Learning Control for Voltage and Frequency Regulation of an Infinite Bus System*

Luis A. Duffaut Espinosa[†]
Department of Electrical and Biomedical Engineering
University of Vermont, Burlington, VT 05405 USA
lduffaut@uvm.edu

W. Steven Gray G. S. Venkatesh
Department of Electrical and Computer Engineering
Old Dominion University, Norfolk, VA 23529 USA
{sgray,sgugg001}@odu.edu

***Abstract*—A new learning methodology in terms of a discretization of a so-called Chen-Fliess series of a control affine nonlinear system was recently proposed, in part, for the purpose of systematically including system structure and expert knowledge into control strategies. The main objective of this paper is to appropriately embed this learning unit as a supporting predictive controller for power dynamical systems. In particular, an infinite bus system is used for the prototype design of a smart and active control policy to regulate voltage and frequency. It is demonstrated by simulation how a controller employing a Chen-Fliess learning unit can recover from a fault and address modeling mismatch.**

***Index Terms*—power dynamical systems, infinite bus, learning/adaptive control, Chen-Fliess series**

I. Introduction

Today's standards for managing power systems require high levels of learning and adaptation to the conditions under which they operate. This is particularly the case when there is an abundance of renewable resources available. Controllers regulating these systems must predict and respond quickly to a constantly changing environment. Machine learning algorithms have conventionally used artificial neural networks for their realization. In the context of power/control systems, the closest approximation for their dynamic behavior is given by so called recurrent networks but with well documented theoretical limitations [14], [16], [19], [23]. However, other less complex architectures with stronger theoretical foundations are desired.

The authors in [7]–[9] have employed the discretization of the notion of a Chen-Fliess functional series (or Fliess operator [3]) to create a novel type of learning algorithm for control affine nonlinear systems. It is known that arbitrary control affine systems in continuous-time have a representation as the one given by Fliess in [3], [4], [13]. Therefore, the proposed learning system structure is comprised of learning units that are capable of approximating such systems to arbitrary desired accuracy [6] Furthermore, this new technology is suited for taking advantage of a known model description or can be employed for control purposes using solely real-time data. The former allows one to include expert knowledge in the control strategy. When these learning units are employed in the context of predictive control, they have been shown to robustify the closed-loop system when the plant is poorly modelled or even completely unknown modulo the assumption of being in the class of control affine nonlinear systems [10], [21]. In this vein, power systems have been analyzed in the context of predictive control [12], [15], [20]. These works addressed stability and voltage/frequency regulation from the perspectives of one step ahead prediction and centralized/decentralized distributed consensus. But the concept of learning was not utilized in these works.

The main objective of this paper is to appropriately embed this new type of learning unit into the standard framework of power system operations. As a starting point, the focus will be on a simple power system model comprised of a synchronous generator connected to an infinite bus (SMIB) [18]. The learning unit will be configured to work in tandem with standard power systems controllers such as droop and automatic voltage regulators (AVRs). It will be shown that these systems do not perform well when there is significant model mismatch. The learning unit will recognize such deficiencies and adapt to these conditions by introducing additional compensation for voltage and/or frequency regulation. In particular, this paper focuses on a synchronous generator described by the flux-decay model connected to the afore mentioned infinite bus. The objective is to first show that when there is a mismatch between the inertia of the plant and the model used for control, the SMIB system settles to an undesired voltage. However, if the learning controller developed in [9], [10] is used in tandem, the closed-loop system can achieve the desired voltage despite the inertia error in the SMIB model. Additionally and under the same conditions, the case in which the controller has no knowledge of the plant (free model) is also presented. The latter shows that the approximation provided by the discrete-time Fliess operator theory performs reasonable well in providing an input-output model for one-step ahead predictive control for power system regulation.

The organization of the work in this manuscript is as follows. Section II provides preliminaries and describes prior work with the objective of keeping the manuscript succinct and self-contained. It includes a brief introduction to discrete-time Fliess operators and a description of the learning unit. A high level implementation of the learning algorithm is also given. This is followed by Section III, where the SMIB system is

*This research was supported by the National Science Foundation under grants CMMI-1839378 and CMMI-1839387.
[†]Corresponding author.

described. Section IV presents simulations results when the predictive learning controller is used for two cases: expert knowledge of the plant is available and the free-model case. Conclusions and some future research directions are provided in the final section.

II. Preliminaries

The learning controller employed in this manuscript is based on the theory of Fliess operators and their discretization. In the next section, a concise summary of discrete-time Fliess operators is given. See [2], [6] for more specific content.

