ASME Journal of Dynamic Systems, Measurement, and Control
Online journal at:
https://asmedigitalcollection.asme.org/dynamicsystems

Zihang Zhang
Department of Engineering Technology & Industrial Distribution,
Texas A&M University,
College Station, TX 77843;
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: zzhangau@tamu.edu

Xingyong Song[1]
Department of Engineering Technology & Industrial Distribution,
Texas A&M University,
College Station, TX 77843;
Department of Mechanical Engineering,
Texas A&M University,
College Station, TX 77843;
Department of Electrical and Computer Engineering,
Texas A&M University,
College Station, TX 77843
e-mail: songxy@tamu.edu

Designing Hybrid Neural Network Using Physical Neurons—A Case Study of Drill Bit-Rock Interaction Modeling

Neural networks have been widely applied in system dynamics modeling. One particular type of networks, hybrid neural networks, combines a neural network model with a physical model, which can increase rate of convergence in training. However, most existing hybrid neural network methods require an explicit physical model constructed, which sometimes might not be feasible in practice or could weaken the capability of capturing complex and hidden physical phenomena. In this paper, we propose a novel approach to construct a hybrid neural network. The new method incorporates the physical information to the structure of network construction, but does not need an explicit physical model constructed. The method is then applied to modeling of bit-rock interaction in the down-hole drilling system as a case study, to demonstrate its effectiveness in modeling complex process and efficiency of convergence in training. [DOI: 10.1115/1.4062631]

Keywords: neural network bit–rock interaction

1 Introduction

In recent years, neural network (NN) has increasingly been used to model complicated physical system. Unlike the pure physics-based models, NN is a data-based method where the intrinsic physical characteristics within the data are learned by layers of neurons and their connections. Its capability of modeling complex systems and identifying their hidden dynamical features makes the method widely applied in various fields such as agriculture, medical science, engineering, and management [1].

The idea of combining the physics-based approach and the neural networks approach together for modeling leads to an emerging category of methods called hybrid neural networks (HNN). By leveraging the known physics information, the training of neural networks can be more efficient with faster convergence. There are primarily two groups of hybrid neural networks. The first group [2,3] directly adds a neural network model on top of a physical model, where the neural network model is trained using the error between the physical model and the data used for training. The second group [4–10] is still primarily based on a physics model, where some parameters which are hard to observe or compute are trained and obtained using neural networks. However, most existing hybrid networks methods require a physical model to be explicitly constructed. This can cause two issues. First, in many cases, due to the complex physical process behind, constructing an explicit physical model may not be feasible, while only certain physical characteristics can be extracted. Second, directly using a physical model can make the hybrid model more dependent on the physical model and induce bias towards the preconstructed physical model. If the physical model is not accurately constructed, it can negatively affect the capability of neural networks to capture the actual physical process behind.

In this paper, we intend to explore a new direction of constructing hybrid neural networks. Instead of requiring a physical model to be available, we only assume some physical features of the system can be reliably captured. We then interpret this physical information into specific neurons and incorporate them in the overall neural network structure. The new method can strengthen the traditional neural networks with physical information but avoids explicit construction of a physical model.

The new hybrid modeling method will be explained in the context of drill bit–rock interaction modeling as a case study. Down-hole drilling is a critical technology that has been used not only in oil and gas production but also in enhanced geothermal energy systems (EGS) and specimen extraction in outer space exploration. To analyze, optimize, and control the drilling process, it is important to have an accurate and reliable drilling dynamics model [11]. One particular part that is challenging to model is the interaction between the drill bit and rock. There are some existing studies to model the bit–rock interaction physically [12–17]. However, due to variations in rock/earth property, complex down-hole conditions, it is hard to ensure accurate and reliable modeling results in practical operating conditions. Therefore, in this paper, we will model the bit–rock interaction using a new hybrid neural network framework to demonstrate its effectiveness.

The rest of the paper is organized as follows: First, the physical process of the bit-rock interaction is explained in Sec. 2. Second, Sec. 3 discusses the method to build the HNN structure. Then, Sec. 4 explains how the training data are obtained using a laboratory scale setup. Finally, the training results are presented in Sec. 5. The model is compared with the traditional neural network and pure physical

[1]Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received August 25, 2022; final manuscript received May 16, 2023; published online July 4, 2023. Assoc. Editor: Dongmei Chen.

model on test data. It is concluded that the proposed method has better performance in terms of convergence and accuracy. Section 6 concludes the paper.

2 Drill String Dynamics and Physics of Bit–Rock Interaction

According to Ref. [16], the drilling dynamics can be essentially described by a low dimension model. Based on the modeling method described in Ref. [18], the drilling system is simulated as a low-dimensional lumped-parameter model. As shown in Fig. 1, the drilling system is divided into two portions with top drive and drill string as the top portion and the bottom hole assembly (BHA) with the drill bit as the bottom portion. The axial and torsional displacement of the top and bottom portion are denoted as x_i and θ_i , respectively, and the equation of motion is expressed as

$$\begin{aligned} M_1\ddot{x}_1(t) &= C_a(\dot{x}_2 - \dot{x}_1) + K_a(x_2 - x_1) + M_1 g - W_b(t) \\ M_2\ddot{x}_2(t) &= -C_a(\dot{x}_2 - \dot{x}_1) - K_a(x_2 - x_1) + M_2 g - W(t) \\ I_1\ddot{\theta}_1(t) &= C_t(\dot{\theta}_2 - \dot{\theta}_1) + K_t(\theta_2 - \theta_1) - T_b(t) \\ I_2\ddot{\theta}_2(t) &= -C_t(\dot{\theta}_2 - \dot{\theta}_1) - K_t(\theta_2 - \theta_1) + T(t) \end{aligned} \tag{1}$$

where M_i and I_i are masses and inertias of the two portions, respectively. The connection of the two portions are modeled by spring and damping coefficients, where the axial and torsional coefficients are denoted as K_a, C_a and K_t, C_t, respectively. W and T are force and torque generated by the top drive (the control inputs), W_b and T_b are weight and torque on bit from the bit–rock interaction.

