mathematics

MDPI

Article

Recurrent Neural Network with Finite Time Sampling for Dynamics Identification in Rehabilitation Robots

Ahmed Alotaibi [1,2] and Hajid Alsubaie [1,2,*]

[1] Department of Mechanical Engineering, College of Engineering, Taif University, Taif 21944, Saudi Arabia; am.alotaibi@tu.edu.sa
[2] King Salman Center for Disability Research, Riyadh 11614, Saudi Arabia
* Correspondence: h.alsubaie@tu.edu.sa

Abstract: Rehabilitation robots can establish a direct connection between the user's nerve signals and the robot's actuators by integrating with the human nervous system. However, uncertainties in these systems limit their performance and accuracy. To address this challenge, the current study introduces an algorithm that effectively identifies and predicts unfamiliar dynamics in lower-limb rehabilitation robots. To accomplish this, the current study initially presents the dynamic model of a knee rehabilitation robot. Then, a finite time sampler is developed and the algorithm is proposed. In the proposed algorithm, the electromyographic signals are input into the rehabilitation robot. Via the use of a guaranteed stable sampler, samples from the unknown dynamics are extracted. By training the recurrent neural network with the acquired samples, the algorithm effectively learns and captures the underlying patterns of the unknown dynamics. The proposed recurrent neural network is enhanced with a self-attention mechanism, which plays a vital role in devising effective strategies for practical applications. Numerical simulation demonstrates the algorithm's effectiveness, highlighting its excellent performance in identifying the system's unknown dynamics.

Keywords: knee rehabilitation robot; finite time sampling; recurrent neural network; self-attention mechanism

MSC: 92C60; 93C85; 68T05; 68T45; 93C40; 93B40

Citation: Alotaibi, A.; Alsubaie, H. Recurrent Neural Network with Finite Time Sampling for Dynamics Identification in Rehabilitation Robots. *Mathematics* **2023**, *11*, 3731. https://doi.org/10.3390/math11173731

Academic Editors: Rongwei Guo, Cuimei Jiang and Ruimin Xu

Received: 13 July 2023
Revised: 18 August 2023
Accepted: 25 August 2023
Published: 30 August 2023

1. Introduction

In recent years, there has been a growing focus on the research and development of rehabilitation robots due to their potential in providing personalized and effective rehabilitation for individuals with mobility impairments resulting from neurological disorders such as stroke, spinal cord injury, or brain injury [1,2]. These robots offer a new approach to traditional rehabilitation methods by addressing their limitations. Rehabilitation robots can establish a direct connection between the user's nerve signals and the robot's actuators by integrating with the human nervous system [3–5]. This integration allows for precise and natural movements, enhancing the effectiveness of rehabilitation exercises [6]. They enable more intensive and repetitive training, crucial to the formation of new neural pathways in the brain. Rehabilitation robots also provide customized and targeted training, which is challenging to achieve with traditional methods, and offer objective measurements of patient progress, allowing clinicians to monitor advancements and tailor treatment regimens accordingly [7]. Additionally, these robots create a safe and controlled environment for patients to exercise and recover, minimizing the risk of further injury. With ongoing advancements in this field, the future holds immense promise for the integration of rehabilitation robots with the human nervous system, ushering in a new era of personalized and efficient rehabilitation for individuals with mobility impairments [8,9].

The presence of unknown dynamics within rehabilitation robots can have a detrimental impact on the system's performance and raise safety concerns for patients [10].

Consequently, it is vital to accurately identify and estimate these disturbances to develop effective control strategies that can minimize their effects [11]. Despite the inherent challenges, conventional techniques such as adaptive control strategies [12,13] and Kalman filters [14,15] have been extensively employed for estimating and predicting disturbances in rehabilitation robots. While these methods provide theoretical guarantees and facilitate the creation of adaptable control strategies that can accommodate the system's dynamic behavior, it is worth emphasizing that the majority of existing algorithms rely on conventional approaches and lack the incorporation of state-of-the-art techniques. Thus, there remains ample opportunity for advancements in the field of dynamic identification within rehabilitation robotics.

In this regard, utilizing neural networks to estimate the dynamics of rehabilitation robots offers distinct advantages over traditional methods. Neural networks, given their universal approximation capabilities, can adeptly estimate complex dynamics without presuming the nature of unknown disturbances, a limitation often seen in conventional techniques. Furthermore, neural networks provide a broader, controller-independent approach, ensuring versatility. Their ability to model intricate nonlinear relationships might also yield more accurate representations of robotic dynamics, especially in contexts with nonlinear or complex events. The dynamic nature of human–robot interactions in rehabilitation settings necessitates an adaptable system, a requirement neural networks meet by adjusting to incoming data in real-time. However, the application of such networks demands extensive data and computational resources. Thus, while neural networks hold promise, their deployment mandates careful consideration.

One of the significant challenges in identifying non-linear real-world systems lies in the sampling process [16,17]. The accuracy of samples is not always guaranteed, and there are instances when the required information is not readily accessible. The process of acquiring precise samples from the unknown dynamics of rehabilitation systems presents a substantial challenge in their dynamic identification. Consequently, numerous studies within this field tend to address the unknown dynamics as disturbances in control applications, rather than undertaking their identification. Nevertheless, recent advancements have paved the way for accurate estimation of the unknown dynamics of these systems, allowing the acquired information, as well as insights from the physics of the problem, to be effectively utilized in subsequent tasks of rehabilitation robots. By integrating these advanced techniques into dynamic identification, along with leveraging the inherent knowledge of the system's physical behavior, there exists a significant potential to enhance the control and overall performance of rehabilitation robots. Such an approach recognizes the critical role played by precise dynamic identification in ensuring the efficient operation of these robots.

One of the state-of-the-art techniques barely used for identification in rehabilitation robots is self-attention-based neural networks [18]. This powerful technique captures relationships in sequential or spatial data, improving accuracy in recognizing body movements and coordinating rehabilitation activities. In recent years, there has been significant research and development in the field of self-attention-based neural networks, with promising results in various domains such as natural language processing [19,20], computer vision [21,22], and reinforcement learning [23,24]. However, their application in rehabilitation robots for identification purposes is still underexplored. One of the key advantages of self-attention-based neural networks is their ability to consider the relationships between different parts of the input sequence or spatial configuration. For instance, in the case of human motion analysis, the robot needs to understand how various body parts interact and move in coordination. By using self-attention, the neural network can effectively learn and capture these dependencies, enabling more accurate identification and tracking of body movements. Further research is needed to fully explore its potential in this field.

Advancements in rehabilitation robot research have the potential to significantly improve patients' quality of life. In line with this objective, our study introduces a novel algorithm that utilizes a finite time sampler and neural network to uncover the unknown

dynamics of rehabilitation robots. To the best of our knowledge, there is no prior research that applies finite time sampling to rehabilitation robots and then utilizes this information within a neural network to uncover the robot's undisclosed dynamics. What distinguishes our proposed algorithm is its robustness, physics-informed nature, and integration of a distinctive self-attention layer within the neural network, ensuring its robustness and dependability. The employed self-attention mechanism allows the neural network to assign different weights to different elements of the input sequence or spatial layout, emphasizing the most relevant information for the identification task. This attention mechanism enables the network to focus on important features while suppressing irrelevant or noisy inputs, leading to improved performance and robustness. Additionally, the algorithm combines a finite time sampler with the self-attention recurrent neural network (RNN) to provide real-time samples of unknown dynamics.

The structure of the current study is as follows. In Section 2, a description of the dynamical model used in a 2-DOF knee rehabilitation robot is provided. Section 3 outlines the finite time sampling process and establishes its convergence. In Section 4, the proposed algorithm is delineated and its advantages are discussed. Section 5 presents the testing of the proposed algorithm in two experiments, highlighting its excellent performance. Finally, in Section 6, the study concludes by presenting the findings and suggesting future research directions.

2. Dynamic of the System

The general mathematical formulation of a rehabilitation robotic system, comprising a collection of n interconnected elements, is presented with the following second-order ordinary differential equation (ODE) [25]:

$$I(\phi)\ddot{\phi} + \zeta\left(\phi, \dot{\phi}\right)\dot{\phi} + K(\phi) + J^T(\phi)h(t) = \tau(t) \tag{1}$$

The dynamics of the robotic system are represented by the position (ϕ), velocity ($\dot{\phi}$), and acceleration ($\ddot{\phi}$) vectors, which exist in a multidimensional real number space (R^n). The input torque ($\tau(t)$)) is also a part of this space, while the inertia matrix ($I(\phi)$) is a symmetric and positive definite matrix with dimensions n × n. Moreover, ($\zeta\left(\phi, \dot{\phi}\right)\dot{\phi}$) denotes the combined effects of Coriolis and centripetal forces, and $K(\phi)$ indicates the gravitational forces exerted on the system. Additionally, $J(\phi)$ is the Jacobian matrix and is assumed to be nonsingular. Additionally, the vector representing the constrained force exerted by constraints is denoted by $h(t)$. It is important to mention that the matrix $I(\phi)$ meets the requirements of symmetry as stated in the study by Lee and Harris [26]. Additionally, this dynamic equation of the systems can be reformulated using a linear parameterization as follows:

$$I(\phi)\ddot{\phi} + \zeta\left(\phi, \dot{\phi}\right)\dot{\phi} + K(\phi) = R\left(\phi, \dot{\phi}, \ddot{\phi}\right) \tag{2}$$

In this context, ϕ belongs to the p-dimensional real number space (R^p). The $R\left(\phi, \dot{\phi}, \ddot{\phi}\right)$ is of dimensions ($n \times p$) and is called the regression matrix. This matrix encompasses all unknown functions of the signals $\phi(t)$, $\dot{\phi}(t)$, and $\ddot{\phi}(t)$. It is assumed that the constrained force $h(t)$ remains within a bounded range. From an engineering perspective, this assumption is considered reasonable, as the time-varying constrained force $h(t)$ is expected to be limited.

