Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-15 (2019) 412–417

Efficient networked UAV control using event-triggered predictive control [⋆]

Dohyun Jang [*] **Clark Youngdong Son** [*] **Jaehyun Yoo** [**]
H. Jin Kim [*] **Karl H. Johansson** [***]

[*] *Department of Mechanical and Aerospace Engineering, Seoul National University (SNU), Seoul, South Korea (e-mail: dohyun,clark.y.d.son,hjinkim@snu.ac.kr)*
[**] *Department of Electrical, Electronic and Control Engineering, Hankyong National University, Anseong, South Korea (e-mail: jhyoo@hknu.ac.kr)*
[***] *The School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: kallej@kth.se)*

Abstract: In this paper, we propose a method to improve the networked UAV control system using event-triggered control and model predictive control (MPC). Although the UAV control over the network has many advantages, it involves a long-time delay and packet loss, which adversely affect real-time control performance. Delay compensation algorithms in the networked control system (NCS) have been proposed to address such issues, however, they do not consider the resource limit of the network so that the network congestion may occur. In that case, the packet loss and network delay issues can even be worsened. In this study, we propose a method to reduce the generation of less important control signals and to use the network more efficiently by using event-triggered control. Since the event-triggered control method is also influenced by the network delay, an event trigger function suitable for NCS is designed. We validated the effectiveness of networked UAV control system and event-triggered control by simulation.

Keywords: Aerospace Systems, Networked Control System, Event-triggered Control, Model Predictive Control, Delay Compensation

1. INTRODUCTION

A networked UAV control system, i.e., an UAV integrated with a networked control system (NCS), has many advantages for various UAV applications [Yoo and Johansson (2017); Quaritsch et al. (2010)]. Networked environments such as Ethernet provide an attractive option for UAV operations because of low cost, widely available infrastructure, and multiple-access efficiency [Tang and de Silva (2006)].

However, alongside these advantages, typical characteristics of communication networks present limitations for the networked UAV control. The issues such as network latency, packet dropout, jitter, and band limitation prevent the accurate state observation of the plant from the server and deteriorate the control performance by restricting the control signal of the server.

Several studies have been conducted to address these problems of NCS. In [Seiler and Sengupta (2001)], Markovian jump linear system (MJLS) is considered to cope with packet dropout. It can design a controller such that the closed loop is mean-square stable using linear matrix inequality (LMI). Zhang and Yu (2008) presents a technique for simultaneously considering network delay and packet loss using a switched system model and an average dwell time method. Under the constant latency assumption, jitter and band limitation are not considered. Tipsuwan and Chow (2003); Zuo et al. (2017) tried to model the network delays using least squares support vector machines (LSSVM). However, they only model the approximate tendency of network delays, thus cannot cope with other network issues.

Model predictive control (MPC)-based techniques have also been studied for NCS. In [Liu (2008); Tang and de Silva (2006); Cortes et al. (2012); Montestruque and Antsaklis (2003)], the predictive control provides a local plant with a sequence of predicted control inputs. The predicted signal is converted to an appropriate signal reflecting the network delays and sent to the plant. This method can cope with network latency, jitter, and packet dropout, but does not consider limited network resources.

In order to recognize band limitation problems and efficiently use network resources, an event-based control method has been introduced [Miskowicz (2018); Heemels et al. (2012); Dimarogonas et al. (2012); Eqtami et al. (2010)]. The event control method generates control signals in a non-periodic manner in accordance with a specific criterion, as opposed to conventional periodic control. This type of resource-aware implementation can reduce network

---

[⋆] This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2019-0-00399, Development of A.I. based recognition, judgement and control solution for autonomous vehicle corresponding to atypical driving environment)

10.1016/j.ifacol.2019.11.710

utilization while maintaining plant stability, which can be synergetic with NCS. The event-based control has been applied to UAV in Durand et al. (2014); Vega-Alonzo et al. (2016), but they do not consider various issues related to NCS.

The main contribution of this paper is as follows. We propose an event-based networked UAV control to deal with the issues of NCS mentioned above. We design a control structure to compensate the network delays and packet dropout. MPC is applied to the trajectory generation problem to predict the future UAV state that will be used for delay compensation. Although Yoo and Johansson (2017); Jang et al. (2018) dealt with techniques for utilizing NCS to UAV control, network band restriction has not been considered. To avoid the congestion of the network, we utilize the event-based control with hierarchical MPC. The event generator compares the estimated state of the UAV to the previous predicted state using the MPC trajectory generator. If the event trigger function exceeds a threshold, a new MPC control input set is sent to the UAV.