A. Discrete-Time Fliess Operators

Consider a nonempty set $X = \{x_0, x_1, \ldots, x_m\}$ known as an *alphabet* and its elements as *letters*. A juxtaposition (or concatenation) of letters is called a *word* (e.g., $\eta = x_{i_1} x_{i_2} \cdots x_{i_k}$). One denotes the number of letters in a word by $|\eta|$. This is referred to as the *length* of η. The special word having length zero is the empty word, written as $\emptyset$. X^k is the set of all words whose length is k. Define $X^* = \bigcup_{k\geq 0} X^k$ and $X^{\leq J} = \bigcup_{k=0}^{J} X^k$. X^* forms a monoid under concatenation. A *formal power series* is any mapping $c : X^* \to \mathbb{R}^\ell$. It is usually expressed as $c = \sum_{\eta \in X^*} (c, \eta)\eta$, which should be viewed as a formal sum. In this sum, (c, η) represents the *coefficient* of c. The set of all such noncommutative formal power series over X is denoted by $\mathbb{R}^\ell\langle\langle X\rangle\rangle$.

An input in this paper is taken to be a sequence in

$$l_\infty^{m+1}(N_0) := \{\hat{u} = (\hat{u}(N_0), \hat{u}(N_0 + 1), \ldots) : \|\hat{u}\|_\infty < \infty\},$$

where $\hat{u}(N) := [\hat{u}_0(N), \hat{u}_1(N), \ldots, \hat{u}_m(N)]^T$, $N \geq N_0$ with $\hat{u}_i(N) \in \mathbb{R}$, $|\hat{u}(N)| := \max_{i\in\{0,1,\ldots,m\}} |\hat{u}_i(N)|$, and $\|\hat{u}\|_\infty := \sup_{N\geq N_0} |\hat{u}(N)|$. The subspace of finite sequences over $[N_0, N_f]$ is denoted by $l_\infty^{m+1}[N_0, N_f]$.

Definition 1: A *discrete-time Fliess operator* for a generating series $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$ is defined as

$$\hat{F}_c[\hat{u}](N) = \sum_{\eta\in X^*} (c, \eta) S_\eta[\hat{u}](N)$$

for any $N \geq N_0$, where

$$S_{x_i\eta}[\hat{u}](N) = \sum_{k=N_0}^{N} \hat{u}_i(k) S_\eta[\hat{u}](k)$$

with $x_i \in X$, $\eta \in X^*$, and $\hat{u} \in l_\infty^{m+1}[N_0]$. Here it is assumed that $S_\emptyset[\hat{u}](N) := 1$.

Following [11], pick a fixed $u \in L_1^m[0, T]$ for $T > 0$ finite. Select an integer $L \geq 1$, let $\Delta := T/L$, and define the real-valued sequence

$$\hat{u}_i(N) = \int_{(N-1)\Delta}^{N\Delta} u_i(t)\, dt, \quad i = 0, 1, \ldots, m,$$

where $N \in \{1, 2, \ldots, L\}$. By assuming $u_0 = 1$ one has that $\hat{u}_0(N) = \Delta$. For applications, the series $\hat{F}_c$ must be truncated. That is, for any $J \geq 0$ let

$$\hat{y}(N) = \hat{F}_c^J[\hat{u}](N) := \sum_{\eta\in X^{\leq J}} (c, \eta) S_\eta[\hat{u}](N). \tag{1}$$

In [6, Theorems 6 and 7], the class of discrete-time Fliess operators truncated to finite length was shown to provide a set of universal approximators for which the approximation error with respect to its continuous-time counterpart ([3], [4], [13]) could be explicitly computed. Thus, these truncated operators can be used to approximate the behavior of any control affine state space realization

$$\dot{z}(t) = g_0(z(t)) + \sum_{i=1}^{m} g_i(z(t))\, u_i(t), \quad z(t_0) = z_0 \tag{2a}$$

$$y_j(t) = h_j(z(t)), \quad j = 1, \ldots, \ell. \tag{2b}$$

Needless to say, the accuracy of the approximation increases as L and J increase. A novel learning unit is described in the next section that leverages this universal approximation property to process input-output information produced by systems having the form (2).

B. Learning Unit Based on Discrete-Time Fliess Operator

In this section, the structure of the learning unit is described first, and then the algorithm for its inductive implementation is presented. For simplicity and without loss of generality, only the case where $\ell = 1$ is considered. The most complete treatment of this subject appears in [10].

Assuming some ordering has been imposed on the elements of X^*, equation (1) can be expressed as

$$\hat{y}(N) = \phi^T(N)\theta_0, \quad N \geq 1, \tag{3}$$

assuming $N_0 = 1$, and where

$$\phi(N) = [S_{\eta_1}[\hat{u}](N)\ S_{\eta_2}[\hat{u}](N) \cdots S_{\eta_l}[\hat{u}](N)]^T$$
$$\theta_0 = [(c, \eta_1)\ (c, \eta_2) \cdots (c, \eta_l)]^T$$

with $l = \mathrm{card}(X^{\leq J}) = \sum_{k=0}^{J} (m + 1)^k = ((m + 1)^{J+1} - 1)/m$. Therefore, if at time $N-1$ there is an estimate $\hat{\theta}(N-1)$ of θ_0, then (3) provides an estimate of $\hat{y}(N)$:

$$\hat{y}_p(N) := \phi^T(N)\hat{\theta}(N - 1). \tag{4}$$

The coefficients of the truncated operator are then updated with a standard recursive least-squares algorithm [5, p. 65]. A block diagram of the learning unit is shown in Figure 1. Here the unit is fed with input-output data (u, y) coming from a plant that is running in continuous-time (or the error between some model of the plant and the plant itself). The primary underlying assumption is that the data comes from a plant that can be express as a Fliess operator, for instance, any plant described by (2). Observe also that there is no reason for $\hat{\theta}(N)$ to converge exactly to the coefficients of c since the approximator is a truncated version of $\hat{F}_c$. Nevertheless, since the objective is solely to approximate F_c with $\hat{F}^J_{\hat{\theta}(N)}$ as accurately as possible, this is not a problem. But one can argue from the theory presented in [6] that the series c is at least a feasible limit point for the sequence $\hat{\theta}(N)$, $N \geq 0$.

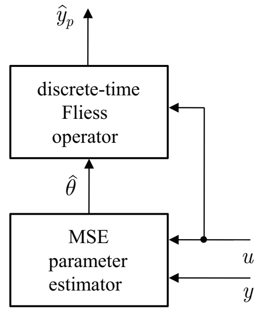

Fig. 1. Learning unit based on a discrete-time Fliess operator

C. Inductive Implementation of Learning Algorithm

The inductive implementation of the learning algorithm requires some knowledge of the underlying algebra of iterated sums appearing in (1). This is best seen using the following concept.

Definition 2: [6] Given any $N \geq N_0$ and $\hat{u} \in l_\infty^{m+1}(N_0)$, a *discrete-time Chen series* is given by

$$S[\hat{u}](N, N_0) := \sum_{\eta \in X^*} \eta S_\eta[\hat{u}](N, N_0),$$

where

$$S_{x_i\eta}[\hat{u}](N, N_0) = \sum_{k=N_0}^{N} \hat{u}_i(k) S_\eta[\hat{u}](k, N_0)$$

with $x_i \in X$, $\eta \in X^*$, and $S_\emptyset[\hat{u}](N, N_0) := 1$. If $N_0 = 0$ then $S[\hat{u}](N, 0)$ is written as $S[\hat{u}](N)$.

Let X be arbitrary and define $\hat{u}_\eta(N) = \hat{u}_{i_k}(N) \cdots \hat{u}_{i_1}(N)$ for any $\eta = x_{i_k} \cdots x_{i_1} \in X^*$ and $N \geq N_0$ with $\hat{u}_\emptyset(N) := 1$. In addition, $c_u(N) := \sum_{\eta \in X^*} \hat{u}_\eta(N)\eta$. Then

$$S_{x_i\eta}[\hat{u}](N_0, N_0) = \hat{u}_{x_i}(N_0) S_\eta[\hat{u}](N_0, N_0)$$

so that $S_\eta[\hat{u}](N_0, N_0) = \hat{u}_\eta(N_0)$, and thus, $S[\hat{u}](N_0, N_0) = c_u(N_0)$. It is important to note that a discrete-time Chen series $S[\hat{u}](N, N_0)$ satisfies a difference equation as described in the next theorem.

Theorem 1: [8] For any $\hat{u} \in l_\infty^{m+1}(N_0)$ and $N \geq N_0$

$$S[\hat{u}](N+1, N_0) = c_u(N+1) S[\hat{u}](N, N_0)$$

with $S[\hat{u}](N_0, N_0) = c_u(N_0)$ so that $S[\hat{u}](N, N_0) = \overleftarrow{\prod}_{i=N_0}^{N} c_u(i)$, where $\overleftarrow{\prod}$ denotes a directed product from right to left.

Consider next two input sequences $(\hat{u}, \hat{v}) \in l_\infty^{m+1}[N_a, N_b] \times l_\infty^{m+1}[N_c, N_d]$ with $N_b > N_a$ and $N_d > N_c$. The *concatenation* of $\hat{u}$ and $\hat{v}$ at $M \in [N_a, N_b]$ is taken to be

$$(\hat{v} \#_M \hat{u})(N) = \begin{cases} \hat{u}(N) & : \; N_a \leq N \leq M \\ \hat{v}((N-M)+N_c) & : \; M < N \leq M + (N_d - N_c). \end{cases}$$

Define the set of sequences

$$l_{\infty,e}^{m+1}(0) := l_\infty^{m+1}(0) \cup \{\hat{\mathbf{0}}\},$$

where $\hat{\mathbf{0}}$ denotes the empty sequence with duration zero so that formally $\hat{v} \#_M \hat{\mathbf{0}} = \hat{\mathbf{0}} \#_M \hat{v} := \hat{v}$ for all $\hat{v} \in l_{\infty,e}^{m+1}(0)$. In which case, $l_{\infty,e}^{m+1}(0)$ is a monoid under this input concatenation operator. Define $S[\hat{\mathbf{0}}] = 1$. The following is a straightforward generalization of Theorem 1.