In Figure 1, we observe a knee rehabilitation robot with a 2-degree-of-freedom (DOF) configuration, specifically designed for vertical plane movements. The mechanical structure of the robot closely resembles that of a human leg, replicating its functionality. The knee joint, the focal point of the system, possesses two revolute joints. To actuate these joints, two motors are employed, connected via a transmission system. The motors are under the control of a computer, which receives input from sensors and electromyography signals. These sensors are responsible for measuring crucial parameters such as joint position, velocity, and acceleration, along with motor torque and current. Based on the acquired sensor

data and electromyography signals the controller performs calculations and determines the necessary torque inputs required to achieve a desired trajectory. Subsequently, these calculated torque inputs are sent back to the motors, enabling precise control over the robot's movements.

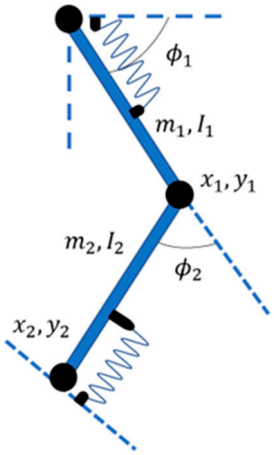

Figure 1. The structure of the 2-DOF knee rehabilitation robotic.

To obtain the equation of motion that corresponds to the parameters in Equation (1), we make use of the Euler–Lagrange equations. In this analysis, we designate ϕ as the vector $\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}$, which represents the generalized coordinates. These coordinates are equivalent to the variables ϕ_1 and ϕ_2 in Figure 1. The kinetic energy of the system can be mathematically formulated as follows:

$$T\left(\phi,\dot{\phi}\right)=\frac{1}{2}\left[\left(I_1+m_1l_{e1}^2+m_2l_1^2+m_2l_1l_{e2}\right)\dot{\phi}_1^2+m_2l_1l_{e2}\dot{\phi}_1\dot{\phi}_2\cos(\phi_2)+\left(I_2+m_2l_{e2}^2\right)\left(\dot{\phi}_1+\dot{\phi}_2\right)^2\right] \tag{3}$$

In this context, the distance from joint $i-1$ to the center of mass of link i is denoted by e_i, in which i can have values of 1 and 2. m_1 and m_2 denote the masses of links 1 and 2, respectively. Additionally, I_i represents the moment of inertia of link i around an axis perpendicular to the page and located at the center of mass of link i. The potential energy is mathematically expressed as:

$$V_p(\phi)=(m_1gl_{e2}+m_2gl_1)\sin(\phi_1)+m_2gl_{e2}\sin(\phi_1+\phi_2). \tag{4}$$

By applying Lagrange's equation, we derive the system's dynamics as follows:

$$\begin{aligned} I(\phi)&=\begin{bmatrix} m_1l_{e1}^2+m_2\left(l_1^2+l_{e2}^2+2l_1l_{e2}\cos\phi_2\right)+I_1+I_2 & m_2\left(l_{e2}^2+l_1L_{e2}\cos\phi_2\right)+I_2\\ m_2\left(l_{e2}^2+l_1l_{e2}\cos\phi_2\right)+I_2 & m_2l_{e2}^2+I_2\end{bmatrix}\\ K(q)&=\begin{bmatrix}(m_1l_{e2}+m_2l_1)g\cos\phi_1+m_2l_{e2}g\cos(\phi_1+\phi_2)\\ m_2l_{e2}g\cos(\phi_1+\phi_2)\end{bmatrix}\\ \zeta\left(\theta,\dot{\theta}\right)&=\begin{bmatrix}-m_2l_1l_{e2}\dot{\phi}_2\sin(\phi_2) & -m_2l_1l_{e2}\left(\dot{\phi}_1+\dot{\phi}_2\right)\sin\phi_2\\ m_2l_1l_{e2}\dot{\phi}_1\sin\phi_2 & 0\end{bmatrix}\end{aligned} \tag{5}$$

The Jacobian matrix is described as follows:

$$J(\phi)=\begin{bmatrix}-l_1\sin\phi_1+l_2\sin(\phi_1+\phi_2) & -l_2\sin(\phi_1+\phi_2)\\ l_1\cos\phi_1+l_2\cos(\phi_1+\phi_2) & l_2\cos(\phi_1+\phi_2)\end{bmatrix} \tag{6}$$

3. Finite Time Sampling

In this section, we present a finite time sampling technique for the rehabilitation robot and establish its convergence using the principles of Lyapunov stability theory. Consider the overall state space equation of a rehabilitation robot by introducing the notation $Q_1 = [\phi_1, \phi_2]^T$ for $Q_2 = \left[\dot{\phi}_1, \dot{\phi}_2\right]^T$, we can express the dynamics of the robot as follows:

$$\begin{aligned} \dot{Q}_1 &= Q_2, \\ \dot{Q}_2 &= f(t, Q_1, Q_2) + I(Q_1)^{-1}\tau(t) + f_u(t, Q_1, Q_2) \end{aligned} \tag{7}$$

Taking into account Equation (1), one can derive the following expression:

$$f(t, Q_1, Q_2) = I(Q_1)^{-1}(-\zeta(Q_1, Q_2)Q_1 - K(Q_1)) \tag{8}$$

$$f_u(t, Q_1, Q_2) = I(Q_1)^{-1}J^T(Q_1)h(t) \tag{9}$$

To encompass both the uncertain terms of the system and the unknown dynamics within a single term, we introduce $f_u(t, x)$, which the proposed sampling mechanism should estimate in finite time.

Unlike most sampling methods, which do not guarantee the accuracy of their samples, here we prove the accuracy of our sampling algorithm by leveraging Lemmas 1 and 2, which are given by:

Lemma 1 ([27]). *Suppose a continuously differentiable positive definite function $V(t)$ that fulfills the following inequality*:

$$\dot{V}(t) + aV(t) + bV(t)^c \leq 0, \qquad \forall t > t_0 \tag{10}$$

Assuming that $a > 0 > b$, and $0 < c < 1$ the function $V(t)$ demonstrates convergence to the equilibrium point within a finite time period.

Proof. Starting from the differential inequality (10), we have:

$$\dot{V}(t) \leq -aV(t) - bV(t)^c \qquad \forall t > t_0 \tag{11}$$

The derivative satisfies $\dot{V}(t) \leq 0$, since we know both $-aV(t)$ and $-bV(t)^c$ terms are either negative or zero. Our objective is to establish finite time convergence of $V(t)$. To accomplish this, first, we separate the terms in Equation (11) and then integrate both sides over the interval $[t_0, t]$:

$$\int_{V(t_0)}^{V(t)} \frac{dV}{aV(t) + bV(t)^c} \leq \int_{t_0}^{t} -dt = t_0 - t \tag{12}$$

Since $\dot{V}(t) \leq 0$, we know $V(t)$ is either decreasing or it is fixed; thus, we have $V(t) \leq V(t_0)$. Considering this and due to the fact that $V(t)$ is definitely positive, we have $V(t)^c \leq V(t_0)^c$. We use this inequality, and this gives an upper bound on the integral as follows:

$$\int_{V(t_0)}^{V(t)} \frac{dV}{aV(t) + bV(t_0)^c} \leq t_0 - t \tag{13}$$

Integrating the left-hand side of Equation (13), we obtain:

$$\int_{V(t_0)}^{V(t)} \frac{dV}{aV(t) + bV(t_0)^c} = \frac{1}{a}\ln\left(aV(t) + bV(t_0)^c\right) \tag{14}$$

Plugging the limits of the integral in Equation (14), we reach:

$$\int_{V(t_0)}^{V(t)} \frac{dV}{aV(t) + bV(t_0)^c} = \frac{1}{a}\ln\left(\frac{aV(t) + bV(t_0)^c}{aV(t_0) + bV(t_0)^c}\right) \tag{15}$$

Inserting Equation (15) in Equation (13) results in:

$$\frac{1}{a}\ln\left(\frac{aV(t) + bV(t_0)^c}{aV(t_0) + bV(t_0)^c}\right) \leq t_0 - t \tag{16}$$