The remainder of this paper is comprised as follows. Section 2 deals with the basic structure of model predictive control and its modification for networked predictive control. Section 3 proposes an algorithm to compensate the network-induced problems. Section 4 introduces an event-triggered control method and proposes a control method to reduce network congestion. Section 5 explains the effectiveness of the network compensation algorithm and event triggering technique, using simulation. The final section discusses the results and improved control performance.

2. NETWORKED PREDICTIVE CONTROL

In this section, we show how to formulate the model predictive control techniques for the design of networked UAV control structure. The prediction horizon and control inputs resulting from MPC are applied to the network compensation algorithm in Section 3 and the event trigger function in Section 4. First, we describe the multi-rotor type UAV model used in MPC and explain the optimal control process. Then, we propose a modified MPC formulation for networked UAV control.

2.1 Dynamic model

We consider a multi-rotor type UAV. The position and velocities of the UAV in the inertial frame are defined as $\mathbf{p} := [x\ y] \in \mathbb{R}^2$, and $\mathbf{v} := [\dot{x}\ \dot{y}] \in \mathbb{R}^2$. The state variables of the UAV are defined as $\mathbf{x} := [\mathbf{p}^T\ \mathbf{v}^T]^T := [x\ y\ \dot{x}\ \dot{y}]^T \in \mathbb{R}^4$. Even though the proposed setting works in the same manner in 3-D, we use 2-D notation for brevity. The state variables include only the position and velocity variables of the UAV because the attitude of the UAV changes rapidly, so either estimating or measuring the current attitude in the network environment with network delay is not reasonable.

As derived in Jang et al. (2018), we approximate the UAV model as a 1_{st} order linear dynamics and set the control input $\mathbf{u} := [u_x\ u_y]^T \in \mathbb{R}^2$ as the desired velocity to be sent to the UAV's velocity controller. As a result, the dynamics of the UAV is given as

$$\dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u}, \tag{1}$$

$$A_c \triangleq \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1/\tau_x & 0 \\ 0 & 0 & 0 & -1/\tau_y \end{bmatrix}, \tag{2}$$

$$B_c \triangleq \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/\tau_x & 0 \\ 0 & 1/\tau_y \end{bmatrix}. \tag{3}$$

The time constants τ_x and τ_y in (2) and (3) can be determined experimentally [Kamel et al. (2017)]. The following discretized system is obtained from (1):

$$\begin{aligned} \mathbf{x}(k+1) &= A_d\mathbf{x}(k) + B_d\mathbf{u}(k) \\ &= f(\mathbf{x}(k), \mathbf{u}(k)), \end{aligned} \tag{4}$$

where A_d and B_d in (4) correspond to A_c and B_c in continuous system, respectively.

2.2 Model predictive control

To predict the future state of the UAV, we rewrite the trajectory generation as an optimal control problem, with boundary conditions defined by the UAV's initial and desired states. The cost function J_k to be minimized is chosen as

$$\begin{aligned} \min_{\mathbf{u}(k+i|k), 0 \le i < H} J_k &= ||\mathbf{x}(k+H) - \mathbf{x}_d(k+H)||_P^2 \\ + \sum_{i=0}^{H-1} &(||\mathbf{x}(k+i) - \mathbf{x}_d(k+i)||_Q^2 + ||\mathbf{u}(k+i|k)||_R^2) \end{aligned} \tag{5}$$

subject to

$$\begin{aligned} \mathbf{x}(k+i+1) &= f(\mathbf{x}(k+i), \mathbf{u}(k+i|k)) \\ |\mathbf{u}(k+i|k)| &\le \mathbf{u}_{max} \\ i &= 0, \cdots, H-1 \\ \mathbf{x}(k) &= \hat{\mathbf{x}}(k). \end{aligned} \tag{6}$$

where $||\cdot||_A^2$ denotes the square of a weighted Euclidean norm with a positive semi-definite weighting matrix A. We use (4) as the constraint in this optimization. $\hat{\mathbf{x}}(k)$ is the estimated state obtained in the downlink delay compensator described in Section 3.3 and $\mathbf{x}_d$ is a desired positon. MPC calculates the predicted state up to the look-ahead horizon of H steps. P, Q, and R are weighting matrices for the final state error, state error at the i_{th} time step , and control input, respectively. The vector $\mathbf{u}_{max}$ denotes the limit of control input. The results of MPC are prediction horizon and control horizon denoted as

$$X(k) = \{\mathbf{x}(k+1|k), \mathbf{x}(k+2|k), \cdots, \mathbf{x}(k+H|k)\}, \tag{7}$$

$$U(k) = \{\mathbf{u}(k|k), \mathbf{u}(k+1|k), \cdots, \mathbf{u}(k+H-1|k)\}, \tag{8}$$

i.e. the predicted state variables and corresponding optimal control input during H. We use a CVXGEN library, which generates a fast custom code for small, QP-representable convex optimization problems [Mattingley and Boyd (2012)].

