NONLINEAR TEMPERATURE CONTROL OF ADDITIVE FRICTION STIR DEPOSITION EVALUATED ON AN ECHO STATE NETWORK*

Glen R. Merritt, Christian A. Cousin, Hwan-Sik Yoon*

ABSTRACT

Additive friction stir deposition is a recent innovation in additive manufacturing allowing the deposition of metallic alloys onto a metallic deposit bed, creating a purely mechanical metallic bond. The deposition can be done in a layer-by-layer manner, and the purely mechanical process eliminates the need for high energy consumption and can be deposited at a much higher rate than beam-based welding. The mechanical nature of the process allows the bonding of dissimilar alloys and a reduction in size of the heat affected zone. The additive friction stir deposition process is difficult to model and existing literature has focused on numerical analysis, which is not amenable to online closed-loop control. In this work, a form of reservoir computing called an echo state network is used to model the additive friction stir deposition process from online process data, and validation is performed on a reserved data set. Subsequently, a model free controller using Lyapunov-derived combination of the robust integral of the sign error, and a single hidden layer neural network design is developed to control the additive friction stir deposition process. Control efficacy is given by way of a Lyapunov analysis which shows the system is globally exponentially stable, and simulation results with the echo state networks. Stability proof shows that under one assumption, the controller can be extrapolated to the real system. The mean squared error of the tracking result using the controller and echo state network simulation is 2.05 degrees Celsius.

1 Introduction

Additive friction stir deposition (AFSD) [1, 2] is a highly nonlinear, nonanalytically modeled, and until recently open loop process [3]. The AFSD process is characterized by non-liquid state mechanical bonding, wherein a metallic feedrod is fed through a hollow rotating tooltip by a linear actuator onto a deposit site. The feedrod is deposited in a layer-by-layer manner resulting in a higher speed deposition process and stronger material bonds [4, 5]. Further, the process allows the deposition of non-uniform materials such as metal matrix composites [6], and even allows the joining of dissimilar metals [7], as the bonding process is purely mechanical. Due to the nonlinearities, complexities, and difficulties in interacting forces, the process has sparse modeling, mostly limited to mesh-free numerical models [8,9]. The process lacks closed-form analytical models that can be used for model-based feedback. Previous work with nonmodel-based control [3] would suggest that there are usable solutions without explicit modeling, but when dealing with large-scale and costly industrial systems, it is inadvisable to implement complex control structures without any form of validation.

Reservoir computing [10–12] is a paradigm wherein a dynamic reservoir is treated as an intermediate computational resource, be it organized oscillating structures such as tensegrity or interconnected spring mass damper systems. Instead of training and tuning the reservoir, which would be intractable for mechanical systems, a single readout layer is fit to the output of the reservoir. One useful digital example of the reservoir computing paradigm is the echo state network (ESN) [13, 14], its continuous time extension [15], and especially deep ESN [16, 17]. ESNs are a form of recurrent neural network that has been extensively studied for their approximation capabilities. Deep ESNs are defined by multiple recurrent layers that are not updated and one final output layer that is updated in a

**DEPARTMENT OF MECHANICAL ENGINEERING, UNIVERSITY OF ALABAMA, TUSCALOOSA, AL 35401, USA EMAIL: GMERRITT@CRIMSON.UA.EDU; CACOUSIN@ENG.UA; HYOON@ENG.UA.EDU

one-shot fashion using Tihknonov regularization [18] (also known as ridge regression) over the entire data set. ESNs are based on the observation that during learning, most of the adaptation in neural networks occurs in the output layer [19]. Functionally, all work of tuning and design is encapsulated in the design of the reservoir (the recurrent network), and therefore sidesteps the need for learning on various layers, early stopping, over fitting, etc., [20]. Due to their fixed structure and one-shot update procedure, they can be deployed on more limited datasets and quickly tuned and trained, in contrast to typical deep neural networks. For compact sets, ESNs are universal approximators [21]. For example, an industrial process such as AFSD with a defined operating region and constant set point represents an excellent candidate for implicit modeling and subsequent control of the developed model [22].

Control Lyapunov neural networks, especially when used in conjunction with a RISE component [23], have previous success in control of smooth, continuous, control affine systems. Multi-layer neural networks have universal approximation capabilities [24, 25] and extensive use in application [26, 27] Control based on RISE [28] has been recently proven to be exponentially stable [29], and thus the combination of RISE and neural nets has a firm grounding in exponentially control of unmodeled, nonlinear, and continuous time systems. Therefore, Lyapunov neural nets and RISE are exploited for their exponential stability properties.

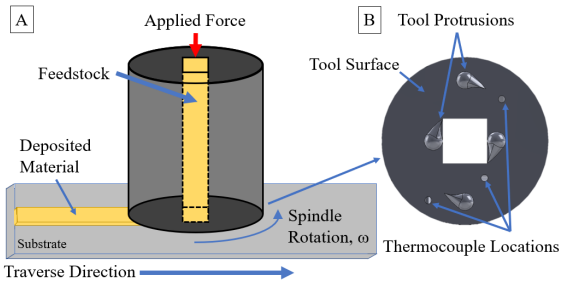

Figure 1. (A) The AFSD process. (B) The AFSD tool-shoulder surface.

The contributions of this work are as follows:
1) First neural network model of the AFSD process with data validation based on process data,
2) Introduction of the echo state network to model control affine systems,
3) First nonlinear Lyapunov based control of the AFSD model, accomplished using Lyapunov neural networks and RISE,
4) Validation of the controller using the ESN model in simulation.