Theorem 2: [8] (Discrete-time Chen's identity) Given $(\hat{u}, \hat{v}) \in l_\infty^{m+1}[N_a, N_b] \times l_\infty^{m+1}[N_c, N_d]$, $M \in [N_a, N_b]$, and $N \in [M, M + (N_d - N_c)]$ it follows that

$$S[\hat{v}]((N-M)+N_c, N_c) S[\hat{u}](M, N_a) = S[\hat{v} \#_M \hat{u}](N, N_a).$$

In particular, $S[\hat{v}](N-M) S[\hat{u}](M) = S[\hat{v} \#_M \hat{u}](N)$ when $N_a = N_c = 0$.

In [8], the theorem above was used to show that (4) can be written in the form

$$\begin{aligned} \hat{y}_p(N+1) &= \hat{\theta}^T(N) \Pi(S[\hat{u}](N+1)) e_1 \\ &= \hat{\theta}^T(N) \mathcal{S}(N+1) \Pi(S[\hat{u}](N)) e_1 \end{aligned} \tag{5}$$

for $N \geq N_0$, where $\mathcal{S}(N+1)$ and $S[\hat{u}](N)$ have been suitable truncated, and $e_1 := [1\,0\,0 \cdots 0]^T \in \mathbb{R}^l$. Another form for (5) is $\hat{y}_p(N+1) = Q(\hat{u}(N+1))$, where Q is a polynomial with maximum degree $l-1$ in the components of $\hat{u}(N+1)$. Furthermore, the matrix $\mathcal{S}^{J+1}(N+1)$ can be expressed in terms of $\mathcal{S}^J(N+1)$ as:

$$\mathcal{S}^{J+1}(N+1) = \left[\begin{array}{c|c} 1 & 0 \cdots 0 \\ \hline \hat{u}(N+1) \otimes (\mathcal{S}^J(N+1)e_1) & \begin{array}{c} \text{block diag}(\mathcal{S}^J(N+1), \\ \ldots, \mathcal{S}^J(N+1)) \end{array} \end{array}\right],$$

where '$\otimes$' denotes the Kronecker product, and the bottom right block is a block diagonal matrix with $m+1$ blocks. The pseudo-code for the generation of $\mathcal{S}^J(N+1)$ is given in [21, Section III].

D. Predictive Controller with Learning

Next it is shown how the learning units in Figure 1 running the learning algorithm described in the previous section can be used for predictive control. Consider an arbitrary plant and a model of the plant given by a dynamical system of the form (2). The latter can be an approximation of the plant over some range of operation. There is no additional complication if the plant is allowed to be time varying. The proposed one-step ahead predicative controller with learning, first proposed in [21], is shown in Figure 2 assuming the plant has two outputs. One learning unit is required for each output, while the number of inputs can be arbitrary. The learning units process the error data in order to build approximations of the input-output system $u \mapsto e$. These error predications are then combined with the assumed plant model to construct an input to make the plant output y track some desired output y_d via one-step ahead predictive control. If the model is set to zero so that $\hat{y} = 0$, then the learning units have to learn the full plant input-output model $u \mapsto y$. The general expectation is that some reasonable model for the plant should improve performance. Some interesting case studying comparing model based control to model free control using this scheme appear in [9], [10], [21]. In the next section, this type of controller is applied for the case of a synchronous generator connected to an infinite bus.

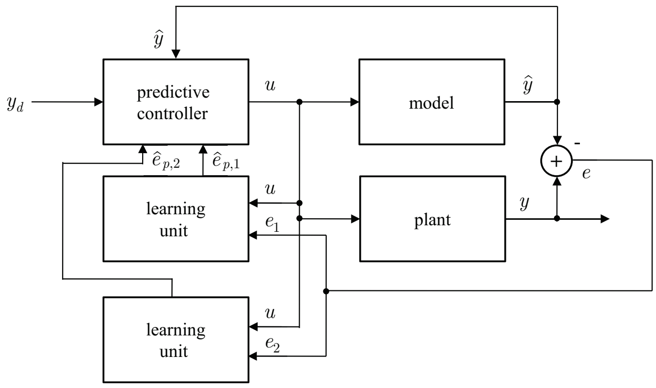

Fig. 2. Closed-loop system with MIMO predictive controller and two learning units

III. SINGLE MACHINE INFINITE BUS SYSTEM

The single machine infinite bus system (SMIB) shown in Figure 3 has been classically used to study the interaction between a large power network (acting as an unmovable voltage and frequency sink) and a single generator. With this type of system it is also possible to understand the small signal stability of synchronous generators [1] and to determine the efficacy of control algorithms for the purpose of improving voltage regulation and transient stability [22]. A complete