In Section 3.3, the downlink delay compensator transfers $\mathbf{x}(k-i)$ to the next step $\mathbf{x}(k-i+1)$ at each $k-i$ time step using $\mathbf{u}(k-i|k-i)$. However, the UAV actually uses $\mathbf{u}(k-i|k-i-\tau_u) \in U(k-i-\tau_u)$ as a result of the uplink

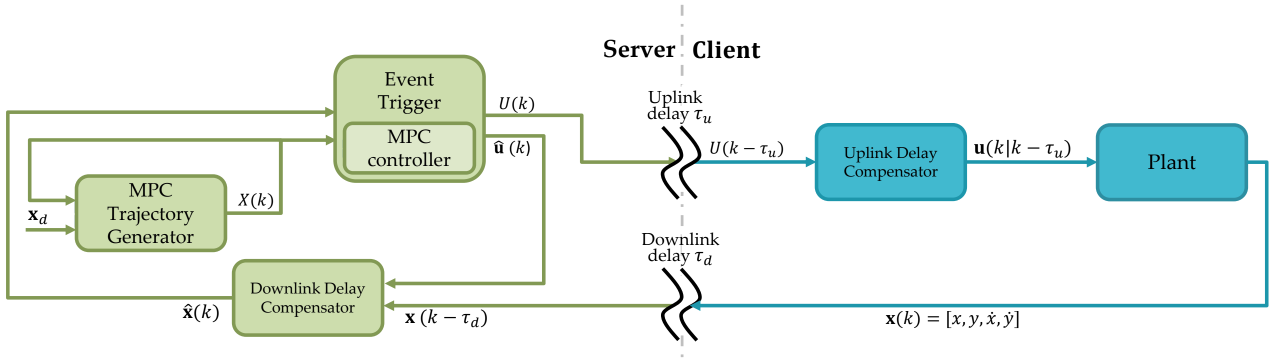


Fig. 1. The overall structure of the proposed algorithm: ① MPC trajectory generator (in server) makes prediction horizon $X(k)$, ② the event trigger (in server) checks the condition of signal generation and sends control horizon $U(k)$, ③ the uplink delay compensator (in client) chooses a proper control input $\mathbf{u}^*$ in the delayed control horizon $U(k-\tau_u)$ according to the current time, ④ the inner control loop (in client) performs cascade control and sends full-state observation $\mathbf{x}(k)$ of the UAV, ⑤ the downlink delay compensator (in server) estimates the current state of UAV $\hat{\mathbf{x}}(k)$ by compensating for downlink delay

delay compensator, and the downlink delay compensator does not know τ_u as it changes every moment. The best strategy is to make $\mathbf{u}(k+i|k)$ and $\mathbf{u}(k+i|k-1)$ as close as possible. Consequently, the server can choose the $(k-i)$-th step control input to minimize the difference from the actual control input used. To account for this change to the MPC constraint, redefine the cost function of (5) as follows:

$$\begin{aligned} \min_{\mathbf{u}(k+i|k),0\le i<H} \; & J_k = ||\mathbf{x}(k+H)-\mathbf{x}_d(k+H)||_P^2 \\ + \sum_{i=0}^{H-1} & (||\mathbf{x}(k+i)-\mathbf{x}_d(k+i)||_Q^2 + ||\mathbf{u}(k+i|k)||_R^2) \\ & + \sum_{i=0}^{H-2}(||\mathbf{u}(k+i|k)-\mathbf{u}(k+i|k-1)||_S^2) \end{aligned} \quad (9)$$

Including the last term will reduce optimal performance, but will improve stability of NCS. (9) can be represented as function for brevity:

$$\{X(k), U(k)\} \leftarrow MPC\{\{\mathbf{x}_d(k+i), \mathbf{x}(k+i)\}_{i=0}^{H}\}. \quad (10)$$

3. NETWORK DELAY COMPENSATION

In this section, we analyze network-induced problems arising from the networked control and introduce a structure to compensate them. The overall structure shown in Fig. 1 is divided into a server part and a client part. The server is responsible for trajectory generation and control input generation, and the client includes the UAV. The network connecting the two parts is divided into an uplink channel and a downlink channel, and an uplink delay compensator and a downlink delay compensator are designed for each.