2 AFSD Process

The additive friction stir deposition process is elucidated below in terms of constituent elements. Explanations for the potential nonlinearities and difficulties in process modeling are included to formulate the need for nonanalytical modeling. Subsequently, an explanation of the success of the reservoir computing paradigm and the echo state network motivation is described. Next, implementation and training of the echo state network are described. Finally, data validation is performed on the echo state network as a modeling performance metric.

2.1 AFSD Process Elements

From an operator's perspective, the AFSD process is governed by two interacting inputs and two measureable outputs. A steady state desired temperature and interaction force are two factors which determine the material properties and strength of the deposit. The temperature can be measured and has been measured by embedding thermocouples in the tooltip. The force is estimated using the current of the linear actuator driving the feedstock. As the rotation rate of the tooltip increases, frictional heat generated increases, the feedstock plasticity increases, and the necessary driving force decreases for a given feedrate. As the applied force increases, the amount of heat generated by friction at the interaction zone also increases. As an increase in force contributes to and increase in friction, and an increase in friction decreases the amount of applied force for a given feedrate, the process can be considered to be intricately and nonlinearly cross-coupled. The control inputs are the rate of rotation of the tooltip, and the feedrate of the feedstock.

In addition to the observable variables, internal, unmeasured states can play a role in the AFSD process. For example, the quality of the weld is related to the strain rate tensor. The ambient temperature, and temperature of the deposit bed can act as elements that remove an unpredictable amount of heat from the system. Beading or galling, common weld defects, can affect the quality and dynamics of the

next layer. Therefore it is important to consider such contributions as an unmodeled error in dynamics, and to be addressed in future work.

2.2 Temperature Dynamics

To model the AFSD process, let $x \in \mathbb{R}$ denote the temperature (i.e., state) of the deposition. The dynamics of the AFSD process are then assumed to be modeled by

$$\dot{x} = f(x) + g(x)\,u, \tag{1}$$

where $f : \mathbb{R} \to \mathbb{R}$ represents an unknown nonlinear state-dependent mapping, $g : \mathbb{R} \to \mathbb{R}$ represents an unknown nonlinear control effectiveness, and $u \in \mathbb{R}$ denotes the control input, i.e., the spindle's angular velocity.

Assumption 1. The function $f(x)$ is locally Lipschitz and bounded in the sense that $\mathtt{f}_1 \leq f(x) \leq \mathtt{f}_2$, with positive, known constants $\mathtt{f}_1$, $\mathtt{f}_2$. Moreover, the function $\frac{df(x)}{dx}$ is bounded in the sense that $\left|\frac{df(x)}{dx}\right| \leq \mathtt{f}_3$, with the positive, known constant $\mathtt{f}_3$.

Assumption 2. The function $g(x)$ is locally Lipschitz and bounded in the sense that $\mathtt{g}_1 \leq g(x) \leq \mathtt{g}_2$, with positive, known constants $\mathtt{g}_1$, $\mathtt{g}_2$. Moreover, $g(x)$ is invertable and static in the sense that $\frac{d}{dt}g(x) = 0$. Lastly, the function $\frac{dg^{-1}(x)}{dx}$ is bounded in the sense that $\left|\frac{dg^{-1}(x)}{dx}\right| \leq \mathtt{g}_3$, with the positive, known constant $\mathtt{g}_3$.

3 Control Development

3.1 Error System Development

Because the process dynamics in (1) are unknown, a model-free temperature controller will be developed. To facilitate the development of this controller, let $e \in \mathbb{R}$ denote a temperature error defined as

$$e \triangleq x - x_d, \tag{2}$$

where x was defined in (1) and $x_d \in \mathbb{R}$ denotes the constant desired temperature (i.e., setpoint) of the deposition. To facilitate the subsequent analysis, let $r \in \mathbb{R}$ denote a filtered tracking error defined as

$$r \triangleq \dot{e} + \alpha e, \tag{3}$$

where $\alpha \in \mathbb{R}_{\geq 0}$ is a user-selected constant. .

3.2 Neural Network Development

Noting that the functions $f(x)$ and $g(x)$ are unknown, they are unable to be used directly in feedback. Moreover, because x does not exist a on closed set, $f(x)$ and $g(x)$ can not be approximated with a neural network without a semi-global analysis [30]. Therefore, instead of estimating $f(x)$ and $g(x)$, the functions $f_d : \mathbb{R} \to \mathbb{R}$ and $g_d : \mathbb{R} \to \mathbb{R}$ are introduced, where

$$f_d \triangleq f(x_d). \tag{4}$$
$$g_d \triangleq g(x_d). \tag{5}$$

Because (4) and (5) are predicated on the desired temperature setpoint, they exist on a closed set and can be estimated using a neural network [31]. The open-loop error system is obtained by taking the derivative of (3), and adding and subtracting $\overline{f}_d \triangleq g_d^{-1} f_d$ to yield

$$g^{-1} r = \chi + \overline{f}_d + u, \tag{6}$$

where $\chi \triangleq g^{-1}f - {g_d}^{-1}f_d + g^{-1}\alpha e$. Using Assumptions 1 and 2, the Mean Value Theorem can be used to show that $|\chi| \leq c_1 |e|$, with the positive, known constant c_1.

