01 Jan 2022

Concurrent Learning-Based Neuro-Adaptive Robust Tracking Control of Wheeled Mobile Robot: An Event-Triggered Design

Krishanu Nath

Manas Kumar Bera

Sarangapani Jagannathan
Missouri University of Science and Technology, sarangap@mst.edu

Recommended Citation

K. Nath et al., "Concurrent Learning-Based Neuro-Adaptive Robust Tracking Control of Wheeled Mobile Robot: An Event-Triggered Design," *IEEE Transactions on Artificial Intelligence*, Institute of Electrical and Electronics Engineers, Jan 2022.
The definitive version is available at https://doi.org/10.1109/TAI.2022.3207133

Concurrent Learning-Based Neuro-Adaptive Robust Tracking Control of Wheeled Mobile Robot: An Event-Triggered Design

Krishanu Nath, *Student Member, IEEE*, Manas Kumar Bera, *Member, IEEE* and Sarangapani Jagannathan, *Fellow, IEEE*

***Abstract*—In this paper, an event-based neuro-adaptive robust tracking controller for a perturbed and networked differential drive mobile robot (DMR) is designed with concurrent learning. A radial basis function neural network, which approximates an unknown perturbation, is used to design an adaptive sliding mode controller (SMC). The RBFNN weights and SMC parameters are estimated online using an adaptive tuning law to ensure performance with reduced chattering. To improve the convergence of RBFNN weight estimation error, a concurrent learning-based adaptive law is derived, which uses measured online and recorded data. Further, a suitable triggering condition is designed to achieve a reduced number of control computations while minimizing network resources without sacrificing the stability of the sampled data closed-loop control system. A finite sampling frequency is guaranteed for the designed triggering condition by establishing a positive lower bound on the inter-event execution time which is equivalent to the Zeno-free behaviour of the system. Finally, the proposed event-based neuro-adaptive robust controller is implemented on a practical system (Q-bot 2e) to show the effectiveness of the proposed design.**

***Impact Statement*—The wheeled mobile robots are used extensively in many indoor and outdoor applications like warehouse automation, agriculture, space mission, etc. With the development of technologies like the internet of things and cyber-physical systems, these robots are a part of a networked system while working in a dynamic environment. The control of the mobile robots in such situations is challenging due to network constraints and dynamic perturbations. In this work, we design an event-triggered robust controller for trajectory tracking of a DMR in the presence of perturbations where a concurrent learning-based adaptive sliding mode control with RBFNN is proposed for the first time. Experimental results verify the effectiveness of the proposed scheme with a 94.54% reduction in control updates compared to the traditional time-based implementation. Also, it is established that concurrent learning improves tracking accuracy with reduced control effort in comparison with baseline adaptive design.**

***Index Terms*—Adaptive sliding mode control, concurrent learning, differential drive mobile robot, event-triggered control, radial basis function neural network.**

K. Nath and M. K. Bera are with the Department of Electronics and Instrumentation Engineering, National Institute of Technology, Silchar, India. India (e-mail: krishanu_rs@ei.nits.ac.in, manas@ei.nits.ac.in).

Sarangapani Jagannathan is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sarangap@mst.edu).

The project or effort is sponsored in part by Office of Naval Research Grant N00014-21-1-2232, Department of Army Cooperative Research Agreement W911NF2120260, Intelligent Systems Center at Missouri University of Science and Technology, Rolla.

This paragraph will include the Associate Editor who handled your paper.

I. Introduction

THE two-wheeled differential-drive mobile robot (DMR) has a large domain of applications such as unmanned space missions, military surveillance, disaster management, and warehouse maintenance, where they work as a single entity or as a formation. These DMR applications require tracking a trajectory in uncertain environments, which is a challenging task. The difficulties arise due to the inherent non-linearity caused by the nonholonomic constraints and Multi-Input-Multi-Output (MIMO) nature of the dynamics. Moreover, the kinematic model of DMR is an under-actuated system. This complex dynamic DMR structure restricts the application of linear controllers [1] necessitating neural network (NN)-based advanced control schemes.

Recently, with the proliferation of wireless sensor networks, such DMR systems are treated as networked control systems (NCS) or more generally cyber-physical systems (CPS). In NCS, controllers are implemented in a discrete-time setting, connected to sensor/ actuator nodes via a shared communication link. In this situation, the efficient utilization of communication, computing and energy capabilities becomes a crucial issue. A major challenge in NCS is to design control schemes that can save communication and computational resources while guaranteeing the overall performance in the presence of network-induced imperfections and external disturbances. In such a scenario, to overcome the shortcomings of time-triggered control, many resource-aware control strategy has been developed, out of which event-triggered control (ETC) is an effective alternative [2].

For a DMR system which suffers from model uncertainties, parameter variations and external disturbances, a nonlinear robust sliding mode control (SMC) [3] is a preferred technique. Nevertheless, to minimize chattering with SMC schemes and to mitigate the network imperfections, event-triggered SMC design is necessary. A detailed discussion on the design of event-triggered SMC can be found in [4] and references therein. However, the works [5]–[7] fail to assure closed-loop system robustness in the presence of the system perturbations.

The event-triggered control design presented in [8] guarantees the robust performance of DMR with matched bounded disturbances, whereas it suffers from limitations such as: a) the need for an upper bound of the unknown matched disturbance, b) singularity issues and c) the controller switching gain using ad hoc design introduces chattering affecting the actuator

health. An event-triggered super-twisting controller for DMR [9], [10] is introduced under the assumption that the first time derivative of the disturbance is known, which is the bottleneck.

Since SMC designs are unsuitable when the uncertainty bound is unknown, an NN-based robust controller design was formulated with the aim of reducing the chattering [11] by estimating the non-linearity using a NN and designing an adaptive law such that the overall system is stable.

To guarantee the convergence of the adaptive NN weights, the input signals must satisfy a stringent persistence of excitation (PE) condition [12] which can be relaxed through σ-modification [13] or ϵ- modification [14]. In contrast, it was shown that measured and recorded data during the operation of an adaptive controller improves the performance [15] without needed PE. An exhaustive investigation of adaptive controller design with current and recorded data, referred to as concurrent learning, can be found in [16], [17]. The design of adaptive NN-based SMC with concurrent learning is not reported.

In this work, it is established that the design of NN-based SMC with concurrent learning will improve the convergence of the estimation error, which in turn will reduce the discontinuous gain requirement to ensure robust stability. The event-triggered adaptive SMC design with NN [18] for a class of single-input single-output systems with matched disturbance cannot be directly employed for DMR as it introduces controller singularity. Also, the design is restricted to the control of systems with matched perturbation.

To relax the controller singularity and address the unmatched uncertainty, in [19], the authors proposed a continuous time sliding mode discontinuous control for DMR. However, the knowledge of the upper bound of the disturbance to implement the SMC is needed [8], [19] and a continuous-time controller design is not preferred for networked DMRs with constrained communication and computational resources.

In order to overcome the above-mentioned limitations, the design of an event-triggered adaptive SMC is introduced with a radial basis function NN (RBFNN) for uncertain MIMO networked DMR to track the desired trajectory. The RBFNN estimates the unmatched uncertainty without needing its bound and the adaptive SMC is used to compensate for the mismatch in estimation which reduces chattering in comparison to [8], [19]. The design is further improved by using concurrent learning for updating the RBFNN weights.

We summarized the major contributions of the paper as:

- A novel RBFNN SMC design in the event-triggered framework is introduced, for DMR with both matched and unmatched uncertainty, so as to overcome the controller singularity over [8] via the nonlinear sliding surface. The sliding surface helps DMR to track both positive and negative reference velocities unlike [19].
- Our work combined the concurrent learning-based adaptation with the event-based adaptive SMC to relax the PE condition and guarantee an enhanced convergence of RBFNN weight estimation error in comparison to traditional NN-based SMC design in [11]
- A suitable triggering condition is derived using weighted absolute of state error vector to ensure practical stability, thus avoiding Zeno behaviour unlike ET-SMC [4].

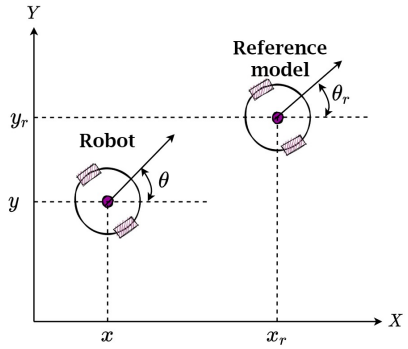

Fig. 1. A DMR tracking a reference model in Cartesian coordinates.

- Finally, the proposed event-triggered concurrent learning-based neuro-adaptive SMC law is verified on hardware demonstrating a reduction in the number of control updates and chattering.

II. Problem Formulation

Consider a DMR in Fig. 1 where it has to track a reference trajectory of a virtual DMR. The mobile robot is considered to be in the XY-plane whose position is expressed with the triplet (x, y, θ) where x, y are the coordinates in X-axis and Y-axis and θ is the orientation angle between the heading direction which is measured counter-clockwise with respect to the X-axis. The aim is to maneuver the DMR such that it can follow a reference trajectory generated by a virtual reference model whose position is represented by the triplet (x_r, y_r, θ_r) in the same XY-plane. The unperturbed kinematic model of DMR [19] is given by

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}, \tag{1}$$

where $\boldsymbol{q} = \begin{bmatrix}x & y & \theta\end{bmatrix}^\top$ are the system states, v and ω are linear and angular velocities driving the DMR.

The non-ideal environmental conditions may lead to skidding motion which violates the nonholonomic constraint. Assuming a disturbance μ, the nonholonomic constraint can be introduced as

$$\dot{x}\sin\theta - \dot{y}\cos\theta = \mu. \tag{2}$$

The input velocities to the kinematic model is imperfect [20] which can be modeled as perturbations in the input velocities as $v + \delta v$ and $\omega + \delta\omega$, where δv and $\delta\omega$ represent the perturbation terms.

