.2402.03656

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[5] M. Jung, P. R. da Costa Mendes, M. Önnheim, and E. Gustavsson, "Model predictive control when utilizing lstm as dynamic models," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106226, 2023.

[6] B. Dinh, P. Baron, and J. Duhamet, "Modeling the PUREX Process: The PAREX Code," in *Treatment and Recycling of Spent Nuclear Fuel*. Le Moniteur, 2008, pp. 55–70.

[7] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[8] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[9] D. J. Gardner, D. R. Reynolds, C. S. Woodward, and C. J. Balos, "Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, 2022.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, and al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] M. Abadi, A. Agarwal, P. Barham, and al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.