Since $x_d \in \mathcal{L}_\infty$, let $\mathbb{F}$ be a compact simply connected set of $\mathbb{R}$ with map $\overline{f}_d : \mathbb{F} \to \mathbb{R}$, where $\overline{f}_d$ is continuous. Then, there exist weights and biases such that the function $\overline{f}_d$ can be represented by a neural network as [30]

$$\overline{f}_d = W^\top \sigma\left(V^\top X_d\right) + \varepsilon, \tag{7}$$

where $X_d \triangleq \begin{bmatrix} 1 & x_d \end{bmatrix}^\top \in \mathbb{F}$, $V \in \mathbb{R}^{(2n+1)\times L}$ and $W \in \mathbb{R}^{(L+1)\times 1}$ are bounded constant ideal weight matrices of the neural network, and L is a user-defined constant denoting the number of neurons in the hidden layer. The function $\sigma : \mathbb{R}^L \to \mathbb{R}^{L+1}$ is defined as $\sigma \triangleq \begin{bmatrix} 1 & \sigma_1 & \sigma_2 & ... & \sigma_L \end{bmatrix}^\top$, where σ_i, $\forall i = \{1,\ 2,\ ...,\ L\}$ represents the differentiable sigmoid activation function for each neuron, and the function reconstruction error is denoted by $\varepsilon : \mathbb{F} \to \mathbb{R}$. Since the weights W and V are unknown, an approximated version of (7) is generated as

$$\widehat{f}_d \triangleq \widehat{W}^\top \sigma\left(\widehat{V}^\top X_d\right), \tag{8}$$

where $\widehat{V} \in \mathbb{R}^{1\times L}$ and $\widehat{W} \in \mathbb{R}^{(L+1)\times 1}$ are the estimates of V and W, respectively. By approximating (7) with (8), the controller can estimate the unknown dynamics for use in feedback, negating the need for potentially more aggressive methods such as high-gain or high-frequency feedback.

To facilitate the following development, let $\widetilde{V} \triangleq V - \widehat{V}$ and $\widetilde{W} \triangleq W - \widehat{W}$ denote estimation errors of the weights of the neural network. Then, $\sigma\left(V^\top X_d\right)$ may be approximated at $\sigma\left(\widehat{V}^\top X_d\right)$ using a Taylor Series Expansion as

$$\sigma\left(V^\top X_d\right) = \widehat{\sigma} + \widehat{\sigma}'\widetilde{V}^\top X_d + O^2, \tag{9}$$

where $\widehat{\sigma} \triangleq \sigma\left(\widehat{V}^\top X_d\right)$, $\widehat{\sigma}' \triangleq \left.\frac{\partial \sigma\left(V^\top X_d\right)}{\partial V^\top X_d}\right|_{\widehat{V}^\top X_d}$ denotes the partial derivative, and O^2 denotes the higher order terms of the expansion.

Assumption 3. The ideal weights, biases, and function approximation error of (7), and higher order terms of (9) are assumed to be bounded [30, pgs. 7 and 44], [31]. This assumption is typical in neural network literature, although texts such as [30] may also assume explicit knowledge of these bounds.

3.3 Control Development

Based on the open-loop error system in (6), and the subsequent stability analysis, the controller is designed as

$$u \triangleq -\widehat{f}_d - \mu, \tag{10}$$

where μ denotes a RISE-like control term. Substituting (10) into (6) yields the closed-loop error system

$$g^{-1}r = \chi + \overline{f}_d - \widehat{f}_d - \alpha e - \mu. \tag{11}$$

After employing the approximation properties of the neural network from equations (7),(8), and (9), this can be further written as

$$g^{-1}r = \chi + \widetilde{W}^\top \sigma\left(\widehat{V}^\top X_d\right) + \widehat{W}^\top \sigma'\left(\widehat{V}^\top X_d\right)\widetilde{V}^\top X_d + N - \alpha e - \mu,$$

where $N \triangleq \widetilde{W}^{\top}\sigma'\left(\widehat{V}^{\top}X_d\right)\widetilde{V}^{\top}X_d + W^{\top}O^2 + \varepsilon(x_d)$.

Based on the subsequent stability analysis, the update laws for the neural network are given as

$$\dot{\hat{W}} = \text{proj}\left(\Gamma_1\widehat{\sigma}e^{\top}\right), \tag{12}$$

$$\dot{\hat{V}} = \text{proj}\left(\Gamma_2 X_d e^{\top}\widehat{W}^{\top}\widehat{\sigma}'\right), \tag{13}$$

where $\Gamma_1 \in \mathbb{R}^{L\times L}, \Gamma_2 \in \mathbb{R}^{(2n+1)\times(2n+1)}$ are user selected constants.

The derivative of the closed loop error system in (11), which motivates the RISE element, can be written as

$$g^{-1}\dot{r} = \dot{\chi} + \dot{\overline{f}}_d - \dot{\hat{f}}_d - \dot{\mu}. \tag{14}$$

The RISE control element is denoted as

$$\mu \triangleq -\alpha e + \widehat{d}, \tag{15}$$

where $e(t_0)$ is the initial state error, and $\widehat{d}$ is the Filippov solution to the differential inclusion[1]

$$\dot{\widehat{d}} = kr + e + \beta_1 \text{sgn}(e), \widehat{d}(t_0) = 0. \tag{16}$$

where the signum function is defined as

$$\text{sgn}(b) = \begin{cases} 1, & b > 0 \\ 0, & b = 0 \\ -1, & b < 0 \end{cases}.$$