Equation (16) gives a relation between time and the value of $V(t)$. Now, we want to calculate the convergence time (t_f), which means $V\left(t_f\right) = 0$. We plug this value into Equation (16), and obtain:

$$\frac{1}{a}\ln\left(\frac{bV(t_0)^c}{aV(t_0) + bV(t_0)^c}\right) \leq t_0 - t_f \tag{17}$$

Therefore,

$$\frac{1}{a}\ln\left(\frac{bV(t_0)^c}{aV(t_0) + bV(t_0)^c}\right) - t_0 \leq -t_f \tag{18}$$

and as a result,

$$t_f \leq t_0 - \frac{1}{a}\ln\left(\frac{bV(t_0)^c}{aV(t_0) + bV(t_0)^c}\right) = t_0 + \frac{1}{a}\ln\left(\frac{aV(t_0) + bV(t_0)^c}{bV(t_0)^c}\right) = t_0 + \frac{1}{a}\ln\left(\frac{aV(t_0)^{1-c} + b}{b}\right) \tag{19}$$

We know $0 \leq V(t_0) \leq \infty$; therefore, $1 \leq \frac{aV(t_0)^{1-c}+b}{b} \leq \infty$ and $\frac{1}{a}\ln\left(\frac{aV(t_0)^{1-c}+b}{b}\right)$ is positive and finite. As a result, we found a finite value for the upper bound of t_f, which concludes the proof of Lemma 1. □

Lemma 2. *When $0 < \alpha < 1$ and $a_\Delta > 0$ for $\Delta = 1, 2, \ldots, n$, then the triangle inequality holds as follows:*

$$\sum_{\Delta=1}^{n} a_\Delta^\alpha \geq \left(\sum_{\Delta=1}^{n} a_\Delta\right)^\alpha \tag{20}$$

Now, we introduce variable ψ as follows:

$$\psi = \xi - Q_2 \tag{21}$$

where we define variable ξ in the following manner:

$$\dot{\xi} = -k_d\psi - \delta\text{sign}(\psi) - \varepsilon\psi^{p_0/q_0} - |f(t, Q_1, Q_2)|\text{sign}(\psi) + I(Q_1)^{-1}\tau(t) \tag{22}$$

The variable μ is mathematically defined based on the positive design parameters k_d and ε. It is important to note that δ is chosen to be greater than $\|f_u\|_1$ to ensure the

validity of the expression. Furthermore, considering odd positive integers p_0 and q_0, where $p_0 < q_0$, the estimation of the $\hat{N}$ is derived according to the following expression:

$$f_s = -k_d\psi - \delta \text{sign}(\psi) - \varepsilon\psi^{p_0/q_0} - |f(t, Q_1, Q_2)|\text{sign}(\psi) - f(t, Q_1, Q_2) \tag{23}$$

By considering Equations (7), (21) and (22), the following equation is derived:

$$\dot{\psi} = \dot{\xi} - \dot{Q}_2 = -k_d\psi - \delta \text{sign}(\psi) - \varepsilon\psi^{\frac{p_0}{q_0}} - |f(t, Q_1, Q_2)|\text{sign}(\psi) - f(t, Q_1, Q_2) - f_u(t, Q_1, Q_2) \tag{24}$$

By utilizing Equations (9), (23) and (24), we can deduce the following result:

$$\begin{aligned} \widetilde{f}(t, Q_1, Q_2) = f_s \quad & (t, Q_1, Q_2) - f_u(t, Q_1, Q_2) \\ & = -k_d\psi - \delta \text{sign}(\psi) - \varepsilon\psi^{\frac{p_0}{q_0}} - |f(t, Q_1, Q_2)|\text{sign}(\psi) - f(t, Q_1, Q_2) - f_u(t, Q_1, Q_2) \\ & = -k_d\psi - \delta \text{sign}(\psi) - \varepsilon {s_d}^{\frac{p_0}{q_0}} - |f(t, Q_1, Q_2)|\text{sign}(\psi) - f(t, Q_1, Q_2) - \dot{Q}_2 + f(t, Q_1, Q_2) \\ & + I(Q_1)^{-1}\tau(t) = -k_d\psi - \delta \text{sign}(\psi) - \varepsilon\psi^{\frac{p_0}{q_0}} - |f(t, Q)|\text{sign}(\psi) + I(Q_1)^{-1}\tau(t) - \dot{x} = \dot{\xi} - \dot{Q}_2 \\ & = \dot{\psi} \end{aligned} \tag{25}$$

Theorem 1. *By employing the proposed sampling algorithm characterized by Equations (21)–(23) on the MIMO uncertain nonlinear system represented by Equation (9), the estimation error, denoted as* $\widetilde{f}$*, converges to zero within a finite duration.*

Proof. We consider the following positive definite Lyapunov function candidate:

$$V_0 = \frac{1}{2}\psi^T\psi \tag{26}$$

The expression for the time derivative of the function V_0 is as follows:

$$\begin{aligned} \dot{V}_0 = \psi^T\dot{\psi} = {s_d}^T \quad & \left(-k_d\psi - \delta \text{sign}(\psi) - \varepsilon\psi^{\frac{p_0}{q_0}} - \|f(t, Q_1, Q_2)\|_1\text{sign}(\psi) - f(t, Q_1, Q_2) - f_u(t, Q_1, Q_2)\right) \\ & \leq -k_d\psi^T\psi - \delta\psi^T\text{sign}(\psi) - \varepsilon {s_d}^T\psi^{\frac{p_0}{q_0}} - \|f(t, Q_1, Q_2)\|_1\psi^T\text{sign}(\psi) - \psi^T f(t, Q_1, Q_2) \\ & -\psi^T f_u(t, Q_1, Q_2) \\ & \leq -k_d\psi^T\psi - \beta\left\|\psi^T\right\|_1 - \varepsilon\psi^T\psi^{\frac{p_0}{q_0}} - \|f(t, Q_1, Q_2)\|_1\left\|\psi^T\right\|_1 - \psi^T f(t, Q_1, Q_2) \\ & + \left\|\psi^T\right\|_1\|f_u(t, Q_1, Q_2)\|_1 \leq -k_d\psi^T\psi - \varepsilon\psi^T\psi^{\frac{p_0}{q_0}} \leq -2kV_0 - 2^{(p_0+q_0)/2q_0}\varepsilon V_0^{(p_0+q_0)/2q_0} \end{aligned} \tag{27}$$

The final line of Equation (27) is obtained by applying Lemma 3.

Lemma 3. *Given the following function:*

$$\begin{aligned} V_0 = \tfrac{1}{2}{s_d}^T s_d = \tfrac{1}{2}\left(\psi_1^2 + \psi_2^2 + \ldots + \psi_n^2\right) \\ V_0^{(p_0+q_0)/2q_0} = \left(\tfrac{1}{2}\left(\psi_1^2 + \psi_2^2 + \ldots + \psi_n^2\right)\right)^{(p_0+q_0)/2q_0} \\ \leq \tfrac{1}{2^{(p_0+q_0)/2q_0}}\left(\psi_1^{(p_0+q_0)/2q_0} + \psi_2^{(p_0+q_0)/2q_0} + \ldots + \psi_n^{(p_0+q_0)/2q_0}\right) \end{aligned} \tag{28}$$

By considering the given function, we can derive the following equation:

$$2^{(p_0+q_0)/2q_0}V_0^{(p_0+q_0)/2q_0} \leq \psi_1^{(p_0+q_0)/q_0} + \psi_2^{(p_0+q_0)/q_0} + \ldots + \psi_n^{(p_0+q_0)/q_0} \tag{29}$$

As we know $\psi_1^{(p_0+q_0)/q_0} + \psi_2^{(p_0+q_0)/q_0} + \ldots + \psi_n^{(p_0+q_0)/q_0} = \psi^T\psi^{p_0/q_0}$, as a results

$$\begin{aligned} & 2^{(p_0+q_0)/2q_0}V_0^{(p_0+q_0)/2q_0} \leq \psi^T\psi^{\frac{p0}{q_0}} \\ \overset{yields}{\rightarrow} \; & -\varepsilon\psi^T\psi^{p0/q_0} \leq -\varepsilon 2^{(p_0+q_0)/2q_0}V_0^{(p_0+q_0)/2q_0} \end{aligned} \tag{30}$$

Therefore, based on Lemmas 1–3, along with the satisfaction of the Lyapunov condition as expressed in Equation (30), we establish the finite time convergence of the sampling approximation $\tilde{f}$ to zero. □

4. Dynamic Approximation

Accurate estimation of uncertainties is crucial for robust and reliable control of robotic systems. In the past, conventional methods such as finite time disturbance observers have been commonly employed for uncertainty estimation in robotics. Conventional methods and finite time disturbance observers often rely on pre-defined mathematical models or heuristics for uncertainty estimation. These approaches may struggle to accurately capture nonlinear uncertainties that arise in complex robotic systems. Neural networks, on the other hand, excel at nonlinear uncertainty modeling by approximating complex functions. They have the capacity to capture intricate uncertainty patterns, enabling more accurate estimation in real-world robotic applications.