Considering (2) and the imperfect velocity tracking, the perturbed kinematic model of DMR can be represented as

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\left(\begin{bmatrix}v\\ \omega\end{bmatrix} + \boldsymbol{\psi_m}\right) + \boldsymbol{\psi_u}, \tag{3}$$

where $\boldsymbol{\psi_m}$ is the matched uncertainty and $\boldsymbol{\psi_u}$ is the unmatched uncertainty. The unmatched uncertainty has the form $\boldsymbol{\psi_u} = \begin{bmatrix}\mu\sin\theta & -\mu\cos\theta & 0\end{bmatrix}^\top$ which arises due to the violation of the nonholonomic constraint (2) and the matched

uncertainty $\boldsymbol{\psi_m} = \begin{bmatrix}\delta v & \delta\omega\end{bmatrix}^\top$ is induced due to the imperfect velocity input. The model of the virtual reference DMR is given by

$$\begin{bmatrix}\dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r\end{bmatrix} = \begin{bmatrix}\cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1\end{bmatrix}\begin{bmatrix}v_r \\ \omega_r\end{bmatrix}, \tag{4}$$

where $\boldsymbol{q_r} = \begin{bmatrix}x_r & y_r & \theta_r\end{bmatrix}^\top$ are the reference states generated from the reference linear velocity v_r and angular velocity ω_r. The reference model has the same dynamics as that of the unperturbed model of the system (1). To achieve perfect tracking, a robust control law has to be designed such that $\|\boldsymbol{q} - \boldsymbol{q_r}\|$ decays to zero in the presence of uncertainties. The error in posture between the reference model in the coordinates of the DMR can be represented using a transformation as [5]

$$\boldsymbol{e} = \begin{bmatrix}x_e & y_e & \theta_e\end{bmatrix}^\top = \boldsymbol{T}(\boldsymbol{q})(\boldsymbol{q_r} - \boldsymbol{q}) = \bar{\boldsymbol{T}}(\boldsymbol{q}) \tag{5}$$

where $\boldsymbol{T}(\boldsymbol{q}) = \begin{bmatrix}\cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1\end{bmatrix}$ (refer to [21] for more details). Taking the time derivative of e and using (3), (4) and (5), the error dynamics can be expressed as

$$\dot{\boldsymbol{e}} = \boldsymbol{F}(\boldsymbol{e}) + \boldsymbol{G}(\boldsymbol{e})\left(\boldsymbol{u} + \boldsymbol{\varphi}\right) + \boldsymbol{\xi}, \tag{6}$$

where $\boldsymbol{F}(\boldsymbol{e}) := \begin{bmatrix}v_r\cos\theta_e \\ v_r\sin\theta_e \\ \omega_r\end{bmatrix}$, $\boldsymbol{G}(\boldsymbol{e}) := \begin{bmatrix}-1 & y_e \\ 0 & -x_e \\ 0 & -1\end{bmatrix}$ and $\boldsymbol{u} := \begin{bmatrix} v & \omega \end{bmatrix}^\top$. The matched disturbance in error coordinate is represented as $\boldsymbol{\varphi} = \begin{bmatrix}\varphi_v & \varphi_\omega\end{bmatrix}^\top$ and the unmatched disturbance in error coordinate is represented as $\boldsymbol{\xi} = \frac{\partial}{\partial \boldsymbol{q}}\bar{\boldsymbol{T}}(\boldsymbol{q})\psi_u|_{\boldsymbol{q}=\bar{\boldsymbol{T}}^{-1}(\boldsymbol{e})} = \begin{bmatrix}0 & \mu & 0\end{bmatrix}^\top$.

The problem of tracking is now addressed by designing a control law for stabilizing the error $\boldsymbol{e}$. As, the transformation matrix $\boldsymbol{T}(\boldsymbol{q})$ is non-singular with $\boldsymbol{T}(\boldsymbol{q})^{-1}$ bounded then it can be assured that if the error dynamics can be stabilized, the DMR will achieve a perfect tracking as $\|\boldsymbol{q} - \boldsymbol{q_r}\| \leq \|\boldsymbol{T}(\boldsymbol{q})^{-1}\|\|\boldsymbol{e}\|$.

The following assumptions are true for the unmatched uncertainty:

Assumption 1: The unmatched uncertainty acting on the kinematics of the DMR is bounded as $\|\mu\| \leq \left(\sin^2(\theta_e) + 0.1|x_e y_e|\right)\sqrt{x_e^2 + \chi}$, where χ is a positive constant [19].

Remark 1: The unmatched perturbation in the channel of θ is zero due to the fact that the nonholonomic constraint (2) doesn't involve $\dot{\theta}$. The unmatched uncertainty in the dynamics of x_e is equal to zero and thus it acts only in the dynamics of y_e due to the diffeomorphism property [22] of $\bar{\boldsymbol{T}}(\boldsymbol{q})$.

In this work, the DMR is connected to the controller over a shared wireless communication network as shown in Fig. 2. The communication network is considered to be ideal and free from imperfections such as packet loss and delays. The DMR is operating in a dynamic environment and is subjected to perturbations, where it is required to track a reference trajectory. The upper bound of the perturbations is unknown and a resource-aware robust controller needs to be designed to accomplish the tracking objective with the reduced number of control computations and the over-usages of the network. The proposed control scheme to achieve these objectives is depicted in Fig. 2. The steps to design the proposed control methodologies are described in the subsequent sections. As an immediate step, an adaptive SMC in the continuous domain is designed in the following section, under the assumption that the upper bound of the disturbances is not known.

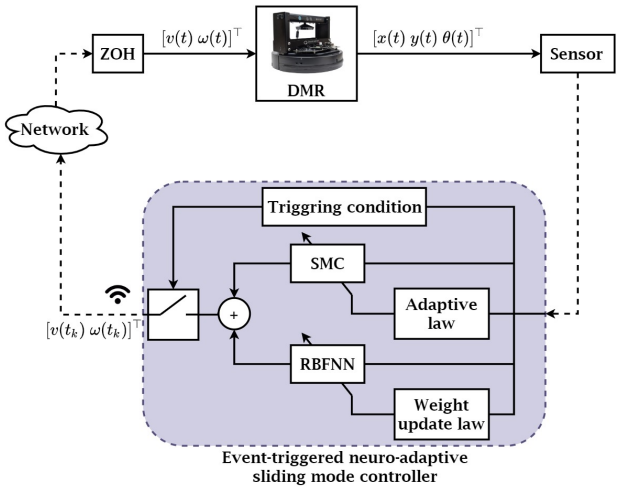

Fig. 2. Block diagram of the networked DMR.

III. Traditional Neuro-adaptive SMC Design

This section is devoted to designing the adaptive SMC with RBFNN to approximate the unknown disturbances acting on DMR. To design the SMC, a nonlinear sliding surface is used to avoid the singularity in control. The following assumptions are introduced to facilitate the controller design.

Assumption 2: The reference velocities v_r and ω_r are bounded and satisfy $0 < |v_r| \leq v_M$ and $|\omega_r| \leq \omega_M$. Also the first time derivative of v_r is bounded as $|\dot{v}_r| \leq \varsigma_v$.

Assumption 3: The maximum linear velocity satisfies the following inequality $v_M > \sqrt{\chi}$ [19].

Assumption 4: The system error trajectories $\begin{bmatrix}x_e & y_e & \theta_e\end{bmatrix}^\top$ are contained in the domain $\mathfrak{D} \subset \mathbb{R}^3$ given by

$$\mathfrak{D} = \left\{\boldsymbol{e} \in \mathbb{R}^3 : |x_e| < x_M, |y_e| < y_M, |\theta_e| \leq \pi\right\}. \tag{7}$$

Typically, in the design of the SMC, we proceed by designing a stable sliding manifold followed by the design of a control law such that the error trajectories can reach the sliding manifold in finite time. The choice of the sliding variable must be done such that once the sliding variable is zero, the reduced order dynamics are stable. Based on the existing work [19], a modified sliding variable is chosen as

$$\boldsymbol{s} = \begin{bmatrix}s_1 \\ s_2\end{bmatrix} = \begin{bmatrix}k_1 x_e \\ k_2\theta_e + \frac{v_r y_e}{\sqrt{c + x_e^2 + y_e^2}}\end{bmatrix}, \tag{8}$$

where $k_1 > 0$, $k_2 > \frac{v_M}{\pi}$ and $c > 0$ are the design parameters. The dynamics of the sliding variable along the trajectories of

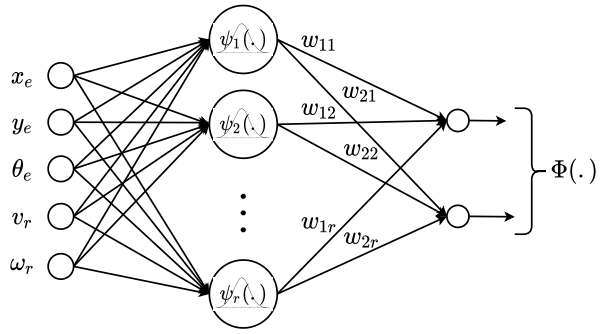

Fig. 3. Schematic representation of RBFNN

(6) is given by

$$\begin{aligned}\dot{\boldsymbol{s}} &= \begin{bmatrix} k_1\dot{x}_e & k_2\dot{\theta}_e + \frac{d}{dt}\left\{\frac{v_r y_e}{\sqrt{c+x_e^2+y_e^2}}\right\}\end{bmatrix}^\top \\ &= \mathcal{A}(\boldsymbol{e}) + \mathcal{B}(\boldsymbol{e})u + \Phi(\boldsymbol{\varphi},\boldsymbol{\xi}), \end{aligned} \tag{9}$$

where $\mathcal{A}(\boldsymbol{e}) := \begin{bmatrix} k_1 v_r \cos\theta_e \\ \frac{(c+x_e^2+y_e^2)y_e\dot{v}_r + (c+x_e^2)v_r^2\sin\theta_e - v_r^2 x_e y_e\cos\theta_e}{(c+x_e^2+y_e^2)^{1.5}} + k_2\omega_r \end{bmatrix}$, $\mathcal{B}(\boldsymbol{e}) := \begin{bmatrix} -k_1 & k_1 y_e \\ \frac{v_r x_e y_e}{(c+x_e^2+y_e^2)^{1.5}} & -\frac{v_r x_e}{\sqrt{c+x_e^2+y_e^2}} - k_2 \end{bmatrix}$, and $\Phi(\boldsymbol{\varphi},\boldsymbol{\xi})$ is the total uncertainty acting on the system. The control law which can drive the system to the sliding manifold is given by

$$\boldsymbol{u} = -\mathcal{B}^{-1}(\boldsymbol{e})\Big(\mathcal{A}(\boldsymbol{e}) + \bar{K}\mathrm{sgn}(\boldsymbol{s})\Big), \tag{10}$$

where $\bar{K} > \|\Phi(\boldsymbol{\varphi},\boldsymbol{\xi})\|$ is the switching gain to ensure finite time reaching of the system to the sliding manifold

$$\mathbb{S} = \Big\{\boldsymbol{e} \in \mathfrak{D} : s_1 = 0,\ s_2 = 0\Big\}, \tag{11}$$

which in turn stabilizes the error asymptotically.

Remark 2: The sliding surface is designed such that the stability of the reduced order dynamics of y_e and θ_e can be ensured. Also, it is worth mentioning that the design of the sliding surface for the DMR error dynamics given by (6) is challenging as the linear sliding surface introduces singularity in control [8]. Also, the proposed sliding surface in this work can track positive and negative reference linear velocity v_r which was not addressed in [19].

Remark 3: The control law (10) uses a discontinuous injection term with high gain to nullify the effect of the total uncertainty Φ which may amplify chattering. Alternatively, it is better to design a control scheme where the estimation of Φ will be used to nullify its effect and the resultant estimation error can be taken care of using a robust injection term with lower gain.

