FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 35, Nº 4, December 2022, pp. 557-570
https://doi.org/10.2298/FUEE2204557S

Original scientific paper

EVENT-TRIGGERED SLIDING MODE CONTROL FOR CONSTRAINED NETWORKED CONTROL SYSTEMS*

Andrej Sarjaš, Dušan Gleich

University of Maribor, Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia

Abstract. *The paper describes a Non-linear Control (ETNC) approach for constrained Networked Feedback Control Systems (NFCS). The real-time controller execution is implemented based on the Event-triggering paradigm. A nonlinear variable structure is used for the controller design. The nonlinear approach is based on the predefined sliding variable defined by the system states with a nonlinear switching function. The system's stability is analyzed regarding the evolution of the sliding variable. The Event-Triggered operation of the nonlinear controller is based on the prescribed triggering rule. The stability boundary of the sliding variable is subject to the preselected triggering condition, whose selection is a tradeoff of system performance, networks constraints and transmission capabilities. The main focus of the Event triggering approach is lowering network resources utilization in the steady-state behavior of the NFCS. The presented approach ensures a non-zero inter-event time of controller execution, which enables scheduling and optimization of the network operation regarding the network constraints and real-time system performance. The efficiency of the presented method is presented with a comparison of the classical time triggering approach. The real measurement supports the results.*

Key words: *Event-triggering, networked control system, variable structure control, sliding mode control*

1. INTRODUCTION

Networked feedback control systems have been researched extensively over the last two decades [1]. New communication technologies integrated into tiny devices with decent computational capability offer vast, remote applications in distributed or decentralized structures. Regarding the network structure and amount of connected devices, the implementation of the NFCS is critical. New methods are derived that improve network usage and ensure system performance according to the controller implementation and execution. The paper introduces the nonlinear control law with event triggering execution.

Received March 23, 2022; revised April 29, 2022; accepted June 15, 2022
Corresponding author: Andrej Sarjaš
University of Maribor, Faculty of Electrical Engineering and Computer Science, Koroška cesta 46, SI-2000 Maribor, Slovenia
E-mail: andrej.sarjas@um.si
* An earlier version of this paper was presented at the 15thInternational Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS 2021), October 20-22, 2021, in Niš, Serbia [1]

Sliding mode control (SMC) is an effective approach to ensure the prescribed performance of a closed-loop system, despite external disturbances and system uncertainty [1]-[4]. Depending on the controller structure, the sliding mode controller is straightforward to implement and requires much computational time. All controllers in the real-time system are implemented in a discrete form, which results in a hybrid system where the continuous and discrete systems are interconnected [3]-[6]. The most commonly used approach for controller implementation is a sample and hold technique, or time triggering approach (TT). Time triggering means that the controller output is updated at equidistant time intervals, also known as a sampling time. Such TT closed-loop system is more suitable to design, due to the vast amount of ` developed techniques and approaches for time sampled systems. On the other hand, the TT system requires constant resources` utilization and data transmissions over the network system.

The Event-Triggering (ET) approach of a closed-loop system offers an alternative to the TT [7]. Regarding the TT in the ET system, the closed-loop system is updated based on the trigger rule evaluation. In other words, the controller is updated when the system states violate the triggering rule, which means that the controller is no longer updated periodically with fixed time intervals. Such an implementation of the controller is more efficient than the TT implementation, and requires fewer computational resources, especially when the sliding manifold is reached. Regarding the latter, ET is beneficial for the networked control system (NCS), where the trigger mechanism reduces network transmission and is suitable for systems with data-rate constraints [8]-[10]. The network constraints with variable Round Trip Time (RTT), limited data transmission, and package drops are insufficient for the NCS[11], [12]. The mentioned network parameters reduce system performances considerably, and can lead to unstable operation. The presented work introduced an SMC controller design with an associated triggering rule, which ensures NCS stability and takes all the network parameters into account during the design procedure. The derived event-triggered sliding mode controller (ET-SMC) introduces triggering boundaries regarding the admissible lower inter-event time value and network delay [13], [14]. The ET-SMC for NCS is divided into two steps. The first step introduces an SMC controller design with preselected system dynamics and parametrized sliding variables [15]-[17]. The second step involves triggering boundary selection regarding the system tracking performance and NCS uncertainty robustness. In comparison to the similar linear ET paradigms, the presented approach still ensure SMC properties and lowers the computational burden and network usage effectively.

The controller parameter selection can be presented as an optimization procedure. The optimal parameter selection can be evaluated as a tradeoff between network utilization regarding NCS uncertainties and closed-loop performance, such as tracking capability, transient performance, network delay, etc. The assessment of the admissible lower inter-event time of the ET shows the direct influence of the ET-controller on the network utilization during the reaching and sliding phase of the sliding variable evolution. The efficiency of the proposed controller is evaluated on a real-time system.