Noise is an inescapable component of real-world data, often adding complexity to various applications. In response, neural networks, especially RNNs, have become instrumental in distinguishing signal from noise due to their adeptness at processing sequential data patterns. Recently, advanced techniques have been developed to handle noise more effectively with neural networks [28,29]. For a comprehensive understanding and more detailed insights into these innovative methods, one can refer to [30].

For rehabilitation robots, the presence of inputs and outputs is crucial for obtaining the system's states when the states are designed to be observable. However, neural networks alone are limited to modeling the relationship between given inputs and outputs and cannot capture the unknown dynamic which is a part of the system. Moreover, their performance is optimal only under ideal conditions. This obstacle remains an open question in the field of robotics and identification. To overcome this challenge, we propose an RNN integrated with a finite time sampler, offering a novel approach to addressing these limitations.

4.1. The Proposed Algorithm

The structure of our proposed algorithm is illustrated in Figure 2. As depicted in the figure, during the training process, electromyographic signals obtained from the nervous system are applied as inputs to the rehabilitation robot. These signals are time-dependent and serve to update the state of the rehabilitation robot. The states, along with their derivatives and time information, are then fed into both the RNN and the finite time sampler. The RNN's loss function is defined based on the acquired samples and the estimated values generated by the RNN. Via backpropagation, the RNN is trained, and its weights and biases in all layers, including the self-attention layers, are updated. Once the training process is completed, the sampler is no longer required. Using the states of the systems and time information, the RNN is capable of estimating the unknown dynamics of the systems.

This algorithm incorporates physics principles with machine learning techniques to yield more precise system predictions. It can be summarized as follows:

1. An electromyographic signal is introduced into the system, and it acts as a torque according to Equation 1.
2. This electromyographic influence is then applied to the system dynamics, as visualized in Figure 2.
3. The dynamic state of the system, along with the time, are used as inputs for the neural network. Simultaneously, this system state is also input to a finite time estimator.
4. The finite time estimator operates according to Equations 12, 13, and 14 to estimate an unknown dynamic factor, referred to as f_s.
5. This estimated f_s is then compared with the output of the neural network. The discrepancy between the estimated f_s and the neural network's output forms a loss function.
6. This loss function is instrumental in updating the weights and biases of the neural network during the training phase.

7. Once training concludes, the now refined neural network serves as a reliable estimator for the system.

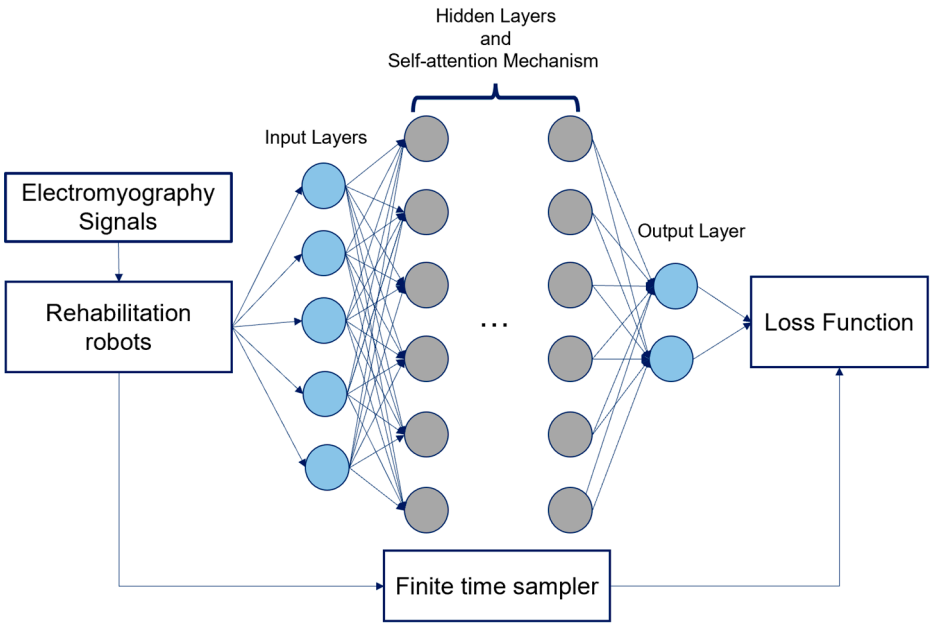

Figure 2. The structure of the proposed technique for dynamic identification in Rehabilitation robot.

Additionally, to train the neural network, we complete the following 5 steps:

I. Forward Propagation:

For a given input vector, the network generates an output prediction utilizing its current parameter configuration (weights and biases). This prediction is the network's approximation of the expected f_s.

II. Loss Quantification:

The computed output is contrasted against the estimated f_s. We compute the loss function by calculating the root-mean-square error (RMSE) between f_s and its prediction.

III. Gradient Computation (Backpropagation):

Leveraging the chain rule of differentiation, gradients of the loss function with respect to each weight and bias in the neural network are calculated. These gradients denote the sensitivity of the loss to changes in each parameter.

IV. Update Weights and Biases:

Using the gradients computed in the backpropagation step, the weights and biases of the neural network are updated using an optimization algorithm.

V. Iterative Refinement:

The aforementioned steps are reiterated across multiple epochs, ensuring progressive optimization of the model's parameters. In our neural network, the stop condition is indicated by a minimal fluctuation in the loss function. For our specific model, termination criteria are defined by the stability of the loss function: we halt training when the change in the normalized loss across five consecutive iterations is less than 10^{-3}.

Our approach harnesses the power of self-attention algorithms, which offer distinct advantages in estimation tasks. Self-attention layers enable the model to simultaneously consider all input positions, capturing global dependencies efficiently. This capability proves especially beneficial in estimating quantities that rely on non-local relationships within the data. By attending to relevant information from the entire input, self-attention layers provide a comprehensive understanding that facilitates accurate estimations. These

advantages can lead to a more accurate estimation of unknown dynamics in the knee rehabilitation robot. However, this approach also comes with limitations: it requires considerable memory, especially for long sequences; it lacks an inherent bias towards neighboring elements in a sequence, which may lead to missing out on localized patterns; the attention maps produced are not always easy to interpret; and the model may be distracted by noisy or irrelevant data, necessitating careful preprocessing. Balancing these benefits and limitations is crucial in developing an effective and efficient model. In what follows, we elaborate on the advantages of our proposed algorithm.

4.2. Accurate Sampling

In situations where explicit information or samples are unavailable, our proposed algorithm overcomes this limitation by incorporating a robust mechanism for obtaining accurate samples from unknown dynamics. This mechanism ensures the reliability and trustworthiness of the acquired samples for the neural network. By generating reliable samples for the RNN, our method demonstrates its practical applicability in real-world scenarios. This significant contribution directly addresses a crucial problem in the field, providing a solution that enables accurate sampling even in the absence of explicit information or samples.

4.3. Adaptability and Generalization

Conventional methods and finite time disturbance observers typically require the explicit specification of model parameters or tuning of observer gains. This limits their adaptability to varying operating conditions and environments. In contrast, neural networks can adaptively learn from data, making them more flexible and capable of generalizing to different robotic system configurations and dynamic environments. Via training on diverse datasets, our algorithm is capable to capture the underlying dynamics of uncertainties and adapt their estimation accordingly, resulting in improved performance across various scenarios.

4.4. Robustness to Complex Dynamics

Robotic systems often operate in complex and dynamic environments, where uncertainties can vary significantly. Conventional methods and finite time disturbance observers may struggle to handle the complexity of these dynamics and may exhibit limited robustness. Neural networks possess inherent robustness to complex dynamics due to their ability to learn from diverse and large-scale data. Hence, our algorithm can effectively model uncertainties arising from intricate and time-varying dynamics, providing more reliable and robust estimation capabilities in challenging operating conditions.

4.5. Integration of Sensor Data

Uncertainty estimation in robotics often requires the fusion of data from multiple sensors. For instance, here we use two electromyographic signals which are displayed in Section 5. Conventional methods and finite time disturbance observers may face challenges in integrating heterogeneous sensor data, such as noisy or incomplete measurements. Neural networks excel in data fusion tasks by leveraging their ability to process and extract meaningful features from multiple sensor inputs. By incorporating sensor fusion capabilities, neural networks enhance uncertainty estimation by effectively utilizing diverse sensor information, leading to more accurate and reliable estimates compared to conventional methods.