A RBFNN can be used to approximate an unknown function with sufficient accuracy [23]. The RBFNN is a single-layer neural network which has only one hidden layer comprising radial basis functions and the associated weights as shown in Fig. 3. The output of the RBFNN is the weighted sum of the output of these basis functions which is used to approximate the unknown disturbances $\Phi(\boldsymbol{\varphi},\boldsymbol{\xi})$. Mathematically, the uncertainty function $\Phi(\boldsymbol{\varphi},\boldsymbol{\xi})$ can be represented with the help of RBFNN [24] as

$$\Phi(\boldsymbol{\varphi},\boldsymbol{\xi}) = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1r} \\ w_{21} & w_{22} & \cdots & w_{2r} \end{bmatrix} \begin{bmatrix} \psi_1(\boldsymbol{z}) \\ \psi_2(\boldsymbol{z}) \\ \vdots \\ \psi_r(\boldsymbol{z}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1(\boldsymbol{z}) \\ \boldsymbol{\epsilon}_2(\boldsymbol{z}) \end{bmatrix},$$

$$\Phi(\boldsymbol{\varphi},\boldsymbol{\xi}) = \mathcal{W}^\top\Psi(\boldsymbol{z}) + \boldsymbol{\epsilon}(\boldsymbol{z}), \tag{12}$$

where $\mathcal{W} \in \mathbb{R}^{r\times 2}$ represents the ideal weighting matrix and the vector $\Psi(\boldsymbol{z}) \in \mathbb{R}^r$ is the basis function vector, $\boldsymbol{z} = \begin{bmatrix} x_e & y_e & \theta_e & v_r & \omega_r \end{bmatrix}^\top$ respectively and $\boldsymbol{\epsilon}(\boldsymbol{z}) \in \mathbb{R}^r$ is the function reconstruction error. The ideal weighting matrix is represented by

$$\mathcal{W} = \arg\min_{\mathcal{W}} \|\Phi(\boldsymbol{\varphi},\boldsymbol{\xi}) - \mathcal{W}^\top\Psi(\boldsymbol{z})\|. \tag{13}$$

The estimation of the uncertainty function can be represented by $\hat{\Phi}(\boldsymbol{\varphi},\boldsymbol{\xi}) = \hat{\mathcal{W}}^\top\Psi(\boldsymbol{z})$, where $\hat{\mathcal{W}}$ is the estimation of $\mathcal{W}$. In this article, the basis function is chosen to Gaussian expressed as $\psi_i(\boldsymbol{z}) = \exp\left(\frac{\|\boldsymbol{z}-m_i\|}{2v_i}\right)$ with m_i and v_i being the mean and variance respectively. The weight estimation error is defined as $\tilde{\mathcal{W}} = \hat{\mathcal{W}} - \mathcal{W}$. To ensure the convergence of the estimation error $\tilde{\mathcal{W}}$ to a arbitrarily small value, an adaptive law is designed to update $\hat{\mathcal{W}}$ subsequently. The following assumptions hold true for the RBFNN-based function approximation.

Assumption 5: The basis function $\Psi(\cdot)$ is bounded satisfying $\|\Psi(\cdot)\| \le \Psi_M$ [25].

Assumption 6: The reconstruction error in (12) is always upper bounded by $\|\boldsymbol{\epsilon}(\boldsymbol{z})\| \le \varrho$ [18].

Remark 4: The choice of the structure of RBFNN for implementing the controller needs to be decided by the designer. One may choose a dense number of neurons in the hidden layer for higher accuracy in the function approximation. But the greater number of neurons in the hidden layer will surely give rise to higher parameters of the RBFNN to be estimated incurring greater computational load. The section of the Gaussian function's center/ means is to be decided such that the input to the RBFNN is close to the center of the function. The variance represents the width of a Gaussian function, which must be selected such that the range of the inputs can excite the function (For more details see [26]).

The neuro-adaptive SMC law is given by

$$\boldsymbol{u} = -\mathcal{B}^{-1}(\boldsymbol{e})\Big(\mathcal{A}(\boldsymbol{e}) + \hat{\mathcal{W}}^\top\Psi(\boldsymbol{z}) + (\hat{\varrho} + \bar{K})\mathrm{sgn}(\boldsymbol{s})\Big), \tag{14}$$

where $\hat{\mathcal{W}}^\top\Psi(\boldsymbol{z})$ is the output of the RBFNN and $\hat{\varrho}$ is the estimation of ϱ. The adaptive weight update law is given by

$$\dot{\hat{\mathcal{W}}} = \Gamma_1\Psi(\boldsymbol{z})\boldsymbol{s}^\top, \tag{15}$$

where $\Gamma_1 > 0$ is the associated learning diagonal matrix and the $\hat{\varrho}$ is updated adaptively with

$$\dot{\hat{\varrho}} = \gamma_2\|\boldsymbol{s}\|, \tag{16}$$

where γ_2 is the associated learning rate. The adaptive laws are incorporated such that the unknown parameters of the RBFNN and the controller can be estimated online. The adaptive law

(16) ensures the convergence of the estimation error $\tilde{\varrho} := \varrho - \hat{\varrho}$ which is established in the following Theorem.

Theorem 1: Consider the error dynamics (6) with the sliding variable (8), then the RBFNN-based SMC scheme (14) with the adaptive laws (15) and (16) satisfying the switching gain as $\bar{K} \geq \eta_1$, with $\eta_1 > 0$, drives the error trajectories to the sliding manifold (11) in finite time and stabilizes the error dynamics asymptotically.

Proof: Refer to the Appendix ∎

Remark 5: From the result of Theorem 1, it can be assured that the RBFNN-based control law (14) guarantees the convergence of the sliding variables $s_1 = s_2 = 0$. Once this is established, using further analysis, it can be proved that the error in x, y and θ coordinates are convergent to zero. Also, we can observe that with the inclusion of the RBFNN in the controller with the adaptive laws (15) and (16), the design is free from the knowledge of the bound of uncertainty.

Remark 6: Recall the dynamics of $\boldsymbol{s}$ from (9), with the control law (14), which can be expressed as $\dot{\boldsymbol{s}} = \Phi - \hat{\Phi} - \bar{K}\text{sgn}(\boldsymbol{s})$. The mismatch between Φ and $\hat{\Phi}$, is compensated using $\bar{K}\text{sgn}(\boldsymbol{s})$. As the accuracy of estimation is not guaranteed by the online estimation law (15), a greater magnitude of $\bar{K}$ is required for the mismatch in approximation. This can be relaxed by improving the accuracy of estimation.

The RBFNN-based adaptive SMC law developed in (14) for the trajectory control of networked DMR has two main shortcomings. Firstly, the networked DMR will be controlled using embedded hardware with periodic sampling resulting in large quantities of data through the network causing congestion leading to a drop in packets (either sensor data or control input) and random delays–both will create stability issues for control. Moreover, the lifespan of the battery-powered DMR is reduced.

Secondly, the objective with which the RBFNN was introduced, is not completely fulfilled as the ultimate bounds of the estimation error are not ensured. The larger mismatch in $\tilde{\Phi}$, leads to the requirement of a larger amplitude of discontinuous control to ensure robust tracking performance, which is likely to induce chattering. To overcome the first shortcoming, control with event-trigger sampling is introduced to reduce the network usage thus lowering packet losses and delays. Further, an adaptive weight update law using historically stacked experience data will be employed for enhanced estimation. The next section discusses the proposed controller design.

IV. Event-Triggered Neuro-Adaptive SMC Design

In event triggered design, the control is updated at time $t = t_k$ where $k \in \mathbb{N}$ if and only if the triggering condition is violated at $t = t_k$ and the control is computed as $u(t) = u(t_k)$; for all $t \in [t_k; t_{k+1})$. For DMR, the sampled state information is denoted as $\check{\boldsymbol{e}} = \begin{bmatrix}\check{x}_e & \check{y}_e & \check{\theta}_e\end{bmatrix}^\top$, the sampled values of reference input is denoted by $\check{v}_r$, $\check{\omega}_r$ and the sampled information of the sliding variable is $\check{\boldsymbol{s}}$. The control law (14) can be rewritten in terms of the sampled variable as

$$\boldsymbol{u} = -\mathcal{B}^{-1}(\check{\boldsymbol{e}})\Big(\mathcal{A}(\check{\boldsymbol{e}}) + \hat{\mathcal{W}}^\top(t_k)\Psi(\check{\boldsymbol{z}}) + (\hat{\varrho}(t_k) + K)\text{sgn}(\check{\boldsymbol{s}})\Big), \tag{17}$$

where K is the switching gain different from $\bar{K}$. This gain must be designed such that the stability is retained by the closed-loop system with the sampled data control law (17). This control is updated by the aperiodic triggering sequence $\{t_k\}_{k\in\mathbb{Z}_{\geq 0}}$ generated by a suitable triggering mechanism. The control is updated at instance t_k and applied to the actuator of the DMR through a zero-order hold. This constant actuation signal is applied to the DMR until any further sampling occurs and the plant evolves in an open loop fashion during the time interval $[t_k, t_{k+1})$. The triggering mechanism is designed using the information of the current states and the last sampled value. As the realization of the scheme will be on the digital platform, it has to be ensured that the sampling frequency is finite. This is done by ensuring the inter-event execution time defined by $\Delta T_k := t_{k+1} - t_k$ is nonzero and positive, i.e., $\Delta T_k > 0$.

Remark 7: It is important to notice that the closed-loop stability of the event-triggered system cannot be established using the control law (14) with the sampled state information. The modified control law is introduced as (17), where a new gain K is to be designed to ensure the stability of sampled data closed loop system.

We define

$$\lambda(t) = c_1|x_e - \check{x}_e| + c_2|y_e - \check{y}_e| + c_3|\theta_e - \check{\theta}_e|, \tag{18}$$

where the choice of the weighting constants c_1, c_2 and c_3 must satisfy

$$\begin{aligned} c_1 &\geq \max(k_1 + Lv_M, 1), \\ c_2 &\geq \max(k_2, 1), \\ c_3 &\geq \max(Lv_M, 1). \end{aligned} \tag{19}$$

The function $\lambda(t)$ will be utilized to design the triggering condition which evaluates the tracking error of DMR to determine the next sampling instance and whenever the variable $\lambda(t)$ grows beyond some threshold an event will be generated. The triggering condition will be introduced in Theorem 2.