2. SLIDING MODE CONTROLLER DESIGN

For the sliding mode controller (SMC) synthesis, the given system is used,

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -bx_2 + gv + d,\end{aligned} \tag{1}$$

where $x(t) = [x_1(t)\ x_2(t)]^T \in \mathbb{R}^2$ is a state vector and $v(t) \in \mathbb{R}$ is the control variable. The parameters $g: \mathbb{R} \to \mathbb{R}$ and $b: \mathbb{R} \to \mathbb{R}$ are system parameters, where $d: \mathbb{R} \to \mathbb{R}$ is a disturbance. For SMC design, the boundary of the system parameters are given, $0 < b < b_{\max}$, $g_{\min} < g < g_{\max}$, $[g_{\min}, g_{\max}, b_{\max}] \in \mathbb{R}_{>0/\infty}$. For system tracking capability, new system states are introduced, $\xi_1 = x_d - x_1$, $\xi_2 = \dot{x}_d - x_2$, where x_d is the desired value with its derivative $\dot{x}_d$. The transformed system is given as,

$$\begin{aligned}\dot{\xi}_1 &= \xi_2\\ \dot{\xi}_2 &= -b\xi_2 - gv + \tilde{d},\end{aligned} \tag{2}$$

where $\xi = [\xi_1, \xi_2]^T$, $\tilde{d} = -d + \ddot{x}_d + b\dot{x}_d$ and holds $\sup_{t\geq 0}\left|\tilde{d}(t)\right| \leq \Delta_d < \infty$. The sliding variable is designed as $s = c\xi$ for $c \in \mathbb{R}^2$, where $c = [c_1\ \ 1],\ c_1 > 0$. Differentiating of $s = c\xi$ with respect to time gives,

$$\begin{aligned}\dot{s} &= \dot{\xi}_2 + c_1\dot{\xi}_1 =\\ &= (c_1 - b)\xi_2 - gv + \tilde{d} < 0.\end{aligned} \tag{3}$$

Regarding (3), $\Delta_d < \infty$ and the sliding property, which brings the sliding variable to the sliding manifold, $s, \dot{s} = 0$ the SMC controller can select as,

$$v = g^{-1}((c_1 - b)\xi_2 + \rho sign(s)), \tag{4}$$

where $\rho > \Delta_d$ holds. After the SMC controller design (4), the ET mechanism will be introduced in the next section. The controller (4) contains a nonlinear term, the solution of the feedback system (2),(3) with controller (4) is understood in the Filippov sense [18].

3. EVENT-TRIGGERED SLIDING MODE CONTROL FOR NCS

The event-triggered rule derivation is based on the analysis of the reaching phase stability of the sliding variable [2]. It is worthy of mentioning that the discrete implementation of SMC can not reach a sliding manifold completely. As a result, the quasi-sliding mode is obtained [16], [19], where the sliding variable is limited with boundary $|s| \leq \Omega,\ \Omega \in \mathbb{R}_+/\mathbb{R}_\infty$, where Ω it is subject to the sampling time, sliding parameter, and disturbance Δ_d. Furthermore, the presented work is limited to the ET approach, where the band Ω will be determined regarding the trigger mechanism and preselected inter-event time. The ET-SMC after two consecutive updates is given as

$$v_{ET}(t) = g^{-1}\left(\left(c_1 - b\right)\xi_2\left(t_n\right) + \rho sign\left(s\left(t_n\right)\right)\right), \tag{5}$$

where t_n is the last update, t is the current time between two updates, and is $t \in [t_n, t_{n+1})$.

Theorem 1: *Consider system (2) with the sliding manifold* s = 0. *The parameter β is given so that*

$$v_{ET}(t) = g^{-1}\left((c_1 - b)\xi_2(t_n) + \rho sign(s(t_n))\right), \tag{6}$$

for all $t > 0$, *where* $e_2(t) = \xi_2(t) - \xi_2(t_n)$. *The event triggering is established if the controller gain is selected as*

$$\rho > \beta + \Delta_d \tag{7}$$

where holds $\beta > 0$.

Proof: Before continuing to prove, the remaining ET error variables are introduced, $e_1(t) = \xi_1(t) - \xi_1(t_n)$, and $e(t) = \xi(t) - \xi(t_n)$. For the stability test, the Lyapunov function is presented $V(t) = s(t)^2/2$ for the time interval $t \in [t_n, t_{n+1})$, where $n \in \mathbb{Z}_{\geq 0}$. Differentiation V with respect to time t the derivative $\dot{V}$ is given as

$$\dot{V} = s\dot{s} = s((c_1 - b)\xi_2 - gv_{ET} + \tilde{d}). \tag{8}$$

Substituting the controller (5) in (8) gives

$$\begin{aligned}
\dot{V} &= s((c_1 - b)\xi_2(t) - gv_{ET}(t_n) + \tilde{d}(t)) \\
&= s((c_1 - b)\xi_2(t) - (c_1 - b)\xi_2(t_n) - \rho sign(s(t_n)) + \tilde{d}(t)) \\
&= s((c_1 - b)(\xi_2(t) - \xi_2(t_n)) - \rho sign(s(t_n)) + \tilde{d}(t)) \\
&\leq s(c_1 - b)\underbrace{(\xi_2(t) - \xi_2(t_n))}_{e_2(t)} - |s|\rho + |s|\Delta_d \\
&\leq s(c_1 - b)e_2(t) - |s|\rho + |s|\Delta_d \\
&\leq |s|\beta - |s|\rho + |s|\Delta_d \\
&\leq -|s|\left(\rho - \beta - \Delta_d\right) \\
&\leq -\psi|s|,
\end{aligned}$$

where $\psi > 0$. Concerning the condition (7) and assumption $sign(s(t_n)) = sign(s(t))$, it is to be noted that the sliding variable is approaching the sliding manifold, where $s = 0$. The above is true if at the time of triggering $t = t_n$ holds $e_2(t_n) = e_2(t) = 0$, then the sliding variable s is bounded with Ω, where,

$$\begin{aligned}
|s(t) - s(t_n)| &= |c\xi(t) - c\xi(t_n)| = \|c\|\|e\| \\
&\leq \|c\|\|k\|\|e_2\| \\
&< k\frac{\|c\|}{\|(c_1 - b)\|}\beta = \tilde{k}\beta,
\end{aligned} \tag{9}$$

regarding $\|e_2\| \leq \|e\|$ and $k\|e_2\| = \|e\|$. The parameter k is defined as $k = \sqrt{1 + \frac{\|(c_1 - b)\|^2 \alpha^2}{\beta^2}}$ and α is an upper limit of the $\sup_{t \geq 0}|e_1(t)| \leq \alpha < \infty$. The boundary Ω is defined as $\Omega = \{\xi \in \mathbb{R}, |s| = |c\xi| < \tilde{k}\beta\}$, where the triggering rule in (6) can be defined as,

$$\|e_2(t)\| > \beta\|(c_1 - b)\|^{-1}, \tag{10}$$

which is the end of the proof.

