2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Redundancy-Aware Physics Informed Neural Networks (R-PINNs) based Learning of Nonlinear Algebraic Systems with Non-Measurable States

Mayank Shekhar Jha*, Hugues Garnier, Didier Theilliol

Abstract— **The paper presents Redundancy-Aware Physics Informed Neural Networks (R-PINNs) for learning of unknown model parameters of nonlinear algebraic systems in continuous time with non-measurable state variables. R-PINNs accomplish the learning task in presence of non-measurable states of the system by incorporating input-output representation of the *a priori* available physics based laws, generally in form of nonlinear differential (partial) equations within the NN based learning procedure, leading to learning of a set of optimal parameters that determine the optimal mapping between input-output data while adhering to the known physics. Analytical Redundancy Relationships (ARRs) are able to express input-output representation of system using solely the measured/known variables by exploring the redundancy within the analytical structure of the system. The paper proposes a methodology that includes ARR derivation and suitable integration within PINNs framework to develop R-PINNs. Mathematically rigorous novel proofs on uniform and ultimate boundedness (UUB) of the output and parametric estimation errors in Lyapunov sense is provided. Finally a DC motor enabled friction drive system based simulation study is presented to demonstrate the effectiveness of the approach.**

I. INTRODUCTION

The predictive ability of physics based models is dependent on the available knowledge of governing laws, identification / calibration of model parameters under given conditions and the model uncertainties, to name a few. With the advent of modern deep learning, there has been an unprecedented progress in the field of data driven modelling [9] and nonlinear identification [12]. Two factors that remain extremely crucial for accurate learning are: large high quality data sets (rich and diverse) and immense computing power. While the former leads to variance reduction, the latter, allows for large models to be trained leading to a certain bias reduction. However, the availability of usable high quality data is not always guaranteed leading to less generalization capability and high variance. Moreover, model calibration under limited data introduces high bias. To reduce model bias and increase generalization capability, various approaches have been proposed including learning of governing equations and generalised coordinates [3], semi physical models and recently, the focus has been on the incorporation of physics based knowledge within the neural network (NN) structure [14].

Authors are with Centre de Recherche en Automatique de Nancy (CRAN), UMR 7039, CNRS, University of Lorraine, 54506 Vandoeuvre Cedex, France.
*corresponding author
mayank-shekhar.jha@univ-lorraine.fr
[2] hugues.garnier@univ-lorraine.fr
[2] didier.theilliol@univ-lorraine.fr

Past semi decade has witnessed an ever growing interest and extensive work using physics informed neural networks (PINNs) for the discovery of governing laws (forward problems) as well as learning of model parameters using the available data (inverse problems) in several fields of engineering including discovery of Navier-Stokes equation [14], Schrodinger equation [14], Burger [14], Hamilton and Euler-Lagrange equation [14] to name a few. Readers are referred to [5] for the latest comprehensive review on PINNs and their application in various engineering domains starting from 2017. Most existing NN(deep) based approaches for parametric learning or system identification, target learning of the mapping function between input and output data (black-box approach) that neglect inclusion of physics laws within learning paradigm[10][12]. On the other hand, only very few notable works based on variants of PINNs have appeared recently for system identification [16] .
In general, PINNs based approaches accomplish the learning task by incorporating the *a priori* available physics based laws, generally in form of nonlinear differential (partial) equations within the NN based learning procedure, leading to learning of a set of optimal parameters that determine the optimal mapping between input-output data, while adhering to the known physics. Generally the feature that makes a classical NN into PINNs is the incorporation of nonlinear differential(partial) equation(s) within the loss function leading to minimisation of the corresponding residual. The loss is indicative of consistency (measure of difference from zero). The latter must be minimised to accomplish the learning task. These residuals are constructed using system variables that are usually assumed to be measurable/available. However, in reality, dynamical systems may have only a set of states measurable. In this context, existing PINNs approaches neglect the problem arising from the presence of unknown/non-measurable state variables. This becomes crucial especially for the construction of nonlinear differential residual as the presence of unknown state variables calls for their estimation along with unknown parameters. As per the authors, there is no existing work that formally enables inclusion of only known (measurable) information along with unknown parameters within PINNs for parametric learning.

On the other hand, Analytical Redundancy Relationships (ARRs) are able to express input-output representation of system using solely the measured/known variables, by exploring the redundancy within the analytical structure of the system [15]. Under nominal functioning of the system, ARRs

979-8-3503-0123-6/23/$31.00 ©2023 IEEE

must be consistent and as such, have been extensively exploited for residual generation. ARRs have been extensively investigated in Fault detection and isolation (FDI) literature wherein the main utility remains in generation of residuals using only the measured variables [17][1]. It must be noted that ARRs are closely related to input-output representation of dynamic systems and as such, ARR derivation techniques have been heavily inspired by the former that involve only measured variables of system [17].

