Event-Sampled Output Feedback Control of Robot Manipulators Using Neural Networks

Vignesh Narayanan, *Member, IEEE*, Sarangapani Jagannathan, *Fellow, IEEE*, and Kannan Ramkumar, *Member, IEEE*

***Abstract*— In this paper, adaptive neural networks (NNs) are employed in the event-triggered feedback control framework to enable a robot manipulator to track a predefined trajectory. In the proposed output feedback control scheme, the joint velocities of the robot manipulator are reconstructed using a nonlinear NN observer by using the joint position measurements. Two different configurations are proposed for the implementation of the controller depending on whether the observer is co-located with the sensor or the controller in the feedback control loop. Besides the observer NN, a second NN is utilized to compensate the effects of nonlinearities in the robot dynamics via the feedback control. For both the configurations, by utilizing observer NN and the second NN, torque input is computed by the controller. The Lyapunov stability method is employed to determine the event-triggering condition, weight update rules for the controller, and the observer for both the configurations. The tracking performance of the robot manipulator with the two configurations is analyzed, wherein it is demonstrated that all the signals in the closed-loop system composed of the robotic system, the observer, the event-sampling mechanism, and the controller are locally uniformly ultimately bounded in the presence of bounded disturbance torque. To demonstrate the efficacy of the proposed design, simulation results are presented.**

***Index Terms*— Event-triggering, neural-network (NN) control, robotic manipulator.**

I. Introduction

THE design of torque input for robot manipulators has been one of the widely studied problems among the control researchers [1]–[3]. For example, in the areas ranging from nano/micro scale applications in biological systems [4] to large-scale applications in engineering systems [5], and for medical environments [6], [7], the robotic manipulators are extensively used, and therefore, various control schemes are proposed for these applications in the past couple of decades.

Manuscript received January 12, 2018; revised July 8, 2018; accepted September 2, 2018. Date of publication October 12, 2018; date of current version May 23, 2019. This work was supported by NSF under Grant ECCS 1406533 and Grant CMMI 1547042. *(Corresponding author: Vignesh Narayanan.)*

V. Narayanan is with the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: vnxv4@mst.edu).

S. Jagannathan is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409-0040 USA.

K. Ramkumar is with the Department of Electronics and Instrumentation Engineering, Shanmugha Arts, Science, Technology & Research Academy, Thanjavur 613401, India.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2870661

The computed torque (CT) controllers convert the nonlinear robot manipulator dynamics into a linear system via the feedback-linearization technique. In the CT control scheme, a filtered tracking error [2] is first defined, and the controller design process is restated using the filtered tracking error dynamics under the assumption that the dynamics are accurately known. To obviate the requirement of the complete knowledge of the manipulator dynamics, adaptive controllers [2] are introduced under the assumption that the effects of nonlinear dynamics can be represented as a linear combination of known regression functions (linear in the unknown parameters [2]).

With the introduction of the artificial neural networks (NNs) [1], [2], [8] and their role as function approximators, adaptive NN controllers are introduced to replace the known regression function with NN activation functions. In all the above mentioned works [2]–[8], the controllers require continuous/periodic measurements of the joint positions and velocities, though the requirement of joint velocity measurements is later relaxed in [1]. Various control schemes for a robot manipulator are presented in the literature taking into account the network delays [9], finite-time control task [10], learning and adaptation to map complex tasks and to generate appropriate manipulation [11]–[16], input constraint by considering saturation nonlinearity [17], time-varying output constraints [18], and the fast adaptation with performance guarantee through adaptation [19].

In addition to the application specific requirements, robotic manipulator systems with remote interoperability enabling network integration are desired in many engineering applications, wherein continuous feedback information to compute appropriate torque input is not always desirable. In this context, event-based sampling and control implementation have become relevant [20]–[25]. In the event-triggered control framework, the sensor measurements are sampled by an event-triggering mechanism (ETM) via an event-triggering condition that is derived using the stability criterion and the controller is updated with the current feedback information only when the ETM generates an event.

Traditionally, in the event-triggered control schemes [21], the ETM is developed for a known, globally stabilizing control input, and the objective is to ensure that this ETM does not exhibit Zeno behavior. Later, in [20], the performance of the approximation-based NN controller under the event-sampled feedback framework is introduced. In the case of adaptive NN controllers under event-triggered feedback framework,

the unknown parameters are updated at aperiodic sampling instants, introducing a trade-off between feedback frequency and learning accuracy. In these efforts [20], [21] and the references therein, full state vector is used for the design of controller and the ETM. However, in practice, only a part of the state vector is available for measurement.

For a robot manipulator, the joint velocities, which are typically not measured, are required for designing the torque input. Therefore, an observer is required to reconstruct the joint velocities from the joint position measurements. Output feedback control design using observers that utilize an NN to approximate the dynamics with periodic/continuous feedback is addressed in [1], [4], and [26]. However, event-sampled implementation of observer together with the controller for a robot manipulator with uncertain dynamics using NNs is nontrivial. Furthermore, with the aperiodic estimation of states, the location of observers plays an important role in the tracking performance and, therefore, needs to be investigated.

In this paper, we propose an adaptive NN-based control scheme for a robot manipulator system utilizing event-sampled measurements of joint position vector. An NN observer is utilized to reconstruct the joint velocity vector. The nonlinear uncertainties capturing the effects of inertia, Coriolis/centripetal forces, frictional forces, and gravitational forces on the robotic manipulator are learned by using a second NN for generating the torque input. In the proposed framework, two controller configurations are presented based on the observer location in the feedback loop. The first configuration considers that the NN observer is co-located with the sensor, whereas in the second configuration, the NN observer is co-located with the controller. Due to the location of the NN observer, the convergence properties of the closed-loop system are not identical for the two configurations. Moreover, the ETM has to be redesigned for these two configurations based on the information available at the ETM.

Therefore, the design and analysis of the NN observer and the NN controller along with the ETM for the two configurations are studied in detail. In addition, adaptive event-triggering conditions are developed for both the configurations based on the stability criterion, and their computational efficiency and control performance are analyzed. Furthermore, though the primary contribution in this paper is the derivation of an output-feedback controller, the state-feedback case is also briefly presented as a corollary. Finally, local uniform ultimate boundedness (UUB) of the tracking errors, observer errors, and the NN weight estimation errors is demonstrated.