Since the drill string dynamics is known and well-established, this research will focus on modeling the bit–rock interaction W_b and T_b, which are typically hard to model due to the complex downhole condition and rock cutting process. The method in Refs. [15] and [18] is chosen as the baseline physical bit–rock interaction model. Note that, the baseline physical model referred here is only used for benchmark purpose in the result analysis and is not required in the construction of the hybrid neural network model. The description of the physical model below also explains some well-known physics of the bit–rock interaction that will be incorporated into the hybrid model construction in Sec. 3. Nevertheless, only part of the physics, rather than the full physical model, will be used in the hybrid model. The magnitudes of the force and torque components are determined by a physical variable called depth-of-cut (DOC). Depth-of-cut is defined as the total contact depth of the drill bit with the rock-cutting surface, which is governed by

$$d(t) = n(x_1(t) - x_1(t - \tau(t))) \tag{2}$$

where n is the number of the blades and $\tau(t)$ represents the delay for the blade to rotate the angle difference between two successive blades, described as

$$\frac{2\pi}{n} = \theta_1(t) - \theta_1(t - \tau(t)) \tag{3}$$

When $d < 0$, the drill bit is not in contact with the rock thus the reaction force and torque are both zero. Consider $d \geq 0$, the contact force and torque are separated into frictional and cutting components, denoted as

$$W_b(t) = W_b^f(t) + W_b^c(t) \tag{4}$$

$$T_b(t) = T_b^f(t) + T_b^c(t) \tag{5}$$

The cutting component is contributed by the removal of the rock in the cutting process, which is expressed as

$$W_b^c(t) = a\zeta\epsilon d(t) \tag{6}$$

$$T_b^c(t) = \frac{1}{2}a^2\epsilon d(t) \tag{7}$$

where a denotes the radius of the drill bit, ζ is a characteristic coefficient of the cutting surface orientation, and ϵ is the intrinsic specific energy to destroy a unit volume of rock.

The frictional component will first increase as the depth-of-cut increases, and become saturated when the depth-of-cut reaches a threshold value. Let d^* be the threshold value, then the frictional component is given as

$$W_b^f(t) = \begin{cases} W_{b1}^f(t) = a\sigma\kappa d(t)\dfrac{1 + \text{sign}(\dot{x}_1)}{2} & d < d^* \\ W_{b2}^f(t) = a\sigma\kappa d^* & d \geq d^* \end{cases} \tag{8}$$

$$T_b^f(t) = \frac{1}{2}a\mu\gamma W_b^f(t) \tag{9}$$

where σ denotes the maximum normal stress, κ represents the rate of variance of the contact length, μ is the coefficient of friction, and γ is a characteristic parameter representing the orientation and distribution of the contact forces.

In order to have a smooth derivative, a smooth transition is used between the two phases at d^*, the frictional force components is rewritten as

$$\begin{aligned} W_b^f(t) = W_{b1}^f(t)\frac{1 - \tanh\lambda(d(t) - d^* + \delta)}{2} \\ + W_{b2}^f(t)\frac{1 + \tanh\lambda(d(t) - d^* - \delta)}{2} \end{aligned} \tag{10}$$

where λ and δ are smoothing factors.

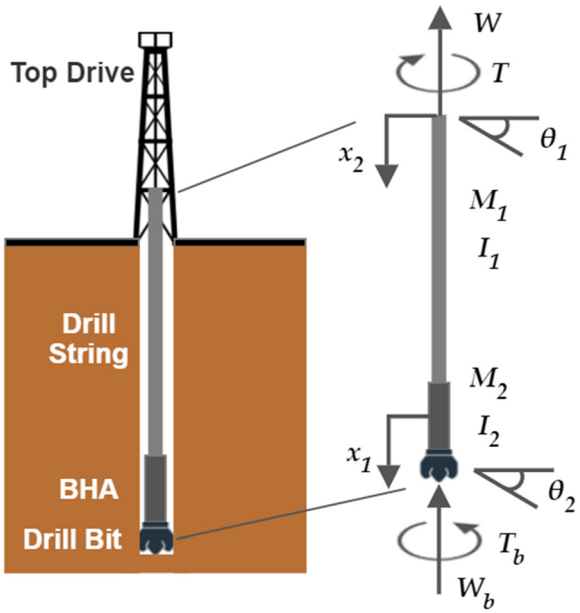

Fig. 1 Drill string schematic

3 Neural Network Structure

The limitation of the physical model above is that it can only be used to describe the process in an ideal condition. However, there can be other physical processes such as the flush of drilling fluid and possible accumulation of drilling rocks close to the bit that can impact the bit rock interaction but are hard to be explicitly modeled. To address this limitation, we propose to use a HNN model to model the bit-rock interaction process. The HNN does not need to rely on

the physical model explained in Sec. 2 but only incorporates some well-known physics in bit–rock interaction such as the depth of cut.

3.1 Incorporating Physics to Network Layers. As implied in Fig. 2, the physical process of bit-rock interaction is first decomposed to several processes, each of the process is defined as a layer of the network and outputs some physical parameters or states. If a process involves memories of states, it is interpreted as a recurrent layer. Otherwise a general fully connected layer is applied.

Based on this method, the network structure to model the bit-rock interaction is shown in Fig. 3(*a*): It has two layers, and the first layer is the recurrent layer that takes drill bit states as inputs and outputs a cut profile, which is the height of the rock relative to the drill bit, at sampled angles between two successive blades. The second layer outputs the reaction force and torque based on depth-of-cut information from the first layer output. Denote the inputs of the layers as $\mathbf{y}_1$, $\mathbf{y}_2$, and the outputs of the layers as $\mathbf{z}_1$, $\mathbf{z}_2$. The input to the network $\mathbf{y}_1$ consists of the axial displacement and velocity, torsional displacement, and velocity. The details of each layer are shown below.

3.1.1 Cut Profile Layer. This layer is a recurrent layer since the cut profile, which is the trajectory of the blade in $\frac{2\pi}{n}$ rad range, is a dynamic memory required to derive the next cut profile. The forward method of this layer at time t is written as

$$\mathbf{z}_1^{t+1} = F_1(\mathbf{y}_1^t, \mathbf{z}_1^t) \tag{11}$$

Below, the function F_1 is derived based on the interpolation formula.