The stability of the remaining system in (2) with controller (5) needs to be assessed after the stability analysis of the sliding variable with triggering condition. Regarding the reaching phase boundary (9), it can be derived $\xi_2 = s - c_1\xi_1$, where $\dot{\xi}_1 = s - c_1\xi_1$ holds. With the introduction of the Lyapunov function $V = \xi_1^2 / 2$, the stability can be assessed as,

$$\begin{aligned}\dot{V} &= \xi_1\dot{\xi}_1\\ &= \xi_1(s - c_1\xi_1)\\ &= -c_1\|\xi_1\|\left(\|\xi_1\| - \frac{1}{c_1}\|s\|\right),\end{aligned}$$

With respect to conditions (6),(9), the system is stable if it holds that $\|\xi_1\| - c_1^{-1}\|s\| > 0$. Thus, the closed-loop system is stable with respect to *s*, and the system trajectory ξ_1 is bounded by

$$\|\xi_1\| < \frac{k\|c\|}{\|c_1\|\|(c_1 - b)\|}\beta. \tag{11}$$

4. Event-Triggered Sliding Mode Control for NCS

The structure of the network control system is depicted in Fig. 1. The controller algorithm is executed on the network computer, where the triggering rule is evaluated on the plant. We assume that the plant has a real-time system with computational ability and communication interfaces. The real-time system on the plant side is used for noncomplex computation such as triggering condition evaluation, signal conditioning, and communication capability. The User Datagram Protocol (UDP) is used for the given ET-SMC implementation. The data have been transmitted over different network hops, where additional time delay and package loss may occur. The package loss in the network is modeled as a loss delay [12], [13], where the maximal allowed Round Trip Time (RTT) of the network is used for package loss detection. The plant side uses a dedicated package-loss timer, and if the watchdog timer is expired, then the request for new data is demanded from the server. We assume that two consecutive losses can not be accrued for the package loss occurrence.

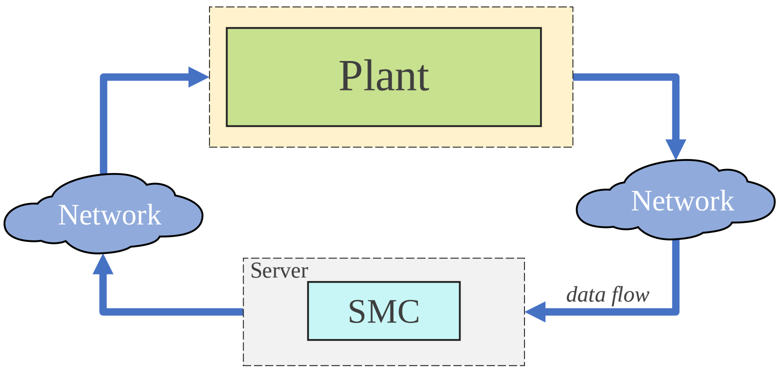

Fig. 1 Networked controller structure with ET-SMC

The controller feedback structure is presented in Fig. 2, where the triggering condition determines the network usage. The controller (5) is implemented on the server, and the triggering mechanism is on the plant side.

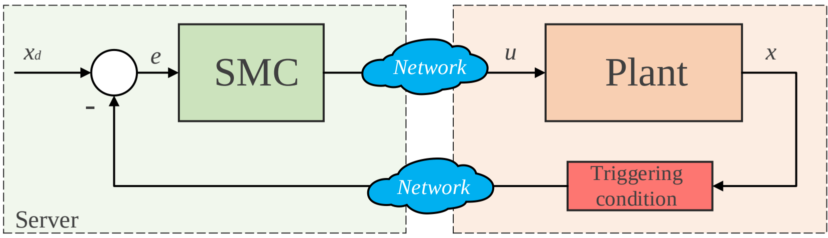

Fig. 2 ET-SMC feedback configuration

The inter-event time of the ET-SMC is determined regarding the error analysis of the two consecutive sampled states,

$$\frac{d}{dt}\|e(t)\| \leq \left\|\frac{d}{dt}e(t)\right\| = \left\|\frac{d}{dt}\begin{bmatrix}\xi_1(t)-\xi_1(t_n)\\ \xi_2(t)-\xi_2(t_n)\end{bmatrix}\right\| = \left\|\frac{d}{dt}\begin{bmatrix}\xi_1(t)\\ \xi_2(t)\end{bmatrix}\right\|, \tag{12}$$

where $\xi(t_n)=0$, according to the last update. Substitute (12) in (2), (5) which gives

$$\frac{d}{dt}\|e(t)\| \leq \left\|\begin{bmatrix}0 & 1\\ 0 & -b\end{bmatrix}\xi(t) + \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\tilde{d}(t) - \begin{bmatrix}0\\ g\end{bmatrix}v_{ET}(t_n)\right\|,$$

$$= \left\|\underbrace{\begin{bmatrix}0 & 1\\ 0 & -c_1\end{bmatrix}}_{A_c}\xi(t) + \underbrace{\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}}_{B_d}\tilde{d}(t) - \underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{B_c}\rho sign(s(t_n))\right\|$$