The contributions of this paper include: 1) event-driven NN-based control scheme with output feedback under two configurations; 2) design of the NN observers based on the location of the observer in the feedback control loop; 3) event-triggering conditions are developed based on observer location, and the trade-off between frequency of feedback and learning efficiency is analyzed; and 4) the numerical analysis based on the Lyapunov stability theory for the two configurations is presented.

In this paper, standard mathematical notations are used. For a vector x, $\|x\|$ denotes the Euclidean-norm while for a matrix A, $\|A\|$ denotes the Frobenius norm. The smallest singular value is denoted by $\lambda_{\min}$, and the right limit operator is denoted by x^+, i.e., $x^+ = \lim_{r \downarrow t} x(r)$. A brief background on event-sampling, the tracking problem formulation are presented in Section II.

II. Background and Problem Formulation

A. Event-Based Sampling and Control Implementation

In contrast to the traditional periodic feedback control design, in the event-based control implementation, an ETM is co-located with the sensor to determine when the sensor samples should be released to the controller.

To denote the time instants when the controller has access to the sensor samples, a sequence, $\{t_k\}_{k=0}^{\infty}$ with $t_0 = 0$, is defined. These time instants are referred to as the event sampling or event-triggering instants. At the time instant t_k, the sensor measurement denoted by $\gamma(t_k)$ is updated at the controller and between any two sampling instants, a zero-order-hold (ZOH) is utilized at the controller to hold the feedback information. This is represented as $\breve{\gamma}(t) = \gamma(t_k),\ t_k \leq t < t_k + 1,\ \forall k \in \{0, \mathbb{N}\}$. Lack of current sensor measurement at the controller leads to an error $\bar{e}_{\mathrm{ET}}(t) = \gamma(t) - \breve{\gamma}(t),\ t_k \leq t < t_{k+1}$, with $\bar{e}_{\mathrm{ET}}(t_k) = 0,\ \forall k \in \{0, \mathbb{N}\}$, and this error is called the measurement error or event-sampling error. Here, we assume that the computational delays are negligible.

Remark 1: Similar to the ZOH used at the controller, the actuator is also equipped with a ZOH so that a piecewise constant control input is applied to the dynamic system.

The design of controller in the event-triggered feedback framework is challenging partially due to the requirement of designing an ETM which constructs the sequence $\{t_k\}$ dynamically, along with the controller. However, this challenge is further reinforced by the introduction of a learning mechanism in the controller. In this case, in addition to considering the stability of the system, the learning mechanism and its efficiency should be accounted for when the ETM is developed. In the following Lemma, the event-sampled NN approximation property [20] is recalled.

Lemma 1: Given $\chi : A \to \mathbb{R}$, a smooth, real-valued function with a compact support, $A \subset \mathbb{R}^n$, and a positive constant ϵ_M, there exists $W^* \in \mathbb{R}^{p \times 1}$ so that

$$\chi(x) = W^{*T}\phi(\breve{x}) + \breve{\epsilon} \tag{1}$$

holds, where $\|\breve{\epsilon}\| < \epsilon_M$, $\forall \breve{x} \in A$; x, $\breve{x}$ are the continuous and event-sampled variables, $\phi(\breve{x})$ is the bounded event-sampled activation function which forms a basis set of functions in A [2], p is the number of hidden-layer neurons, and $\breve{\epsilon} = W^{*T}(\phi(\breve{x} + \bar{e}_{\mathrm{ET}}) - \phi(\breve{x})) + \epsilon$ is the event-driven reconstruction error where ϵ is the function approximation error [20].

Remark 2: Note that the activation function $\phi(x)$ is used to denote $\phi(\omega x)$, where ω is the input to hidden-layer weights which are randomly set to form a stochastic basis [2]. Since the values of ω are fixed upon initialization, they are not explicitly stated. The approximation of a nonlinear function using NN given in (1) reveals that when the input to the NN is corrupted by the measurement error, the accuracy of the approximation is affected, creating a trade-off between the estimation accuracy and the frequency of events.

Next, the robot manipulator dynamics are introduced followed by a brief background on event-triggered tracking control problem.

B. Robot Manipulator System

The $n-$link robot manipulator dynamics with rigid links [2] are expressed as

$$M(q)\ddot{q}(t) + V_m(q,\dot{q})\dot{q}(t) + G(q) + F(\dot{q}) + \tau_d(t) = \tau(t) \tag{2}$$

where $q(t) \in \mathbb{R}^n$ is the joint variable vector, $V_m(q,\dot{q})$ is the Coriolis/centripetal matrix, $M(q)$ is the inertia matrix, $F(\dot{q})$ models the effects of friction, $G(q)$ is the gravity vector, and $\tau_d(t)$ represents the disturbance torque. The control input vector $\tau(t)$ has components of torque for revolute joints and force for prismatic joints. With the following facts, the tracking control problem for the robot manipulator dynamics as in (2) is introduced.

Facts [2]: The matrix $M(q)$ satisfies $B_{m_1} I \leq M(q) \leq B_{m_2} I$, for some known positive constants B_{m_1}, B_{m_2}, where I is the identity matrix of appropriate dimension, and the Coriolis/centripetal matrix $V_m(q,\dot{q})$ is bounded such that $\|V_m(q,\dot{q})\| \leq V_b\|\dot{q}\|$ with $V_b > 0$. There exists constants $g_B, B_F, B_f > 0$ such that $\|G(q)\| \leq g_B$, and $\|F(\dot{q})\| \leq B_F\|\dot{q}\| + B_f$. The matrix $\dot{M} - 2V_m$ is skew-symmetric, and finally, there exists a positive constant, $\tau_{dM} > 0$, such that $\|\tau_d(t)\| \leq \tau_{dM}$, uniformly for all $t \in \mathbb{R}^+$.

C. Tracking Control Problem

By defining the joint positions and velocities as the state vector, the robot manipulator system dynamics can be represented in a compact form [2] as

$$\begin{bmatrix}\dot{q}\\ \ddot{q}\end{bmatrix} = \begin{bmatrix}\dot{q}\\ -M^{-1}(q)(V_m(q,\dot{q})\dot{q} + F(\dot{q}) + G(q))\end{bmatrix} + \begin{bmatrix}0\\ M^{-1}(q)\end{bmatrix}\tau(t) + \begin{bmatrix}0\\ -M^{-1}(q)\end{bmatrix}\tau_d(t). \tag{3}$$

The objective in a tracking control problem is to design the torque input $\tau(t)$ such that the joint positions track a desired trajectory $q_d(t)$. The desired trajectory is restricted to satisfy the following standard assumption.