First, a general interpolation formula is defined as follows: Write $\mathbf{a} = [a_1, a_2, \ldots, a_i]$, $\mathbf{b} = [b_1, b_2, \ldots, b_i]$, assume $a_1 < a_2 < a_3 < \cdots < a_i$, define the interpolated trajectory of points (a_1, b_1), (a_2, b_2), …, (a_i, b_i) as

$$I(x; \mathbf{a}, \mathbf{b}) : [a_1, a_i] \to \mathbb{R} \tag{12}$$

For simplicity, we use linear interpolation in this case study, which is

$$I(x; \mathbf{a}, \mathbf{b}) = b_j + \frac{(x - a_j)(b_{j+1} - b_j)}{a_{j+1} - a_j} \text{ if } a_j \le x < a_{j+1} \tag{13}$$

Second, the sampled points are computed. Assume there are p hidden layers (p outputs) in this layer, then at time t, they are written as $\mathbf{z}_1^t = [z_1^t, z_2^t, \ldots, z_p^t]^T$, and each hidden state represents the heights of rock relative to the drill bit at sampled angles, i.e., $\phi_1 = \frac{2\pi 1}{n\ p}$, $\phi_2 = \frac{2\pi 2}{n\ p}$, …, $\phi_{p-1} = \frac{2\pi p-1}{n\ p}$, $\phi_p = \frac{2\pi}{n}$ in rad. Let $\phi = [\phi_1, \phi_2, \ldots, \phi_p]^T$, the sampled points are denoted as $(0,0)$, $(\phi_1, z1^t)$, (ϕ_2, z_2^t), …, (ϕ_p, z_p^t). The sampled points coordinates are denoted as $\tilde{\phi} = [0, \phi_1, \phi_2, \ldots, \phi_p]$ and $\tilde{\mathbf{z}} = [0, z_1^t, z_2^t, \ldots, z_p^t]$.

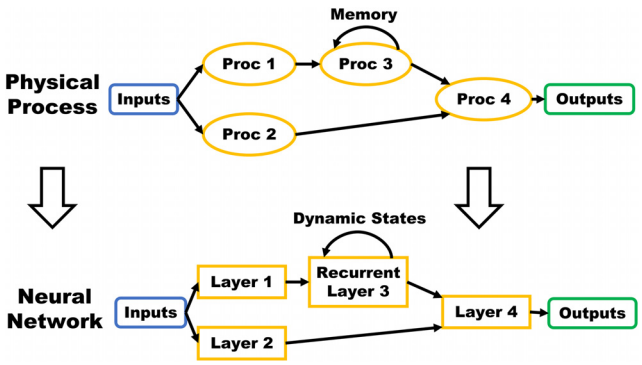

Fig. 2 Interpreting physics to network structure

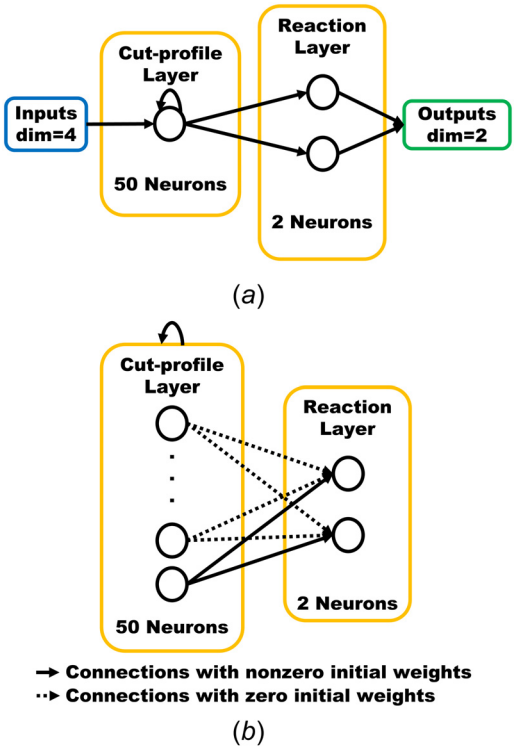

Fig. 3 Layers of the proposed hybrid neural network: (*a*) overview of layers and (*b*) details of connections

Third, the new sample points are computed when the layer receives a new input. The inputs can be decomposed to the axial and the torsional displacements of the drill bit, denoted as δx_1^t, $\delta\theta_1^t$, by which the sampling points are shifted. The new sampling points coordinates after the displacement are derived as

$$\hat{\phi} = [0, \delta\theta_1^t, \phi_1 + \delta\theta_1^t, \phi_2 + \delta\theta_1^t, \ldots, \phi_p + \delta\theta_1^t] \tag{14}$$

$$\hat{\mathbf{z}} = [0, \delta x_1^t, z_1^t + \delta x_1^t, z_2^t + \delta x_1^t, \ldots, z_p^t + \delta x_1^t] \tag{15}$$

Finally, the outputs are derived based on the interpolated trajectory of the new points, which is $I(\theta; \hat{\phi}, \hat{\mathbf{z}})$, thus the forward method can be derived as

$$\mathbf{z}_1^{t+1} = F_1(\mathbf{y}_1^t, \mathbf{z}_1^t) = I(\phi; \hat{\phi}, \hat{\mathbf{z}}) \tag{16}$$

which is to apply the interpolation at sampled angles ϕ to the sampling points $(\hat{\phi}_i, \hat{z}_i)$.

The backpropagation of the layer, which is similar to that of a recurrent layer, can be derived by chain rule

$$\frac{\partial L}{\partial y_1^n} = \frac{\partial L}{\partial z_1^{n+1}} \frac{\partial z_1^{n+1}}{\partial y_1^n} + \frac{\partial L}{\partial z_1^{n+2}} \frac{\partial z_1^{n+2}}{\partial z_1^{n+1}} \frac{\partial z_1^{n+1}}{\partial y_1^n} + \cdots \tag{17}$$

where L is the loss of the network.

3.1.2 Reaction Layer. This layer outputs the weight-on-bit and torque on bit based on the cut profile from the last layer. According to the definitions, DOC is equal to the axial displacement of drill bit within $\frac{2\pi}{n}$ torsional displacement, that is $d = \mathbf{y}_{2,p}$ where $\mathbf{y}_{2,i}$ denotes the ith element of $\mathbf{y}_2$. Based on the Eqs. (4)–(10), when drill bit velocities are positive, the reaction force and torque can be denoted as a function of depth-of-cut $W_b(d; \Theta)$, $T_b(d; \Theta)$ where Θ is the set of physical parameters in bit–rock interaction and the forward method is written as

$$\mathbf{z}_2 = \begin{bmatrix} W_b(\mathbf{y}_{2,p};\Theta) \\ T_b(\mathbf{y}_{2,p};\Theta) \end{bmatrix} + W_R\mathbf{y} + b_R \tag{18}$$

where W_R and b are weights and bias, respectively.