To improve the accuracy in estimation of the uncertainty Φ, a data-driven learning scheme is adopted. The formalism of using stored data along with current data concurrently in adaptive control is termed concurrent learning. The data is recorded selectively such that it maintains a sparse representation of the operating region of the system keeping a good richness of data. It is established in [17], that concurrent learning for structured uncertainty is an effective method as it renders exponential convergence of parameter estimation error while relaxing the stringent PE condition. For an unstructured uncertainty where an RBFNN can be used to approximate the unknown function, it guarantees the convergence of the estimated weights. This learning can be introduced by modifying the weight update (15) as

$$\dot{\hat{\mathcal{W}}} = \begin{cases} \Gamma_1\Psi(\boldsymbol{z})\boldsymbol{s}^\top - \sum_{i=1}^{p_r}\Gamma_1\Psi(\boldsymbol{z}_i)\boldsymbol{\epsilon}_i^\top - \sigma\hat{W} \\ \text{if rank}(\mathcal{Z}) < r \\ \Gamma_1\Psi(\boldsymbol{z})\boldsymbol{s}^\top - \sum_{i=1}^{p_r}\Gamma_1\Psi(\boldsymbol{z}_i)\boldsymbol{\epsilon}_i^\top \\ \text{otherwise} \end{cases} \tag{20}$$

where $\boldsymbol{\epsilon}_i = \hat{\mathcal{W}}^\top\Psi(\boldsymbol{z}_i) - \Phi(\boldsymbol{z}_i)$ and $\sigma > 0$ is the σ-modification term to ensure boundedness RBFNN weights. The mismatch

in estimation $\boldsymbol{\epsilon}_i$ can be expressed as

$$\boldsymbol{\epsilon}_i = \tilde{\mathcal{W}}^\top \Psi(\boldsymbol{z}_i) - \boldsymbol{\varepsilon}(\boldsymbol{z}_i). \tag{21}$$

The estimation law (20) is designed using online and stored data such that the accuracy in estimation increases, causing $\tilde{\mathcal{W}}$ to be very small. The experience data is stored in the history stack matrix $\mathcal{Z}$ as $\mathcal{Z} = [\Psi(z_1)\ \Psi(z_2)\ \cdots\ \Psi(z_{p_r})]$ where p_r is the index of last stored data. The corresponding $\boldsymbol{\epsilon}_i$ is stored in a matrix $\mathcal{E} = [\boldsymbol{\epsilon}_1\ \boldsymbol{\epsilon}_2\ \cdots\ \boldsymbol{\epsilon}_{p_r}]$, where $i = 1, \cdots, p_r$. To implement the RBFNN weight update law (20), the information of $\boldsymbol{\varepsilon}_i$ is required. Recall the dynamics of the sliding variable (9) from where $\Phi(\boldsymbol{z}_i)$ can be calculated by estimating $\dot{\boldsymbol{s}}$. Thus $\Phi(\boldsymbol{z}_i) = \dot{\boldsymbol{s}}_i - \mathcal{A}(\boldsymbol{e}_i) - \mathcal{B}(\boldsymbol{e}_i)\boldsymbol{u}_i$. The $\dot{\boldsymbol{s}}$ can be calculated using fixed point smoothing along with a backward Kalman filter (see [17]). In this work to estimate the $\dot{\boldsymbol{s}}$, a robust exact differentiator toolbox [27] is used. The selection of the data points for the recording is crucial for learning. In this regard, the SVD (singular value decomposition) maximization-based data recording algorithm is adopted [28]. The matrix $\mathcal{Z}$ will be rank deficient during the initial period of data recording in the history stack, which causes the weight update law to act with the σ-modification term. The dynamics of the RBFNN weight estimation error during the rank deficient condition can be calculated using (20) and (21) as

$$\begin{aligned}\dot{\tilde{\mathcal{W}}}(t) = &-\Gamma_1 \left(\sum_{i=1}^{p_r} \Psi(\boldsymbol{z}_i)\Psi^\top(\boldsymbol{z}_i) + \sigma I \right) \tilde{\mathcal{W}}(t) \\ &- \Gamma_1 \sum_{i=1}^{pr} \Psi(\boldsymbol{z}_i)\boldsymbol{\varepsilon}^\top(\boldsymbol{z}_i) + \Gamma_1 \Psi(\boldsymbol{z})\boldsymbol{s}^\top - \sigma \mathcal{W}. \end{aligned} \tag{22}$$

As the history stack collects more data at some time $t = t_r \in (0, \infty)$, the $\mathcal{Z}$ becomes the full rank matrix. This implies that the recorded data is sufficiently rich to ensure enhanced learning, which leads to better estimation of RBFNN weights. Due to the full rank of $\mathcal{Z}$, the RBFNN weight estimation error dynamics can be written using (20) and (21) as

$$\begin{aligned}\dot{\tilde{\mathcal{W}}}(t) = &-\Gamma_1 \left(\sum_{i=1}^{p_r} \Psi(\boldsymbol{z}_i)\Psi^\top(\boldsymbol{z}_i) \right) \tilde{\mathcal{W}}(t) \\ &- \Gamma_1 \sum_{i=1}^{pr} \Psi(\boldsymbol{z}_i)\boldsymbol{\varepsilon}^\top(\boldsymbol{z}_i) + \Gamma_1 \Psi(\boldsymbol{z})\boldsymbol{s}^\top. \end{aligned} \tag{23}$$

At this point, it is worth mentioning that generally, the event-triggered SMC can ensure the practical sliding motion (PSM) in contrast to the ideal sliding motion which can be achieved with the continuous control law. This notion of PSM can also be found in [29], [30].

With these preliminaries, we now state our main result in the next Theorem. To facilitate a simpler proof, the following assumption is considered.

Assumption 7: The variation in the matrix $\mathcal{B}(\boldsymbol{e})$ is negligible in the interval $t \in [t_k, t_{k+1})$.

Note : *We exclusively use the argument t with the continuous variables $\hat{\mathcal{W}}$ and $\hat{\varrho}$ to distinguish them from their sampled values $\hat{\mathcal{W}}(t_k)$ and $\hat{\varrho}(t_k)$.*

Theorem 2: The error dynamics (6) with the sliding variable (8) and the event-triggered control law (17) satisfying the event condition defined as

$$\lambda(t) \leq \alpha \tag{24}$$

where $\alpha > 0$ is an event design parameter, will achieve PSM and the error trajectories will be confined in the set given by

$$\mathfrak{B}_s = \{e \in \Omega : \|\boldsymbol{s}\| \leq \alpha\} \tag{25}$$

with the adaptive weight update law for the RBFNN (20) and with the adaptive law to estimate the upper bound of reconstruction error as

$$\dot{\hat{\varrho}}(t) = \begin{cases} \gamma_2 \|\boldsymbol{s}\| & \text{if } \ \boldsymbol{s} \notin \mathfrak{B}_s \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

provided, the discontinuous injection gain K is selected as $K > L_{\mathcal{A}}\alpha + L_{\mathcal{W}} + L_{\varrho} + \eta$, where η is a positive scalar; $L_{\mathcal{A}}(\boldsymbol{e})$ is the Lipchitz constant of $\mathcal{A}$; $L_{\mathcal{W}}$ and L_{ϱ} are the bounds associated with $\tilde{\mathcal{W}}^\top \Psi(\boldsymbol{z})$ and $\hat{\varrho}$. The derivation of these bounds is discussed in the Appendix section. Additionally, the RBFNN weight estimation error is contained in the set given by

$$\mathfrak{B}_{\tilde{\mathcal{W}}} = \left\{ \mathcal{W} \in \mathbb{R}^{2\times r} : \|\tilde{W}(t)\| \leq \max(r_1, r_2) \right\} \tag{27}$$

where $r_1 = \frac{\sigma\|\mathcal{W}\| + r p_r \Psi_M \|\boldsymbol{\varepsilon}(\boldsymbol{z})\|}{\sigma}$ $r_2 = \frac{r p_r \Psi_M \|\boldsymbol{\varepsilon}(\boldsymbol{z})\|}{\zeta_{min}(\Xi)}$ and $\Xi = \sum_{i=1}^{p_r} \Psi(\boldsymbol{z}_i)\Psi^\top(\boldsymbol{z}_i) = \mathcal{Z}\mathcal{Z}^\top$.

Proof: Refer to the Appendix ■

Remark 8: Using the results of Theorem 2, one may observe that the proposed control law, including the concurrent learning, RBFNN and event-triggered scheme with adaptive SMC relaxes the system from periodic sampling while maintaining robust tracking performance. Also, the sliding variable achieves PSM, which results in $\|\boldsymbol{s}\|$ being bounded by α. Due to the inclusion of concurrent learning, the RBFNN weight estimation error now is guaranteed to be upper bounded within a ball of radius $\max(r_1, r_2)$. As the boundedness of estimation is now established, this will result in a lower requirement of discontinuous gain for achieving stability in the presence of uncertainty.

Remark 9: The triggering condition is designed in a non-conventional way based on the function $\lambda(t)$ and not using the measurement error e as in traditional ET-SMC [4] since we use a nonlinear sliding surface (8). The coefficients c_1, c_2 and c_3 in (18) are chosen as (19) such that $\lambda(t) \geq \max(\|\boldsymbol{s} - \check{\boldsymbol{s}}\|, \|\boldsymbol{e} - \check{\boldsymbol{e}}\|)$. Due to the event-condition (24), it is assured that $\lambda(t)$ is always bounded by α, which also ensures the boundedness of $\|\boldsymbol{s} - \check{\boldsymbol{s}}\|$ and $\|\boldsymbol{e} - \check{\boldsymbol{e}}\|$.

Using Theorem 2, it can be ensured that the sliding variables s_1 and s_2 will reach the set $\mathfrak{B}_s$ in finite time, which means that the error x_e will reach it's bound in finite time and remains uniformly ultimately bounded (UUB) thereafter. The boundedness of y_e and θ_e are introduced in the following proposition.

Proposition 1: The event-based control (17) with the triggering condition (24) drives the error trajectories of (6) to the vicinity of the sliding manifold (11) and subsequently, the error trajectories will achieve uniform boundedness in the set

defined by

$$\mathfrak{B}_e = \left\{ \boldsymbol{e} \in \mathfrak{D} : |x_e| \leq \frac{\alpha}{k_1}, |y_e| \leq \frac{\alpha}{c_2}, |\theta_e| \leq \frac{\alpha}{c_3} \right\}. \quad (28)$$

Proof: From Theorem 2, the PSM is achieved by the error dynamics which ensures the boundedness of s_1 and s_2. Using (8) and (27) it can be written as

$$|s_1| \leq \|\boldsymbol{s}\| \leq \alpha \Rightarrow k_1|x_e| \leq \alpha.$$

The above relation ensures the bound of x_e. To prove the stability of the tracking error y_e and θ_e, the results of Theorem 1 and 2 are used. The PSM is achieved by the error dynamics with the control law (17) stated in Theorem 2. Theorem 1 confirms that, when the error trajectories are in the vicinity of the sliding manifold, they converge towards the origin. With the combination of these two facts, it can be concluded that the vector field around the sliding manifold is attractive towards the origin. Therefore, from the Regularization Theorem [31], it can be said that the error trajectories will converge to $\mathfrak{B}_e$ in finite time with a zig-zag motion around the manifold $\mathbb{S}$. Next, when the error trajectories will try to escape the ball $\mathfrak{B}_e$, the $\lambda(t)$ grows to its maximum value α which causes the violation of triggering condition (24). At that instance of time, the control (17) will be updated and it forces the trajectories to be inside the ball. Now, if we assume that a control signal excites motion only in y_e, using (24) we can write

$$c_2|y_e - \check{y}_e| \leq \alpha \Rightarrow |y_e - \check{y}_e| \leq \frac{\alpha}{c_2}. \quad (29)$$

Using the same argument for θ_e, we have

$$c_3|\theta_e - \check{\theta}_e| \leq \alpha \Rightarrow |\theta_e - \check{\theta}_e| \leq \frac{\alpha}{c_3}. \quad (30)$$

Thus the tracking errors are finally confined in the set $\mathfrak{B}_e$ which completes the proof. ∎

The boundedness of the tracking errors of DMR is established using event-triggered neuro-adaptive SMC with concurrent learning. Next, to ensure the existence of a finite sampling rate with the proposed event condition, the minimum inter-event time ΔT_k is shown positive.