3.1 Network delay analysis

There are typically three disadvantages that the network has on control: time delay, packet dropout, and band limitation. The time delay is the time it takes for a packet transmitted from the source to arrive at the destination, which is the sum of the latency of each node. It can also have a different time delay per transmission, including jitter. Packet dropout means that the packet being transmitted is corrupted during transmission or fails to reach its destination. This irregular long delay can degrade control performance due to outdated signal feedback. Band limitation is also an important issue. If the data rate exceeds the capacity threshold, it can cause network congestion, which leads to the time delay and packet dropout problems mentioned above. We design compensators for uplink channel and downlink channel to solve random packet arrival problems in this chapter, and apply an event trigger with hierarchical MPC to solve band limitation in Chapter 4.

3.2 Uplink delay compensation

The client receives the packet including the control horizon and the time stamp. However, due to the uplink delay τ_u, the client receives a packet at τ_u time later than the time it was created. τ_u can be calculated by comparing the time stamp included in the data packet with the time when the client receives the packet. During this delay, the UAV follows the previous trajectory so that the current UAV state is expected to be at the predicted position $\mathbf{x}(k|k-\tau_u) \in X(k-\tau_u)$. The uplink delay compensator chooses the proper control input $\mathbf{u}^*$ in the delayed control horizon $U(k-\tau_u)$ according to the current time,

$$\mathbf{u}^* = \mathbf{u}(min(k, k-\tau_u+H-1)|k-\tau_u) \in U(k-\tau_u). \quad (11)$$

$\mathbf{u}^*$ is given to the UAV velocity controller. Since we assume that there are intermittent packet dropouts, the delayed control horizon $U(k-\tau_u)$ may not reach the client in every step. In this case, the uplink delay is recalculated based on the most recently received packet, and the delay compensation algorithm is applied. Also, jitter may occur, and the timing of the received packets may be mixed. In this case, the delay compensation algorithm is applied based on the latest generated packet among the received packets.

3.3 Downlink delay compensation

The downlink delay τ_d can be calculated by comparing the time stamp included in the data packet with the time when the server receives the packet. In (4), we assume that the

UAV follows the linear model. The server stores the history of past control inputs $\{\hat{\mathbf{u}}(i)\}_{i=k_0}^{k}$ as follows:

$$\hat{\mathbf{u}}(k) = \mathbf{u}(k|k) \in U(k). \tag{12}$$

When the delayed state observation $\mathbf{x}(k-\tau_d)$ is given, $\hat{\mathbf{x}}(k)$ is calculated by the following equation:

$$\hat{\mathbf{x}}(k) \triangleq A_d^{\tau_d}\mathbf{x}(k-\tau_d) + \sum_{i=1}^{\tau_d} A_d^{i-1} B_d \hat{\mathbf{u}}(k-i). \tag{13}$$

As with the uplink delay compensator, the delayed state observation $\mathbf{x}(k-\tau_d)$ may not reach the server in every step because of the packet dropout. The uplink delay is recalculated based on the most recently received packet, and the delay compensation algorithm is applied. The response of the downlink compensator to jitter is also the same as that of the uplink compensator.

4. EVENT-TRIGGERED CONTROL

The network delay compensation algorithms introduced in the previous chapter are suitable for compensating for network delays, jitter and packet loss, but do not consider network band limitation. In this section, an event trigger function is designed, and an event-triggered control technique is introduced. The main idea of our event-triggered control is that the knowledge of the UAV dynamics is utilized to reduce the usage of the network bands. We propose a hierarchical model predictive control for event-triggered control by modifying MPC introduced in Section 2, and design event trigger function that can be applied to the networked UAV control system.

4.1 Hierarchical MPC

In this paper, MPC described in Section 2 is used for two purposes in a hierarchical manner. The first one is the trajectory generator for the desired position, and the other is the networked predictive control for the uplink delay compensation. Unlike common MPC settings where trajectory generation and control input generation are performed in a single process, the two MPC blocks are utilized. The first MPC is used for trajectory generation based on the estimated state of the UAV, and the second MPC is used for judging the trajectory tracking error of the UAV and utilizing it as a criterion of event occurrence. These are described in Section 4.2.