In this layer, Θ, W_R and b are the learnable parameters. Based on the physics of bit–rock interaction described in Sec. 2, Θ can be described by four independent parameters $\left[a\zeta\epsilon \quad \frac{1}{2}a^2\epsilon \quad a\sigma\kappa \quad \frac{1}{2}a^2\mu\gamma\sigma\kappa\right]$. W_R and b are both initialized as zeros to keep the physical information.

Note that Eqs. (4)–(10) provide an initial guess of the structure and parameters but the network does not rely on that. The physical parameters are assumed to be inaccurate and are learned during training. Even without a physical model, this layer can be replaced with a general function approximator. Nevertheless, the embedded known information will help improve the convergence and accuracy.

Finally a RELU activation function is added to this layer since if the depth-of-cut is negative, the drill bit does not contact the rock and there is no reaction force or torque.

After building the layers, normalization of data should be processed to stabilize training progress and improve convergence rate. In addition to normalization of the input data, the hidden layer also needs to be normalized since they are generated based on physical information and inappropriate scales will cause vanishing derivatives at the hidden layer. This is achieved by scaling the weights and bias of each layer.

3.2 Increasing the Network Complexity. The neural network with only physical information incorporated may not be sufficient to address any hidden dynamics or uncertainties within the physical process. Therefore, it is necessary to increase the network complexity to improve its capability of modeling complex process. This can be achieved by adding additional neurons and connections as well as additional layers.

3.2.1 Additional Neurons and Connections. As depicted at the top of Fig. 4, additional neurons and connections can be added to an existing layer. The weights of the additional connections to the former layer are initialized to zeros, making the initial values of the new neurons be zero, and the weights to the next layer are randomly initialized. This maintains the original physical properties of the neuron network and at the same time add network complexity to address uncertainties.

In the proposed network, new connections are added in cut-profile layer, which originally is a recurrent layer without learnable parameters. The neurons are now connected the same way as gated

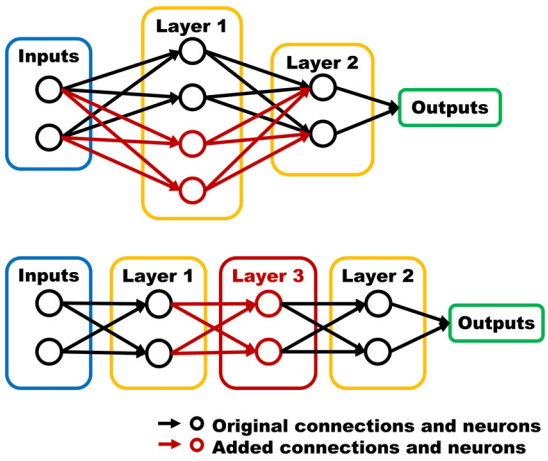

Fig. 4 Ways to add structures to increase network complexity. Top: additional neurons and bottom: additional layers.

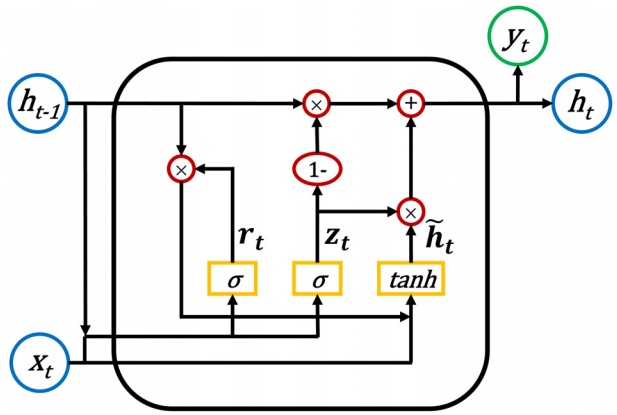

Fig. 5 Structure of gated recurrent unit

recurrent unit (GRU), with TANH activation function. GRU connections include an update and a reset gate [19] as shown in Fig. 5, which is used to process the memory of the layer. Its forward method is governed by

$$\begin{aligned} z_t &= \sigma_z(W_z x_t + R_z h_{t-1} + b_z) \\ r_t &= \sigma_r(W_r x_t + R_r h_{t-1} + b_r) \\ \hat{h}_t &= \sigma_h(W_h x_t + R_h(r_t \odot h_{t-1}) + b_h) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \end{aligned} \tag{19}$$

where z_t is the update gate, r_t is the reset gate, $\hat{h}_t$ is the candidate hidden state, and h_t is the hidden state and output. σ's represent activation functions, W's and R's are weights of input states and hidden states, respectively, b's are biases.

Note that GRU is chosen because of its simplicity and capability of avoiding vanishing and exploding gradient problem in recurrent neural network. The long short-term memory (LSTM) is another alternative if the computational power is sufficient [20].

3.2.2 Additional Layers. The bottom figure in Fig. 4 shows another way to add complexity which is to add additional fully connected layer (Layer 3) between existing layers (Layers 1 and 2) in series. The initial weights and bias of the new layer are designed to maintain the physical information in the original network, so that the outputs of Layer 1 in the original network (Fig. 4) are also included in the outputs of the new layer added. This helps keep the physical information transmitted between Layer 1 and Layer 2 in the original network, and at the same time, any unmodeled process or dynamics between Layers 1 and 2 can be captured by the new layer (Layer 3 in Fig. 4).

Specifically, in our proposed network, a new layer with 50 neurons is added between the cut-profile layer and the reaction layer, shown in Fig. 6. The forward method is written as

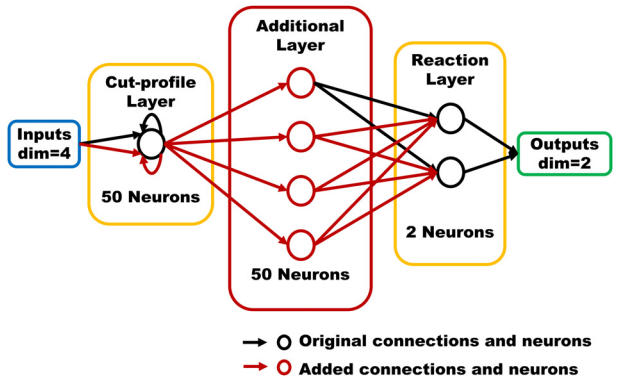

Fig. 6 Final structure of the proposed hybrid neural network

$$\mathbf{z}' = W\mathbf{y}' + B \tag{20}$$

where $\mathbf{z}' = \mathbf{y}_2$ is the output, and $\mathbf{y}' = \mathbf{z}_1$ is the input, $W \in \mathbb{R}^{50\times 50}$ and $B \in \mathbb{R}^{50\times 1}$ are weights and bias, respectively.