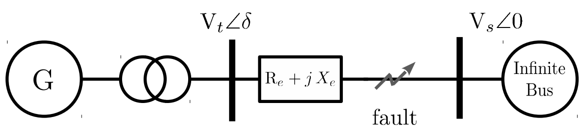

Fig. 3. Single machine infinite bus system

general model of a single synchronous machine is described in [18]. This manuscript focuses in particular on the one axis flux decay model. This results in the control affine dynamical system:

$$\frac{d\delta}{dt} = \omega - \omega_s \tag{6a}$$

$$\frac{2H}{\omega_s}\frac{d\omega}{dt} = T_m - E'_q I_q - (x_q - x'_d) I_d I_q - D(\omega - \omega_s) \tag{6b}$$

$$\tau'_{d0}\frac{dE'_q}{dt} = -E'_q - (x_d - x'_d) I_d + E_{fd} \tag{6c}$$

$$\tau_e \frac{dE_{fd}}{dt} = -k_e E_{fd} + V_r \tag{6d}$$

$$\tau_a \frac{dV_r}{dt} = -V_r - K_a (V_{ref} - V_s - V_{pss}) \tag{6e}$$

$$\tau_{ch}\frac{dT_m}{dt} = -T_m + P_{sv} \tag{6f}$$

$$\tau_{sv}\frac{dP_{sv}}{dt} = -P_{sv} + P_c - \frac{1}{R_d}\left(\frac{\omega}{\omega_s} - 1\right), \tag{6g}$$

TABLE I
SYNCHRONOUS MACHINE - INFINITE BUS PARAMETERS

Field winding time constant	τ'_{d0} = 3.0 s
Inertia constant (model)	H = 3 s
Inertia constant (plant)	H = 6 s
Damping constant	D = 0.0125 $\frac{\text{p.u}}{\text{rad/s}}$
AVR time constant	τ_a = 0.01 s
Direct axis reactance	x_d = 0.8958 p.u.
Infinite bus frequency	w_s = 60 Hz
AVR gain	k_a = 20
Transient direct-axis reactance	x'_d = 0.1198 p.u.
Transmission network resistance	r_e = 0.025 p.u.
Transmission network reactance	x_e = .6 p.u.
Quadrature axis reactance	x_q = 0.8645 p.u.
Excitation time constant	τ_e = 0.314 s
AVR integral gain	k_{ai} = 10
Mechanical torque time constant	τ_{ch} = 1 s
Valve time constant	τ_{sv} = 2 s
Droop gain	R_d = 0.05
Ininite bus voltage	V_s = 1 p.u.

where δ is the rotor's angle, ω is the angular frequency, E'_q is the quadrature axis transient voltage, E_{fd} is the excitation voltage, V_r is the automatic voltage regulator (AVR) state, T_m is the mechanical torque, and P_{sv} is the valve on the turbine gate state. The definitions and assumed values of the parameters in (6) are summarized in Table I. The model above includes the valve dynamics in (6g) which regulate the force (e.g., steam) applied to the turbine and produces mechanical torque obeying (6f) in the synchronous machine [17]. The AVR is realized by (6e). The system is naturally limited by how much the gate is open to the turbine and by how much AVR voltage can be supplied for regulation. That is,

$$0 \le P_{sv} \le 1$$
$$V_r^{\min} \le V_r \le V_r^{\max}.$$

Also observe that (6e) contains the input V_{pss}, which acts as a power system stabilizer control action on the system. In addition, the input P_c is a control input that can either be constant or driven by the output of an automatic controller. In this paper, V_{pss} and P_c are the points of interaction between the infinite bus system, the learning unit, and the predictive controller described in Section II. However, the droop controller and AVR are built-in linear controllers that one always encounters in a synchronous generator and as such these *inner feedback loops* will have to be taken into account when designing the predicative controller in the *outer feedback loop*. The droop gain is usually given by the relation

$$R_d = \frac{\text{No load frequency} \;-\; \text{Full load frequency}}{\text{No load frequency}} = 0.05\%.$$

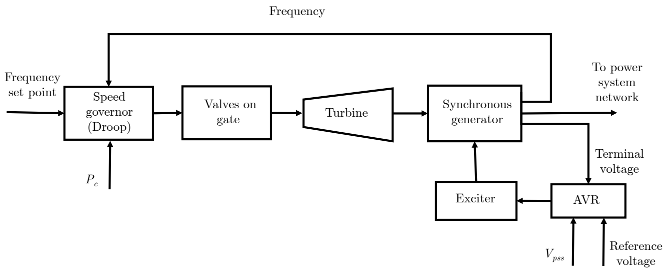

Fig. 4. Synchronous machine block diagram with standard droop and AVR controllers