Assumption 1 ([1], [2]): Define $q_d(t) \in \mathbb{R}^n$ such that $\|Q_d\| \leq q_B$, where $Q_d(t) = [q_d^T(t)\ \dot{q}_d^T(t)\ \ddot{q}_d^T(t)]^T$, and $q_B > 0$.

In real-world applications, for example, in manufacturing plants, the robotic manipulator is expected to perform tasks which are characterized by position, velocity, and acceleration that are bounded. Therefore, it is reasonable to make the assumption stated above in the design and analysis.

To develop a controller that enables the robot manipulator to track a desired trajectory, begin by defining the tracking error as

$$e(t) = q_d(t) - q(t) \tag{4}$$

and the filtered tracking error [2] as

$$r(t) = \dot{e}(t) + \lambda e(t) \tag{5}$$

with a symmetric matrix $\lambda > 0$ of compatible dimension.

Observe from (5) that a positive choice of λ ensures the stability of $\dot{e}(t)$ if the filtered tracking error is bounded.

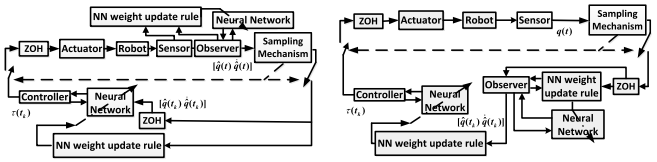

Fig. 1. Controller block diagram. (a) Configuration 1. (b) Configuration 2.

To design the torque input and to ensure boundedness of the filtered tracking error, differentiate (5) with respect to time to reveal $\dot{r}(t) = \ddot{e}(t) + \lambda\dot{e}(t)$. By incorporating the dynamics of the robotic manipulator (2) and utilizing (4) and (5), the dynamics of the filtered tracking error is obtained as [2]

$$M\dot{r}(t) = -V_m r(t) - \tau(t) + f(z) + \tau_d(t) \tag{6}$$

with $f(z) = M(q)(\ddot{q}_d + \lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \lambda e) + F(\dot{q}) + G(q)$ and $z = [e^{\mathrm{T}}\ \dot{e}^{\mathrm{T}}\ q_d^{\mathrm{T}}\ \dot{q}_d^{\mathrm{T}}\ \ddot{q}_d^{\mathrm{T}}]^{\mathrm{T}}$. If the function $f(z)$ is known, it is straightforward to generate a control input, $\tau(t) = f(z) + k_v r(t)$, where $k_v \in \mathbb{R}^{n\times n}$ is a positive definite design matrix, such that the filtered tracking error dynamics in (6) is stable. However, the nonlinear function $f(z)$ is unknown due to the nonlinearities such as friction and inertia matrix.

Furthermore, in the event-triggered feedback framework, the torque input using the event-sampled state is designed as

$$\tau(t) = \hat{f}(\breve{z}) + k_v\breve{r}\ (t), \quad t_k \leq t < t_{k+1} \tag{7}$$

where $\breve{r}(t) = r(t) - e_{\mathrm{ET}}(t)$ is the event-sampled filtered tracking error, e_{ET} is the error due to event-based sampling, and $\hat{f}(\breve{z})$ denotes an estimate of $f(z)$ with $\breve{z}(t) = z(t_k)$. By using (7) in (6), we get

$$M\dot{r} = -V_m r - \hat{f}(\breve{z}) - k_v\breve{r} + f(z) + \tau_d. \tag{8}$$

The filtered tracking error dynamics (8) reveals that the tracking performance of the robotic manipulator is influenced by the measurement error due to event-based sampling and the mismatch between $f(z)$ and its estimate.

Moreover, all the controller equations (4)–(8) assume the availability of both the joint position and velocities. In Section III, an observer is designed first to generate the estimate of joint velocities, and then, a controller which utilizes the estimated state vector is developed to design (7).

III. NN Observer and Controller Design

Before developing the controller, two output feedback control schemes, which are represented using block diagrams in Fig. 1, are introduced. In the first architecture [Fig. 1(a)], the observer is co-located with the sensor. This ensures that the observer has access to the sensor data continuously and hence, the convergence of the observer estimation error will be fast. When an event is triggered, the estimated joint velocities and the joint positions are fed back to the controller. The controller has access to the event-sampled estimated states to generate the required torque. Moreover, the ETM can be designed as a function of all the estimated states.

In the second architecture [Fig. 1(b)], the observer is co-located with the controller. In this configuration, only the output information is transmitted from the sensor to the

controller. The observer NN and the controller are updated with current sensor readings only when there is an event. This configuration introduces two issues: one, the observer is injected with an additional error due to event-triggered feedback and second, the ETM now cannot be a function of the estimated state vector as only the outputs are continuously available. However, this configuration is preferable since the observer and controller are usually implemented together in software. Furthermore, it is easy to implement such control schemes when multiple sensors are present at different joint locations.

In both cases, to determine the sequence $\{t_k\}_{k=0}^{\infty}$, the ETM should be synthesized. In [20], an inequality condition which is a function of the NN parameter is developed which resulted in frequent events during the transient learning phase. This also required an additional mirror NN estimator at the ETM, increasing the computations.

Next, the analytical design of the controller, the ETM, and the observer are presented for configuration 1.

A. Observer and Controller Dynamics (Configuration 1)

The robot manipulator dynamics (3) can be represented as

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = \bar{f}(x(t), \tau(t)) \tag{9}$$

where x_1, x_2 denote the joint variables and joint velocities, respectively, $x = [x_1^{\mathrm{T}}, x_2^{\mathrm{T}}]^{\mathrm{T}}$, and $\bar{f}(x, \tau) = -M^{-1}(q)(V_m(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d(t) - \tau(t))$. Using the function approximation property of the NN [2], the nonlinear map $\bar{f}(.)$ can be represented as $\bar{f}(x, \tau) = W_o\sigma_o(x, \tau) + \varepsilon_o(x)$, where W_o is the unknown, constant NN weights, $\sigma_o(\cdot)$ is the activation function, and $\varepsilon_o(x)$ is the reconstruction error.

Assumption 2 [2]: The NN approximation/reconstruction error $(\varepsilon_o(.))$ is bounded such that $\|\varepsilon_o\| \leq \varepsilon_{oM}$, for $\varepsilon_{oM} > 0$. The nonlinear NN activation function, $\sigma_o(.)$, is chosen such that $\|\sigma_o(\cdot)\| \leq \sqrt{N_o}$, where N_o denotes the number of neurons in the hidden layer. Also, there exists $W_{oM} > 0$ such that $\|W_o\| \leq W_{oM}$ holds. Finally, the nonlinear activation function satisfies $\|\sigma_o(\alpha) - \sigma_o(\beta)\| \leq L_\sigma \|\alpha - \beta\|$, $\forall \alpha, \beta \in A$, where A is the compact domain for the activation function.