Theorem 3: Consider the function $\lambda(t)$ (18) with the tracking error dynamics (6), sliding variable (8), event based control (17) satisfying the triggering condition (24), then the inter-event execution time satisfies the following inequality.

$$\Delta T_k \geq \frac{\alpha}{\Theta(\hat{\mathcal{W}}, \hat{\varrho})}, \quad (31)$$

where $\Theta(\hat{W}, \hat{\varrho})$ is the $\sup\{\dot{\lambda}(t)\}$ whose expression is given in (51) in the Appendix section.

Proof: To find the minimum time between two events, we find the time required for $\lambda(t)$ to grow from 0 to α. Just after an event violation, at $t = t_k^+$, the function $\lambda(t) = 0$. Now in the interval $t \in [t_k,\ t_{k+1})$, $\lambda(t)$ grows due to the evolution of the states x_e, y_e and θ_e and when it attains its threshold value α, the next triggering occurs. The minimum time for the two events can be computed using the maximum growth rate of $\lambda(t)$ which is calculated in Appendix and denoted by $\Theta(\hat{W}, \hat{\varrho})$ (51). The growth of $\lambda(t)$ in the interval $t \in (t_k, t_{k+1}]$ satisfies the following inequality

$$\frac{d}{dt}\lambda(t) \leq \sup\left\{ c_1|\dot{x}_e|| + c_2|\dot{y}_e| + c_3|\dot{\theta}_e| \right\} \leq \Theta(\hat{W}, \hat{\varrho})$$

Integrating in the above inequality

$$\lambda(t_{k+1}) - \lambda(t_k^+) \leq \Theta(\hat{W}, \hat{\varrho}) \int_{t_k}^{t_{k+1}} dt$$
$$\alpha \leq \Theta(\hat{W}, \hat{\varrho})(t_{k+1} - t_k).$$

Using (51) and the above inequality, the minimum inter-event time can be expressed as (31). Thus the positive lower bound of inter-event execution is established which completes the proof. ∎

In the next section, to evaluate the performance of the proposed controller, the experimental results are presented using hardware-in-loop testing.

V. Experimental Results

In this section, the performance of the proposed event-based neuro-adaptive SMC with concurrent learning is verified to track a reference trajectory by a DMR, namely the Quanser QBot 2e. The video of the experiment conducted to implement the proposed controller on the QBot can be found in [32]. Though there exist various mobile robot platforms for hardware-in-loop (HIL) testing of control algorithms like Pionner3DX and Khepera IV, the Quanser QBot 2e gives a ready-to-use platform integrated with Simulink/MATLAB using the QUARC software [33]. The QBot has a Raspberry Pi as an onboard computer which communicates with the controlling computer using a WiFi (2.4GHz and 5Hz IEEE 802.11. b/g/n/ac wireless LAN). The hardware-in-loop testing of the controller is done by using Simulink/ MATLAB (2018b) as a platform to implement the controller. The experiment is performed to track an elliptical trajectory in the XY-plane, which can be generated with the reference velocities using

$$v_r = \left(\varpi^2 l_1^2 \sin^2(\varpi t) + \varpi^2 l_2^2 \cos^2(\varpi t)\right)^{0.5} \quad (32)$$

$$\omega_r = \left(\frac{\varpi l_2}{l_1} + \frac{\varpi l_2 \cos^2(\varpi t)}{l_1 \sin^2(\varpi t)} \right) / \left(\frac{l_2^2 \cos^2(\varpi t)}{l_1^2 \sin^2(\varpi t)} + 1 \right). \quad (33)$$

From the expression of the reference trajectories given in (32) that v_r is upper bound by $\varpi \times \max(l_1, l_2)$ and ω_r is upper pounded by $\frac{\varpi l_2}{l_1}$ (Since $a\sin^2 x + b\cos^2 x \leq \max(a, b)$ and $\frac{1+\cot^2 x}{(l_2/l_1)^2 \cot^2 x + 1} \leq 1$ for $l_2 > l_1$). Using the above arguments, one can verify that the above expressions of the reference trajectory satisfy Assumption 2. The parameters are selected as $\varpi = 0.5$, $l_1 = 0.5$ and $l_2 = 0.8$ from where the upper bound on the velocities are calculated as $v_M = 0.4$m/s and $\omega_M = 0.8$rad/s. To start with the controller design, the domain of interest is chosen arbitrarily large enough as $|x_e| \leq 1.5$m, $|y_e| \leq 1.5$m and $|\theta_e| \leq \pi$ rad to accommodate the constraint space available to conduct the experiment. The initial tracking error is set as $\begin{bmatrix} 0.2 & 0.2 & 0.2\pi \end{bmatrix}^\top$.

As per the design requirement, the parameters of the sliding surface must be chosen first. The choice of k_1, is impacting the ultimate bound of the system state x_e whereas k_2 and c govern the response of the system and also affect the

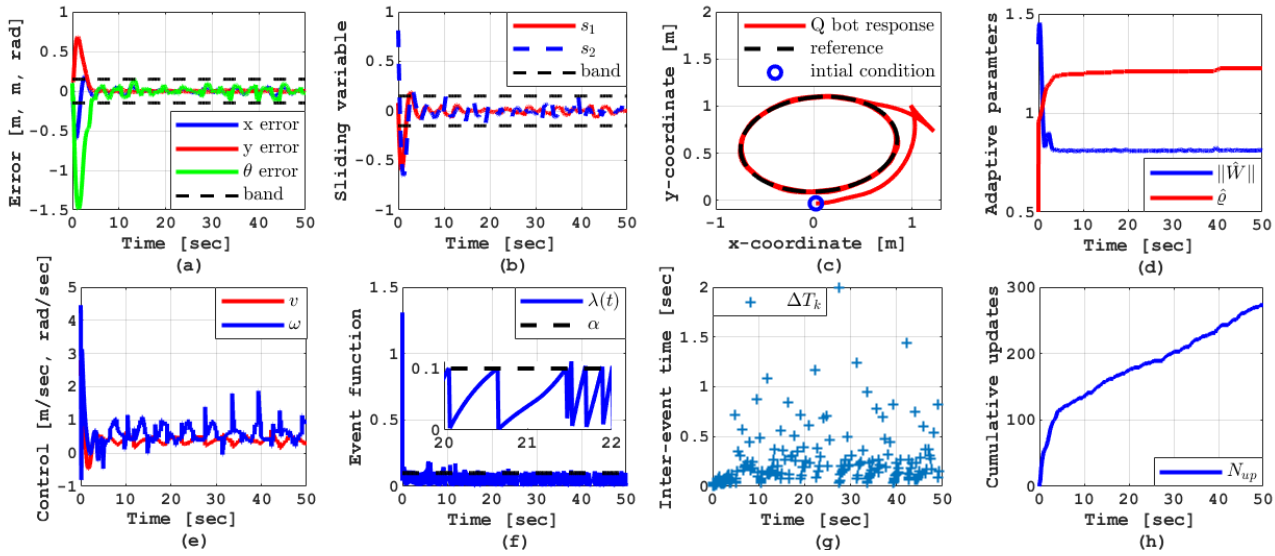

Fig. 4. Evolution of (a) tracking error (b) sliding variable (c) xy-plot of trajectory tracking (d) estimated parameters (e) control effort (f) triggering function $\lambda(t)$ (g) inter-event execution time (h) cumulative number of control updates for event-triggered RBFNN based adaptive SMC.

choice of parameters associated with the triggering function $\lambda(t)$. A greater value of c will induce sluggish behaviour in the response so, it is selected with a nominal positive value as $c = 0.1$. For simplicity, while maintaining the design requirements k_1 and k_2 were selected as unity.

Next, the weights associated with the triggering condition are selected as $c_1 = 2.1$, $c_2 = 1$ and $c_3 = 1.1$, such that the design constraint (19) is satisfied. To ensure that the PSM occurs within a band of 0.15, the event design parameter is chosen as $\alpha = 0.15$. A direct implication on this design parameter α can be observed in the inter-sampling behaviour of the closed-loop system. Too small value will generate a greater number of events while maintaining a small band of the sliding variable increasing the accuracy in the tracking and vice-versa.

Next, we proceed with the choice of parameters for the RBFNN and the adaptive gain $\hat{\varrho}$. The Gaussian basis function for the RBFNN is chosen as $\psi_i(\boldsymbol{z}) = \exp\left(\frac{\|z-m_i\|}{2v_i}\right)$, with dimension 5×1. The mean of each of the Gaussian functions are $m_i = -2,\ -1,\ 0,\ 1,\ 2$ and the variance is $v_i = 1$ where $i = 1$ to 5. This is selected to approximate the disturbance locally around the region of operation keeping the centers symmetric to the origin. The variance is selected such that each of the hidden layer neurons is spread over the sufficient region where the system may be excited. In (20), the parameters are chosen as $\Gamma_1 = 0.8I$ and $\sigma = 0.01$. The data recording is done using the SVD maximization algorithm as in [17] and the data is recorded in the $\mathcal{Z}_t$ matrix. The tuning parameter in (26) is selected as $\gamma = 0.15$. The values of adaptive learning parameters, Γ_1 and γ are decided based on multiple simulation trials. The larger values associated with adaptive laws may introduce oscillatory behaviours in the estimation of the parameters. The switching gain K is selected as 2.5 such that the gain condition in Theorem 2 is satisfied. For safety of the mechanical parts of the Qbot 2e, the $\mathrm{sgn}(\cdot)$ function is approximated in the control law (17) by a $\tanh(\cdot)$. The experimental results are shown in Fig. 4.

It can be observed from Fig 4 (a) that during the initial phase of the response the error trajectories grow due to the learning phase of the controller. As time progress, the adaptive parameters are tuned and the error trajectories start converging inside the ultimate band. The evolution of the sliding variables is shown in Fig 4 (b). Once the estimated gain $\hat{\varrho}$ is sufficient to drive the trajectories towards the manifold, the sliding variables start to converge and hit the manifold in finite time. The tracking error x_e converges as s_1 converges to the ultimate band. The convergence of y_e and θ_e is guaranteed after the convergence of s_2.