For nonlinear systems, ARR derivation principally involves state elimination techniques [1], broadly based on three methods: *elimination theory* that rests primarily on Euclidean division and successive derivation [6], *Grobner bases* which uses Euclidean division and the computation of the so-called S-polynomials [4] and *Characteristic sets* (also called Ritt's algorithm) that involves direct elimination of state to find minimum order differential input-output constraints [7]. In the case of algebraic dynamic systems, elimination theory provides the most straightforward and intuitive tool for state elimination [1].

To bridge the existing scientific gap, this paper blends the benefits of ARR with PINNs framework in order to address the parametric learning problem for dynamical systems with non-measurable state variables (section II). To that end, as novelty, the paper presents Redundancy aware PINNs (R-PINNs) for nonlinear algebraic systems in continuous time with non-measurable state variables (section III). Therein, the redundancy within system variables is exploited to construct ARRs which are then suitably integrated within PINNs structure to learn unknown parameters. The paper also presents mathematically rigorous novel proofs on uniform and ultimate boundedness (UUB) of the output and parametric estimation errors in Lyapunov sense and provides a DC motor enabled friction drive system based simulation study to demonstrate the effectiveness of the approach (section IV).

II. Problem formulation

Consider a dynamical system in continuous time with physics laws given by a set of nonlinear differential algebraic equations (DAEs) [2][11] leading to state space model as:

$$\begin{aligned} \dot{x} &= f(x, u, \lambda) \\ y &= g(x, u) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n \in \Omega$ is the state variable vector, which is non available, Ω being a compact set, $\dot{x}$ is the time derivative of state variable vector, $u \in \mathbb{R}^m \in \Omega$ is the control input vector with $\lambda \in \mathbb{R}^d$ are the unknown parameter vector, $y \in \mathbb{R}^p$ is the output measurement vector. This work considers $f(\cdot)$ and $g(\cdot)$ as nonlinear DAEs such that $f(\cdot)$ and $g(\cdot)$ are C^1 where C^1 denotes the set of continuously differentiable functions over the compact set Ω.

Remark 1. Restricting the known system dynamics of (1) to nonlinear DAEs makes it possible to use mathematical properties from differential algebra. Moreover, the assumption is not as restrictive as non-polynomial non-linearity can be transformed to equivalent polynomial DAEs. For example, y =cos(x) can be equivalently written as a DAE as $y^2\dot{x}^2 + \dot{y}^2 = \dot{x}^2$.

1) Known and unknown variables: The system variables sets $\mathcal{Z} = x \cup y \cup u$ can be decomposed in two categories: measurable variables $\mathcal{K} = y \cup u$ and unknown non-measurable variable(s) $\mathcal{X} = x$.

2) Residual Generation: Using traditional PINNs based approach, the nonlinear differential residual corresponding to system (1) will be generated as:

$$\dot{x} - f(x, u, \lambda) \tag{2}$$

Readers are referred to [5] for more details. The existing PINNs based approaches oblige the implication of state variables x into the residual expression. As such, in presence of unknown(partially) variables x which is the case for most of the nonlinear dynamic systems, it becomes imperative to resort to state estimation methods in order to facilitate parametric estimation approaches.

III. Proposed Methodology

To alleviate the difficulty arising from unknown state variables, this paper proposes a variable elimination based approach via generation of ARRs by exploiting the analytical redundancy in the system mathematical model (1). Elimination of x leads to transformation of the original mathematical model to an equivalent input-output representation of the system. To that end, this paper proposes ARR derivation followed by construction of R-PINNs.

A. ARR derivation with known inputs

Consider the system (1), wherein the inputs are known and all functions are differentiable (as (1) is DAE). Then, the steps below are followed to obtain the ARRs [1].