In this paper, standard assumptions [2] are made in the NN design and analysis as listed in Assumption 2. It is proven in [2] that as the number of hidden-layer neurons is increased, i.e., as $N_o \to \infty$, the reconstruction error, $\varepsilon \to 0$. In addition, commonly used activation functions such as the sigmoidal function and radial basis functions satisfy the bounds and the smoothness properties stated above [2].

Let $\hat{W}_o$ represent the weight estimates of W_o. The approximated nonlinear function is given by $\hat{\bar{f}}(x, \tau) = \hat{W}_o\sigma_o(\hat{x}, \tau)$. Let the estimated joint variable be denoted by $\hat{x}_1$ and the estimated joint velocity be represented using the vector $\hat{x}_2$ and let $\hat{x} = [\hat{x}_1^{\mathrm{T}}\ \hat{x}_2^{\mathrm{T}}]^{\mathrm{T}}$. Using $\hat{W}_o$, define the NN observer as

$$\begin{aligned}\dot{\hat{x}}_1(t) &= \hat{x}_2(t) + K_d\tilde{x}_1(t)\\ \dot{z}_2(t) &= \hat{W}_o\sigma_o(\hat{x}, \tau) + K\tilde{x}_1(t), \quad \hat{x}_2(t) = z_2(t) + K_p\tilde{x}_1(t).\end{aligned} \tag{10}$$

Now, rewriting (10) using the estimated state variables $\hat{x}_1, \hat{x}_2$, one has

$$\dot{\hat{x}}_1 = \hat{x}_2 + K_d\tilde{x}_1, \quad \dot{\hat{x}}_2 = \hat{W}_o\sigma_o(\hat{x}, \tau) + K\tilde{x}_1 + K_p\dot{\tilde{x}}_1 \tag{11}$$

where K_d, K, and K_p are observer gains, and the output estimation error, $\tilde{x}_1(t) = x_1(t) - \hat{x}_1(t)$ and $\hat{x}(0)$, is the initial state estimate. Note that the dynamics in terms of a new variable $\dot{z}_2$ is introduced in (10) and is utilized to get the estimate of x_1, x_2 as in [1]. This change of variable enables both the implementation of the observer using (10) and the convergence, stability analysis using (11).

To analyze the stability of the observer, first, define the state estimation error as $\tilde{x} = [\tilde{x}_1^{\mathrm{T}}\ \tilde{x}_2^{\mathrm{T}}]^{\mathrm{T}}$, where $\tilde{x}_2 = x_2 - \hat{x}_2$. Since the observer is co-located with the sensor, the observer dynamics are not injected with the measurement error and the estimation error dynamics are obtained as

$$\begin{aligned}\dot{\tilde{x}}_1 &= \tilde{x}_2 - K_d\tilde{x}_1\\ \dot{\tilde{x}}_2 &= -K_p\tilde{x}_2 - (K - K_pK_d)\tilde{x}_1 + \tilde{W}_o\sigma_o(\hat{x}) + \varepsilon_1(\tilde{x})\end{aligned} \tag{12}$$

where $\varepsilon_1(\tilde{x}) = W_o(\sigma_o(x) - \sigma_o(\hat{x})) + \varepsilon_o(x)$. The stability results of the observer are presented next.

Lemma 2 (Configuration 1): Consider the dynamics of the robotic manipulator (9) and the observer dynamics (10). Let the weight tuning rule for the observer NN be defined as

$$\dot{\hat{W}}_o = -\Gamma_oK_d\sigma_o(\hat{x})\tilde{x}_1^{\mathrm{T}} - \Gamma_o\kappa_o\hat{W}_o \tag{13}$$

where $\Gamma_o, \kappa_o > 0$ are design parameters. Then, the error dynamics (12) and the weights of the NN observer are locally UUB when $K_dK > (1/2)N_o + (1/2)K_p^2$, $K_p > (1/2)K_d^2 + (1/2)N_o + (1/2)$, $\kappa_o > (1/2) + (1/2)K_d^2$, and the bound is a function of the approximation error, target NN observer weights given by $\mathcal{B}_o = (1/2)\|\varepsilon_1\|^2 + \kappa_o^2W_{oM}^2$.

Proof: See Appendix in the Supplementary file.

To develop the control equations, begin by defining the tracking error using the estimated states as $\hat{e}(t) = x_d(t) - \hat{x}_1(t)$, $\dot{\hat{e}}(t) = \dot{x}_d(t) - \hat{x}_2(t)$. Now, the estimated filtered tracking error can be defined as $\hat{r}(t) = \dot{\hat{e}}(t) + \lambda e(t)$. Note that the filtered tracking error is calculated using the estimated joint velocity tracking error and the actual tracking error (4).

Next, the nonlinear map in (6) is parameterized using an NN as $f(z) = W^{\mathrm{T}}\phi(z) + \varepsilon$, where the NN weights W denotes the target parameter for the learning scheme, the activation function $\phi(.)$, and the reconstruction error ε with $f(z)$ defined as in (6). Using the estimated states, the tracking errors are redefined and the input to the NN is redefined as $\hat{z} = [\hat{e}^{\mathrm{T}}\ \dot{\hat{e}}^{\mathrm{T}}\ q_d^{\mathrm{T}}\ \dot{q}_d^{\mathrm{T}}\ \ddot{q}_d^{\mathrm{T}}]^{\mathrm{T}}$. The estimate of the nonlinear map in the event-triggered feedback framework is given as

$$\hat{f}(\breve{z}) = \hat{W}^{\mathrm{T}}\phi(\breve{z}) \tag{14}$$

with the estimated NN weights, $\hat{W}$ and $\breve{z} = \hat{z}(t_k)$. Let the weight estimation error be defined as the difference between the target weights and the estimated weights, $\tilde{W}^{\mathrm{T}} = W^{\mathrm{T}} - \hat{W}^{\mathrm{T}}$. Using the estimated states, the estimated filtered tracking error dynamics [1] is revealed as

$$M\dot{\hat{r}} = -V_m\hat{r} - \tau + f(z) + \tau_d \tag{15}$$

with $f(z) = M(q)(\ddot{q}_d + \dot{\tilde{x}}_2 + \lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \tilde{x}_2 + \lambda e) + F(\dot{q}) + G(q)$. Using the control torque (7)

$$M\dot{\hat{r}} = (-V_m - k_v)\hat{r} + \tau_d + \tilde{W}^{\mathrm{T}}\phi(\hat{z}_e) + W^{\mathrm{T}}[\phi(z) - \phi(\hat{z}_e)] + \varepsilon + k_v e_{\mathrm{ET}}. \quad (16)$$