To maintain the physical information transferred from the cut profile layer to the reaction layer, the first rows of W and B are initialized as

$$\begin{aligned} W_1 &= [0 \quad 0 \quad 0 \quad \ldots \quad 0 \quad 1] \\ B_1 &= 0 \end{aligned} \tag{21}$$

which transfers the original physical information in the last neuron of the cut-profile layer to the first neuron of the additional layer. Other rows of W and B are randomly initialized.

This will also change connections bonded to the reaction layer. According to the original connections explained in Sec. 3.1.2, the reaction layer will extract the original physical information from the input out of the first neuron of the additional layer. $y_{2,p}$ in Eq. (18) will be changed to $y_{2,1}$ assuming y still represents the input to this layer.

The final network structure for bit–rock interaction model is shown in Fig. 6. It consists of three layers: the recurrent cut-profile layer has 50 hidden units/outputs and a total of $3 \times 50 \times 4 + 3 \times 50 \times 50 + 3 \times 50 = 8250$ learnable parameters, including the learnable parameters in GRU structure. The depth-of-cut layer is a fully connected layer with 50 outputs and totally $50 \times 50 + 50 = 2550$ learnable parameters. The reaction layer outputs axial and torsional control inputs and has $4 + 50 \times 2 = 104$ learnable parameters.

4 Training and Testing Data

Typically, it is hard to access field testing data for down-hole drilling system, and real-time force/torque data close to the drill bit are even more rare. Therefore, a lab scale down-hole drilling rig was built in our lab, and testing data were obtained to validate the proposed model.

Figure 7 shows the lab-scale drilling system. The driving torque is supplied on the top of the rig and transmitted to the drill bit at the bottom through drilling pipes with universal joints to allow direction change. Rubber is used to connect upper and lower drilling pipes to simulate the lowered effective stiffness for a large-scale drill pipe (the longer the drill pipe, the lower the effectiveness stiffness is). Load is placed on top of the rig and presses the drill pipe to simulate the load added on drill bit to push it to drill forward. Multiple sensors are installed close to the drill bit, including encoders to measure the bit rotational speed, load sensor to measure weight on bit, torque sensor to measure torque on bit. An accelerometer and a gyro are also available to measure axial and lateral velocities.

Due to the vibrating environment in drilling, the data collected from accelerometer and gyro cannot be directly used due to heavy high frequency noise. A Kalman filter is applied to estimate the displacement and velocity of drill bit from the accelerometer and gyroreadings [21].

First consider the axial direction, two measurements are obtained which are the velocity of the top drive v_t and the acceleration of the drill bit a_b. Consider the states as $\mathbf{X}_a = [x_b \quad v_b \quad a_b]^T$ including the axial displacement, velocity, and acceleration of the drill bit, the transition function can be derived as

$$\begin{aligned} x_b(n+1) &= x_b(n) + v_b(n)\delta t + \frac{1}{2}a_b(n)\delta t^2 \\ v_b(n+1) &= v_b(n) + a_b(n)\delta t \\ a_b(n+1) &= N_0 \end{aligned} \tag{22}$$

where N_0 is a Gaussian noise. The transition function can be written into matrix form

$$\mathbf{X}_a(n+1) = A_a\mathbf{X}_a(n) + N_a(n)$$

$$A_a = \begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 0 \end{bmatrix}, \quad N_a(n) = \begin{bmatrix} 0 \\ 0 \\ N_0 \end{bmatrix} \sim (0, Q_a) \tag{23}$$

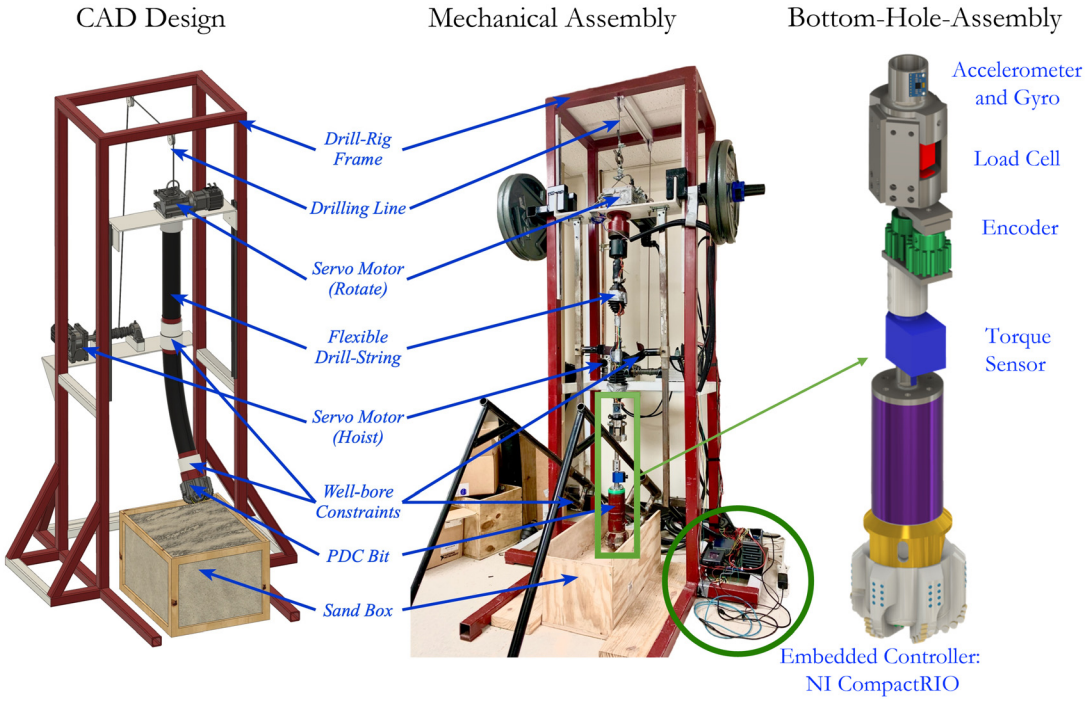

Fig. 7 Drilling experiment setup

The measurement model can be expressed as

$$\mathbf{Z}_a(n) = C_a\mathbf{X}_a(n) + V_a(n)$$
$$C_a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad V_a(n) = \begin{bmatrix} N_v \\ N_a \end{bmatrix} \sim (0, R_a) \tag{24}$$

where N_v is the noise of the velocity relative to the top drive and N_a is the noise from the accelerometer measurement.