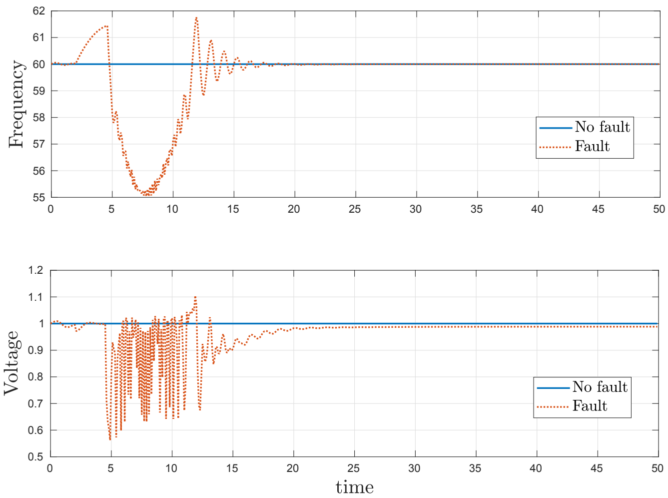

Fig. 5. SMIB system running with standard controllers with and without fault

The terminal voltage of the system is computed as

$$V_t = \sqrt{V_d^2 + V_q^2},$$

where the direct-axis and quadrature-axis terminal voltages are given, respectively, by

$$V_d = I_q x_q, \quad V_q = E_q' - I_d x_d'$$

with I_d and I_q (the direct-axis and quadrature-axis currents) defined as

$$\begin{aligned} I_d =& - V_s x_e \cos(\delta) - (x_q + x_e)E_p' + V_s x_q \cos(\delta) \\ &+ \frac{r_e V_s \sin(\delta)}{(x_e x_d' + x_e x_q + x_d' x_q + r_e^2 + x_e^2)} \\ I_q =& r_e E_q' - r_e V_s \cos(\delta) + V_s x_e \sin(\delta) \\ &+ \frac{x_d' V_s \sin(\delta)}{(x_e x_d' + x_e x_q + x_d' x_q + r_e^2 + x_e^2)}. \end{aligned}$$

Also, one can add integral action to (6e) on the error signal $e_V := V_{ref} - V_s - V_{pss}$ inside the AVR block in Figure 4. It is assumed that the integral gain is fixed at $k_{ia} = 10$.

Figure 5 shows the frequency and voltage response for a SMIB modeled as in (6) when a fault occurs at $t = 2$ s for a duration of 2.5 s. The fault constitutes a short circuit on the transmission line (the resistance and reactance suddenly and dramatically increase) between the synchronous generator and the infinite bus. Once the fault occurs, the frequency and terminal voltage begin to oscillate and drop below acceptable levels. The internal linear controllers eventually stabilize the system, as they are designed to do, but the terminal voltage set point (≈ 0.98 p.u.) never fully recovers. More importantly, the transient response as a result of the fault is extremely poor, not acceptable in most cases. This points out the limitations of linear control in the face of large disturbances. To further complicate the situation, prediction, as mentioned in the introduction, is key to handling challenges such as the penetration of renewable resources. So adding an outer prediction loop to (6) while maintaining acceptable disturbance responses is a nontrivial problem. The next section deals with this issue.

IV. Predictive Learning Control for SMIB System

In this section, the model described in Section III is used together with two learning units and a one-step ahead predictive controller as shown in Figure 6 for the purpose of augmenting the linear controllers embedded in the SMIB system. The objective is not to compete with these internal controllers but rather to compensate them whenever the system leaves the linear regime for which these controllers are known to be most effective. To achieve this goal, a simple switching rule was implemented so that the learning units and predictive controllers are inactive when the droop controller and AVR are working properly. That is,

$$u = \begin{cases} 0, & |y - y_d| \le 0.02 \text{ p.u.} \\ \bar{u}, & |y - y_d| > 0.02 \text{ p.u.}, \end{cases}$$

where in Figure 2, $u = (V_{pss}, P_c)^\top$, $y = (\omega, V_t)^\top$, and the desired regulation set point is $y_d = (\omega_d, V_{t,d})^\top$. Here the switching threshold was tuned empirically. Two scenarios are considered next: the case in which one has erroneous knowledge of the plant and the case where no knowledge of the plant is assumed a priori.

A. Model Mismatch Scenario

It is rare to have complete knowledge of the plant in a control problem since machines will age, vary with operating conditions, and are subject to measurement uncertainly. Although the approach described in the previous section can address mismatch on any subset of parameters in (6), the focus here is on parametric error in the generator inertia H. Inertia represents the energy stored in the spinning plant. Its presence limits the rate at which frequency can change. Assuming that the expert model does not know H perfectly or does not know it at all, any mismatch between plant and expert model produces an output error $e(t)$ (see Figure 2), which activates the learning units. In this situation, an error system is being fitted using a discrete-time Fliess operator approximator as described in Sections II-A and II-B. The predictive controller is then realized using the assumed model and this identified error model. Figure 7 shows the result of regulation under the fault described in the previous section