$$= \left\|A_c(e(t)+\xi(t_n)) - B_c\rho sign(s(t_n)) + B_d\tilde{d}(t)\right\|$$

$$\leq \|A_c\|\|e(t)\| + \|A_c\|\|\xi(t_n)\| + \|B_c\|\rho + \|B_d\|\Delta_d.$$

The solution of the differential equations is

$$\|e(t)\| \leq \frac{\|A_c\|\|\xi(t_n)\| + \|B_c\|\rho + \|B_d\|\Delta_d}{\|A_c\|}(e^{\|A_c\|(t-t_n)}-1), \tag{13}$$

where the minimal inter-event time $\tau = t - t_n$ is determined as

$$\tau_{\min} \geq \frac{1}{\|A_c\|}\ln\left(\frac{k\beta\|A_c\|}{\|(c_1-b)\|\left(\|A_c\|\|\xi(t_n)\| + \|B_c\|\rho + \|B_d\|\Delta_d\right)} + 1\right) \tag{14}$$

It can be seen that the inter-event time depends on triggering condition β and selected controller parameters c_1 and ρ. Regarding the uncertainty of the network, the delay η_n is introduced with the update time t_n. The update sequence $\{t_n+\eta_n\}_{n=0}^{\infty}$ corresponds to the update time t_n and means that the controller is not updated with the last states, wherein the inter-event time is extended by delay value η_n. Hence the error (13) grows till the next update time t_{n+1}. The triggering sequence is admissible, regarding if $t_{n+1} > t_n + \eta_n,\ n \in \mathbb{Z}_{\geq 0}$ and the triggering rules (6),(10) ensure system stability. The derivation of the delay boundary, where the triggering rule ensures the system stability, is similar to the derivation of the inter-event time in (13),(14). For a given derivation, we assumed that the controller (5) at the time $t \in [t_n, t_n + \eta_n)$ is not updated with the current state $\xi(t_n)$, whereby the further updates are executed at $t \in [t_n + \eta_n, t_{n+1} + \eta_{n+1})$, and the analysis involves the controller structure with past value $v(t) = g^{-1}((c_1 - b)\xi_2(t_{n-1}) + \rho\, sign(s(t_{n-1})))$. The admissible inter-event time is caused by the delay, which ensures that the system stability with triggering condition (10) is,

$$\tau_n = \frac{1}{\|A_c\|} \ln\left(\frac{k\beta\|A_c\|}{\|(c_1 - b)\|\left(\|A_c\|\left(\|\xi(t_n)\| + \|\xi(t_{n-1})\|\right) + \|B_c\|\rho + \|B_d\|\Delta_d\right)} + 1 \right) \tag{15}$$

The system is stable, and the boundary (11) is preserved if $\eta_n \leq \tau_n$ it holds. For proper parameter selection, it is necessary to assume the maximally allowed delay in the network. The delay boundary is given as $\sup_{n\geq 0}|\eta_n| \leq \Delta_\eta < \infty$.

The network structure and the used protocol for communication are designed after derivation of the crucial parameters for event-triggering implementation. The focal point of the network system is a protocol that needs to ensure simple transmission and minimal package loss with low *RTT*. All transmitted data must be transparent to the server and the client, whereas the measured data should not be ambiguous. The designed protocol enables package lost detection and possible adaptation of the controller execution in a classical TT or ET manner. The package loss algorithm is essential for controller output recovery. If the package loss is detected or the *RTT* timer reaches the threshold, the controller output must be updated. Otherwise, the closed-loop system is running in an open-loop. The update can be done with a new data transmission request from the server or an internal model-based approach. The recurrent request sent is a straightforward task for the controller update, whereas the model-based approach is more complex and advanced. In the model-based approach, the system data are obtained from the model or system approximation algorithms such as fuzzy sets and neural networks. The model-based approach requires more computational resources on the server or the client-side. Such an approach can ensure faster output recovery than sending a new transmission request. The model-based update regarding the computational resources can act as a redundant system in the case of irregularities on the network or system. The structure of the designed protocol for the client communication is presented in Fig. 3.

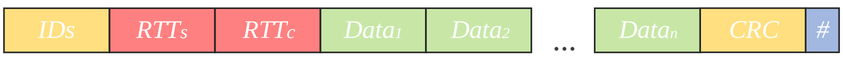

Fig. 3 The communication protocol of the client message

The *IDs* presents the server address, which is the main system of the NCS. Tags RTT_s and *RTTc* are timing data of the network *RTT*, one on the server-side and the other on the client-side. Both sides are measured with their own *RTT*, where the server`s *RTTs* is the time from server send to server received, and the client RTT_c is similar to *RTTs* with beginning on the client send and received. The package loss and network irregularities can be detected with comparisons of the RTT_c and RTT_s. Tags $Data_{1,2,n}$ are transmitted states of the system. The estimation and detection of the network irregularities through different measured parameters are not the main objective of the presented work and will not be discussed hereinafter. All additional parameters of the protocol, which are not directly involved in the NCS operation, are just starting points for the further research of a network`s quality and reliability assessment. The protocol is concluded with a cyclic redundancy check *CRC* and the delimiter #. The response message from the server to the client is presented in Fig. 4.

Fig. 4 The communication protocol of the server message

The *IDc* presents the client address, where *RTT*, *CRC* and # are the same parameters as in the client message presented in Fig. 3. The tags $\text{Cont}_{1,2,..}$ are controllers update values. All the time values and data are presented in *4*bytes float format. The *ID* and *CRC* are presented with 32-bit integer values. The length of the message is determined with a number of transceived system states (data), whereas *ID*, RTT_s, RTT_c, *CRC*, and # values are mandatory and are the control parameters of the used protocol.