The subscript $(\cdot)_e$ indicates the event-sampled variable. The slope of the filtered tracking error (16) reveals that the measurement error introduced by event-based sampling and the error due to the observer affects the tracking performance of the manipulator. The NN is designed by taking into account the standard assumptions as stated in Assumption 2. Specifically, there exists positive constants ε_M, $W_M > 0$ so that $\|\varepsilon\| \leq \varepsilon_M$, $\|W\| \leq W_M$ holds. The activation function, $\phi(\cdot)$, satisfies $\|\phi(\cdot)\| \leq \sqrt{N_0}$, $\|\phi(\alpha) - \phi(\beta)\| \leq L_\phi\|\alpha - \beta\|$, $\forall \alpha, \beta \in A$, where N_0 denotes the number of neurons in the hidden layer.

The stability results of the closed-loop system with continuous feedback is briefly presented next.

Theorem 1: Let the estimated filtered tracking error dynamics for the robot manipulator, using the approximation-based torque input $\tau(t) = \hat{f}(z) + k_v\hat{r}(t)$ and the NN approximation (14), be defined as $M\dot{\hat{r}} = (-V_m - k_v)\hat{r} + \tau_d + \tilde{W}^{\mathrm{T}}\phi(\hat{z}) + W^{\mathrm{T}}[\phi(z) - \phi(\hat{z})] + \varepsilon$. Let the dynamics of the observer be given by (11). Select the NN weight tuning rule as

$$\dot{\hat{W}} = \Gamma\phi(\hat{z})\hat{r}^{\mathrm{T}} - \kappa\Gamma\hat{W} \quad (17)$$

with the design parameters $\Gamma > 0, \kappa > 0.5, \lambda_{\min}(k_v) > (1/2)$. Let the NN observer weight update rule be given by (13) with the observer design parameters satisfying the conditions given in Lemma 2. Under Assumptions 1 and 2, the NN observer, tracking errors, and the error in the NN weight estimates are locally UUB.

Proof: See Appendix in the Supplementary file..

In the following theorem, the ETM is designed and the stability results for the event-sampled controller will be presented in two cases following an approach similar to [20].

Theorem 2 (Configuration 1): Consider the estimated filtered tracking error dynamics given by (16) and the observer dynamics given by (11). Define the approximated torque input (7) and the weight tuning law

$$\hat{W}^{+} = \hat{W}(t) + \hat{\Gamma}\phi(\hat{z})\hat{r}^{\mathrm{T}} - \kappa\Gamma\hat{W}(t), \quad t = t_k \quad (18)$$

for $k \in \mathbb{N}$. Let Assumptions 1 and 2 hold and let an event be triggered such that the following condition holds during the inter-sampling period

$$\|e_{\mathrm{ET}}\|^2 \leq \sigma\mu_k\|\hat{r}\|^2. \quad (19)$$

Then, the observer estimation error, NN weight estimation error, and the tracking errors are locally ultimately bounded provided the design parameters satisfy $\sigma, \mu_k \in (0,1)$, $\kappa > 0$, $\Gamma > 0$, $\lambda_{\min}(k_v) \geq (3/2) + \sigma$, and $\hat{\Gamma} = (\Gamma/1 + \|\hat{r}\|^2)$.

Proof: See Appendix in the Supplementary file.

Remark 3: In Theorem 2, two different values for μ_k in (19) are derived. In condition 1, $\mu_k = 2\sigma/\|k_v\|^2$, is a constant and it does not require the information regarding the NN weights to determine the event-based sampling instants. In contrast, the second sampling condition uses $\mu_k = 2\sigma/[\|\hat{W}^{\mathrm{T}}\|^2 L_\phi^2 + \|k_v\|^2]$. This condition, similar to [20], utilizes the NN weights.

B. Observer and Controller Dynamics (Configuration 2)

Consider the second control configuration as represented in Fig. 1(b). To reconstruct the joint velocities when the observer is co-located with the controller, define the event-driven observer dynamics as

$$\dot{\hat{x}}_1(t) = \hat{x}_2(t) + K_d\tilde{x}_{1,e}(t)$$
$$\dot{z}_2(t) = \hat{W}_o\sigma_o(\hat{x}_e, \tau) + K\tilde{x}_{1,e}(t), \quad \hat{x}_2(t) = z_2(t) + K_p\tilde{x}_{1,e}(t). \quad (20)$$

Rewriting the observer dynamics (20) using the coordinate variables $\hat{x}_1(t), \hat{x}_2(t)$, we get

$$\dot{\hat{x}}_1 = \hat{x}_2 + K_d\tilde{x}_{1,e}, \quad \dot{\hat{x}}_2 = \hat{W}_o\sigma_o(\hat{x}_e) + K\tilde{x}_{1,e} + K_p\dot{\tilde{x}}_{1,e}. \quad (21)$$

Note that the observer given by (20) has access to the outputs only at the event-triggering instants, introducing an error in (21). From (21), the error dynamics are obtained as

$$\dot{\tilde{x}}_1 = \tilde{x}_2 - K_d\tilde{x}_{1,e}$$
$$\dot{\tilde{x}}_2 = \tilde{W}_o\sigma_o(\hat{x}) - K\tilde{x}_{1,e} - K_p\dot{\tilde{x}}_{1,e} + \varepsilon_1(\tilde{x}) \quad (22)$$

where $\varepsilon_1(\tilde{x}) = W_o(\sigma_o(x) - \sigma_o(\hat{x})) + \varepsilon_o(x)$. The stability results of the observer (20) are presented next.