Step I: **Derivation of outputs.** It is possible to obtain the total derivative of the output signal as:

$$\dot{y} = \frac{\partial g(\cdot)}{\partial x}\dot{x} + \frac{\partial g(\cdot)}{\partial u}\dot{u} \tag{3}$$

Substituting (1) in (3) leads to:

$$\begin{aligned} \dot{y} &= \frac{\partial g(\cdot)}{\partial x} f(x, u, \lambda) + \frac{\partial g(\cdot)}{\partial u}\dot{u} \\ &:= g_1(x, \bar{u}^{(1)}, \lambda) \end{aligned} \tag{4}$$

where for the sake of ease of notation, $\bar{u}^{(1)} = (u', \dot{u}')'$. Then, through iterative derivation up-to some order of derivation q one obtains:

$$\bar{y}^{(q)} = G^{(q)}(x, \bar{u}^{(q)}, \lambda) \tag{5}$$

with $\bar{u}^{(q)} \in \mathbb{R}^{(q+1)m}$, is set of $(q+1)p$ DAE based constrains that must be satisfied under nominal functioning of the system.

Step II: **Elimination of the state.** It is assumed that $(q+1)p > n$ and the Jacobian $\frac{\partial G^{(q)}(\cdot)}{\partial x}$ is of rank n (see [15][6] for a more mathematically rigorous treatment omitted here). Then, (5) can be decomposed into two subsystems:

$$\begin{pmatrix} \bar{y}_I^{(q)} \\ \bar{y}_{II}^{(q)} \end{pmatrix} - \begin{pmatrix} G_I^{(q)}(x, \bar{u}^{(q)}, \lambda) \\ G_{II}^{(q)}(x, \bar{u}^{(q)}, \lambda) \end{pmatrix} = 0 \tag{6}$$

where subsystem $(\bar{y}_I^{(q)}, G_I^{(q)}(x, \bar{u}^{(q)}, \lambda)) \in \mathbb{R}^n$ and $(\bar{y}_{II}^{(q)}, G_{II}^{(q)}(x, \bar{u}^{(q)}, \lambda)) \in \mathbb{R}^{(q+1)p-n}$ Then, x can be computed as a function of other variables that are known leading to:

$$x = \phi(\bar{y}_I^{(q)}, \bar{u}^{(q)}, \lambda) \tag{7}$$

where $\phi(\cdot) \in \mathbb{R}^n$ is some nonlinear DAE.

Further, using *implicit function theorem* [8] one can substitute (7) in (6) to obtain (5) sensitive to only known variables as:

$$\begin{aligned} r(\bar{y}^{(q)}, \bar{u}^{(q)}, \lambda) &= \bar{y}_{II}^{(q)} - G_{II}^{(q)} \left(\phi(\bar{y}_I^{(q)}, \bar{u}^{(q)}, \lambda), \bar{u}^{(q)}, \lambda \right) \\ &= 0 \end{aligned} \tag{8}$$

where $r(\bar{y}^{(q)}, \bar{u}^{(q)}, \lambda)$ is a set of constraints or ARRs that are sensitive to only known system variables and unknown parameters that must be satisfied.

B. R-PINNs

To address the inverse problem for system (1), i.e. estimation of unknown parameters λ given measured data y, this paper proposes the latter, along with ARRs (input-output representation of original system (1)) to constitute R-PINNs as:

$$\begin{aligned} y &= h(z) \\ r(\bar{y}^{(q)}, \bar{u}^{(q)}, \lambda) &= 0 \end{aligned} \tag{9}$$

where $z := [u_1, u_2, ...u_m; t]$ is the input space-time vector with $z \in \Omega$, $h(\cdot) \in \mathbb{R}^m$ is a hidden nonlinear function that maps the available measured data to input time series and $r(\cdot)$ are the derived set of ARRs in (8).

To solve (9), this paper proposes R-PINNs that considers as input, z to computationally predict the hidden relation $h(z)$ using a NN, parameterised by a set of parameters $\theta \in \mathbb{R}^{(m+1)\times p}$ leading to the approximation:

$$\hat{y}_\theta = \hat{h}_\theta(z) \tag{10}$$

where $(\cdot)_\theta$ denotes the NN approximation realised with parameter set θ. Further, since ARRs must be satisfied at all times, an approximation of $r(\cdot)$ can be generated using $\hat{y}_\theta$ and its (partial)derivative(s) as:

$$r_{\hat{\theta},\lambda}(\hat{\bar{y}}_\theta^{(q)}, \bar{u}^{(q)}, \lambda) = 0 \tag{11}$$

Consider the R-PINNs parameter set Θ as tuple of shared parameters $\Theta = (\theta, \lambda)$ that consists of NN approximation parameters θ and unknown system parameters λ. Then, an R-PINNs parameter set Θ^*can be found through minimisation of total loss funciton $\mathcal{L}(\Theta)$ as:

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(\Theta) \tag{12}$$

where the R-PINNs based loss function can be defined as:

$$\mathcal{L}(\Theta) = (w_y \mathcal{L}_y + w_r \mathcal{L}_r) \tag{13}$$

considers the weighted sum of: loss $\mathcal{L}_y$ due to true measurement data y and its respective estimation by NN; and loss $\mathcal{L}_r$ based on the corresponding estimated residual which must be equal to zero (11), with w_y and w_r being the weight coefficients.
The building blocks of proposed R-PINNs are shown in Fig.1.