The tracking performance can be visualized from the xy-plot given in Fig. 4 (c). Initially, the Qbot manoeuvres around the reference trajectory and gradually it converges to the reference trajectory. The adaptive gain $\hat{\varrho}$ and the norm of RBFNN weights $\|\hat{\mathcal{W}}\|$ are shown in Fig. 4 (d) and found to be bounded. In Fig. 4 (e), the control signal is plotted. The evolution of the threshold function $\lambda(t)$ whose evolution governs the inter-sampling behaviour of the system is shown in Fig. 4 (f).

During the first measurement, the sampled values are set to zero. Thus the value of $\lambda(t)$ at the first iteration is the weighted difference between the initial condition and the sampled values, which is greater than the threshold value that causes the first triggering. Subsequently, further triggering occurs as $\lambda(t)$ grows to α between each of the events. The inter-event execution time is shown in Fig. 4 (g), which is bounded by a positive value.

The evolution of the total number of events occurring represented by N_{up} are shown in Fig. 4 (h), which is observed to be steep in the early stages of the simulation due to the transient in the system. Once the tracking error has settled down and the system is in a steady state, the growth of N_{up} is reduced.

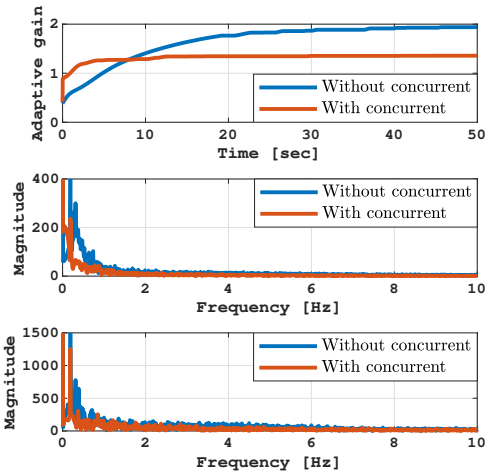

Fig. 5. Evolution of (a) the adaptive gains and frequency spectrum for the actuating signal (b) linear velocity and (c) angular velocity for implementation of the controller with and without concurrent learning.

A total number of 273 triggers occur during the 50-second simulation. The time-triggered implementation for the same with a sampling rate of 0.01-second results in 5001 control updates. Thus, the event-triggered implementation reduces the network usage by 94.54% compared to the time-triggered design.

TABLE I
EFFECT OF CONCURRENT LEARNING ON NETWORK USAGE, TRACKING ERROR AND CONTROL EFFORT

RBFNN weight update	$\mathcal{N}_{up}$	IAE_{ss}	IAC_{ss}
Without concurrent Learning	316	2.11	22.49
With concurrent Learning	273	1.38	18.72

To show the effectiveness of the proposed design, three performance indices are introduced, i.e., the total number of control updates, $\mathcal{N}_{up}$, integral of absolute error at steady state, $IAE_{ss} = \int_{\frac{1}{2}\mathcal{T}_{sim}}^{\mathcal{T}_{sim}} \|e\| dt$ and integral of absolute control effort at steady state, $IAC_{ss} = \int_{\frac{1}{2}\mathcal{T}_{sim}}^{\mathcal{T}_{sim}} \|u\| dt$, where $\mathcal{T}_{sim}$ is the total simulation time. The concurrent learning-based proposed scheme is compared with the same controller introduced in Section IV, where instead of using the concurrent learning-based law (20), the standard weight update law (15) is used.

The performance indices are tabulated in Table I for the implementation of an event-triggered controller with and without concurrent learning. It is observed that concurrent learning improves tracking performance.

Further to highlight the improvements, by the use of concurrent learning, we plot the evolution of the adaptive discontinuous gain and the frequency spectrum of the actuating signals for the two cases. The adaptive gain with and with concurrent learning are shown in Fig. 5 (a). It is easily visible that using concurrent learning the estimated discontinuous gain $\hat{\varrho}$ is lower in comparison to the controller without concurrent learning. This lower switching gain implies a reduced amount of control effort, which is responsible for the growth of the threshold function $\lambda(t)$.

As the discontinuous gain is reduced, the growth rate of $\lambda(t)$ decreases. Due to this, the number of occurrences of events will be reduced, which is reflected in Table I. The evolution of the adaptive discontinuous gains for controller implementation with and without concurrent learning are plotted in Fig. 5 (a). It is observed that the discontinuous gain requirement without concurrent learning is greater than the gain estimated in the case with concurrent learning. The higher discontinuous gain may introduce chattering in the system. The frequency versus magnitude plot for the control signal v and ω are shown in Fig. 5 (b)-(c) for the controller implementations with and without concurrent learning.

From the magnitude plot shown in Fig. 5 (b)-(c), it can be visualized that magnitude of the frequency component of the actuation signal in the lower frequency range has reduced for the controller with concurrent learning in comparison to the non-concurrent case. Also, in the higher range of the frequency, the magnitude is slightly lesser for the concurrent learning-based controller. This observation establishes that concurrent learning-based event-triggered neuro SMC is successful in reducing the chattering in the system.

Thus, it is guaranteed that the proposed control strategy can ensure improved performance in terms of tracking accuracy, reduced control effort, discounted chattering and lower usage of the communication network which is vital for a networked battery-powered mechanical device like DMR.

VI. CONCLUSION

The tracking control of a networked DMR is addressed in this paper using an event-based neuro-adaptive SMC with RBFNN and concurrent learning. To achieve the control task in the presence of disturbances and minimal usage of the network resources, the proposed design includes an event-based sampling mechanism along with an RBFNN-based sliding mode controller. A nonlinear sliding surface was adopted to avoid singularity in control law and an SMC was designed to guarantee robustness. The RBFNN was used to approximate the perturbations, which relaxes the prior knowledge of the upper bound of the perturbations to design the SMC and an adaptive gain of the controller was used to compensate for the reconstruction error. To ensure improved estimation of the uncertainty, a concurrent learning-based adaptive law is designed, which also relaxes the PE condition.

It was established that the inclusion of concurrent learning and RBFNN in the SMC scheme helped to reduce chattering by improving the accuracy of the online estimation of parameters. Moreover, the practical implementation of the event-based adaptation of the proposed control law on DMR showed that the reduced number of control computations improved tracking performance with less control effort and reduced chattering. As a future scope, the design technique proposed here using concurrent learning with event-based SMC can be further extended to control nonlinear systems like robotic manipulators, tethered satellites, unmanned aerial vehicles etc. The design is restricted here to control a single networked DMR, the proposed control scheme has the potential for solving multi-robot control problems such as leader-follower tracking problems, consensus problems and formation control.

APPENDIX

PROOF OF THEOREM 1

Proof: To prove the closed loop stability, first the finite time reaching of the error trajectories to the sliding manifold $\mathbb{S}$ is established followed by the convergence of the error trajectories. Consider a Lyapunov candidate function given by

$$V_1 = \frac{1}{2}\boldsymbol{s}^\top s + \frac{1}{2}\text{tr}(\tilde{\mathcal{W}}^\top \Gamma_1^{-1}\tilde{\mathcal{W}}) + \frac{1}{2\gamma_2}\tilde{\varrho}^2. \tag{34}$$

Evaluating the time derivative of V_1 along the trajectories of (9) with the control law (14) we have

$$\begin{aligned}\dot{V}_1 =& \boldsymbol{s}^\top\Big(\Phi(\boldsymbol{\varphi},\boldsymbol{\xi}) - \hat{\mathcal{W}}^\top\Psi(\boldsymbol{z}) - (\hat{\varrho}+\bar{K})\text{sgn}(\boldsymbol{s})\Big)\\ &+ \text{tr}(\tilde{\mathcal{W}}^\top\Gamma_1^{-1}\dot{\hat{\mathcal{W}}}) + \frac{1}{\gamma_2}\tilde{\varrho}\dot{\hat{\varrho}}.\end{aligned}$$

Substituting $\Phi(\boldsymbol{\varphi},\boldsymbol{\xi})$ from (12) in the last expression and using (15) we have

$$\begin{aligned}\dot{V}_1 =& \boldsymbol{s}^\top\Big(\boldsymbol{\epsilon}(\boldsymbol{z}) - \hat{\varrho}\text{sgn}(\boldsymbol{s}) - \tilde{\mathcal{W}}^\top\Psi(\boldsymbol{z}) - \bar{K}\text{sgn}(\boldsymbol{s})\Big)\\ &+ \text{tr}(\tilde{\mathcal{W}}^\top\Gamma_1^{-1}\dot{\hat{\mathcal{W}}}) + \frac{1}{\gamma_2}\tilde{\varrho}\dot{\hat{\varrho}},\\ \leq& \|\boldsymbol{s}\|\|\boldsymbol{\epsilon}(\boldsymbol{z})\| - \hat{\varrho}\|\boldsymbol{s}\| - \bar{K}\|\boldsymbol{s}\| + \frac{1}{\gamma_2}\tilde{\varrho}\dot{\hat{\varrho}}\\ &+ \text{tr}\Big(\tilde{\mathcal{W}}^\top(\Gamma_1^{-1}\dot{\hat{\mathcal{W}}} - \Psi(\boldsymbol{z})\boldsymbol{s}^\top)\Big),\\ =& -\bar{K}\boldsymbol{s}^\top\text{sgn}(\boldsymbol{s}) + \frac{1}{\gamma_2}\tilde{\varrho}(\dot{\hat{\varrho}} - \gamma_2\boldsymbol{s}).\end{aligned} \tag{35}$$

Substituting (16) and $\bar{K} > \eta_1$, the inequality (35) boils down to $\dot{V}_1 \leq -\eta_1\|\boldsymbol{s}\|$. Thus, the control law (14) drives error trajectories to the manifold $\mathbb{S}$ in finite time. Once the error trajectories reaches $\mathbb{S}$, we have $s_1 = s_2 = 0$ in (8) from where we have $x_e = 0$ and

$$\theta_e = -\frac{v_r y_e}{k_2\sqrt{c+y_e^2}}. \tag{36}$$

As x_e is identically zero, we can rewrite the dynamic equation of y_e from (6) as

$$\dot{y}_e = -v_r\sin\theta_e + \mu. \tag{37}$$

Substituting $x_e = 0$ in the expression of $\|\mu\|$ in Assumption 1, we have $\|\mu\| \leq \sqrt{\chi}\sin^2(\theta_e)$. Now, to establish the stability of (37) with the vanishing perturbation μ, first the stability of the unperturbed system, with $\mu = 0$ in (37) is proved and then the stability of the system with vanishing disturbance is established (see chapter 9 [22]). To prove the stability of the unperturbed system, we chose the Lyapunov candidate as $V_2 = \frac{1}{2}y_e^2$. The time derivative of V_2 along the trajectory of (37) with $\mu = 0$ is given by $\dot{V}_2 = -y_e v_r\sin\theta_e$. Substituting θ_e as in (36) we find

$$\dot{V}_2 = -y_e v_r\sin\left(\frac{v_r y_e}{k_2\sqrt{c+y_e^2}}\right). \tag{38}$$