4.6. Equipped with Attention Mechanism

The integration of attention mechanisms has showcased their considerable utility in the domain of time series classification. These mechanisms empower models to dynamically emphasize significant segments of sequences, effectively capturing essential temporal patterns and relationships. Within our tailored RNN architecture for time se-

ries classification, we have introduced a self-attention layer. This augmentation enhances the model's capabilities by enabling it to selectively weigh the relevance of distinct time steps, a particularly advantageous feature for tasks where certain time steps carry more weight in determining class labels. In what follows, we delve into the formal mathematical framework of this self-attention layer within our RNN:

At each time step, i input vector x_i undergoes projection to derive query q_i, key k_i, and value v_i vectors:

$$\begin{aligned} q_i &= W_q\ x_i \\ k_i &= W_k\ x_i \\ v_i &= W_v\ x_i \end{aligned} \tag{31}$$

in which W_q, W_k, and W_v are weights for query, key, and value vectors, respectively. Then, we calculate dot products between the current time step's query vector q_t, and all key vectors generate attention scores:

$$e_{ti}\ = q_t \cdot\ k_i \tag{32}$$

The computed attention scores undergo normalization using the softmax function to yield weights:

$$a_{ti} = softmax(e_{ti}) \tag{33}$$

where a_{ti} denotes the normalized attention score. The context vector c_t emerges as the weighted summation of value vectors as follows:

$$c_t = \sum_{i=1}^{t} a_{ti}\ \ v_i \tag{34}$$

Subsequently, the context vector c_t merges with the RNN's previous hidden state h_{t-1} to generate the new hidden state h_t for the current time step:

$$h_t = RNN(x_t,\ c_t,\ h_{t-1}) \tag{35}$$

Via this incorporation of a self-attention layer, our RNN enhances its temporal dependency-capturing capabilities, ultimately refining classification accuracy by accommodating the varying relevance of different time steps.

In summary, when compared to conventional methods and disturbance observers, our method provides distinct advantages for uncertainty estimation in robotic systems due to its capacity for nonlinear uncertainty modeling, adaptability, and generalization. Moreover, the integration of self-attention layers in neural networks for estimation problems provides several benefits. From modeling complex relationships and handling variable importance to robustness in handling variable length and missing data, self-attention layers enhance the performance and robustness of estimation models.

5. Numerical Results

In this section, we assess the effectiveness of the proposed algorithm by conducting evaluations using two distinct experiments. The system parameters are given by $m_1\ =\ 2$ kg, $m_2 = 0.85$ kg, $l_1 = 0.2$ kgm^2, and $l_1 = 0.25$ kgm^2. The design parameters of the sampler are as follows: $p_0 = 1$, $q_0 = 7\ k_d = 300$, $\delta = 0.1$, and $\varepsilon = 0.55$. For the training process of the RNN, the Adam optimizer is utilized. The activation function employed is the hyperbolic tangent (tanh). The RNN consists of several layers, including a fully connected layer, two long short-term memory (LSTM) layers, and one self-attention layer. In the self-attention layer, the number of heads is set to 8, and the number of key channels is set to 64.

We have allocated 50 percent of the samples for training purposes and reserved the remaining 50 percent for testing to ensure a balanced distribution of data. The architecture of the neural network provided begins with a sequence input layer, which has been designed to receive sequences of data. In this case, the input sequences are vectors of length 5. This layer is followed by an LSTM layer with 128 hidden units. The LSTM layer is capable of

learning patterns over time and is highly suited to time-series data. Subsequently, the data passes through a fully connected layer with 64 units, which is used to transform the dimensionality of the data. A tanh layer follows, where the hyperbolic tangent function is applied to the input, ensuring that the transformed data falls within the range between -1 and 1. After that, a self-attention layer is applied. Following this, another fully connected layer is employed, transforming the data into a 1-dimensional space. Here, the predictions are made based on the final hidden state. The MSE is then calculated as the average of the squared differences between the actual and predicted values. This computed MSE serves as a quantitative measure of the model's performance, with lower values indicating a better fit. During training, the aim is to minimize this MSE value via repeated iterations of backpropagation, adjusting the model's weights to improve prediction accuracy. For the training of this network, the Adam optimizer is employed, sequence padding to the left is incorporated, and data is shuffled before each epoch.

5.1. Experiment 1

In this specific instance, our goal is to evaluate the effectiveness of the suggested algorithm in accurately predicting the undisclosed behavior of a system. To clarify further, the undisclosed behavior can be described as follows:

$$f_u = J(\phi)h(t) = \begin{bmatrix} -l_1 \sin\phi_1 + l_2 \sin(\phi_1+\phi_2) & -l_2 \sin(\phi_1+\phi_2) \\ -l_1 \cos\phi_1 + l_2 \cos(\phi_1+\phi_2) & l_2 \cos(\phi_1+\phi_2) \end{bmatrix} \begin{bmatrix} 0.1t + \cos(t) \\ \frac{1}{t+1} + \sin(t) \end{bmatrix} \quad (36)$$

where $h(t)$ denotes the force exerted by the user. The applied electromyographic signals to the system are displayed in Figure 3. According to these signals, the system is stimulated, samples from unknown dynamics are automatically captured, and the RNN is trained. It is noteworthy that in real-world scenarios, the electromyographic signal is not merely added to torque. Instead, it has to be transformed or converted first. The true value, which is the desired torque to be applied to the system, is obtained after this transformation. This is a calibration process for each electromyographic signal, converting it into the appropriate torque. In the context of this study, the simplifying assumption is made that this transformation is linear. However, it is important to note that the method is not constrained by this assumption; the transformation could be non-linear or follow a different model.

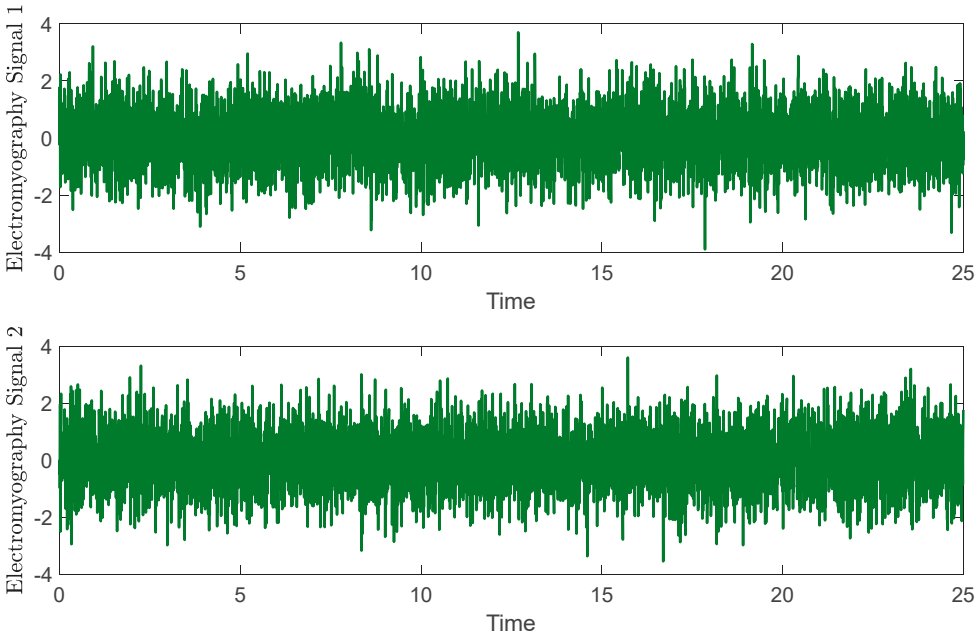

Figure 3. The applied electromyography signals to the system.

The RMSE and loss function during the training process of RNN are illustrated in Figures 4 and 5, respectively. The time history of unknown dynamic and predicted values in the testing dataset using the proposed algorithm is demonstrated in Figure 6. As it is illustrated in this figure, the proposed finite time estimator successfully obtains precise samples, and the RNN effectively learns the intricate unknown dynamics. The performance of the RNN in predicting the test set is particularly noteworthy, showcasing remarkable results.

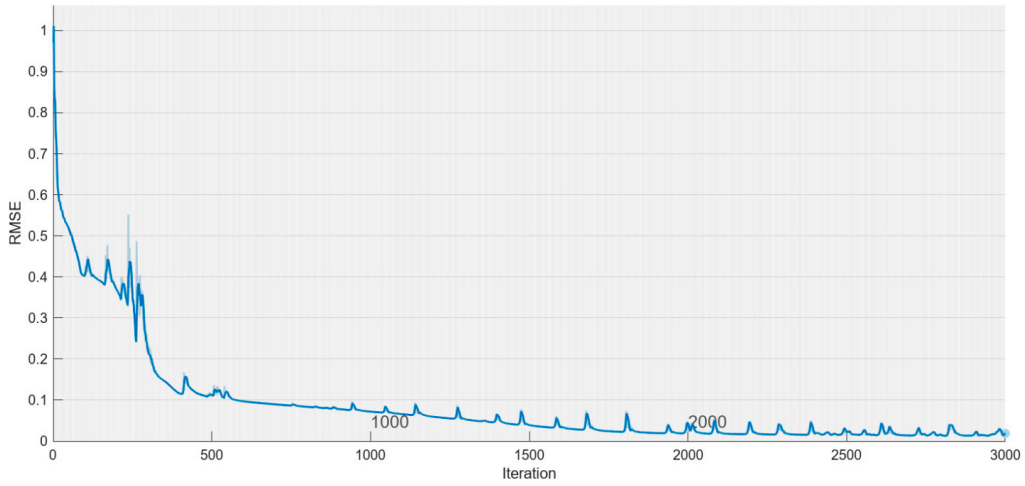

Figure 4. The RMSE of the proposed algorithm for identifying unknown dynamics in experiment 1.