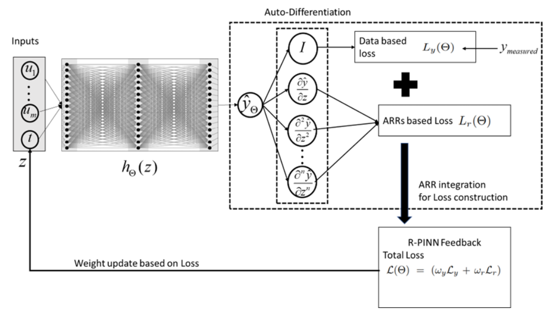

Fig. 1: Proposed R-PINNs Architecture.

Remark 2. R-PINNs based methodology estimates the R-PINNs shared parameter set $\Theta = (\theta, \lambda)$ simultaneously to give optimal parameters corresponding to NN based approximation error and residual based approximation error.

The loss $\mathcal{L}_y$ represents the mismatch between the available measurement data y and its approximation by NN $\hat{y}_\theta$ and attempts to fit the data to learn accurately the mapping $\hat{h}_\theta$ over Ω . On the other hand, $\mathcal{L}_r$ represents the difference of estimated residual (value of ARRs) from zero in order to "orient" the learning in such a way that known physical laws are adhered to. $\mathcal{L}_y$ and $\mathcal{L}_r$ can be expressed as:

$$\begin{aligned} L_y(\Theta) &= \int_\Omega y - \hat{h}_\Theta(z) dz \\ L_r(\Theta) &= \int_\Omega r_{\hat{\theta},\lambda}(\hat{\bar{y}}_\theta^{(q)}, \bar{u}^{(q)}, \lambda) dz \end{aligned} \tag{14}$$

Consider N_d measurement data points such that $\{z_i, y_i\}_{i=1}^{N_d}$ are the observation points. Then, typically, these losses can be numerically obtained using mean squared error (MSE) formulation as:

$$\begin{aligned} L_y(\Theta) &= MSE_y = \frac{1}{N_d} \sum_{i=1}^{N_d} \left\| y_i - \hat{h}_\theta(z_i) \right\|^2 \\ L_r(\Theta) &= MSE_r = \frac{1}{N_d} \sum_{i=1}^{N_d} \left\| (\hat{\bar{y}}_\theta^{(q)}, \bar{u}^{(q)}, \lambda) \right\|^2 \end{aligned} \tag{15}$$

As the ARRs are sensitive to only known/measured data, (12) can be estimated over the domain Ω. It should be noted that the derivatives (partial) required to construct the ARRs (11) can be obtained directly using Automatic Differentiation (AD)procedure [5][14] that evaluates derivatives(partial) in an algorithmic fashion through analytical evaluation. Usage of AD is extensively common in the state of the art works associated with NN, to accomplishing back-propagation based NN weight update. Almost all available Deep Learning packages, such as Pytorch/TensorFlow, provide numeric estimates of derivatives with high accuracy.

C. Boundedness of errors

Universal approximation property of feed-forward NN, enables expression of output of R-PINNs $y_\Theta(z)$ in an exact manner on a compact set Ω, $\forall z \in \Omega$:

$$y = \Theta^{*T}\sigma(z) + \varepsilon_{N_o}(z) \tag{16}$$

where $\Theta^* = (\theta^*, \lambda^*) \in \mathbb{R}^{N_o}$ is the ideal weight vector, N_o is the number of neurons, $\sigma(z) = (\sigma_1(z)), \sigma_1(z), ...\sigma_{N_o}(z))^T \in \mathbb{R}^{N_o}$ is the nonlinear activation function (sigmoid or tanh) with $\sigma_j(z) \in \mathcal{C}^1(z)$, $\sigma(0) = 0$, and the set of activation functions $\{\sigma_j(z)\}_{j=1}^{j=N_o}$ are linearly independent (to assure persistence of excitation) and $\varepsilon_{No}$is a bounded R-PINNs function that represents the reconstruction error. The derivative of 16 can be given as:

$$\dot{y} = \nabla\sigma^T(z)\Theta^* + \nabla\varepsilon_{N_o}(z) \tag{17}$$

where $\nabla\sigma(z) = \frac{\partial\sigma(z)}{\partial(z)}$ and $\nabla\varepsilon_{N_o}(z) = \frac{\partial\varepsilon_N o(z)}{\partial(z)}$. R-PINNs based output estimate can be expressed as:

$$\hat{y} = \hat{\Theta}^T\sigma(z) \tag{18}$$

where $\hat{\Theta}$ is the estimation of Θ^*. Denote weight estimation and output estimation errors respectively as $\tilde{\Theta} = \Theta^* - \hat{\Theta}$ and $\tilde{y} = y^* - \hat{y}$. The parameters $\hat{\Theta}$ of PINNs are updated using gradient descent as:

$$\dot{\hat{\Theta}} = -l_r\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}} \tag{19}$$

where $0 < l_r < 1$ is a non negative learning rate.

Assumption 1. The parameters of R-PINNs are upper bounded by known constants such that $||\Theta|| \leq \Theta_M$ and $\forall z \in \Omega$, $||\sigma(z)|| \leq \sigma_M$, $||\nabla\sigma(z)|| \leq \sigma_{\nabla M}$. Moreover, the R-PINNs reconstruction errors are upper bounded by known constants such that $||\varepsilon_{N_o}|| \leq \varepsilon_M$ and $||\nabla\varepsilon_{N_o}|| \leq \varepsilon_{\nabla M}$

Assumption 2. The gradient of loss $\mathcal{L}(\Theta)$ with respect to parameters is bounded such that $||\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}}|| \leq \delta_{\mathcal{L}_\Updownarrow}$.

It is noted that these assumptions are not as restrictive as various saturating nonlinear activation functions that satisfy these properties such as sigmoid or tanh functions exist.

Theorem 1. *Under the assumptions 1 and 2, given the R-PINNs weight update law as (19), then the R-PINNs based estimation error as well as parameter estimation errors are guaranteed to be uniformly and ultimately bounded (UUB).*

Proof: Consider a Lyapunov function candidate:

$$L(z) = L_1 + L_2 \tag{20}$$

where

$$\begin{aligned} L_1 &= 1/2\tilde{y}^T\tilde{y} \\ L_2 &= 1/2\tilde{\Theta}^T\tilde{\Theta} \end{aligned} \tag{21}$$

Taking derivative of L_1 we have:

$$\begin{aligned} \dot{L_1} &= \tilde{y}^T\dot{\tilde{y}} \\ &= (y - \hat{y})^T.(\dot{y} - \dot{\hat{y}}) \\ &= (\Theta^{*T}\sigma(z) + \varepsilon_{No} - \hat{\Theta}\sigma(z))^T \\ &(\nabla\sigma^T(z)\Theta^* + \nabla\varepsilon_{No} - \nabla\sigma^T(z)\hat{\Theta} - \dot{\hat{\Theta}}^T\sigma(z)) \\ &= (\tilde{\Theta}\sigma(z) + \varepsilon_{No})^T(\nabla\sigma^T(\tilde{\theta}) + \nabla\varepsilon_{No} - \dot{\hat{\Theta}}^T\sigma(z)) \\ &= \sigma^T(z)\tilde{\Theta}^T\nabla\sigma^T(z)\tilde{\Theta} + \sigma^T\tilde{\Theta}^T\nabla\varepsilon_{No} \\ &+ \sigma^T\tilde{\Theta}^T l_r\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}}\sigma(z) + \varepsilon_{No}^T\nabla\sigma^T\tilde{\Theta} + \varepsilon_{No}^T\nabla\varepsilon_{No} \\ &+ \varepsilon_{No}^T l_r\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}}\sigma(z) \end{aligned} \tag{22}$$

Using the Cauchy-Schwartz inequality $a^Tb \leq (1/2)a^Ta + (1/2)b^Tb$ for any two vectors a and b and some mathematical manipulations, (22) yields :

$$\begin{aligned} \dot{L_1} \leq &||\sigma(z)||.||\nabla\sigma(z)||.||\tilde{\Theta}||^2 \\ &+ ||\sigma^T(z)||.(\tilde{\Theta}^T\Theta + \nabla\varepsilon_{No}^T\nabla\varepsilon_{No}) \\ &+ ||\sigma^T(z)||.(\Theta^T\Theta + \sigma^T(z)\sigma(z))l_r||\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}}|| \\ &+ \varepsilon_{No}^T(\nabla\sigma^T\nabla\sigma + \tilde{\Theta}^T\tilde{\Theta}) \\ &+ ||\varepsilon_{No}||.||\nabla\varepsilon_{no}|| \\ &+ ||\varepsilon||.||l_r\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}}||.||\sigma(z)|| \end{aligned} \tag{23}$$