Assumption 4. The target function of the RISE-based controller is denoted as

$$r = d(x,\dot{x},t) - \widehat{d}(x,\dot{x},t), \tag{17}$$

where $d : \mathbb{R} \to \mathbb{R}$ is defined as $d \triangleq \chi + N$, where N, χ were defined previously. Further, the derivative of (17) can be written as

$$\dot{r} = \widetilde{N} + N_B - kr - e - \beta\text{sgn}(e) \tag{18}$$

where for some known $\gamma_1, \gamma_3, \gamma_4 \in \mathbb{R}_{\geq 0}$, and a known strictly increasing function $\rho_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that the elements of (18) can be bounded by $\|N_B\| \leq \gamma_1$, $\|\dot{N}_B\| \leq \gamma_3 + \gamma_4\|z\| + \rho_2(\|z\|), \forall t \in \mathbb{R}_{\geq 0}$,and finally for some known $\gamma_2, \mathbb{R}_{\geq 0}$ and known strictly increasing function $\rho_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, the bound can be developed according to the mean value theorem $\|\widetilde{N}\| \leq \gamma_2 + \rho_1(\|z\|), \forall t \in \mathbb{R}_{\geq 0}$. The concatenated error vector is denoted as

$$z \triangleq \begin{bmatrix} e & r \end{bmatrix}^{\top} \tag{19}$$

[1]Note that since r may not be commonly available, $\widehat{d}$ is evaluated using $\hat{d} = ke(t) - ke(t_0) + \int_{t_0}^{t}((k\alpha+1)e(\tau) + \beta\text{sgn}(e(\tau)))d\tau$ for closed loop implementation.

4 Stability Analysis

Theorem 1. *Given the closed-loop admittance error system in (11), the admittance error system demonstrates global exponential stability in the sense that*

$$\|e(t)\| \leq \kappa_1 \exp\left(-\kappa_2 (t - t_0)\right), \tag{20}$$

where $\kappa_1 = \|z(t_0)\|$, $\kappa_2 = -\frac{1}{\gamma_5} 2\lambda_V \in \mathbb{R}_{>0}$ *are known constants, provided the gains are selected according to the sufficient conditions:* $\alpha > \lambda_P$, $\beta > \gamma_1 + \frac{\gamma_3}{\alpha - \lambda_P}$.

Proof. The Lyapunov function for the observation signal $\eta \triangleq \left[e \; r \; \text{vec}\left(\widetilde{W}\right)^\top \; \text{vec}\left(\widetilde{V}\right)^\top \; \sqrt{P} \right]^\top$ is defined as as

$$\begin{aligned} V_L(\eta) \triangleq & \quad g(x)^{-1} \tfrac{1}{2} e^2 + \tfrac{1}{2} r^2 \\ & + \tfrac{1}{2} \text{tr}\left(\widetilde{W}^\top \Gamma_1^{-1} \widetilde{W}\right) + \tfrac{1}{2} \text{tr}\left(\widetilde{V}^\top \Gamma_2^{-1} \widetilde{V}\right) + P \end{aligned}, \tag{21}$$

and by design satisfies the condition $2\|\eta\|^2 \geq 2V_L \geq \gamma_5 \left(\|z\|^2 + P\right) \geq \gamma_5 \|z\|^2$, where $\gamma_5 = \min\left(1, \overline{g}^{-1}\right)$. The values $e, r, \widetilde{W}, \widetilde{V}$ were defined previously in equations (8), (9), $\text{tr}(\cdot)$ is the matrix trace operator, the scalar quantity P is the Fillipov solution to

$$\dot{P} = -\lambda_L P + L, \, P(t_0) = 0 \tag{22}$$

and the quantity L is given as

$$L \triangleq e^\top N_B + e^\top \beta \text{sgn}(e) - \left(\gamma_4 + \rho_2\left(\|z\|\right)\right) \|z\| \|e\|_1, \tag{23}$$

where $\|\cdot\|_1$ is the one-norm . The analytical form of P can be written as

$$\begin{aligned} P \triangleq & \quad \beta \|e\|_1 - e^\top + e^{-\lambda_p t} * \\ & \left((\alpha - \lambda_p)\left(\beta \|e\|_1 - e^\top N_B\right) + e^\top N_B\right), \\ & + e^{-\lambda_p t} * \left(\gamma_4 + \rho_2\left(\|z\|\right) \|z\| \|e\|_1\right) \end{aligned} \tag{24}$$

where $*$ is the convolution operation defined as $p(t) * q(t) = \int_{t_0}^{t} p(t - \tau) q(\tau) d\tau$. Taking the time derivative of (21), substituting in equations (11), (12), and (24), employing the transpose property, canceling like terms, yields,

$$\begin{aligned} \dot{V} = & \quad e\left(\chi + N - \alpha e - \hat{d}\right). \\ & + r\left(\widetilde{N} + N_B - kr - e - \beta \text{sgn}(e)\right) \\ & - \lambda_L P + r^\top N_B + r^\top \beta \text{sgn}(e) \\ & + \left(\gamma_4 + \rho_2\left(\|z\|\right)\right) \|z\|_1^2 \end{aligned}$$

Using the definition of (3), upper bounding the resultant equation, and utilizing the arguments in [29] yields,

$$\begin{aligned} \dot{V} \leq & \left(-\alpha e^2 - kr^2 - \lambda_L P + \left(\gamma_2 + \rho_1\left(\|z\|\right)\right) \|r\| \|z\|\right) \\ & + \left(\gamma_4 + \rho_2\left(\|z\|\right)\right) \|z\|_1^2. \end{aligned} \tag{25}$$