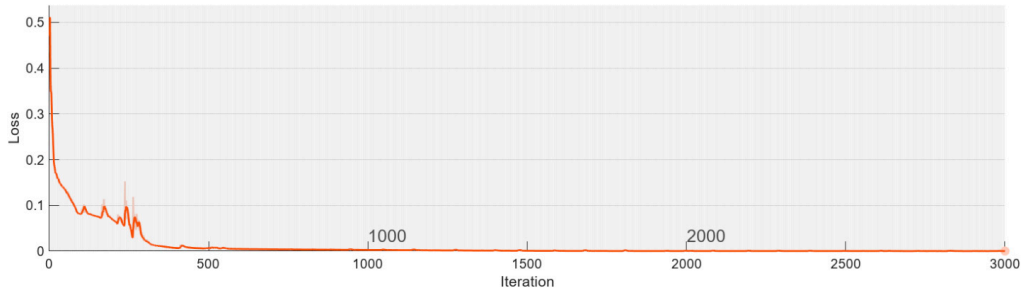

Figure 5. The loss function of the proposed algorithm for identifying unknown dynamics in experiment 1.

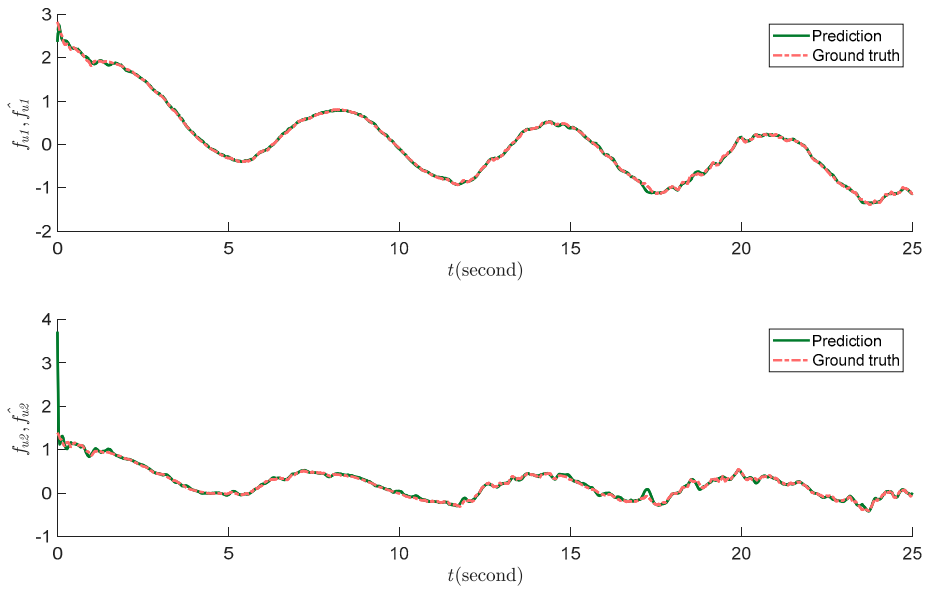

Figure 6. The time history of the unknown dynamic and its predicted value in experiment 1.

It is worth noting that taking into account the input torque range and the uncertainty stipulated in equation 27, our analyses reveal that the uncertainty bounds for f_{u_1} and f_{u_2} are calculated as 2.8 and 1.3 respectively. However, it is crucial to emphasize that this information has not been furnished to the sampler. Nonetheless, despite this absence, the sampler continues to perform adeptly within the current circumstances. This notable aspect of our approach's exceptional performance can be attributed to the integration of self-attention layers in neural networks. In estimation tasks, it is evident that not all input features contribute uniformly to the target variable.

Some features may have a more significant impact, while others may be less relevant. Self-attention layers empower the model to assign varying attention weights to different input elements, effectively determining their importance in the estimation process. This adaptability enables the model to focus on the most influential features while attenuating the influence of less relevant ones, thereby improving overall estimation performance.

5.2. Experiment 2

Conventional approaches face challenges when it comes to identifying intricate dynamics encompassing complex time-varying functions. However, our innovative method possesses a resilient and intelligent framework that excels in handling such dynamics. As a result, it becomes feasible to make accurate predictions for unknown dynamic functions that remain elusive to traditional methods. To further highlight the efficacy of our proposed algorithm, we present an illustrative example featuring a sophisticated unknown dynamic function as follows:

$$f_u = J(\phi)h(t) = \begin{bmatrix} -l_1 \sin\phi_1 + l_2 \sin(\phi_1 + \phi_2) & -l_2 \sin(\phi_1 + \phi_2) \\ -l_1 \cos\phi_1 + l_2 \cos(\phi_1 + \phi_2) & l_2 \cos(\phi_1 + \phi_2) \end{bmatrix} \begin{bmatrix} \tan^{-1}(\mathrm{t}) + \cos(t) \\ \sin(2t) + \sin(0.5t) \end{bmatrix} \tag{37}$$

The presence of the arctangent function ($\tan^{-1}(\mathrm{t})$) and the difference in the frequencies of the time-varying functions $\sin(2t)$ and $\sin(0.5t)$ introduces additional complexity to the signals of the system, making their estimation challenging. In this scenario, considering the input torque range and the uncertainty provided in Equation (28), our analyses indicate that the uncertainty bounds for f_{u1} and f_{u1} are 1.4 and 0.8 respectively.

Here, we utilize the electromyographic signals depicted in Figure 3 to perform our analysis. In Figures 7 and 8, we present the RMSE and loss function values observed during the training process. The identification results in the testing dataset are illustrated in Figure 9, which clearly demonstrates that despite the complexity of the unknown dynamics, our proposed intelligent algorithm successfully detects and captures them. This remarkable achievement emphasizes the outstanding performance of our algorithm in dynamic identification.

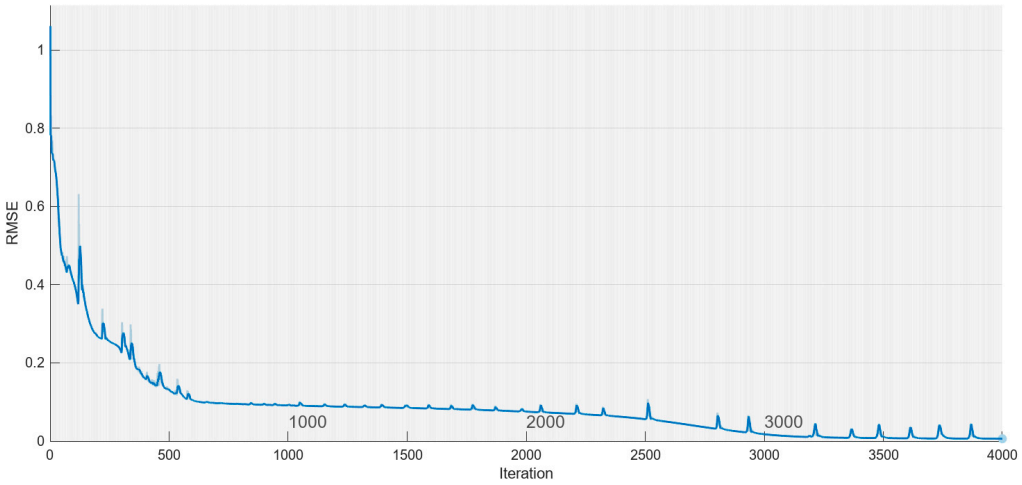

Figure 7. The RMSE of the proposed algorithm for identifying unknown dynamics in experiment 2.

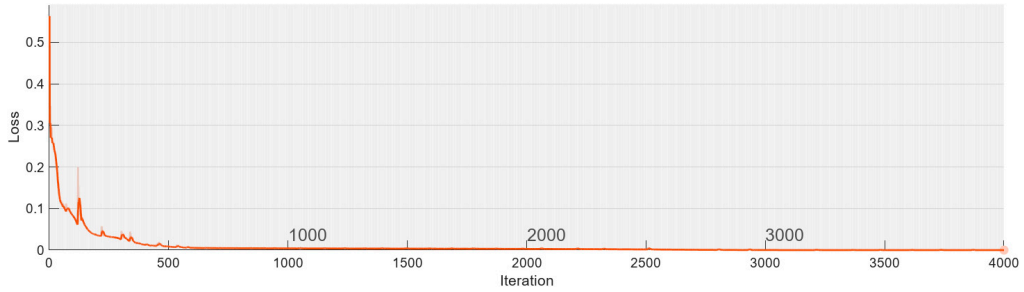

Figure 8. The loss function of the proposed algorithm for identifying unknown dynamics in experiment 2.