Regarding the employed protocol with network RTT time measurement on the server and client-side, it is necessary to acknowledge the possible network uncertainty. The network uncertainty can be presented as network delay, where the network information takes time to spread from the sender to the receiver over different network hoops. The delay can cause an unwanted effect on the feedback system, such as an oscillation, slower response, deteriorated disturbance rejection capability, and even unstable operation. The delayed system needs special awareness in the controller design. In the proposed approach, the delayed system is presented as an additional elapsed time after requesting a new update from the client-side. The delay caused a more extended operation in the unstable region given in (11),(13),(14). The inter-event time (14) is extended, and the permitted state boundary is extended (13). Such time delay lowers the performance of the closed-loop system and tracking capability. The system's stability is ensured with the proper selection of the controller gain given in (7). If the time delay is modeled as a parametric uncertainty with a prescribed bound, Δ_η then the controller gain selection can be lowered for the admissible delay boundary.

$$\rho > \beta + \Delta_d + \Delta_\eta \tag{16}$$

Besides the network delay, package loss can occur in the network. Unlike the network delay, package loss is generally described as information that never arrives at the destination. In the NCS approach, different types of package loss are known; newer arrived, out of order, and multiple package arrivals. In the TT-NCS approach, the state observer with a controller on the server-side is mainly used to recover the loosed packages [8]. In the ET technique, the package loss stability criteria can be analyzed regarding the Lyapunov stability function of the reaching phase in the ET-SMC operation, where the package loss is modeled such as the error, $e_p(t) = \xi(t) - \xi(t_\kappa)$, for time $t \in [t_k, t_{k+1})$, where $\kappa \in \mathbb{Z}_{\geq 2}$. The state $\xi(t_\kappa)$ presents the last update after the package loss. The number of packages lost is equal to $\kappa - 1$, which $\kappa = 2$ means one lost package. The proof of the stability is similar to the proof presented in (8), where the Lyapunov function is equal to $V(t,\kappa) = s(t,\kappa)^2/2$, and its derivative is

$$\begin{aligned}\dot{V} &= s((c_1 - b)\xi_2(t) - g v_{ET}(t_\kappa) + \tilde{d}(t)) \\ &= s\left((c_1 - b)\underbrace{(\xi_2(t) - \xi_2(t_\kappa))}_{e_p(t)} - \rho sign(s(t_\kappa)) + \tilde{d}(t)\right) \\ &\le s(c_1 - b)e_p(t) - |s|\rho + |s|\Delta_d \\ &\le |s|\beta_p - |s|\rho + |s|\Delta_d \\ &\le -|s|(\rho - \beta_p - \Delta_d)\end{aligned}$$

Regarding the assumption $\kappa > n$ it holds $\beta_p \ge \beta$. After a consecutive package lost, the system is stable if the controller gain ensure the given condition,

$$\rho > \beta_p + \Delta_d \tag{17}$$

The ξ_1 trajectory is bounded by

$$\|\xi_1\| < \frac{k_p \|c\|}{\|c_1\| \|(c_1 - b)\|} \beta_p \,, \tag{18}$$

where is $k_p = \sqrt{1 + \frac{\|(c_1 - b)\|^2 \alpha^2}{\beta_p^{\,2}}}$. After solving the differential equation $\frac{d}{dt}\|e(t,\kappa)\|$ given in (12), the minimal inter-event time is

$$\tau_{\min_p} \ge \frac{1}{\|A_c\|} \ln\left(\frac{k_p \beta_p \|A_c\|}{\|(c_1 - b)\| \left(\|A_c\| \|\xi(t_\kappa)\| + \|B_c\|\rho + \|B_d\|\Delta_d \right)} + 1 \right) \tag{19}$$

It is evident that the package loss higher the boundary of the output trajectory ξ_1. If the output boundary needs to be in the prescribed range (11), (18), the controller gain and inter-event time (14), (19) need to be selected at lower values. The closed-loop performance needs to be reduced to ensure higher robustness of the network uncertainties. The package loss can be detected with $TT_{s,c}$ measurement on both sides of the network. With the proper selection of the $maxTT_{s,c}$, and delay parameter Δ_η, the desirable performance of the closed-loop system can be ensured; otherwise, the lowered closed-loop or unstable behavior can occur.

5. Experimental Results

The dual servo system is used for the validation of the presented ET-NCS approach. The servo system is presented in Fig 5. The client is implemented on the ARM® Cortex®-M7 based STM32F7xx MCU with an IwIP stack for transparent UDP communication with the presented NCS protocol presented in Figs. 3 and 4. The IwIP stack on STM32F7 enables 100BASE-TX communication speed. The ARM embedded system is responsible for the measured current, velocity, and angle of the servo system and provides actuation to the motor drive, with Pulse Width Modulation (PWM) at the frequency of *10kHz* and resolution *4mV/duty*. All the measurements before the transmission are preprocessed with different signal processing algorithms to ensure the high fidelity and reliability of the measured data. The used brushed motors in Fig. 5 have a maximal velocity of *3500RPM* at *24V* and max load current *4A*.