Further bounding the constant c and the class $\mathcal{K}$ function ρ yields

$$\dot{V}_L, \leq -\frac{1}{\gamma_5} 2 \left(c - \rho\left(\sqrt{\frac{1}{\gamma_5} 2 V_L}\right)\right) V_L \tag{26}$$

where $c = \min\left\{k_r - \lambda_2 - \gamma_3, \alpha - \gamma_2 - \gamma_3, \frac{\lambda_P}{2}\right\}$ and $\rho(\cdot) = \rho_1(\cdot) + \rho_2(\cdot)$. Given the following property

$$c > \lambda_V + \|z(t_0)\| \tag{27}$$

and the necessary gain conditions are satisfied, the solution to the differential inclusion can be resolved as

$$V_L(\eta), \leq V_L(\eta_0)\exp\left(-\frac{1}{\gamma_5}2\lambda_V(t-t_0)\right) \tag{28}$$

where λ_V is a user defined constant resulting from the primary gain selection. Using the following property restated here

$$\|z\| \leq \sqrt{\frac{2}{\gamma_5}V_L}. \tag{29}$$

bound base on (28) and (29) can be further developed

$$\|z\| \leq \sqrt{\frac{2}{\gamma_5}V_L(t_0)}\exp\left(-\frac{1}{\gamma_5}2\lambda_V(t-t_0)\right), \tag{30}$$

and finally resolved,

$$\|z(t)\| \leq \|z(t_0)\|\exp\left(-\frac{1}{\gamma_5}2\lambda_V(t-t_0)\right). \tag{31}$$

A more detailed derivation of the properties of the RISE controller is found in [29]. □

5 Echo-State Network

The ESN is used to approximate the AFSD dynamics from previously obtained data. The ESN was chosen due to its ability to be quickly trained and tuned on a smaller amount of data as opposed to typical deep learning methods. The training methodology is elucidated below, and validation on a reserved dataset is performed.

5.1 ESN Training

Using an ESN, the dynamics of (1) can be approximated with following assumptions:

Assumption 5. The unknown functions $f(x)$ and $g(x)$ of (1) exist on a bounded and compact set , and can be modeled using the universal approximation property of integral activation fading memory echo state networks [15] .

Let $K_i \in \mathbb{N}$ for all $i \in \{1,2\}$ denote the number of neurons in the ith hidden layer of the ESN. Then, using the recurrent ESN, the unknown dynamics of (1) can be approximated as

$$\dot{x} = W_f\mathcal{F}(x) + W_g\mathcal{F}(x)u + \varepsilon(x), \tag{32}$$

where $\mathcal{F}(x) : \mathbb{R} \to \mathbb{R}^{K_2}$ denotes the inner recurrent network, $W_f \in \mathbb{R}^{1\times K_2}$ and $W_g \in \mathbb{R}^{1\times K_2}$ denote the outer weights mapping the inner recurrent network to the output space of both $f(x)$ and $g(x)$, respectively, and the neural network approximation error is denoted by $\varepsilon(x) \in \mathbb{R}$.

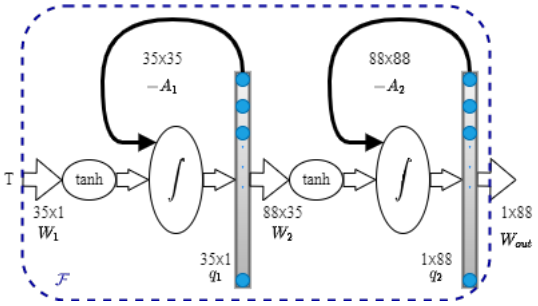

Figure 2. The ESN architecture. The internal reservoir is encapsulated by $\mathcal{F}$, while the internal state vectors are labeled q_1, q_2. The feedforward matrix operations are denoted as W_1, W_2, W_{out}, and the recurrent matrices are denoted as A_1, A_2. The integration and tanh operations are also displayed.

Based on the ESN structure [15], the input of $x = T$ is fed into the reservoir and the output dynamical constants f, g are estimated . The minimizing output matrices W_g and W_f are determined by the ridge regression equation [18]. Letting the observation variable $Y \in \mathbb{R}^{K_2}$ denote the output time series from the ESN before performing regression and let $\dot{x}$ be the numerical derivatives determined from the measured data by way of interpolation and a moving window average of 100 samples, the regression is performed such that

$$W_{f,out} = \dot{x}^\top Y \left(Y^\top Y + \iota I_{K_2 \times K_2} \right)^{-1}, \tag{33}$$

where $\iota \in \mathbb{R}$, is the Tikhinov regularization constant [18], set as $\iota = 0.01$. Increasing ι emphasizes the matrix diagonal and can be used to mitigate noisy measurements. The output data is multiplied elementwise by the time series ω, and shall be denoted by the variable $\mathcal{Y} \in \mathbb{R}^{K_2}$. The output matrix is computed by

$$W_{g,out} = \dot{x}^\top \mathcal{Y} \left(\mathcal{Y}^\top \mathcal{Y} + \iota I_{K_2 \times K_2} \right)^{-1}. \tag{34}$$

The structure of each layer of the ESN is composed of leaky integrator units and the update equation given for each layer is

$$\frac{dq_i}{dt} = \text{-}A_i q_i \text{+}\tanh(W_i q_{i-1}), \forall q_i \in \mathbb{R}^{K_i}, \tag{35}$$

where $K_i \in \mathbb{N}$ denotes the number of neurons in the i^{th} hidden layer, $A_i \in \mathbb{R}^{K_i \times K_i}$ is the leaky state matrix, $W_i \in \mathbb{R}^{K_i \times K_{i-1}}$ is the feedforward matrix from the previous layer, and $q_0 = T$. The number of layers was chosen as two and the hidden neuron counts where selected as $K_1 = 35, K_2 = 88$, based on trial and error. The echo state network used in this work is shown in Fig. 2. It should be noted that q_2 represents the last ESN hidden neuron layer in this work. The development of the echo state network follows the form given in [15, 32] , from which recommendations on leaking rates, neuron size ratios, data preprocessing, and matrix properties can be extrapolated. An example of training is shown in Fig. 3, while the training error from the same data set is shown in Fig. 4.