Then, taking into account the assumptions 1 and 2, following is obtained:

$$\begin{aligned} \dot{L_1} \leq &(\sigma_M + \sigma_{\nabla M} + \sigma_M + \sigma_M)||\tilde{\Theta}||^2 + \sigma_M\sigma_{\nabla M}^2 \\ &+ \sigma_M^3 l_r\delta_{\mathcal{L}_\Updownarrow} + \varepsilon_M\sigma_{\nabla M}^2 \\ &+ \varepsilon_M\varepsilon_{\nabla M} + \varepsilon_M l_r\delta_{\mathcal{L}_\Updownarrow}\sigma_M \end{aligned} \tag{24}$$

Similarly, consider derivative of L_2 as:

$$\begin{aligned} \dot{L_2} &= \tilde{\Theta}^T\dot{\tilde{\Theta}} \\ &= (\Theta^* - \hat{\Theta})^T.(\dot{\Theta^*} - \dot{\hat{\Theta}}) \\ &= (\Theta^* - \hat{\Theta})^T(l_r\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}}) \end{aligned} \tag{25}$$

Using Cauchy-Schwartz inequality and some mathematical manipulations, one obtains:

$$\dot{L_2} \leq (1/2)||\tilde{\Theta}||^2 + (1/2)||\frac{\partial\mathcal{L}(\hat{\Theta})}{\partial\hat{\Theta}}|| \tag{26}$$

Putting (24) and (26) into (20), one obtains $\dot{L}(z) \leq 0$ which implies that $\tilde{y}$ and $\tilde{\Theta}$ are UUB over the compact set Ω if $\tilde{y}(0) = 0$ and $\tilde{\Theta}(0) = 0$ are bounded over Ω. The latter being true, the proof is completed hereby.

IV. Simulations and Results

To demonstrate the effectiveness of the approach, a friction drive system composed by a driver device (a DC motor) and a driven device (a wheel) is considered for simulation based analysis. The system dynamics and details can be consulted in [13] . Here, the DC motor system along with nonlinear friction drive dynamics is considered in form of DAE with states $x(t) = [i(t),\, \omega_1(t),\, \omega_2(t)]^T$ as:

$$L\frac{di}{dt} = U - Ri - K_e\omega_1 \tag{27}$$

$$J_1\frac{d\omega_1}{dt} = K_e i - b_1\omega_1 + T_L \tag{28}$$

$$J_2\frac{d\omega_2}{dt} = T_s - b_2\omega_2 \tag{29}$$

where $U(t)$ is the input voltage, R=0.08 Ohms is rotor resistance, $i(t)$ is the motor current K_e=0.1 being the electromotive force constant, L=0.005 Henry being the electric inductance, J_1 and J_2 are respectively moments of inertia of the driver and the driven device, ω_1 and ω_2 are respectively the angular speeds of driver and driven device and b_1, b_2 are the viscous friction coefficiencts, T_L is a load torque seen from the driver side, T_S is a source of torque observed from the driven side. Both torques can be written in terms of the contact force F_c as:

$$\begin{aligned} T_L(t) &= -F_c r_1 \\ T_s(t) &= F_c r_2 \end{aligned} \tag{30}$$

where F_c is the contact force proportional to the relative tangential speed at the contact level as:

$$F_c = \alpha(r_1\omega_1 - r_2\omega_2)^2 \tag{31}$$

with r_1, r_2 being the radii of driver and driven device respectively, α being a model parameter known as *contact quality coefficient* generally non-measurable and must be estimated. Moreover, viscous friction coefficient b_1 is considered as the unknown parameter. Thus, unknown parameter set is $\lambda = [\alpha, b_1]^T$. Motor current $i(t)$ and driven device speed ω_2 are assumed measurable, leading to the following output measurement vector as:

$$y(t) = [i(t), \omega_2(t)]^T \tag{32}$$

The objective is to estimate α and b_1 using some available data $y(t)$ and ARR based input-output representation using R-PINNs. ARRs are derived through elimination of non-measurable variable ω_1. To that end, (27) is considered so that $\omega_1 = (L\frac{di}{dt} - U + Ri)/K_e$ and substituted in (28) as well as (27) is differentiated with respect to time to give $\dot{\omega}_1$ and substituted in (28) leading to ARR1 as:

$$\begin{aligned} ARR1: K_e i + (1/K_e)(U - Ri - L\frac{di}{dt})(-b_1 - r^2\alpha) \\ -(J_1/K_e)(\dot{U} - R\frac{di}{dt} - L\frac{d^2 i}{dt^2}) - \omega_2 r_1 r_2\alpha = 0 \end{aligned} \tag{33}$$