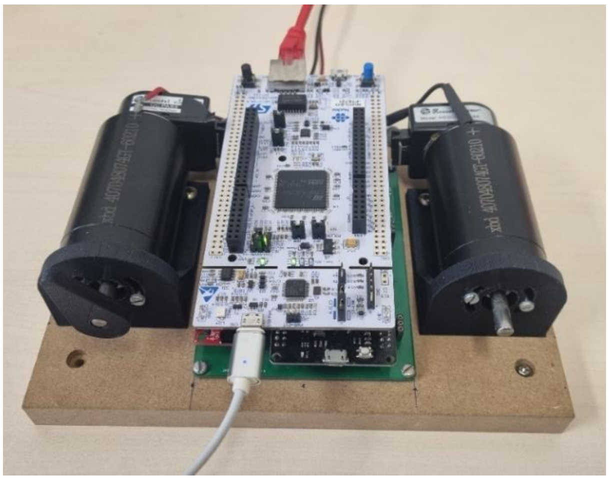

Fig. 5 Real-time system with network socket

The network is composed of an ARM embedded system, router and PC-server. The embedded system provides a request for the controller update, which is sent to the server. The request message structure is defined with the protocol presented in Fig. 3. After the client's received message, the server calculates the new controller output and prepares the server message back to the client, Fig. 4. The used network is presented in Fig. 6.

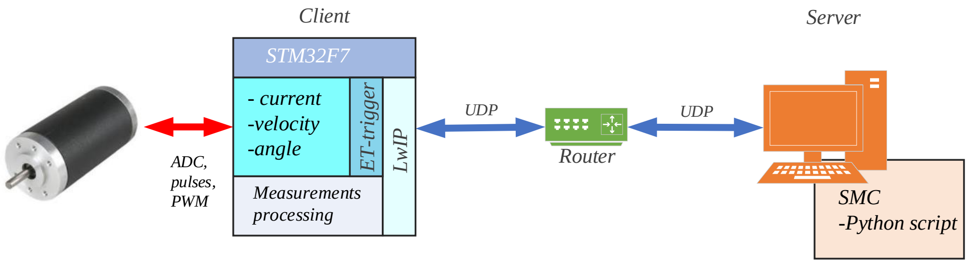

Fig. 6 NCS-Network configuration

The sliding mode controller is implemented with Python 3.7. The main components of the Python script are running the UDP server with additional timer interrupt threats for TT implementation and RTTs measurements.

The closed-loop performance for TT and ET implementation is evaluated with the given performance indices,

$$RMS_w = \sqrt{\frac{1}{n_s}\sum_{k=1}^{n_s} w^2}, \quad w \in \{x_1, s, v_{TT}, v_{ET}\}, \tag{20}$$

$$Flag_v = \sum_{i=0}^{n_s-1} n_i,\ n_i = \begin{cases} 0 \text{ for } u\{\|\xi_2(t)\| \le \|c_1 - a\|^{-1}\beta\} \\ 1 \text{ for } u\{\|\xi_2(t)\| > \|c_1 - a\|^{-1}\beta\} \end{cases}, \tag{21}$$

where n_s and n_i are the numbers of triggering events for controllers v_{TT} and v_{ET} respectively. The controller v_{TT} stands for the TT execution of the controller algorithm presented in (4) as v. The controllers v_{TT} and v_{ET} are tested in the same condition, with equal references values and a sampling time of 10*ms* for TT execution and periodic triggering evaluation for v_{ET} execution $\tau_n \geq 10ms$ (15). The parameters of the system presented in (1), (2) are, $b = 3.3$, $g = 0.897$, $\Delta_d = 7.1$.The selected controller parameters are, $c_1 = 5.2$, $\rho = 16.2$, $\beta = 19.7$, $\beta_p = 19.7$, $TT_{s,c} = 11ms$.

The network performance is presented in Fig. 7.

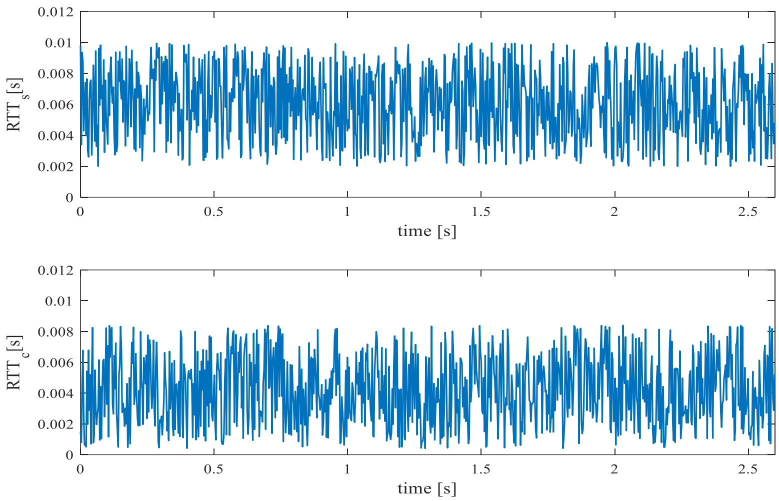

Fig. 7 Measured RTT_s and RTT_c values of the NCS network

The periodic triggering evaluation is selected properly regarding the measured *RTT* values for server and client $\tau_{trigger} = 10ms$. In each $\tau_{trigger}$ period, only measured data are examined concerning the triggering boundary β.

Figs. 8 and 9 present the NCS performance of the controller $v = v_{TT}$.