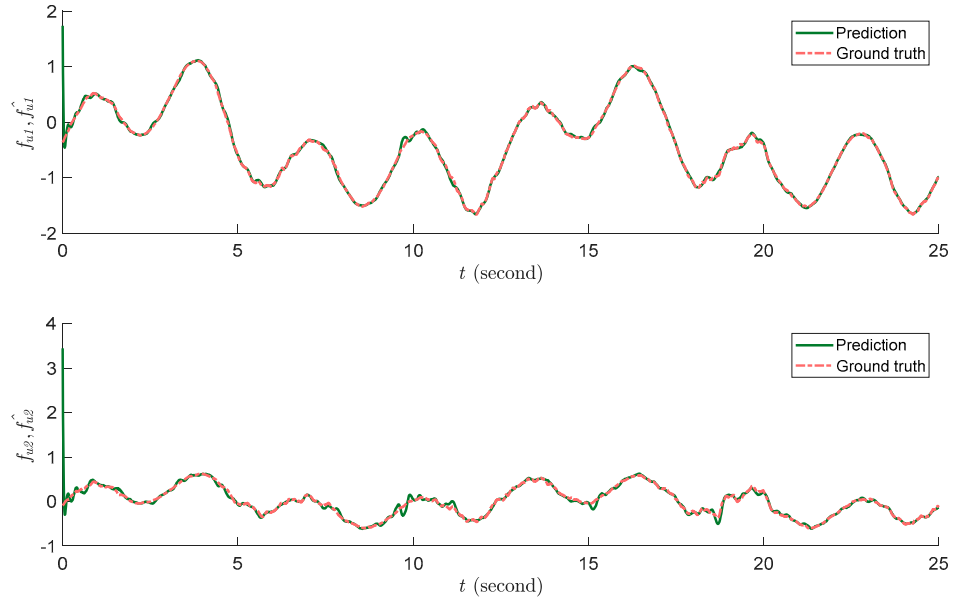

Figure 9. The time history of the unknown dynamic and its predicted value in experiment 2.

5.3. Comparison with Other Techniques

Utilizing neural networks for estimating the unknown dynamics of rehabilitation robots presents substantial advantages over traditional identifiers. Firstly, due to their universal approximation capabilities, neural networks can effectively estimate complex and even discontinuous dynamics, requiring no assumptions about the nature of unknown disturbances. This is in stark contrast to conventional methods, which often make assumptions about the smoothness of these unknowns. Furthermore, unlike traditional estimators that are designed around specific controllers, thus limiting their applicability, neural networks offer a broader approach. They provide a general approximation that is not dependent on any particular controller, offering a significant degree of flexibility. Once a neural network is trained, it can adapt to work with any controller. In addition to this, neural networks have the unique ability to learn intricate, non-linear mappings from inputs to outputs, potentially modeling the robot's dynamics more accurately, especially in scenarios involving non-linear or complex phenomena. Moreover, the unpredictable and dynamic nature of human–robot interactions, particularly in rehabilitation settings, necessitates a system that can adapt in real-time. Neural networks, thanks to their adaptability, can learn from incoming data and adjust their operations accordingly, unlike conventional methods that often rely on a static model. Therefore, the application of neural networks in the control of rehabilitation robots could potentially lead to improved accuracy, robustness, and overall better outcomes in complex, unpredictable environments.

Neural networks offer a compelling suite of advantages, but it is vital to recognize the inherent trade-offs. The process of training these networks necessitates considerable

amounts of data and extensive computational resources. Further, their intricate nature often makes interpretation and validation more challenging than with conventional models. Therefore, while their potential benefits are substantial, the application of neural networks requires a thoughtful evaluation to confirm their appropriateness for a specific task, with an unwavering focus on safety and effectiveness. Interestingly, our proposed approach incorporates the use of a sampler, the accuracy of which carries theoretical guarantees. This aspect allows our method to enjoy benefits typically associated with conventional approaches, presenting a harmonious blend of traditional reliability and innovative potential.

In order to thoroughly assess the effectiveness of our developed method, we carried out a comparative analysis with the integral terminal sliding mode observer (ITSMC), as proposed in [31]. In this comparative analysis, we introduce Gaussian noise with a mean of zero and a standard deviation of 0.01 into the system. We then evaluate the proficiency of our proposed technique in mitigating this interference. For situations with more pronounced noise, we suggest integrating a filter post-sampling, ensuring that the sampler's output is filtered before being fed into the neural network (NN). Our findings are benchmarked against a conventional method. To guarantee an equitable comparison, we fine-tune the parameters of our sampler to correspond with those in the reference while still ensuring compatibility with our system's requirements.

Figure 10 provides a side-by-side comparison of our results with those of the ITSMC observer. Upon closer inspection, as can be discerned from the zoomed-in plot, our method exhibits a more accurate alignment with the ground truth function. In contrast, the results of the ITSMC display considerable scattering. One of the distinct advantages of our method lies in the incorporation of the proposed estimator combined with the regression capabilities of neural networks. This synergy allows our method to deliver smoother and more reliable results. Such characteristics are not just statistically appealing but are of paramount importance for real-world applications, where consistency and reliability are often essential prerequisites.

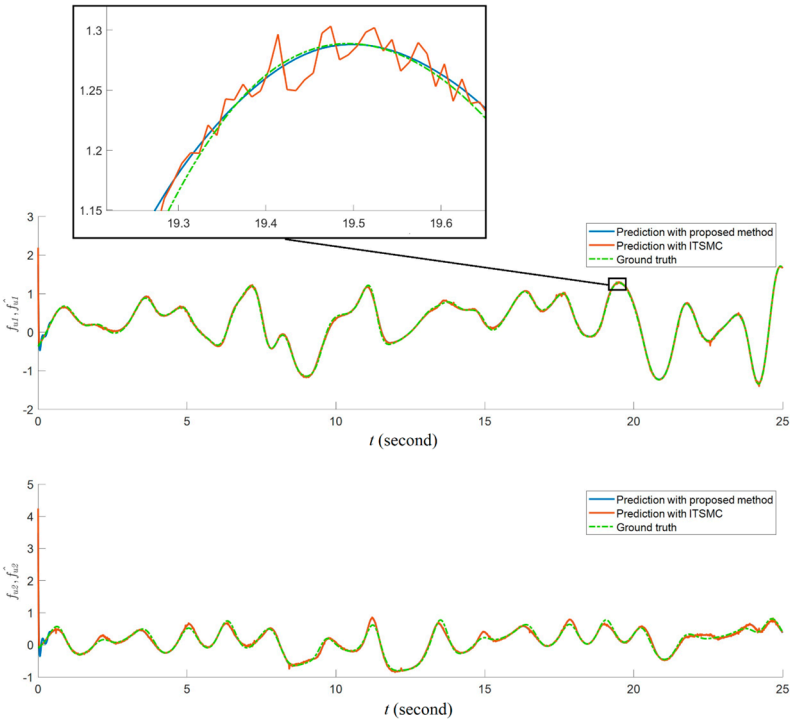

Figure 10. The time history of the unknown dynamic and its predicted value in experiment 2 based on our proposed method and ITSMC.

To further evaluate the performance of our method, we have presented the data in Table 1. This table delineates the numerical results of the average error for both our method

and the ISMC. These results, presented in Table 1, clearly underscore the effectiveness of our approach. Additionally, the proposed method provides smooth results.

Table 1. Comparison the results of the proposed method with those of ITSMC.

Error	The Proposed Approach	ITSMC
Average of $\left\|f_{u1} - \hat{f}_{u1}\right\|$	0.0235	0.0241
Average of $\left\|f_{u2} - \hat{f}_{u2}\right\|$	0.0362	0.0483

For an unbiased comparison, it is noteworthy to mention the advantages of the ITSMC over our method. Specifically, the ISMC does not require any preprocessing or the learning intricacies associated with training a network. In contrast, our method necessitates a training process before it can be deployed in real-time applications. However, the proposed algorithm offers the possibility of end-to-end learning, where uncertainty estimation can be integrated seamlessly with other robotic control tasks. This integration is not readily achievable with conventional methods and disturbance observers, which typically require separate design and tuning. By jointly optimizing uncertainty estimation and control objectives, our technique enables a more holistic and integrated approach to uncertainty-aware robotic control. This end-to-end learning paradigm reduces the need for manual tuning and handcrafted designs, simplifying the development and deployment process while enhancing overall system performance, a capability lacking in conventional methods.