Similarly, substituting ω_1 from (27) in (28) leads to ARR2 as:

$$\begin{aligned} ARR2: -b_2\omega_2 - J_2\frac{d\omega_2}{dt} \\ + r_2\alpha(r_1(1/K_e)(U - Ri - L\frac{di}{dt}) - r_2\omega_2) = 0 \end{aligned} \tag{34}$$

As can be seen, ARR1 and ARR2 are sensitive to unknown parameters $\lambda = [\alpha, b_1]^T$ and only the measurable state variables. A simulated data set comprising of $N_d = 10000$ data points is considered for learning. The simulation is done in continuous time using Runge-Kutta solver with fixed step of integration as 0.0001. Additive white Gaussian noise with signal to noise ratio (SNR) as 50db is considered as noise over measured variables. System input $U(t)$ with white additive Gaussian noise with SNR 50 db, in form of step signals with variable amplitudes and period is injected within the system model to generate the data set. To approximate the output (32), a fully connected NN having 8 hidden layers with neurons in-order as [188,144,64,32,16,8,4,2] and all activation functions as Hyperbolic tangent $tanh(\cdot)$ is considered, leading to the approximation of $y(t)$ as $\hat{y(t)}$. Then, the total loss function (13) constitutes of $L_y(\Theta) = (1/N_d)\sum_{i=1}^{N_d} \left\|y(t) - \hat{y(t)}\right\|^2$ and $L_r(\Theta) = (1/N_d)\sum_{i=1}^{N_d} \|ARR1 + ARR2\|^2$ with the importance weights $w_y = w_r = 0.5$ to attribute equal importance to the former two.

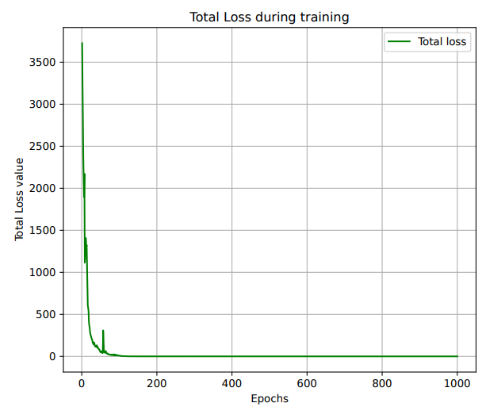

Fig. 2: Total loss per epoch of training

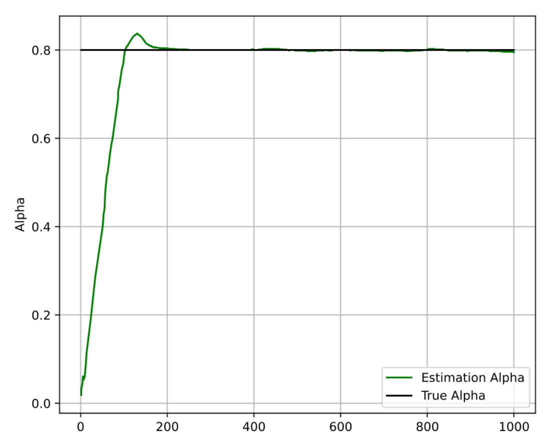

Fig. 3: Estimation of Alpha

The training of R-PINNs is done till 1000 epochs (training iteration) wherein the total loss function saturates to a value close to zero and demonstrates change less than 10^{-6} in value (see Fig. 2). Within each epoch, 500 mini-batches of

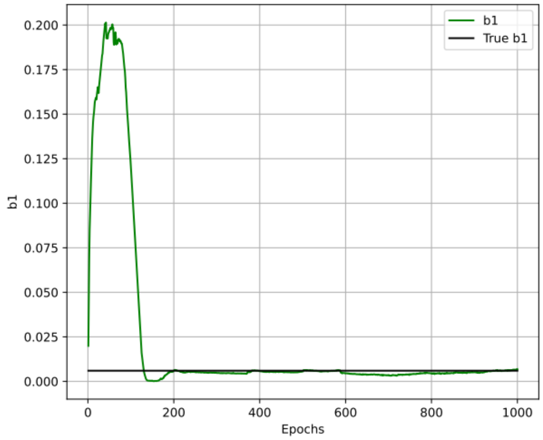

Fig. 4: Estimation of b_1

200 data are considered for gradient-descent based R-PINNs weights update. The R-PINNs is able to approximate the measured noisy variables with high accuracy as shown in Fig. 5. R-PINNs effectively estimate the unknown model parameters as shown in Fig. 3 and Fig. 4.