By design $k_2 > \frac{v_M}{\pi}$, which ensures $\frac{v_r y_e}{k_2\sqrt{c+y_e^2}}$ to be bounded by π. In the region $-\pi$ to 0^- the $\sin(\cdot)$ function is negative and for π to 0^+, it is positive. As the argument of $\sin(.)$ and its coefficients have the variables y_e and x_e, the following inequality holds

$$\dot{V}_2 \leq -|y_e||v_r|\sin\left(\frac{|v_r||y_e|}{k_2\sqrt{c+y_e^2}}\right). \tag{39}$$

This means $\dot{V}_2$ can be zero only when $v_r = 0$ which can be excluded by Assumption 2. Finally, with $\left|\sin\left(\frac{|v_r||y_e|}{k_2\sqrt{c+y_e^2}}\right)\right| \leq 1$, the above inequality can be simplified as

$$\dot{V}_2 \leq -v_M|y_e|. \tag{40}$$

Now to consider the stability of the perturbed system, the time derivative of V_2 along the trajectory (37) is computed as

$$\begin{aligned}\dot{V}_2 =& -y_e v_r\sin\left(\frac{v_r y_e}{k_2\sqrt{c+y_e^2}}\right) + \sqrt{\chi}\sin^2\left(\frac{v_r y_e}{k_2\sqrt{c+y_e^2}}\right)y_e,\\ \leq& -(v_M - \sqrt{\chi})|y_e|.\end{aligned} \tag{41}$$

From Assumption 3 we have $v_M > \sqrt{\chi}$ which ensures that the trajectories of y_e converge to the origin which in turn makes θ_e also stable. Thus the proof is completed. ■

PROOF OF THEOREM 2

Proof: We start with considering the Lyapunov function as $V_3 = V_1$ (defined in (34)) in the interval $t \in [t_k; t_{k+1})$. Obtaining the time derivative of V_3 and substituting (9), (12) and (17) we get

$$\begin{aligned}\dot{V}_3 =& \boldsymbol{s}^\top\Big(\mathcal{A}(\boldsymbol{e}) - \mathcal{A}(\check{\boldsymbol{e}}) + \mathcal{W}^\top\Psi(\boldsymbol{z}) - \hat{\mathcal{W}}^\top(t)\Psi(\boldsymbol{z}) + \boldsymbol{\epsilon}(\boldsymbol{z})\\ &- \hat{\varrho}(t)\text{sgn}(\check{\boldsymbol{s}}) + \hat{\mathcal{W}}^\top(t)\Psi(\boldsymbol{z}) - \hat{\mathcal{W}}^\top(t_k)\Psi(\check{\boldsymbol{z}})\\ &+ (\hat{\varrho}(t) - \hat{\varrho}(t_k))\text{sgn}(\check{\boldsymbol{s}}) - K\text{sgn}(\check{\boldsymbol{s}})\Big)\\ &+ \text{tr}(\tilde{\mathcal{W}}^\top(t)\Gamma_1^{-1}\tilde{\mathcal{W}}^\top(t)) + \frac{1}{\gamma_2}\tilde{\varrho}(t)\dot{\hat{\varrho}}.\end{aligned} \tag{42}$$

The proof is continued by discussing it for two cases. First, when the PSM is not achieved by the error trajectories, i.e., $\boldsymbol{s} \notin \mathfrak{B}_s$ and in the second case the reachability condition is satisfied and the system achieves PSM, i.e., $\boldsymbol{s} \in \mathfrak{B}_s$.

Case 1: **The error trajectories are outside the set $\mathfrak{B}_s$:** In such a scenario, the dynamics of $\tilde{\mathcal{W}}$ depends on the rank condition of the replay buffer matrix $\mathcal{Z}$.

Suppose the rank condition of $\mathcal{Z}$ is not satisfied. In this case, the dynamics of $\tilde{\mathcal{W}}$ is given by (22). Substituting $\hat{\mathcal{W}} - \mathcal{W} = \tilde{\mathcal{W}}$ and using (22) in (42) and simplifying we get

$$\begin{aligned}\dot{V}_3 \leq& \|\boldsymbol{s}\|\Big(\|\mathcal{A}(\boldsymbol{e}) - \mathcal{A}(\check{\boldsymbol{e}})\| + \|\hat{\varrho}(t) - \hat{\varrho}(t_k)\| + \|\hat{\mathcal{W}}^\top(t)\Psi(\boldsymbol{z})\\ &- \hat{\mathcal{W}}^\top(t_k)\Psi(\check{\boldsymbol{z}})\|\Big) - (K + \hat{\varrho}(t))\boldsymbol{s}^\top\text{sgn}(\check{\boldsymbol{s}}) + \|\boldsymbol{\varepsilon}(\boldsymbol{z})\|\|\boldsymbol{s}\|\\ &- \sigma\|\tilde{\mathcal{W}}\|^2 + \sigma\|\tilde{\mathcal{W}}\|\|\mathcal{W}\| + rp_r\Psi_M\|\boldsymbol{\varepsilon}(\boldsymbol{z})\|\|\tilde{\mathcal{W}}\|\\ &+ \frac{1}{\gamma_2}\tilde{\varrho}^\top(t)\dot{\hat{\varrho}}(t).\end{aligned}$$

When $s \notin \mathfrak{B}_s$, it is to be noted that the trajectories of the system will be on either side of the manifold $\mathbb{S}$, where the sampled value of the sliding variable $\text{sgn}(\check{\boldsymbol{s}})$ will have the

same sign with its continuous counterpart s. Thus, it can be assured that $\text{sgn}(\boldsymbol{s}) = \text{sgn}(\check{\boldsymbol{s}})$. Using this fact along with (46), (48), (49) from Appendix, the above inequality can be written further as

$$\begin{aligned}\dot{V}_3 \leq & -\|\boldsymbol{s}\|\bigg(K - L_{\mathcal{A}}\|\boldsymbol{e}-\check{\boldsymbol{e}}\| - L_\rho - L_{\mathcal{W}}\bigg) - \tilde{\varrho}(t)\|\boldsymbol{s}\| \\ & - \|\tilde{\mathcal{W}}\|\bigg(\sigma\|\tilde{\mathcal{W}}(t)\| - \sigma\|\mathcal{W}\| - rp_r\Psi_M\|\boldsymbol{\varepsilon}(\boldsymbol{z})\|\bigg) \\ & + \frac{1}{\gamma_2}\tilde{\varrho}^\top(t)\dot{\hat{\varrho}}(t).\end{aligned}$$

Further with (26) and using (50) from Appendix along with the gain $K > L_{\mathcal{A}}\alpha + L_{\mathcal{W}} + L_\varrho + \eta$, we can write

$$\dot{V}_3 \leq -\eta\|\boldsymbol{s}\| - \sigma\|\tilde{\mathcal{W}}(t)\|\bigg(\|\tilde{\mathcal{W}}(t)\| - r_1\bigg), \tag{43}$$

where r_1 is defined in Theorem 2. The finite time stability of (43) with decay rate $\min(\eta, \sigma)$ is assured when $\|\tilde{\mathcal{W}}(t)\| > r_1$. Thus, the error trajectories hit the manifold $\mathbb{S}$ in finite time and the estimation error is bounded inside a ball with radius r_1. On the other hand, as the replay buffer accumulates the data, after some time $t_r \in (0, \infty)$, the dynamics of the weight estimation error of RBFNN is governed by (23) which can be written with (42) as

$$\begin{aligned}\dot{V}_3 = & \boldsymbol{s}^\top\bigg(\mathcal{A}(\boldsymbol{e}) - \mathcal{A}(\check{\boldsymbol{e}}) - \tilde{\mathcal{W}}^\top(t)\Psi(\boldsymbol{z}) - (\hat{\varrho}(t) + K)\text{sgn}(\check{\boldsymbol{s}}) \\ & + \hat{\mathcal{W}}^\top(t)\Psi(\boldsymbol{z}) - \hat{\mathcal{W}}^\top(t_k)\Psi(\check{\boldsymbol{z}}) + (\hat{\varrho}(t) - \hat{\varrho}(t_k))\text{sgn}(\check{\boldsymbol{s}}) \\ & + \boldsymbol{\epsilon}(\boldsymbol{z})\bigg) - \text{tr}\bigg(\tilde{\mathcal{W}}^\top(t)\bigg[\bigg(\sum_{i=1}^{p_r}\Psi(\boldsymbol{z}_i)\Psi^\top(\boldsymbol{z}_i)\bigg)\tilde{\mathcal{W}}(t) \\ & + \sum_{i=1}^{p_r}\Psi(\boldsymbol{z}_i)\boldsymbol{\varepsilon}^\top(\boldsymbol{z}_i) - \Gamma_1\Psi(\boldsymbol{z})\boldsymbol{s}^\top\bigg]\bigg) + \frac{1}{\gamma_2}\tilde{\varrho}^\top(t)\dot{\hat{\varrho}}(t).\end{aligned}$$

Using similar manipulations as in the case of rank deficient case of $\mathcal{Z}$, we can finally arrive at

$$\dot{V}_3 \leq -\eta\|\boldsymbol{s}\| - \zeta_{min}(\Xi)\|\tilde{\mathcal{W}}(t)\|\bigg(\|\tilde{\mathcal{W}}(t)\| - r_2\bigg), \tag{44}$$

where r_2 is defined in Theorem 2. The error trajectories hit the manifold $\mathbb{S}$ and $\|\tilde{\mathcal{W}}(t)\|$ will be ultimately bounded inside a ball of radius r_2.