6. Conclusions

The current study introduced a new algorithm that combined a finite sampler and RNN to effectively identify and predict unfamiliar dynamics in lower-limb rehabilitation robots. To accomplish this, the study initially presented a dynamic model for a 2-DOF knee rehabilitation robot. Next, a nonlinear estimator was developed to capture samples from the dynamical system. By utilizing the output of the RNN and proposed sampler, the unknown dynamic of the rehabilitation robot is identified. The proposed RNN was enhanced with a self-attention mechanism, which played a vital role in devising effective strategies for practical applications, particularly when the input features held varying levels of importance. The proposed method's effectiveness in identifying two distinct unknown dynamics was clearly demonstrated. Numerical analyses emphasized the algorithm's proficiency in managing both dynamics adeptly, suggesting its broad applicability across diverse rehabilitation contexts. In addition, to more comprehensively assess the efficacy of our method, we presented numerical results juxtaposed with those from a state-of-the-art technique, ITSMC. Our findings highlighted that the proposed algorithm not only yielded smoother outcomes but also consistently achieved a reduced error rate in estimations. To continue advancing this field, we plan to evaluate and enhance the performance of the proposed method by incorporating fuzzy structures. This would allow for further improvement in the accuracy and efficiency of the algorithm, enabling the development of advanced control strategies for rehabilitation robots.

Author Contributions: Conceptualization, A.A. and H.A.; methodology, A.A. and H.A.; software, A.A. and H.A.; validation, A.A. and H.A.; formal analysis, A.A. and H.A.; data curation, A.A. and H.A.; writing—original draft preparation, A.A. and H.A.; writing—review and editing, A.A. and H.A.; visualization, A.A. and H.A.; supervision, A.A. and H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the King Salman Center for Disability Research through Research Group no KSRG-2023-487.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the King Salman center For Disability Research for funding this work through Research Group no KSRG-2023-487.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

DOF	degree-of-freedom
LSTM	long short-term memory
ODE	ordinary differential equation
tanh	hyperbolic tangent
RMSE	root-mean-square error
RNN	recurrent neural network

References

1. Qian, Z.; Bi, Z. Recent Development of Rehabilitation Robots. *Adv. Mech. Eng.* **2015**, *7*, 563062. [CrossRef]
2. Qassim, H.M.; Wan Hasan, W. A Review on Upper Limb Rehabilitation Robots. *Appl. Sci.* **2020**, *10*, 6976. [CrossRef]
3. Xu, J.; Li, Y.; Xu, L.; Peng, C.; Chen, S.; Liu, J.; Xu, C.; Cheng, G.; Xu, H.; Liu, Y. A Multi-Mode Rehabilitation Robot with Magnetorheological Actuators Based on Human Motion Intention Estimation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 2216–2228. [CrossRef]
4. Young, A.J.; Ferris, D.P. State of the Art and Future Directions for Lower Limb Robotic Exoskeletons. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2016**, *25*, 171–182. [CrossRef]
5. Heng, W.; Solomon, S.; Gao, W. Flexible Electronics and Devices as Human–Machine Interfaces for Medical Robotics. *Adv. Mater.* **2022**, *34*, 2107902. [CrossRef]
6. Chen, G.; Chan, C.K.; Guo, Z.; Yu, H. A Review of Lower Extremity Assistive Robotic Exoskeletons in Rehabilitation Therapy. *Crit. Rev. Biomed. Eng.* **2013**, *41*, 343–363. [CrossRef]
7. Dipietro, L.; Ferraro, M.; Palazzolo, J.J.; Krebs, H.I.; Volpe, B.T.; Hogan, N. Customized Interactive Robotic Treatment for Stroke: EMG-Triggered Therapy. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2005**, *13*, 325–334. [CrossRef]
8. Yang, J.; Wang, R.; Ren, Y.; Mao, J.; Wang, Z.; Zhou, Y.; Han, S. Neuromorphic Engineering: From Biological to Spike-based Hardware Nervous Systems. *Adv. Mater.* **2020**, *32*, 2003610. [CrossRef]
9. Lane, S.J.; Mailloux, Z.; Schoen, S.; Bundy, A.; May-Benson, T.A.; Parham, L.D.; Smith Roley, S.; Schaaf, R.C. Neural Foundations of Ayres Sensory Integration®. *Brain Sci.* **2019**, *9*, 153. [CrossRef] [PubMed]
10. Brahmia, A.; Kelaiaia, R. Design of a Human Knee Reeducation Mechanism. *Acta Univ. Sapientiae Electr. Mech. Eng.* **2019**, *11*, 42–53. [CrossRef]
11. Kelaiaia, R.; Chemori, A.; Brahmia, A.; Kerboua, A.; Zaatri, A.; Company, O. Optimal Dimensional Design of Parallel Manipulators with an Illustrative Case Study: A Review. *Mech. Mach. Theory* **2023**, *188*, 105390. [CrossRef]
12. Wu, Q.; Wang, X.; Chen, B.; Wu, H. Development of an RBFN-Based Neural-Fuzzy Adaptive Control Strategy for an Upper Limb Rehabilitation Exoskeleton. *Mechatronics* **2018**, *53*, 85–94. [CrossRef]
13. Rajasekaran, V.; Aranda, J.; Casals, A.; Pons, J.L. An Adaptive Control Strategy for Postural Stability Using a Wearable Robot. *Robot. Auton. Syst.* **2015**, *73*, 16–23. [CrossRef]
14. Elbagoury, B.M.; Vladareanu, L. A Hybrid Real-Time EMG Intelligent Rehabilitation Robot Motions Control Based on Kalman Filter, Support Vector Machines and Particle Swarm Optimization. In Proceedings of the 2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA), Chengdu, China, 15–17 December 2016; pp. 439–444.
15. Nogueira, S.L.; Inoue, R.S.; Terra, M.H.; Siqueira, A.A. Estimation of Lower Limbs Angular Positions Using Kalman Filter and Genetic Algorithm. In Proceedings of the 2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC), Rio de Janeiro, Brazil, 18–20 February 2013; pp. 1–6.
16. Véras, L.G.D.; Medeiros, F.L.; Guimarães, L.N. Systematic Literature Review of Sampling Process in Rapidly-Exploring Random Trees. *IEEE Access* **2019**, *7*, 50933–50953. [CrossRef]
17. Onwuegbuzie, A.J.; Leech, N.L. Sampling Designs in Qualitative Research: Making the Sampling Process More Public. *Qual. Rep.* **2007**, *12*, 238–254. [CrossRef]
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–15.
19. Tang, G.; Müller, M.; Rios, A.; Sennrich, R. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures. *arXiv* **2018**, arXiv:1808.08946.
20. Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Pan, S.; Zhang, C. Disan: Directional Self-Attention Network for Rnn/Cnn-Free Language Understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
21. Fahim, S.R.; Sarker, Y.; Sarker, S.K.; Sheikh, M.R.I.; Das, S.K. Self Attention Convolutional Neural Network with Time Series Imaging Based Feature Extraction for Transmission Line Fault Detection and Classification. *Electr. Power Syst. Res.* **2020**, *187*, 106437. [CrossRef]
22. Wu, Y.; Ma, Y.; Liu, J.; Du, J.; Xing, L. Self-Attention Convolutional Neural Network for Improved MR Image Reconstruction. *Inf. Sci.* **2019**, *490*, 317–328. [CrossRef]

23. Wei, Q.; Yan, Y.; Zhang, J.; Xiao, J.; Wang, C. A Self-Attention-Based Deep Reinforcement Learning Approach for AGV Dispatching Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *1*, 1–12. [CrossRef]
24. Manchin, A.; Abbasnejad, E.; Van Den Hengel, A. *Reinforcement Learning with Attention That Works: A Self-Supervised Approach*; Springer: Berlin, Germany, 2019; pp. 223–230.
25. Li, Y.; Sam Ge, S.; Yang, C. Learning Impedance Control for Physical Robot–Environment Interaction. *Int. J. Control* **2012**, *85*, 182–193. [CrossRef]
26. Lee, T.H.; Harris, C.J. *Adaptive Neural Network Control of Robotic Manipulators*; World Scientific: Singapore, 1998; Volume 19, ISBN 981-02-3452-X.
27. Yu, X.; Zhihong, M. Fast Terminal Sliding-Mode Control Design for Nonlinear Dynamical Systems. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2002**, *49*, 261–264.
28. Chen, X.; Tao, Y.; Xu, W.; Yau, S.S.-T. Recurrent Neural Networks Are Universal Approximators with Stochastic Inputs. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *1*, 1–15. [CrossRef] [PubMed]
29. Qiu, J.; Ma, M.; Wang, T.; Gao, H. Gradient Descent-Based Adaptive Learning Control for Autonomous Underwater Vehicles with Unknown Uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5266–5273. [CrossRef]
30. Tao, Y.; Kang, J.; Yau, S.S.-T. Neural Projection Filter: Learning Unknown Dynamics Driven by Noisy Observations. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *1*, 1–15. [CrossRef]
31. Eskandari, B.; Yousefpour, A.; Ayati, M.; Kyyra, J.; Pouresmaeil, E. Finite-Time Disturbance-Observer-Based Integral Terminal Sliding Mode Controller for Three-Phase Synchronous Rectifier. *IEEE Access* **2020**, *8*, 152116–152130. [CrossRef]