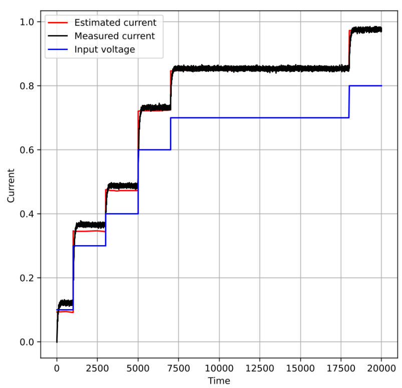

Fig. 5: Estimation of measured variables using R-PINNs

V. CONCLUSIONS

R-PINNs are able to successfully incorporate the ARR based input-output representation of the known physics based laws within the NN based learning paradigm leading to an efficient learning of unknown model parameters in the presence of non measurable state variables. The paper establishes mathematically the uniform and ultimate boundedness of parametric estimation and NN approximation errors. R-PINNs give a framework to exploit only the measurable system variables, thus, avoiding the dependence to unknown states. ARRs have proven to be extremely useful for fault detection and isolation. As such, R-PINNs based approaches promise many possibilities in this direction. Moreover, R-PINNs also provide a means to integrate various Deep NN structures, well adapted to sequential learning within the proposed framework, leading to numerous possibilities for future work.

ACKNOWLEDGMENT

Authors are thankful to anonymous reviewers for their constructive remarks and Soha Kanso for her help in compilation of the compliant version of this paper.

REFERENCES

[1] Mogens Blanke, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki, and Jochen Schröder. *Diagnosis and fault-tolerant control*. Vol. 2. Springer, 2006.

[2] Kathryn Eleda Brenan, Stephen L Campbell, and Linda Ruth Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. SIAM, 1995.

[3] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. "Data-driven discovery of coordinates and governing equations". In: *Proceedings of the National Academy of Sciences* 116.45 (2019), pp. 22445–22451.

[4] DA Cox, JB Little, and D O'Shea. "Ideals, Varieties, and Algorithms, Springer Verlag, New York". In: (1992).

[5] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. "Scientific machine learning through physics–informed neural networks: where we are and what's next". In: *Journal of Scientific Computing* 92.3 (2022), p. 88.

[6] Sette Diop. "Elimination in control theory". In: *Mathematics of control, signals and systems* 4 (1991), pp. 17–32.

[7] ST Glad. "Nonlinear state space and input output descriptions using differential polynomials". In: *New Trends in Nonlinear Control Theory: Proceedings of an International Conference on Nonlinear Systems, Nantes, France, June 13–17, 1988*. Springer. 1989, pp. 182–189.

[8] Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

[9] J. Nathan Kutz, Steven L. Brunton, Bingni W. Brunton, and Joshua L. Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[10] Lennart Ljung, Carl Andersson, Koen Tiels, and Thomas B Schön. "Deep learning and system identification". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 1175–1181.

[11] Lennart Ljung, Torkel Glad, and Anders Hansson. *Modeling and identification of dynamic systems*. Studentlitteratur, 2021.

[12] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. "Deep learning for universal linear embeddings of nonlinear dynamics". In: *Nature communications* 9.1 (2018), p. 4950.

[13] Diego Rodriguez Obando, John J Martinez, and Christophe Bérenguer. "Deterioration estimation for predicting and controlling RUL of a friction drive system". In: *ISA transactions* 113 (2021), pp. 97–110.

[14] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational physics* 378 (2019), pp. 686–707.

[15] Marcel Staroswiecki and G Comtet-Varga. "Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems". In: *Automatica* 37.5 (2001), pp. 687–699.

[16] Jochen Stiasny, George S Misyris, and Spyros Chatzivasileiadis. "Physics-informed neural networks for non-linear system identification for power system dynamics". In: *2021 IEEE Madrid PowerTech*. IEEE. 2021, pp. 1–6.

[17] Qinghua Zhang, Michele Basseville, and Albert Benveniste. "Fault detection and isolation in nonlinear dynamic systems: A combined input–output and local approach". In: *Automatica* 34.11 (1998), pp. 1359–1373.