Case 2: **The error trajectories are inside the set $\mathfrak{B}_s$**: Due to the reachability condition, the error trajectories enter the set $\mathfrak{B}_s$ and hit $\mathbb{S}$. Once this occurs, the $\text{sgn}(\boldsymbol{s}) = \text{sgn}(\check{\boldsymbol{s}})$ is not assured and for which the trajectories will grow away from the manifold $\mathbb{S}$. Once it grows, due to the event condition (24), it can be guaranteed that the next sampling will occur and control will be updated which enforces the trajectories to meet the reachability condition. The maximum growth of the sliding variables can be calculated as

$$\begin{aligned}\|s - \check{\boldsymbol{s}}\| \leq & k_1|x_e - \check{x}_e| + k_2|\theta_e - \check{\theta}_e| \\ & + v_M\left|\frac{y_e}{\sqrt{c + x_e^2 + y_e^2}} - \frac{\check{y}}{\sqrt{c + \check{x}_e^2 + \check{y}_e^2}}\right|.\end{aligned}$$

Using the Lipchitz constant of $\frac{v_r y_e}{\sqrt{c+x_e^2+y_e^2}}$ derived at Appendix and (24) we get

$$\begin{aligned}\|s - \check{\boldsymbol{s}}\| \leq & (k_1 + Lv_m)|x_e - \check{x}_e| + Lv_M|y_e - \check{y}_e| + k_2|\theta_e - \check{\theta}_e|, \\ \leq & \lambda(t).\end{aligned} \tag{45}$$

Hence, it is ensured that the error trajectories will achieve PSM and will be contained in the set $\mathfrak{B}_s$. Hence the proof is completed. ■

Lipchitz constant of $\mathcal{A}(\boldsymbol{e})$ and $\frac{v_r y_e}{\sqrt{c+x_e^2+y_e^2}}$

Consider the vector function as $\mathcal{A}(\boldsymbol{e}) = \begin{bmatrix}\mathcal{A}_1(\boldsymbol{e}) \\ \mathcal{A}_2(\boldsymbol{e})\end{bmatrix}$. The partial derivative with respect to e is given by $\frac{\partial\mathcal{A}(\boldsymbol{e})}{\partial e} = \begin{bmatrix} 0 & 0 & -k_1 v_r \sin\theta_e \\ \frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial x_e} & \frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial y_e} & \frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial\theta_e}\end{bmatrix}$. Hence the Lipchitz constant can be calculated as [22]

$$\begin{aligned}\left\|\frac{\partial\mathcal{A}(\boldsymbol{e})}{\partial\boldsymbol{e}}\right\| = & \max\left\{\left\|\frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial x_e}\right\|, \left\|\frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial y_e}\right\|, k_1 v_M + \left\|\frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial\theta_e}\right\|\right\} \\ = & L_{\mathcal{A}}.\end{aligned} \tag{46}$$

where $\left\|\frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial x_e}\right\| = \frac{2x_M}{c^{1.5}}\left(y_M + 1 + \frac{v_M^2 y_M}{x_M} + \frac{3}{2c}x_M a_M\right)$, $\left\|\frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial y_e}\right\| = \frac{1}{c^{1.5}}\left((c + x_M^2 + 3y_M^2)\varsigma_v + v_M^2 x_M + \frac{3}{c}y_M a_M\right)$ and $\left\|\frac{\partial\mathcal{A}_2(\boldsymbol{e})}{\partial\theta_e}\right\| = c + x_M + v_M^2 x_M y_M$.

The partial derivative of $f = \frac{v_r y_e}{\sqrt{c+x_e^2+y_e^2}}$ with respect to e is given by $\frac{\partial f}{\partial e} = \begin{bmatrix}\frac{2x_e y_e}{(c+x_e^2+y_e^2)^{1.5}} & \frac{2y_e^2}{(c+x_e^2+y_e^2)^{1.5}} & 0\end{bmatrix}$. So, the Lipchitz constant is given by

$$\left\|\frac{\partial f}{\partial e}\right\| = L = \max\left\{\frac{2y_M x_M}{c^{1.5}}, \frac{c + y_M}{c^{1.5}}\right\} \tag{47}$$

Bound of $\|\hat{\varrho}(t) - \hat{\varrho}(t_k)\|$, $\|\hat{\mathcal{W}}^\top(t)h(t) - \hat{\mathcal{W}}^\top(t_k)h(t_k)\|$ and $\|\boldsymbol{e} - \check{\boldsymbol{e}}\|$

Recall the adaptive gain update rule (26) from where we can write

$$\begin{aligned}\|\hat{\varrho}(t) - \hat{\varrho}(t_k)\| = & \gamma_2\int_{t_k}^{t}\|\boldsymbol{s}\|dt \\ \leq & \gamma_2(\|\check{\boldsymbol{s}}\| + \alpha)\Delta T_k = L_\varrho.\end{aligned} \tag{48}$$

Recall the RBFNN weight update rule (15), using this we can write

$$\begin{aligned}\|\hat{\mathcal{W}}^\top(t)h(t) - \hat{\mathcal{W}}^\top(t_k)h(t_k)\| = & \Gamma_1\int_{t_k}^{t}\Psi(\boldsymbol{z})\boldsymbol{s}^\top dt \\ \leq & \Psi_M\|\Gamma_1\|(\|\check{\boldsymbol{s}}\| + \alpha)\Delta T_k \\ = & L_{\mathcal{W}}.\end{aligned} \tag{49}$$

The expression of $\|\boldsymbol{e} - \check{\boldsymbol{e}}\|$ can be expressed as

$$\begin{aligned}\|\boldsymbol{e} - \check{\boldsymbol{e}}\| \leq & |x_e - \check{x}_e| + |y_e - \check{y}_e| + |\theta_e - \check{\theta}_e| \\ \leq & \lambda(t) \leq \alpha.\end{aligned} \tag{50}$$

CALCULATION OF $\sup\{\dot{\lambda}(t)\}$

Recall the definition of $\lambda(t)$ from where we can write

$$\begin{aligned}\dot{\lambda}(t) =& c_1\dot{x}_e + c_2\dot{y}_e + c_3\dot{\theta}_e \\ \leq& \max\{c_1, c_2, c_3\}\|\dot{e}\|\end{aligned}$$

Using $c_m := \max\{c_1, c_2, c_3\}$ and using the error dynamics (6) we can write

$$\begin{aligned}\dot{\lambda}(t) \leq& c_m(\|F\| + \|G\|\|\|u(\hat{\mathcal{W}}, \hat{\varrho})\|\| + \|\Psi\|) \\ \leq& c_m(\sqrt{2v_M^2 + \omega_M^2} + 2 + y_M + x_M + \|u(\hat{\mathcal{W}}, \hat{\varrho})\| \\ &+ \Psi_{max}\|\mathcal{W}\| + \varrho) = \Theta(\hat{\mathcal{W}}, \hat{\varrho})\end{aligned} \tag{51}$$

REFERENCES

[1] R. W. Brockett *et al.*, "Asymptotic stability and feedback stabilization," *Differential geometric control theory*, vol. 27, no. 1, pp. 181–191, 1983.

[2] C. Peng and F. Li, "A survey on recent advances in event-triggered communication and control," *Information Sciences*, vol. 457, pp. 113–125, 2018.

[3] Y. Shtessel, C. Edwards, L. Fridman, A. Levant *et al.*, *Sliding mode control and observation*. Springer, 2014, vol. 10.

[4] A. K. Behera, B. Bandyopadhyay, M. Cucuzzella, A. Ferrara, and X. Yu, "A survey on event-triggered sliding mode control," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, 2021.

[5] R. Postoyan, M. C. Bragagnolo, E. Galbrun, J. Daafouz, D. Nesic, and E. Castelan, "Event-triggered tracking control of unicycle mobile robots," *Automatica*, vol. 52, pp. 302–308, 2015.

[6] Z. Sun, Y. Xia, L. Dai, and P. Campoy, "Tracking of unicycle robots using event-based mpc with adaptive prediction horizon," *IEEE/ASME Transactions on Mechatronics*, vol. 25, pp. 739–749, 2019.

[7] C. Xie, Y. Fan, and J. Qiu, "Event-based tracking control for nonholonomic mobile robots," *Nonlinear Analysis: Hybrid Systems*, vol. 38, p. 100945, 2020.

[8] K. Nath, A. Yesmin, A. Nanda, and M. K. Bera, "Event-triggered sliding mode control of two wheeled mobile robot: An experimental validation," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, 2021.

[9] P. Singh, P. Agrawal, A. Nandanwar, L. Behera, N. K. Verma, S. Nahavandi, and M. Jamshidi, "Multivariable event-triggered generalized super-twisting controller for safe navigation of nonholonomic mobile robot," *IEEE Systems Journal*, vol. 15, no. 1, pp. 454–465, 2020.

[10] P. Singh, A. Nandanwar, L. Behera, N. K. Verma, and S. Nahavandi, "Uncertainty compensator and fault estimator-based exponential super-twisting sliding-mode controller for a mobile robot," *IEEE Transactions on Cybernetics*, 2021.

[11] C. Liu, G. Wen, Z. Zhao, and R. Sedaghati, "Neural-network-based sliding-mode control of an uncertain robot using dynamic model approximated switching gain," *IEEE transactions on cybernetics*, vol. 51, no. 5, pp. 2339–2346, 2020.

[12] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*. Courier Corporation, 2012.

[13] P. Ioannou and K. Tsakalis, "A robust direct adaptive controller," *IEEE Transactions on Automatic control*, vol. 31, no. 11, pp. 1033–1043, 1986.

[14] K. Narendra and A. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic control*, vol. 32, no. 2, pp. 134–145, 1987.

[15] G. Chowdhary and E. Johnson, "Adaptive neural network flight control using both current and recorded data," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, p. 6505.

[16] G. Chowdhary, M. Mühlegg, J. P. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*. Springer, 2013, pp. 29–47.

[17] G. Chowdhary, M. Mühlegg, and E. Johnson, "Exponential parameter and tracking error convergence guarantees for adaptive controllers without persistency of excitation," *International Journal of Control*, vol. 87, no. 8, pp. 1583–1603, 2014.

[18] N. Wang and F. Hao, "Event-triggered sliding mode control with adaptive neural networks for uncertain nonlinear systems," *Neurocomputing*, vol. 436, pp. 184–197, 2021.

[19] J. Mu, X.-G. Yan, S. K. Spurgeon, and Z. Mao, "Generalized regular form based smc for nonlinear systems with application to a wmr," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6714–6723, 2017.

[20] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE transactions on neural networks*, vol. 9, no. 4, pp. 589–600, 1998.

[21] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 384–389.

[22] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

[23] D.-P. Li, Y.-J. Liu, S. Tong, C. P. Chen, and D.-J. Li, "Neural networks-based adaptive control for nonlinear state constrained systems with input delay," *IEEE transactions on cybernetics*, vol. 49, no. 4, pp. 1249–1258, 2018.

[24] N. Zhou, Y. Kawano, and M. Cao, "Neural network-based adaptive control for spacecraft under actuator failures and input saturations," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3696–3710, 2019.

[25] A. Sahoo, H. Xu, and S. Jagannathan, "Neural network-based event-triggered state feedback control of nonlinear continuous-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 497–509, 2015.

[26] J. Liu, *Radial Basis Function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation*. Springer Science & Business Media, 2013.

[27] M. Reichhartinger, S. Spurgeon, M. Forstinger, and M. Wipfler, "A robust exact differentiator toolbox for matlab®/simulink®," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1711–1716, 2017.

[28] G. Chowdhary and E. Johnson, "Lyapunov-based integration of a data recording algorithm in adaptive control," in *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6284.

[29] B. Bandyopadhyay and A. K. Behera, *Event-Triggered Sliding Mode Control A New Approach to Control System Design*. Springer, 2018.

[30] M. Cucuzzella, G. P. Incremona, and A. Ferrara, "Event-triggered variable structure control," *International Journal of Control*, vol. 93, no. 2, pp. 252–260, 2020.

[31] J. G. V. Utkin and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*. Boca Raton: CRC Press, 2009.

[32] Event-triggered concurrent learning-based neuro-adaptive robust tracking control of wheeled mobile robot: An experimental validation. https://www.youtube.com/playlist?list=PLFIBkOT8UMRZTK4es5nriG-1KiW6vWW98.

[33] E. A. Martínez, H. Ríos, and M. Mera, "Robust tracking control design for unicycle mobile robots with input saturation," *Control Engineering Practice*, vol. 107, p. 104676, 2021.