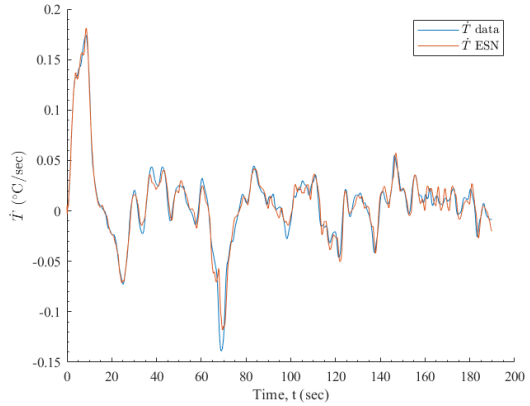

Figure 3. Example training data for the ESN using a single extrusion data set. The orange curve is the predicted $\dot{T}$of the ESN, while the blue curve indicates the numerically obtained $\dot{T}$. The degree of overfitting suggests the inclusion of more data sets.

Of note, when training on a single data set, it is not uncommon to experience overfitting, so nine data sets were utilized. Larger fluctuations are common in the beginning of the prediction time series as the randomized initial states of the ESN need data to reach a steady state. Increasing the constant ι can be used to smooth the output of the network.

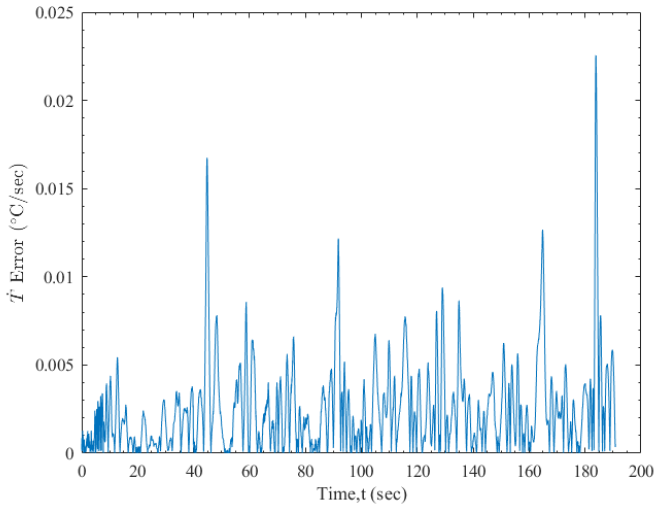

Figure 4. Training error during training for the ESN using a single extrusion data set.

5.2 ESN Validation

Validation was performed by reserving a single data set of the ten available, and performing prediction from the input time series. The prediction output is depicted in Fig. 5, while the prediction error is shown in Fig. 6.

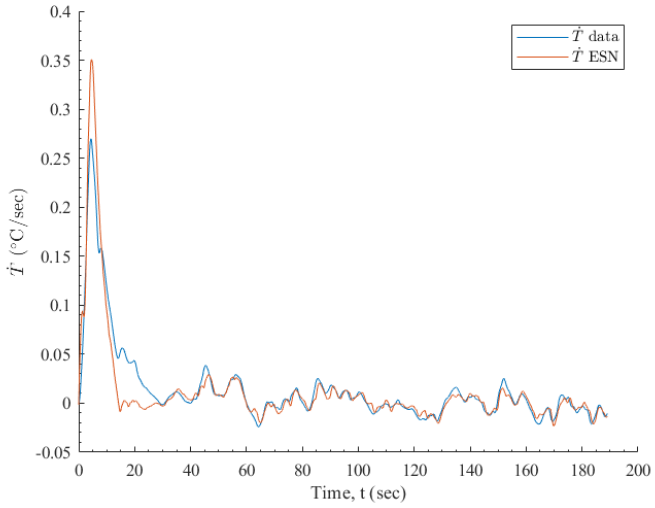

Figure 5. Prediction of the ESN using the reserved data set. The orange curve is the predicted $\dot{T}$ of the ESN, while the blue curve indicates the numerically obtained $\dot{T}$.

The model is capable of predicting the output with RMSE 0.015°C/sec, and Maximum error of 0.10°C/sec

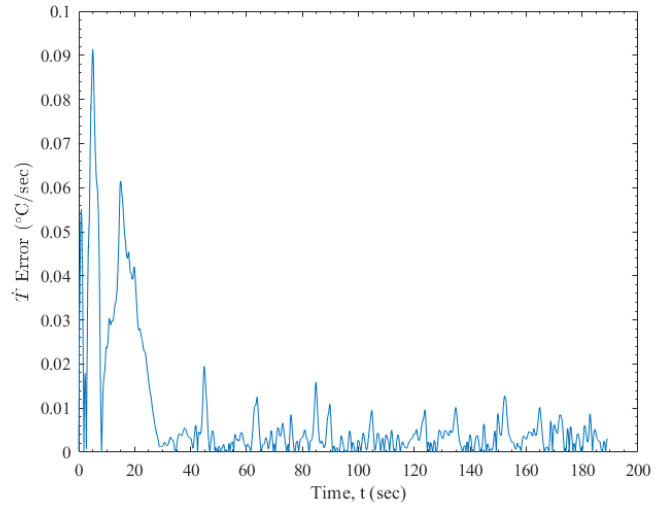

Figure 6. Prediction error for the ESN using the reserved data set.