Lemma 3 (Configuration 2): Consider the dynamics of the robot manipulator defined by (3) and the observer dynamics given by (20). Let the weight tuning rule for the NN observer be defined as

$$\hat{W}_o^{+} = \hat{W}_o(t) - K_d\hat{\Gamma}_o\sigma_o(\hat{x})\tilde{x}_1^{\mathrm{T}} - \kappa_o\Gamma_o\hat{W}_o, \quad t = t_k \quad (23)$$

for $k \in \mathbb{N}$. Then, the observer estimation error and the NN observer weights are locally UB when $K_dK > K_p^2, K_p > K_d^2 + 1$, $\hat{\Gamma}_o = (\Gamma_o/1 + \|\tilde{x}_1\|^2), \kappa_o > 0$ with the adjustable bounds defined by $\mathcal{B}_{3oa} + \psi(e_{\mathrm{ET}})$, (for the inter-event period) and $\mathcal{B}_{3ob}$ (for the trigger instants) as a function of ε, measurement errors, and the W_M and design constants.

Proof: See Appendix in the Supplementary file.

The closed-loop results of configuration 2 are presented next, wherein the event-triggering condition is developed with only the joint position vector and event-triggering errors as opposed to the estimated joint positions and velocities in (19).

Theorem 3 (Configuration 2): Consider the estimated filtered tracking error for the robot manipulator system defined as (16) and the observer dynamics from (20). Define the torque input (7) by utilizing the NN approximation (14). Consider the weight tuning laws (18) and (23), where $\Gamma > 0$, $\kappa > 0$, $k_v \geq 3$. If Assumptions 1 and 2 are satisfied and let an event be triggered such that the following condition holds in the inter-event period

$$\|e_{\mathrm{ET}}\|^2 \leq \sigma\mu_k\|e\|^2 \quad (24)$$

where $\mu_k, \sigma \in (0,1)$, $e_{\mathrm{ET}} = e(t) - e(t_k)$. Then, the tracking error, the observer estimation error, and the NN weight estimation errors are locally UB.

Proof: See Appendix in the Supplementary file.

Remark 4: As $\hat{r}, \tilde{x}$ converges to their bounds, the actual filtered tracking error converges to its bound. The proofs of Theorems 2 and 3 and Lemma 3 considers two cases

which correspond to the event-triggering instants and inter-event period to ensure boundedness of the closed-loop signals. The ETM proposed in this paper does not exhibit Zeno behavior. This is a consequence of the stability results presented in this section, and its proof follows arguments similar to that presented in [20].

Corollary 1 (State-Feedback Controller): Consider the filtered tracking error dynamics (8). In addition to the Assumptions listed (Assumptions 1 and 2), let the joint velocities be measured. Generate the torque input (7) by utilizing the approximated dynamics of the robotic manipulator (6), where the estimated joint velocities are replaced by the actual joint velocities. Consider the weight tuning law (17), where $\Gamma > 0$, $\kappa > 0, k_v \geq 3$. Let the trigger condition (19) be satisfied during the inter-event period with $\mu_k, \sigma \in (0, 1)$. Then, the tracking error and the NN weight estimation error are locally UB [27].

In Section IV, simulations results and discussions based on the results are summarized.

IV. Results and Discussion

In this section, the simulation results for the event-sampled control of an n-link robotic manipulator [2] are presented to illustrate the effectiveness of the proposed analytical design.

The parameters of the robot manipulator dynamics used in the simulation includes [1]: $m_1 = 1$, $m_2 = 2.3$, $g = 9.8$, and $L_i = 1$ for $i = \{1, 2\}$. The controller design variables are selected as $k_v = 30I$ and $\lambda = 5$, $\kappa = 0.0001, \Gamma = 500$, $L_\phi = 1$, and $\sigma = 0.1$. The observer design parameters are selected as $K = 500, K_p = 2, K_d = 0.2$, and $\Gamma_o = 6$, $\kappa_o = 0.1$. The initial values of the manipulator system states and the NN weights are chosen at random from $[0, 1]$. The trajectory to be tracked by the manipulator system is given by $\sin(\omega t), \cos(\omega t)$ with $\omega = 1$. The NN observer is designed with 26 hidden-layer neurons and sigmoid activation [1] and the controller NN is designed with 30 hidden-layer neurons with sigmoid activation function. The input to hidden layer weights are randomly selected in the interval $[0, 0.001]$ and held to form a random vector functional link network [1]. The design parameters for the controller and the observer are similar to those reported in [1] and $\tau_d = [8\sin(2t)5\sin(2t)]^T$.

First, a comparison between the standard event-driven proportional derivative (PD) control (without the NN compensation in the torque) and event-driven NN control are presented in Fig. 2. These can be used to analyze the advantages of the event-sampled NN controller. Fig. 2, as expected, demonstrates that the event-driven NN controller improves the performance of the controlled robotic system over the event-driven PD control.

Case 1: First, the event-triggering condition which is a function of NN weights is chosen with $\mu_k = 2\sigma/[\|\hat{W}^{\mathrm{T}}\|^2 L_\phi^2 + \|k_v\|^2]$, $\sigma = 0.45, L_\phi = 1, k_v = 30I$ and the NN tuning parameters $\Gamma = 0.5$, and $\Gamma_o = 0.6$ with κ, κ_o as chosen above. The initial conditions are the same as those stated before. The actual and desired joint variables and the event-based observer estimation errors are plotted in Fig. 3. Due to the location of the observer, the observer estimation error with event-based feedback remains unaffected. The tracking performance on the other hand converges after the observer error reaches its bound.

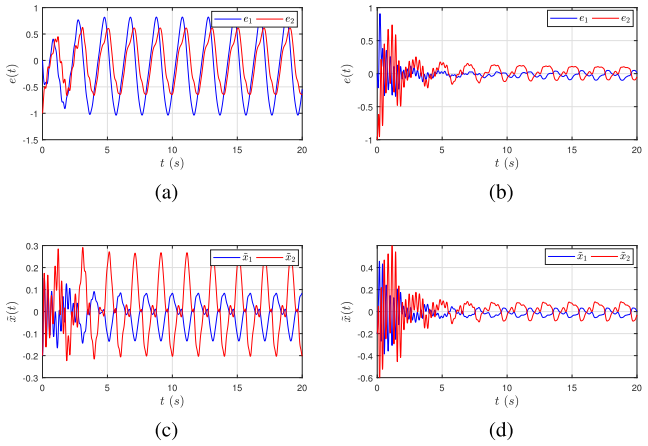

Fig. 2. Tracking error. (a) PD control. (b) NN control. Observer estimation error. (c) PD control. (d) NN control.