Then, the data can be processed by the Kalman filter method. Assume the mean and covariance of states are μ and Σ, respectively, the mean and covariance of the prediction from the transition model are

$$\tilde{\mu}(n) = A_a\mu(n) \tag{25}$$

$$\tilde{\Sigma}(n) = A_a\Sigma(n)A_a^T + Q_a \tag{26}$$

Kalman gain is defined as

$$K(n) = \tilde{\Sigma}(n)C_a^T(C_a\tilde{\Sigma}(n)C_a^T + R_a)^{-1} \tag{27}$$

The mean and covariance of filtered states are derived as

$$\mu(n+1) = \tilde{\mu}(n) + K(n)(\mathbf{Z}_a(n) - C_a\tilde{\mu}(n)) \tag{28}$$

$$\Sigma(n+1) = \tilde{\Sigma}(n) - K(n)C_a\tilde{\Sigma}(n) \tag{29}$$

The same method can be applied to the torsional direction, but the difference is that the gyro measures the velocity rather than acceleration. Consider the state $\mathbf{X}_t = [\,\theta_b \quad \omega_b \quad \alpha_b\,]^T$ including torsional displacement, velocity, and acceleration, the transition model is governed by

$$\mathbf{X}_t(n+1) = A_t\mathbf{X}_t(n) + N_t(n)$$
$$A_t = \begin{bmatrix} 1 & \delta t & \frac{1}{2}\delta t^2 \\ 0 & 1 & \delta t \\ 0 & 0 & 0 \end{bmatrix}, \quad N_t(n) = \begin{bmatrix} 0 \\ 0 \\ N_1 \end{bmatrix} \sim (0, Q_t) \tag{30}$$

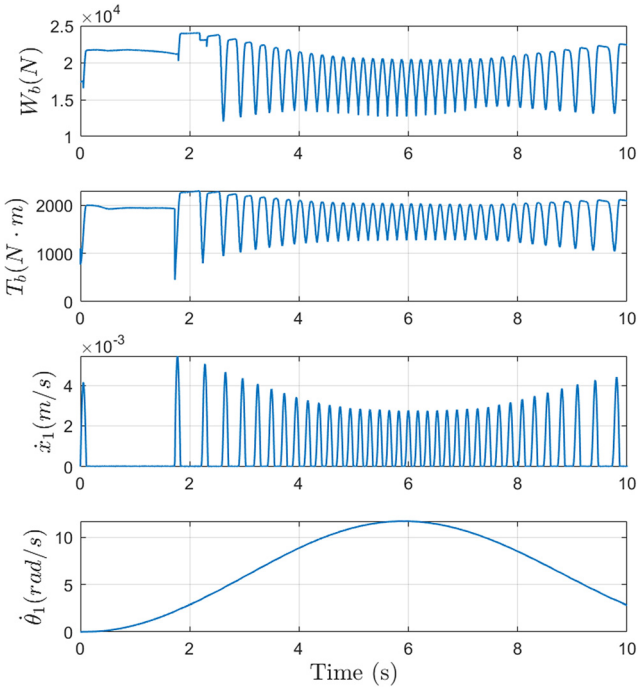

Fig. 8 Data used for training, from top to bottom: force, torque, axial velocity, and torsional velocity

The measurement model is

$$\mathbf{Z}_t(n) = C_t\mathbf{X}_t(n) + V_t(n)$$
$$C_t = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad V_t(n) = \begin{bmatrix} N_{\omega 1} \\ N_{\omega 2} \end{bmatrix} \sim (0, R_t) \tag{31}$$

Then, the states can be estimated using Kalman filter in a similar fashion as the axial direction.

Given the limited load that can be exerted on the lab scale system, the final data used are a combination of data from a physical model and data collected from experiment. The data collected from the physical model are used to ensure the data is in the right scale, and the data from experiment is to replicate uncertainty such as the effect of accumulation of rocks, etc. The dimension of the data is 4×10000, with sampling frequency of 200 Hz, and total time ranging from 0 to 50 s. The first 10 s are used for training while the next 40 s are used for testing. The data for training is shown in Fig. 8.

5 Results

The hybrid neural network, as explained in Sec. 3, is trained on the data obtained in Sec. 4. In comparison, the same data is used to train a traditional neural network. This network has four layers: the first layer is a recurrent GRU layer with 50 neurons and TANH activation function, followed by two fully connected layers each with 50 neurons and RELU activation function, and a fully connected layer with two outputs at last. In addition, the initial conditions of the hybrid network are shifted by random constants. The physical parameters used of the drilling system are shown in Table 1. The learning rates of both networks are set to 0.005. The training results are shown below.

5.1 Training Results. The training progress is shown in Fig. 9 and the results are shown in Figs. 10 and 11. According to Fig. 9, the

Table 1 Parameters of the drilling model

M_1	44,187 kg	I_1	1685 kg m^2
M_2	29,028 kg	I_2	1187 kg m^2
C_a	34,400 Ns/m	C_t	49.5 Nms/rad
K_a	353,000 N/m	K_t	495 Nm/ra
ϵ	77×10^6 Pa	$\gamma\mu$	0.7
ζ	0.64	n	5
κ	5	σ	6.2572×10^7 Pa
a	0.15 m	D^*	1.8721×10^{-4} m
λ	5×4	δ	6×10^{-7}