As is common in many machine learning and deep learning applications, the inclusion of more data or data augmentation can decrease the noise and over fitting commonly found in data sets, the so-called memorization of the training set. The ESN requires fewer training sets than most Deep Learning methods, due to the fixing of the inner layer parameters. Finally, validation was performed against an original temperature dataset using a numerically integrated prediction set from training, the results are depicted in Fig. 7.

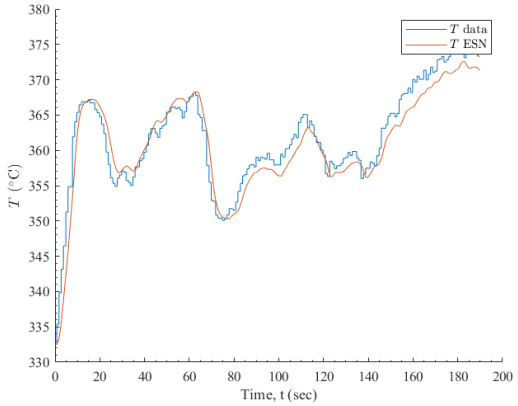

Figure 7. Prediction using trapezoidal integration of the predicted $\dot{T}$ compared to the original T dataset.

6 Simulation

Using the controller developed in Section 3, and the developed and trained ESN of Section 2.2, simulations were performed to validate the controller. The RISE gains were selected as k, α, β, $= 10$, 5, 0.1,, respectively. The neural network learning gains were selected as Γ_1, $\Gamma_2 = 0.1I_{11\times 11}$, $1.0I_{3\times 3}$, respectively. The hidden layer L was selected with 11 neurons. The results of the simulation are shown in Fig. 8, with RMSE 1.98°C,and overshoot of 3.06°C. Limits on ω were enforced in accordance with the limits imposed on the AFSD machine, where $150 \leq \omega \leq 450$ RPM. Integration clamping was also performed on the RISE derivative $\hat{\dot{d}}$ when the saturation limits for ω were hit (therefore only the proportional part of the controller corresponding to e is used), (see Fig. 8).

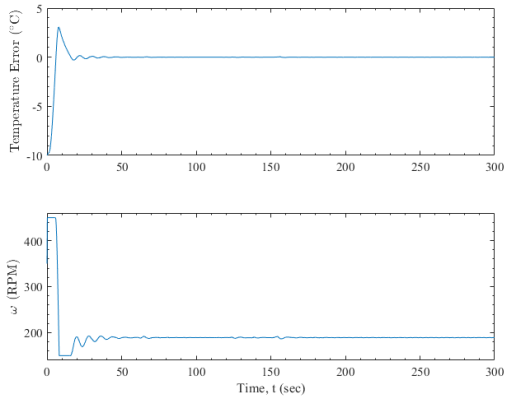

Figure 8. Results temperature control simulation. The lower actuator constraint u_{min} is hit at $t \approx 5s$.

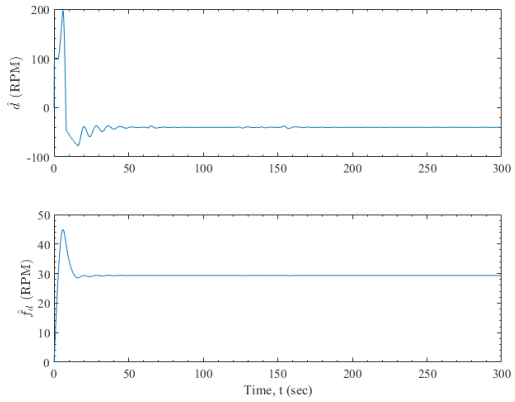

Figure 9. Actuator effort composition added to a nominal spindle speed of 300 RPM. Note that due to the implementation of RISE, the actuator composition includes proportional elements not shown in this figure.

As temperature was approximated as a first order system, the RISE controller does not inherently contain its normal derivative component from the filtered tracking error, which can be tuned to prevent overshoot. Inclusion of a derivative component in the future requires a new control development for RISE on first order systems, and is beyond the scope of this work.

7 Conclusion

In this work, an ESN was developed to approximate the AFSD process in a control affine form. Training was performed on data from nine runs, and validation was performed by reserving the last of the ten data sets. Subsequently, a nonlinear controller based on RISE and a Lyapunov neural network was developed to control the process. A stability analysis was performed to evaluate the approach and proof was given showing that the controller remains stable when extrapolated to the real system given a bounded model error. A simulation was performed on the model developed using the ESN, and results were provided. This work was performed under the data-driven model free paradigm, and control was developed using a model free controller.

REFERENCES

[1] F. Khodabakhshi and A. Gerlich, "Potentials and strategies of solid-state additive friction-stir manufacturing technology: A critical review," *J. Manuf. Process.*, vol. 36, pp. 77–92, 2018.

[2] H. Z. Yu and R. S. Mishra, "Additive friction stir deposition: a deformation processing route to metal additive manufacturing," *Mater. Res. Lett.*, vol. 9, no. 2, pp. 71–83, 2021.