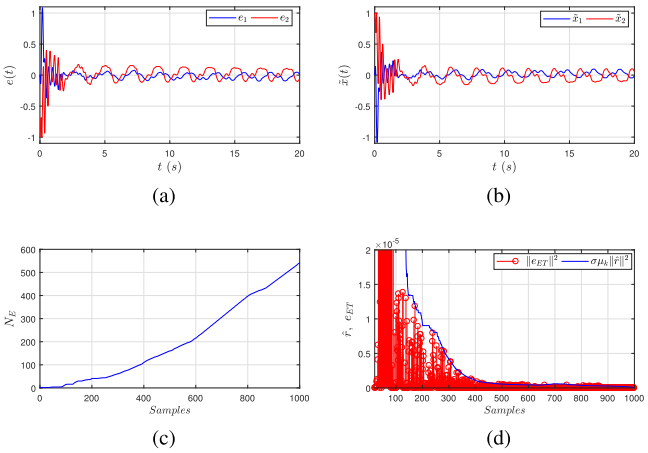

Fig. 3. Case 1. (a) Tracking error. (b) Observer estimation error. (c) Cumulative number of events. (d) Event-triggering error and threshold.

Fig. 3(c) and (d) presents the performance of the event-sampling mechanism. The threshold function which is defined using the Lyapunov analysis ensures that the event-triggering error remains bounded. It can also be observed that a total number of events (N_E) is 540 demonstrating the efficacy of the event-sampling control. The important difference between this ETM with the ETM in case 2 is that the number of events initially is more in case 1 and becomes sporadic later, facilitating the NN learning.

Case 2: In the second case, the event-triggering condition which is not a function of NN weights is evaluated with $\mu_k = 2\sigma/\|k_v\|^2$ and $\sigma = 0.0006$, $k_v = 30I$. In contrast to case 1, the event-triggering condition does not require an NN at the trigger mechanism and, hence, reduces the computations. The initial conditions were kept the same.

The tracking performance and the observer estimation error trajectories are recorded in Fig. 4. Since the design parameter σ which affects the event triggering threshold is adjusted to get a similar cumulative event as in case 1, similar number of events are recorded. However, it is observed that in contrast

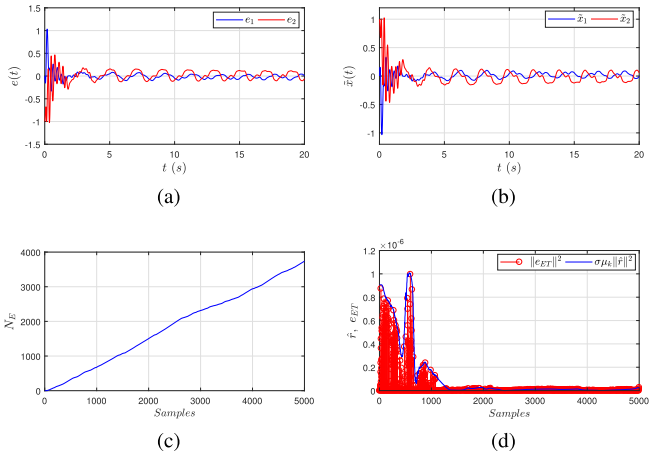

Fig. 4. Case 2. (a) Tracking error. (b) Observer estimation error. (c) Cumulative number of events. (d) Event-triggering error and threshold.

TABLE I

CONFIGURATION 1: ANALYSIS WITH σ

$\mu_k = \frac{2\sigma}{[\|\hat{W}^T\|^2 L_\phi^2 + \|k_v\|^2]}$			$\mu_k = \frac{2\sigma}{\|k_v\|^2}$		
σ	N_E	t_o (s)	σ	N_E	t_o (s)
0.05	865	4.257	0.0002	790	4.27
0.1	762	4.244	0.0006	672	4.25
0.3	743	4.466	0.001	636	4.47
0.4	620	4.525	0.003	583	4.54
0.5	513	4.538	0.0045	572	4.321

TABLE II

CONFIGURATION 2: ANALYSIS WITH σ

$\mu_k = \frac{2\sigma}{[\|\hat{W}^T\|^2 L_\phi^2 + \|k_v\|^2 + L_\psi]}$			$\mu_k = \frac{\sigma}{(\frac{1}{2}\|k_v\|^2 + L_\psi)}$		
σ	N_E	t_o (s)	σ	N_E	t_o (s)
0.05	850	4.240	0.0002	801	4.283
0.1	772	4.242	0.0006	720	4.522
0.3	750	4.457	0.001	663	6.367
0.4	564	6.257	0.003	591	6.518
0.5	525	6.268	0.0045	561	8.576

to case 1, more events are not triggered in the transient NN learning period. The variation in the number of events with the design parameter σ is summarized in Table I (configuration 1). It can be observed that for both the cases, similar cumulative events can result from appropriate choice of σ. The observer convergence time t_o is not affected much, as it has continuous access to the sensor measurements. To avoid redundancy, the results for configuration 2 are summarized in Table II. For the simulations, $L_\psi = 1$ is chosen. In the configuration 2, since the observer is injected with the measurement errors, the convergence time varies with the design parameter σ, as can be seen in Table II.

V. CONCLUSION

In this paper, approximation-based control implementation for the robot manipulator using only the joint position vector is presented in the event-driven NN-based learning formalism. In this context, two control configurations are presented, and two event-triggering conditions are developed in each of these configurations. In the first control configuration, the bounds on the closed-loop signals are smaller when compared to configuration 2. This is expected because the observer when co-located with the sensor has continuous access to the output and is updated continuously. Two conditions that determine the event-triggering instants are obtained: one condition is designed as a function of the NN weights, and this increased computations and facilitated NN learning while the second condition relaxed the computational complexity while generating fewer events. Using the Lyapunov stability theory, analytical bounds are calculated while deriving sufficient conditions for stability of the tracking error system, NN weights, and substantiated with simulation analysis. The inclusion of the effects of network losses, asynchronous and decentralized event-triggering, on the tracking performance of a flexible robot manipulator system will constitute an interesting problem for future research.