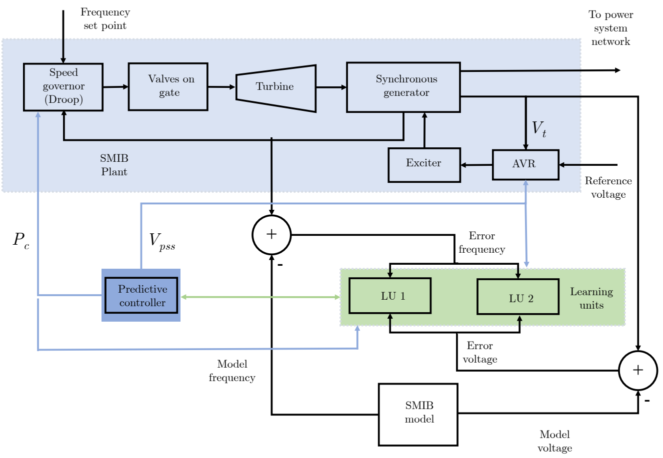

Fig. 6. Synchronous machine block diagram with learning units and predictive controller

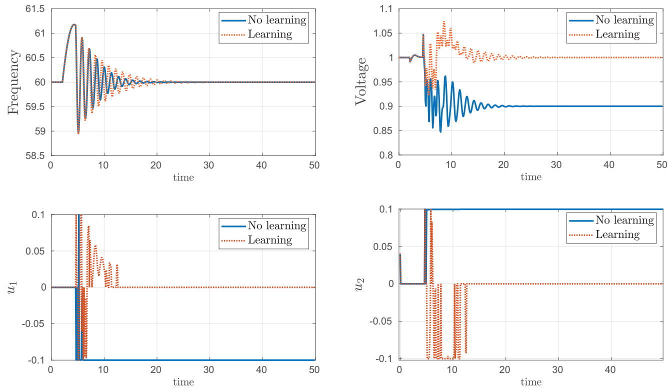

Fig. 7. Predictive learning control with model mismatch on inertia ($H_{\text{model}} = 3$ s and $H_{\text{plant}} = 6$ s) and AVR and droop regulation versus response with only AVR and droop control ($u = (V_{pss}, P_c)^\top = (0, 0)^\top$)

(a 2.5 s duration short circuit in the transmission line) using this predictive learning control (red dotted line) and compares the performance against the case where there is no learning (blue solid line). Specifically, the model inertia is chosen to be $H_{\text{model}} = 3$ s, whereas the plant's real inertia is $H_{\text{plant}} = 6$ s. Observe that the transient response is improved when compared to the case in the previous section (V_t reaches below 0.6 p.u., and ω at points is below 56 Hz), but the steady-state error in the voltage is increased. The new voltage set point ≈ 0.9 p.u. is beyond the allowed range of 1 ± 0.05 p.u.. On the other hand, when the predictive learning controller is enabled, the system recovers to the desired set point of 1 p.u., and thus the closed-loop system is robust to modeling errors in the inertia.

B. Model Free Scenario

In this scenario it is assumed that the outer controller is tuned with no knowledge of the SMIB system parameters and/or structure. That is, $\hat{y}(t) = 0$ in Figure 2. Given that

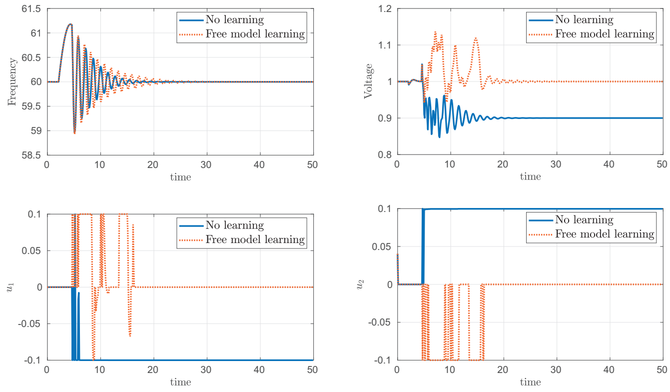

Fig. 8. Predictive model free learning control with AVR and droop regulation versus response with only AVR and droop control ($u = (V_{pss}, P_c)^\top = (0, 0)^\top$)

the plant is modeled by (6), which is control affine, the learning system is mathematically capable of approximating the behavior of the entire plant. Figure 8 shows the simulation results for the same case considered above. As in the previous scenario, the SMIB recovers to the correct set points for voltage and frequency, but not having any model produces a less ideal transient response. In particular, the voltage response has more peaks outside the desired range of ± 0.05 p.u., and around the set point it takes longer to settle. This is essentially the performance cost of knowing less about the plant.

V. Conclusions and Future Work

This paper demonstrated how to employ a predictive learning control system based on discrete-time Fliess operators to control a SMIB with build-in droop control and AVR. The system used a switch to toggle between the linear and nonlinear regimes to avoid inner loop and outer loop control competition. The case studies included a plant with parametric modeling error in the inertia and model free control. It was shown that the addition of learning can eliminate set point errors and improve the transient response, especially if a physical model is available for the plant. In the absence of a plant model, the scheme still produced acceptable regulation, but at the cost of poorer transient response. Future research directions include applying the proposed controller to more realistic power system networks (i.e., ones with several generators) as well as the development of explicit closed-loop stability conditions that can be used during the controller design process.