[3] G. Merritt, M. Williams, P. Allison, B. Jordan, T. Rushing, and C. A. Cousin, "Closed-loop temperature and force control of additive friction stir deposition," *J. Manuf. Mat. Proc.*, 2022.

[4] M. E. Perry, R. J. Griffiths, D. Garcia, J. M. Sietins, Y. Zhu, and H. Z. Yu, "Morphological and microstructural investigation of the non-planar interface formed in solid-state metal additive manufacturing by additive friction stir deposition," *Additive Manufacturing*, vol. 35, p. 101293, oct 2020.

[5] M. B. Williams, T. W. Robinson, C. J. Williamson, R. P. Kinser, N. A. Ashmore, P. G. Allison, and J. B. Jordon, "Elucidating the effect of additive friction stir deposition on the resulting microstructure and mechanical properties of magnesium alloy WE43," *Metals*, vol. 11, no. 11, p. 1739, oct 2021.

[6] R. Griffiths, M. Perry, J. Sietins, Y. Zhu, N. Hardwick, C. Cox, H. Rauch, and H. Yu, "A perspective on solid-state additive manufacturing of aluminum matrix composites using meld," *J. Matter. Eng. Perform.*, vol. 28, 09 2018.

[7] R. J. Griffiths, D. Garcia, J. Song, V. K. Vasudevan, M. A. Steiner, W. Cai, and H. Z. Yu, "Solid-state additive manufacturing of aluminum and copper using additive friction stir deposition: Process-microstructure linkages," *Materialia*, vol. 15, p. 100967, mar 2021.

[8] G. Stubblefield, K. Fraser, B. Phillips, J. Jordon, and P. Allison, "A meshfree computational framework for the numerical simulation of the solid-state additive manufacturing process, additive friction stir-deposition (AFS-d)," *Materials &amp Design*, vol. 202, p. 109514, apr 2021.

[9] Z. Zhang, Z. Tan, J. Li, Y. Zu, W. Liu, and J. Sha, "Experimental and numerical studies of re-stirring and re-heating effects on mechanical properties in friction stir additive manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 1-4, pp. 767–784, jun 2019.
[10] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, "Next generation reservoir computing," *Nature Communications*, vol. 12, no. 1, sep 2021.
[11] B. Schrauwen, D. Verstraeten, and J. Campenhout, "An overview of reservoir computing: Theory, applications and implementations," 01 2007, pp. 471–482.
[12] K. Nakajima and I. Fischer, Eds., *Reservoir Computing*. Springer Singapore, 2021.
[13] H. Jaeger, "Echo state network," *scholarpedia*, vol. 2, no. 9, p. 2330, 2007.
[14] N. Doan, W. Polifke, and L. Magri, "Physics-informed echo state networks," *Journal of Computational Science*, vol. 47, p. 101237, nov 2020.
[15] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, apr 2007.
[16] C. Gallicchio, A. Micheli, and L. Pedrelli, "Design of deep echo state networks," *Neural Networks*, vol. 108, pp. 33–47, dec 2018.
[17] D. Canaday, A. Pomerance, and D. J. Gauthier, "Model-free control of dynamical systems with deep reservoir computing," *Journal of Physics: Complexity*, vol. 2, no. 3, p. 035025, sep 2021.
[18] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. pp. 185–194, 1999.
[19] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the "echo state network" approach," German National Research Center for Information Technology, Bremen, Tech. Rep. 159, 2002.
[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 436444, 2015.
[21] L. Gonon and J.-P. Ortega, "Fading memory echo state networks are universal," *Neural Networks*, vol. 138, pp. 10–13, jun 2021.
[22] A. G. Hart, K. R. Olding, A. M. G. Cox, O. Isupova, and J. H. P. Dawes, "Using echo state networks to approximate value functions for control," 2021.
[23] K. Dupree, P. M. Patre, Z. D. Wilcox, and W. E. Dixon, "Optimal control of uncertain nonlinear systems using a neural network and rise feedback," in *Proc. Am. Control Conf.*, St. Louis, Missouri, Jun. 2009, pp. 361–366.
[24] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, pp. 359–366, 1985.
[25] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, pp. 251–257, 1991.
[26] C. A. Cousin, P. Deptula, C. Rouse, and W. E. Dixon, "Cycling with functional electrical stimulation and adaptive neural network admittance control," in *Proc. Am. Control Conf.*, 2019, pp. 1742–1747.
[27] H. Dinh, S. Bhasin, R. Kamalapurkar, and W. E. Dixon, "Dynamic neural network-based output feedback tracking control for uncertain nonlinear systems," *ASME J. Dyn. Syst. Meas. Control*, vol. 139, no. 7, pp. 074 502–1–074 502–7, July 2017.
[28] N. Fischer, S. Bhasin, and W. E. Dixon, "Nonlinear control of an autonomous underwater vehicle: A RISE-based approach," in *Proc. Am. Control Conf.*, San Francisco, CA, 2011, pp. 3972–3977.
[29] O. S. Patil, A. Isaly, B. Xian, and W. E. Dixon, "Exponential stability with RISE controllers," *IEEE Control Systems Letters*, vol. 6, pp. 1592–1597, 2022.
[30] F. L. Lewis, R. Selmic, and J. Campos, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.
[31] N. Sharma, C. M. Gregory, M. Johnson, and W. E. Dixon, "Closed-loop neural network-based nmes control for human limb tracking," *IEEE Trans. Control Syst. Tech.*, vol. 20, no. 3, pp. 712–725, 2012.
[32] M. Lukoševičius, "A practical guide to applying echo state networks," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 659–686.