REFERENCES

[1] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 301–309, Apr. 1999.
[2] J. S. Lewis, A. Yesildirak, and S. Jagannathan, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Boca Raton, FL, USA: CRC Press, 1998.
[3] T. H. Lee and C. J. Harris, *Adaptive Neural Network Control of Robotic Manipulators*, vol. 19. Singapore: World Scientific, 1998.
[4] C. C. Cheah, X. Li, X. Yan, and D. Sun, "Observer-based optical manipulation of biological cells with robotic tweezers," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 68–80, Feb. 2014.
[5] M. R. Kermani, R. V. Patel, and M. Moallem, "Multimode control of a large-scale robotic manipulator," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1264–1270, Dec. 2007.
[6] C. C. Cheah, X. Li, X. Yan, and D. Sun, "Simple PD control scheme for robotic manipulation of biological cell," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1427–1432, May 2015.
[7] M. Sfakiotakis, A. Kazakidi, T. Evdaimon, A. Chatzidaki, and D. P. Tsakiris, "Multi-arm robotic swimmer actuated by antagonistic SMA springs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep./Oct. 2015, pp. 1540–1545.
[8] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. Boston, MA, USA: Kluwer, 2002.
[9] Y.-C. Liu and N. Chopra, "Control of robotic manipulators under input/output communication delays: Theory and experiments," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 742–751, Jun. 2012.
[10] M. Galicki, "Finite-time control of robotic manipulators," *Automatica*, vol. 51, pp. 49–54, Jan. 2015.
[11] A. Visioli, G. Ziliani, and G. Legnani, "Iterative-learning hybrid force/velocity control for contour tracking," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 388–393, Apr. 2010.
[12] M. Sun, S. S. Ge, and I. M. Y. Mareels, "Adaptive repetitive learning control of robotic manipulators without the requirement for initial repositioning," *IEEE Trans. Robot.*, vol. 22, no. 3, pp. 563–568, Jun. 2006.
[13] C. Sun, H. Gao, W. He, and Y. Yu, "Fuzzy neural network control of a flexible robotic manipulator using assumed mode method," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.
[14] Y. Pan, Y. Liu, B. Xu, and H. Yu, "Hybrid feedback feedforward: An efficient design of adaptive neural network control," *Neural Netw.*, vol. 76, pp. 122–134, Apr. 2016.
[15] J. de Jesús Rubio, "Sliding mode control of robotic arms with deadzone," *IET Control Theory Appl.*, vol. 11, no. 8, pp. 1214–1221, 2016.
[16] J. de Jesús Rubio, "Discrete time control based in neural networks for pendulums," *Appl. Soft Comput.*, vol. 68, pp. 821–832, Jul. 2018.
[17] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 334–344, Mar. 2016.
[18] W. He, H. Huang, and S. S. Ge, "Adaptive neural network control of a robotic manipulator with time-varying output constraints," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3136–3147, Oct. 2017.
[19] A. Pekarovskiy, T. Nierhoff, S. Hirche, and M. Buss, "Dynamically consistent online adaptation of fast motions for robotic manipulators," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 166–182, Feb. 2018.

[20] A. Sahoo, H. Xu, and S. Jagannathan, "Neural network-based event-triggered state feedback control of nonlinear continuous-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 497–509, Mar. 2016.
[21] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.
[22] P. Tallapragada and N. Chopra, "Event-triggered dynamic output feedback control for LTI systems," in *Proc. IEEE 51st IEEE Conf. Decision Control (CDC)*, Dec. 2012, pp. 6597–6602.
[23] X. Zhong and H. He, "An event-triggered ADP control approach for continuous-time system with unknown internal states," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 683–694, Mar. 2017.
[24] D. Wang, H. He, and D. Liu, "Improving the critic learning for event-based nonlinear H_∞ control design," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3417–3428, Oct. 2017.
[25] X. Yang, H. He, and D. Liu, "Event-triggered optimal neuro-controller design with reinforcement learning for unknown nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.
[26] M. Erlic and W.-S. Lu, "A reduced-order adaptive velocity observer for manipulator control," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 293–303, Apr. 1995.
[27] V. Narayanan and S. Jagannathan, "Event-sampled adaptive neural network control of robot manipulators," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 4941–4946.

Vignesh Narayanan (M'17) received the B.Tech. degree in electrical engineering from Shanmugha Arts, Science, Technology & Research Academy, Thanjavur, India, in 2012, the master's degree in electrical engineering from the National Institute of Technology, Kurukshetra, India, in 2014, and the Ph.D. degree from the Missouri University of Science and Technology, Rolla, MO, USA, in 2017.

He is currently a Post-Doctoral Fellow with Washington University, St. Louis, MO, USA. His current research interests include neural network control, and learning and adaptation in systems theory.

Sarangapani Jagannathan (F'16) is currently with the Missouri University of Science and Technology, Rolla, MO, USA, where he is also a Rutledge Emerson Distinguished Professor of electrical and computer engineering and served as the Site Director of the NSF Industry/University Cooperative Research Center on intelligent maintenance systems. He has co-authored, with his students, 157 peer-reviewed journal articles, 267 refereed IEEE conference articles, and several book chapters, and has authored or co-edited six books. He holds 20 U.S. patents, one patent defense publication and several pending. He graduated 29 doctoral and 30 M.S. thesis students, and his total funding is in excess of U.S. $16 million with over U.S. $9.8 million toward his shared credit from federal and industrial entities. His current research interests include neural network control, adaptive event-triggered control/cyber-physical systems, prognostics, and autonomous systems/robotics.

Dr. Jagannathan is a fellow of the Institute of Measurement and Control, U.K., and the Institution of Engineering and Technology, U.K. He received many awards, including the 2000 NSF Career Award, the 2001 Caterpillar Research Excellence Award, the 2007 Boeing Pride Achievement Award, and the 2018 IEEE Control System Society's Transition to Practice Award. He was a Co-Editor of the IET book series on control from 2010 to 2013, and is currently serving on many editorial boards, including the IEEE Systems, Man and Cybernetics. He has been on organizing committees of several IEEE conferences.

Kannan Ramkumar (S'04–M'14) received the B.Tech. degree in instrumentation and control engineering from Madurai Kamaraj University, Madurai, India, in 1997, the M.Tech. degree from the National Institute of Technology, Trichy, India, in 2000, and the Ph.D. degree in control engineering from Shanmugha Arts, Science, Technology & Research Academy (SASTRA University), Thanjavur, India, in 2010.

Since 1998, he has been with the Department of Electronics and Instrumentation, SASTRA University, where he was an Assistant Professor and became an Associate Professor in 2011 and a Professor in 2018. He was the Chair Professor of the Wipro Mission 10X, a pedagogy skills improvement initiative in SASTRA University for the year 2011–2012. He is also heading the Control Artificial Intelligence, Biomedical and Robotics Engineering Group, SASTRA University. His current research interests include mobile robotics, estimation and control theory, electrical drive systems, and solving machine learning, estimation, and control problems related to applications such as electric vehicle, mobile robots, and process control.

Dr. Ramkumar received the prestigious "Innovative Practitioner" a teaching excellence award from Wipro in 2011.