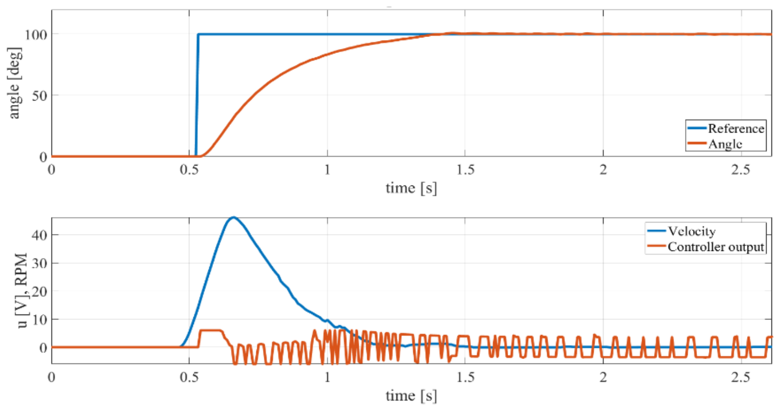

Fig. 8 Tracking capability, RPM value, and v_{TT} controller output of the TT- NCS

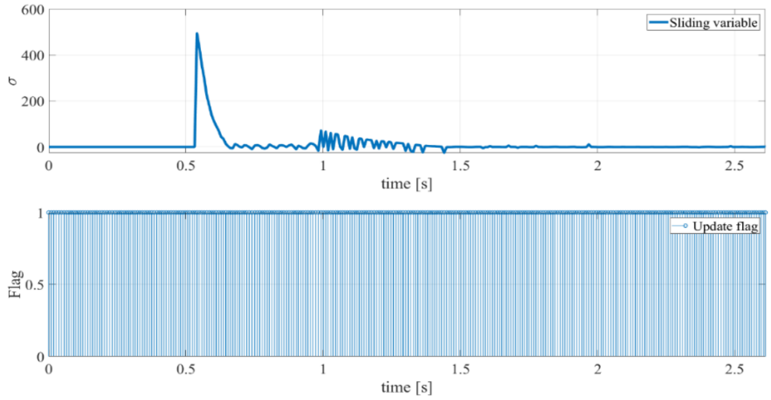

Fig. 9 Sliding variable and controller update flag of the TT- NCS

Figs. 10 and 11 present the NCS performance of the controller v_{ET}

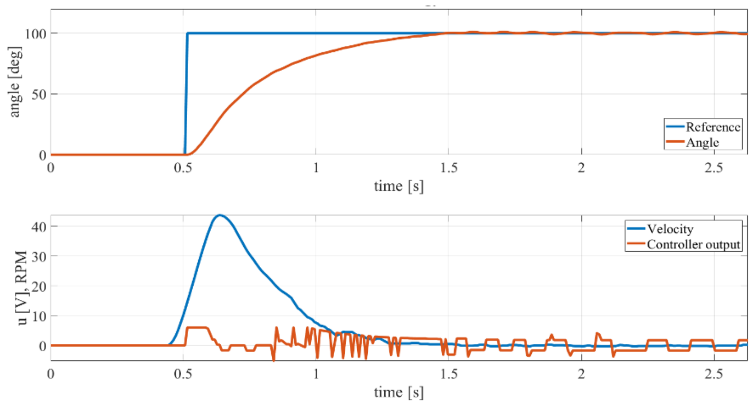

Fig. 10 Tracking capability, RPM value, and v_{ET} controller output of the ET- NCS

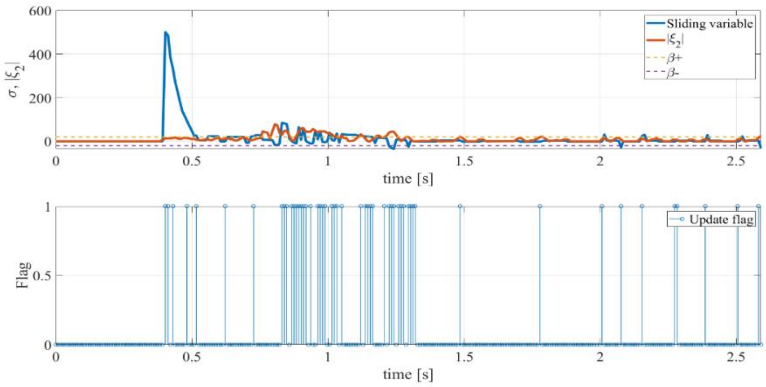

Fig. 11 Sliding variable and controller update flag of the ET- NCS

The estimated indices values (20),(21) are presented in Table 1.

Table 1 Performance indices of TT-NCS and ET-NCS

NCS	RMS_{x1}	RMS_v	RMS_s	$avg(T_s/\tau_n)$	*RTTs*	*RTTc*	*Flag*
v_{TT}	83.2	4.56	57.2	10*ms*	8.23*ms*	3.21*ms*	100%
v_{ET}	85.7	1.82	58.4	41*ms*	8.43*ms*	2.78*ms*	28.7%

Figs. 8-11 show the implementation results of the TT-NCS and ET-NCS strategies. The advantages of both approaches are shown clearly. The TT-NCS has better tracking performance regarding Table 1 and the RMS_{x1} value. This result was expected, due to the constant controller update with a prescribed sampling time of 10ms. On the other hand, the TT approach uses constant network resources. For a given experiment, at least two messages are transmitted in each 10ms time frame. Regarding RMS_{x1} of the ET-approach, the tracking capability has a slightly deteriorated response. The lower performance is the result of the nonlinear switching function of v and the unstable boundary region of the output x_1 variable derived in (11) and the triggering condition. The network usage in the ET-strategy is reduced drastically, especially when the system reaches a sliding manifold. The average update time for ET-NCS is 41ms, presented in column $avg(T_s/\tau_n)$ of Table 1. The average update time is related closely to the preselected triggering boundary and the course of the reference value. The triggering boundary affects the tracking capability of the closed-loop system directly. The employment of the ET-NCS system is a tradeoff between network resources usage and the accuracy of the system. In the given experiment, the network usage of the tracking system is reduced by almost 70%, and the output RMS_v value is reduced drastically. The ET approach can also be considered a chattering alleviation technique for sliding mode controllers with an explicated output signum function, which is studied extensively within different implementation techniques and adaptation algorithms [18]-[21].