References

[1] F. P. Demello and C. Concordia, Concepts of synchronous machine stability as affected by excitation control, *IEEE Trans. Power Apparatus Syst.*, 4 (1969) 316–329.

[2] L. A. Duffaut Espinosa, K. Ebrahimi-Fard, and W. S. Gray, Combinatorial Hopf algebras for interconnected nonlinear input-output systems with a view towards discretization, in *Discrete Mechanics, Geometric Integration and Lie–Butcher Series*, K. Ebrahimi-Fard and M. Barbero Liñán, Eds., Springer Nature Switzerland AG, Cham, Switzerland, 2018, pp. 139–183.

[3] M. Fliess, Fonctionnelles causales non linéaires et indéterminées non commutatives, *Bull. Soc. Math. France*, 109 (1981) 3–40.
[4] M. Fliess, Réalisation locale des systèmes non linéaires, algèbres de Lie filtrées transitives et séries génératrices non commutatives, *Invent. Math.*, 71 (1983) 521–537.
[5] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediciton and Control*, Dover Publications, Inc., Mineola, NY, 2009.
[6] W. S. Gray, L. A. Duffaut Espinosa, and K. Ebrahimi-Fard, Discrete-time approximations of Fliess operators, *Numer. Math.*, 137 (2017) 35–62, arXiv:1510.07901.
[7] W. S. Gray, L. A. Duffaut Espinosa, and L. T. Kell, Data-driven SISO predictive control using adaptive discrete-time Fliess operator approximations, *Proc. 21^{st} Inter. Conf. on System Theory, Control and Computing*, Sinaia, Romania, 2017, pp. 383–388.
[8] W. S. Gray, G. S. Venkatesh, and L. A. Duffaut Espinosa, Discrete-time Chen series for time discretization and machine learning, *Proc. 2019 53rd Conf. on Information Science and Systems*, Baltimore, MD, 2019, pp. 1–6.
[9] W. S. Gray, G. S. Venkatesh, and L. A. Duffaut Espinosa, Combining learning and model based control: Case study for single-input Lotka-Volterra system, *Proc. 2019 American Control Conf.*, Philadelphia, PA, 2019, pp. 928–933.
[10] W. S. Gray, G. S. Venkatesh, and L. Duffaut Espinosa, Nonlinear system identification for multivariable control via discrete-time Chen-Fliess series, *Automatica*, 119 (2020) 109085.
[11] L. Grüne and P. E. Kloeden, Higher order numerical schemes for affinely controlled nonlinear systems, *Numer. Math.*, 89 (2001) 669–690.
[12] I. A. Hiskens and B. Gong, Voltage stability enhancement via model predictive control of load, *Intell. Autom. Soft. Co.*, 12 (2006) 117–124.
[13] A. Isidori, *Nonlinear Control Systems*, 3rd Ed., Springer-Verlag, London, 1995.
[14] L. Jin, P. Nikiforuk, and M. Gupta, Approximation of discrete-time state-space trajectories using dynamic recurrent neural networks, *IEEE Trans. Automat. Control*, 40 (1995) 1266–1270.
[15] A. Jokic, R. M. Hermans, M. Lazar, A. Alessio, P. P. J. van den Bosch, I. A. Hiskens, and A. Bemporad, Assessment of non-centralized model predictive control techniques for electrical power networks, *Int. J. Control*, 85 (2012) 1162–1177.
[16] C. Kambhampati, F. Garces, and K. Warwick, Approximation of non-autonomous dynamic systems by continuous time recurrent neural networks, *Proc. Inter. Joint Conf. on Neural Networks*, Como, Italy, 2000, pp. 64–69.
[17] P. Kundur, *Power System Stability And Control*, McGraw-Hill, NY, 1994.
[18] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*, Prentice Hall, Upper Saddle River, NJ, 1998.
[19] A. Schäfer and H. Zimmermann, Recurrent neural networks are universal approximators, in *Artificial Neural Networks – ICANN 2006*, S. Kollias, A. Stafylopatis, W. Duch, and E. Oja, Eds., Springer, Berlin, 2006, pp. 632–640.
[20] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, Distributed MPC strategies with application to power system automatic generation control, *IEEE Trans. Control Syst. Technol.*, 16 (2008) 1192–1206.
[21] G. S. Venkatesh, W. S. Gray, and L. A. Duffaut Espinosa, Combining learning and model based multivariable control, *Proc. 58th IEEE Conf. on Decision and Control*, Nice, France, 2019, 1013–1018.
[22] Y. Wang, D. J. Hill, R. H. Middleton, and L. Gao, Transient stability enhancement and voltage regulation of power systems, *IEEE Trans. Power Syst.*, 8 (1993) 620–627.
[23] L. Xiao-Dong, J. Ho, and T. Chow, Approximation of dynamical time-variant systems by continuous-time recurrent neural networks, *IEEE Trans. Circuits Syst. II, Exp. Brief*, 52 (2005) 656–660.