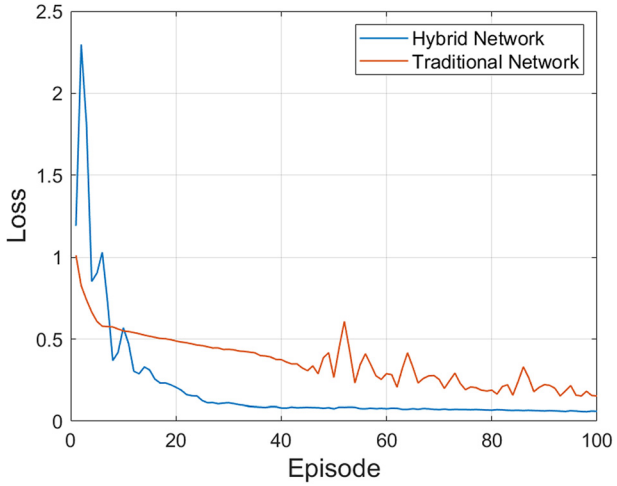

Fig. 9 Training progress of different neural networks

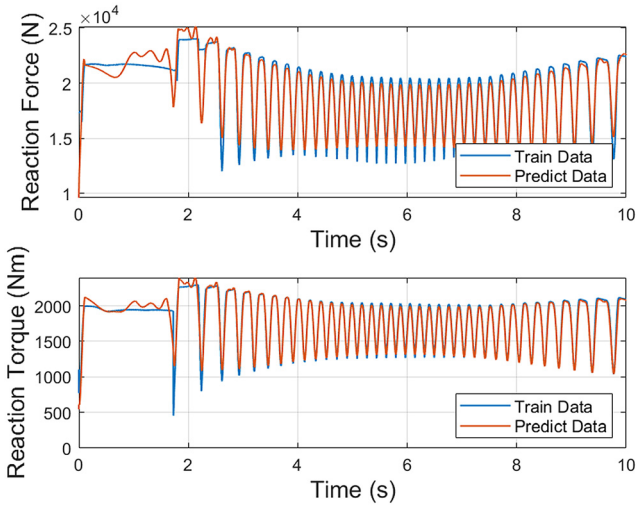

Fig. 10 Training results of hybrid neural network

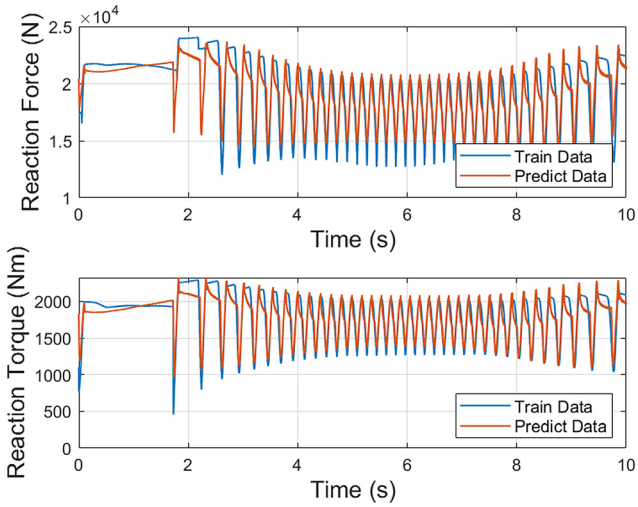

Fig. 11 Training results of traditional neural network

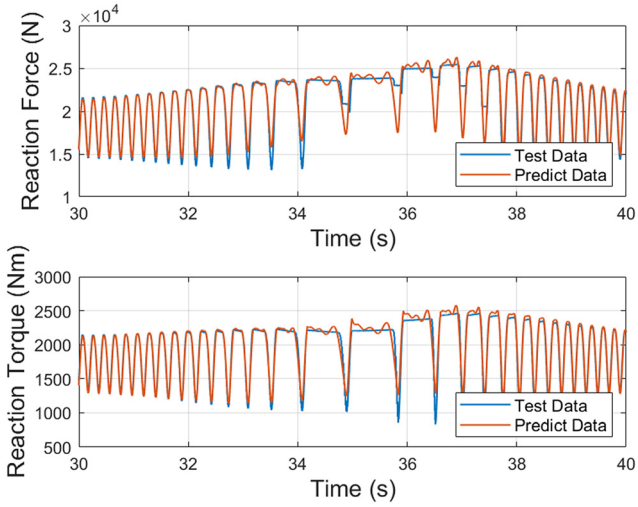

Fig. 12 Test results of hybrid neural network

loss of the traditional network drops during the training but the rate of change gets smaller after five episodes, and the loss remains almost the same after 70 episodes, implying the convergence of the neural network. The final loss is around 0.16.

For the hybrid neural network, it has higher initial loss than the traditional network. The loss increases at first but quickly drops as the episode number increases. It converges to around 0.15 and does not change too much after 40 episodes. This implies a much faster convergence rate and a smaller training loss when using the hybrid network, even with a worse initial condition.

5.2 Testing Results. Both models are applied to testing data for validation, and the results from 30 to 40 s are shown in Figs. 12 and 13. It can be observed that the hybrid network model can achieve much better results than the traditional network.

In addition, the modeling result is also compared to using a pure physical model described in Sec. 2. It can be shown in Fig. 14 that the

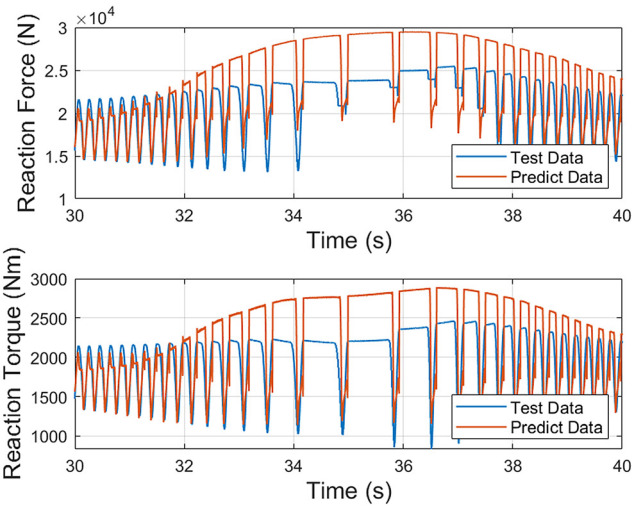

Fig. 13 Test results of traditional neural network

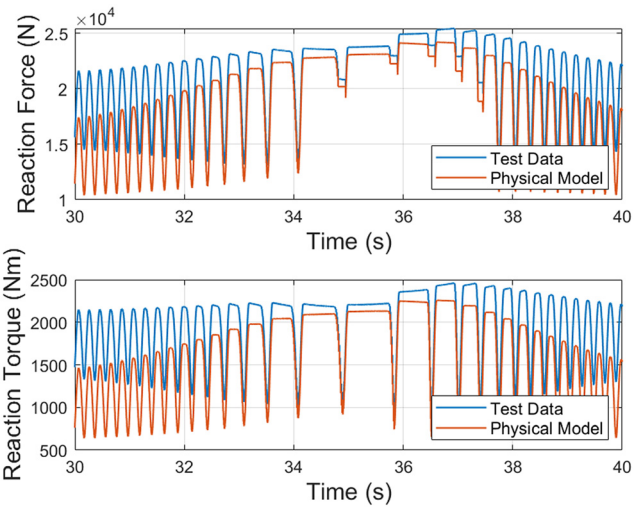

Fig. 14 Simulation results of physical model

Table 2 RMSE of different methods on testing data

Method	Reaction force (N)	Reaction torque (Nm)
Hybrid network	1007.4	75.8
Traditional network	3401.7	355.5
Physical model	2736.5	452.4

accuracy of the proposed approach is also higher than that using physical model only.