6. CONCLUSION

The paper presents the event-triggering nonlinear controller implementation for a networked control system. Compared to the classic time triggering implementation, the approach is beneficial for the NCS system with data rate constraints, where the network constraints can be considered during the controller design. The experimental results confirm the theoretical assumptions of ET-NSC and derivation. The network usage and embedded system utilization are reduced. The ET technique can be a viable alternative for TT feedback systems, especially where the computational and network resources are limited or the optimization subject. The work is a good research starting point for multi-agent, distributed control, and task scheduling in embedded systems. The central supervised server system can share its computation capacity with other distributed systems and control multiple sub-plants remotely, where the relaxation of network requests can be lowered significantly and pre-estimated.

Acknowledgement: *This research was funded by the Slovenian Research Agency (ARRS) Grant number P2-0065.*

REFERENCES

[1] A. Sarjaš and D. Gleich, "Nonlinear Event-Triggered Networked Feedback Control System under Data-rate Constrains", In Proceedings of the 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), 2021, pp. 376-379.

[2] A. K. Behera, B. Bandyopadhyay and X. Yu, "Periodic event-triggered sliding mode", *Automatica*, vol. 96, pp. 1916-1931, Jan. 2018.

[3] V. I. Utkin, *Sliding modes in control and optimization*. New York: Springer-Verlag, 1992.

[4] C. Edwards and S. K. Spurgeon, *Sliding Mode Control: Theory And Applications* Taylor and Francis, 1998.

[5] I. Furtat, Y. Orlov and A. Fradkov, "Finite-time sliding mode stabilization using dirty differentiation and disturbance compensation", *Int. J. Robust Nonlinear Control*, vol. 29, no. 3, pp. 793-809.

[6] K. J. Aström, "Event based control" In A. Astolfi and L. Marconi (Eds.), *Analysis and design of nonlinear control systems*, pp. 127-147, Berlin, Heidelberg, Springer, 2006.

[7] K. J. Åström and B. M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first-order stochastic systems", In Proceedings of the 41st IEEE Conference on Decision and Control (CDC), Las Vegas, NV, USA, 2002, pp. 2011-2016.

[8] A. Ferrara, G. P. Incremona and V. Stocchetti, "Networked sliding mode control with chattering alleviation", In Proceedings of the 53th IEEE Conference on Decision Control, Los Angeles, CA, USA, December 2014, pp. 5542-5547.

[9] E. Kofman and J. H. Braslavsky, "Level Crossing Sampling in Feedback Stabilization under Data-Rate Constraints", In Proceedings of the 45th IEEE Conference on Decision Control (CDC), San Diego, CA, USA, Dec. 2006, pp. 4423-4428.

[10] J. Ludwiger, M. Steinberger, M. Horn, G. Kubin and A. Ferrara, "Discrete Time Sliding Mode Control Strategies for Buffered Networked Systems", In Proceedings of the 57th IEEE Conference on Decision Control, Miami Beach, FL, USA, Dec. 2018, pp. 6735-6740.

[11] M. Cucuzzella, G. P. Incremona and A. Ferrara, "Event-triggered variable structure control", *Int. J. Control*, vol. 93, no. 2, pp. 252-260, Jan. 2019.

[12] J. Ludwiger, M. Steinberger and M. Horn, "Spatially Distributed Networked Sliding Mode Control", *IEEE Control Syst. Lett.*, vol. 3, no. 4, pp. 972-977, May 2019.

[13] J. Ludwiger, M. Steinberger, M. Horn, G. Kubin and A. Ferrara, "Discrete Time Sliding Mode Control Strategies for Buffered Networked Systems", In Proceedings of the 57th IEEE Conference on Decision Control, Miami Beach, FL, USA, Dec. 2018, pp. 6735-6740.

[14] A. K. Behera and B. Bandyopadhyay, "Event-triggered sliding mode control for a class of nonlinear systems", *Int. J. Control*, vol. 89, no. 9, pp. 1916-1931, Jan. 2016.

[15] A. K. Behera, B. Bandyopadhyay and X. Yu, "Periodic event-triggered sliding mode", *Automatica*, vol. 96, pp. 1916-1931, Jan. 2018.

[16] A. K. Behera and B. Bandyopadhyay, "Robust sliding mode control: An event-triggering approach", *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 64, no. 2, pp. 146-150, Feb. 2017.

[17] W. Gao, Y. Wang and A. Homaifa, "Discrete-time Variable Structure Control System", *IEEE Trans. Ind. Electron.*, vol. 42, no. 2, pp. 117-122, April 1995.

[18] S. Koch and M. Reichhartinger, "Discrete-time equivalents of the super-twisting algorithm", *Automatica*, vol. 107, pp. 190-199, 2019.

[19] B. Brogliato and A. Polyakov, "Digital implementation of sliding-mode control via the implicit method: A tutorial", *Int. J. Robust Nonlinear Control*, vol. 31, no. 9, pp. 3528-3586, 2021.

[20] V. Utkin, "Discussion aspects of high-order sliding mode control", *IEEE Trans. Automat. Contr.*, vol. 61, pp. 829-833, 2016.

[21] U. P. Ventura and L. Fridman, "Design of super-twisting control gains: A describing function based methodology", *Automatic*, vol. 99, pp. 175-180, 1990.