A summary of the root-mean-squared error (RMSE) is shown in Table 2 to quantify the performance of different methods.

6 Conclusion

A new method is presented to build a hybrid network by interpreting the physical information into specific neurons and incorporating them in the overall neural network structure. The new hybrid method does not require a physical model to be explicitly available. We then use modeling of bit–rock interaction in the downhole drilling process as a case study. The training results, as compared with the results from a traditional neural network and a benchmark physical model, show that the hybrid network is more effective in terms of convergence rate and modeling accuracy.

Funding Data

- National Science Foundation (Grant No. 2045894; Funder ID: 10.13039/100000001).

Data Availability Statement

The authors attest that all data for this study are included in the paper.

References

[1] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H., 2018, "State-of-the-Art in Artificial Neural Network Applications: A Survey," Heliyon, **4**(11), p. e00938.

[2] Garcia, H. E., and Vilim, R. B., 1998, "Combining Physical Modeling, Neural Processing, and Likelihood Testing for Online Process Monitoring," SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 1, IEEE, San Diego, CA, Oct. 11–14, pp. 806–810.

[3] Forssell, U., and Lindskog, P., 1997, "Combining Semi-Physical and Neural Network Modeling: An Example Ofits Usefulness," IFAC Proc. Vol., **30**(11), pp. 767–770.

[4] Piron, E., Latrille, E., and Rene, F., 1997, "Application of Artificial Neural Networks for Crossflow Microfiltration Modelling: 'Black-Box' and Semi-Physical Approaches," Comput. Chem. Eng., **21**(9), pp. 1021–1030.

[5] Qi, H., Zhou, X.-G., Liu, L.-H., and Yuan, W.-K., 1999, "A Hybrid Neural Network-First Principles Model for Fixed-Bed Reactor," Chem. Eng. Sci., **54**(13–14), pp. 2521–2526.

[6] Wang, S., Wang, F., Devabhaktuni, V. K., and Zhang, Q.-J., 1999, "A Hybrid Neural and Circuit-Based Model Structure for Microwave Modeling," 1999 29th European Microwave Conference, IEEE, Munich, Germany, Oct. 5–7, pp. 174–177.

[7] Dolara, A., Grimaccia, F., Leva, S., Mussetta, M., and Ogliari, E., 2015, "A Physical Hybrid Artificial Neural Network for Short Term Forecasting of pv Plant Power Output," Energies, **8**(2), pp. 1138–1153.

[8] Cao, M., Wang, K., Fujii, Y., and Tobler, W., 2004, "A Hybrid Neural Network Approach for the Development of Friction Component Dynamic Model," ASME J. Dyn. Sys., Meas., Control, **126**(1), pp. 144–153.

[9] Roubos, J., Krabben, P., Setnes, M., Babuska, R., Heijnen, J., and Verbruggen, H., 1999, "Hybrid Model Development for Fed-Batch Bioprocesses; Combining Physical Equations With the Metabolic Network and Black-Box Kinetics," 6th Workshop on Fuzzy Systems, Brunel University, Uxbridge, London, UK, Sept. 8–9.

[10] Olausson, P., Hägsta Hl, D., Arriagada, J., Dahlquist, E., and Assadi, M., 2003, "Hybrid Model of an Evaporative Gas Turbine Power Plant Utilizing Physical Models and Artificial Neural Networks," ASME Paper No. GT2003-38116.

[11] Krama, A., Gharib, M., Refaat, S. S., and Palazzolo, A., 2021, "Design and Hardware in-the-Loop Validation of an Effective Super-Twisting Controller for Stick-Slip Suppression in Drill-String Systems," ASME J. Dyn. Syst., Meas., Control, **143**(11), p. 111008.

[12] Balanov, A., Janson, N., McClintock, P. V., Tucker, R., and Wang, C., 2003, "Bifurcation Analysis of a Neutral Delay Differential Equation Modelling the Torsional Motion of a Driven Drill-String," Chaos, Solitons Fractals, **15**(2), pp. 381–394.

[13] Khulief, Y., Al-Sulaiman, F., and Bashmal, S., 2007, "Vibration Analysis of Drillstrings With Self-Excited Stick–Slip Oscillations," J. Sound Vib., **299**(3), pp. 540–558.

[14] Richard, T., Germay, C., and Detournay, E., 2007, "A Simplified Model to Explore the Root Cause of Stick–Slip Vibrations in Drilling Systems With Drag Bits," J. Sound Vib., **305**(3), pp. 432–456.

[15] Besselink, B., Vromen, T., Kremers, N., and Van De Wouw, N., 2016, "Analysis and Control of Stick-Slip Oscillations in Drilling Systems," IEEE Trans. Control Systems Technol., **24**(5), pp. 1582–1593.

[16] Germay, C., Denoël, V., and Detournay, E., 2009, "Multiple Mode Analysis of the Self-Excited Vibrations of Rotary Drilling Systems," J. Sound Vib., **325**(1–2), pp. 362–381.

[17] Ritto, T. G., Soize, C., and Sampaio, R., 2009, "Non-Linear Dynamics of a Drill-String With Uncertain Model of the Bit–Rock Interaction," Int. J. Non-Linear Mech., **44**(8), pp. 865–876.

[18] Tian, D., and Song, X., 2019, "Control of a Downhole Drilling System Using Integral Barrier Lyapunov Functionals," American Control Conference (ACC), Philadelphia, PA, July 10–12, pp. 1349–1354.

[19] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y., 2014, "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. Association for Computational Linguistics, Oct. 25–29, pp. 1724–1734.

[20] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., 2014, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," CoRR, vol. abs/1412.3555, 2014.

[21] Wang, Y., Sun, Z., and Stelson, K. A., 2011, "Modeling, Control, and Experimental Validation of a Transient Hydrostatic Dynamometer," IEEE Trans. Control Syst. Technol., **19**(6), pp. 1578–1586.