First, we utilize MPC in Section 2 to generate the trajectory leading to the desired position. Using (10), the first MPC is defined as follows:

$$\{X_p(k), \sim\} \leftarrow MPC\{\mathbf{x}_d(k), \mathbf{x}_p(k|k-1)\}. \tag{14}$$

$X_p(k)$ is the prediction horizon for time H, and $\mathbf{x}_d$ is the desired position. $\mathbf{x}_p(k|k-1) \in X_p(k-1)$, which is the predicted position of the previous time $k-1$, is used as the current position of the UAV in MPC, because it assumes that the UAV is following well the predicted trajectory generated at the previous time. By using the event trigger function described in Chapter 4.2, it is possible to discriminate whether the UAV actually follows the predicted trajectory or not, and if not, MPC control is performed to generate the control signal. Using (10), the second MPC is as follows:

$$\{\sim, U(k)\} \leftarrow MPC\{X_p(k), \hat{\mathbf{x}}(k)\}. \tag{15}$$

The generated control signal includes the predicted control inputs of horizon H and is transmitted to the UAV together with the timestamp. Even if the control input is not generated because the event trigger function is not triggered, the uplink delay compensator described in Chapter 3 can select the current control input from the previous control input horizon. As a result, the frequency of the control signal generation is reduced. In other words, you can mitigate network load by reducing the frequency of events.

4.2 Event trigger function

The event trigger function mentioned in the previous section is defined as follows:

$$\begin{aligned} V(k) &= ||\hat{\mathbf{x}}(k) - \mathbf{x}_p(k|k-1)||_P^2 \\ \mathbf{x}_p(k|k-1) &\in X_p(k-1) \\ S(k) &= V(k_i)e^{-\lambda(k-k_i)} + \epsilon \\ k &\in [k_i, k_{i+1}), \end{aligned} \tag{16}$$

where λ is a decay rate, ϵ is a tolerance term, which has the effect of relaxing control signal generation by tolerating a certain level of error. Section 5.2 shows how these parameters affect networked control performance. $V(k)$ represents the error between the current position of the UAV and the position on the predicted trajectory in quadratic form. $\mathbf{x}_p$ is the current position in the predicted trajectory $X(k-1)$ generated by the MPC trajectory generator. However, since the current position $\mathbf{x}(k)$ cannot be confirmed due to the downlink delay, the estimated position $\hat{\mathbf{x}}(k)$ is used.

$S(k)$ is used for the event occurrence condition. It shows whether $V(k_i)$ attenuates at the rate of $e^{-\lambda(k-k_i)}$ after the time of the most recent event occurrence k_i. As a result, $V(k)$ and $S(k)$ are compared, and the event is generated when $S(k)$ becomes smaller than $V(k)$. This entire process can be described as Algorithm 1. Moreover, the whole networked UAV control structure including the event-triggered control process can be summarized as Algorithm 2.

Algorithm 1 Event trigger function

```
Input: $X_p(k-1), \hat{\mathbf{x}}(k)$ for $k \in [k_i, k_{i+1})$
Output: $U(k), \hat{\mathbf{u}}(k)$
Calculate the MPC optimal control horizon
    $\{\sim, U(k)\} \leftarrow MPC\{X_p(k), \hat{\mathbf{x}}(k)\}$
Save $\hat{\mathbf{u}}(k)$ for the downlink delay compensator
    $\hat{\mathbf{u}}(k) \leftarrow \mathbf{u}(k|k) \in U(k)$
Calculate the event trigger functions $V(k), S(k)$
Check whether the event occur or not
if $V(k) > S(k)$ then
    $k_{i+1} \leftarrow k$
    $i \leftarrow i + 1$
    return $U(k), \hat{\mathbf{u}}(k)$
else
    return $\hat{\mathbf{u}}(k)$
end if
Repeat until the UAV state meets the final desired state
```

Algorithm 2 Networked UAV control

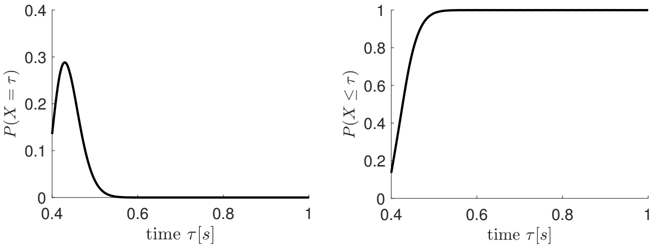
```
1: Trajectory generation (in server)
       $\{X_p(k), \sim\} \leftarrow MPC\{\mathbf{x}_t(k), \mathbf{x}_p(k|k-1)\}$
2: Event trigger (in server)
       if event triggered then
           $\{U(k_i), \hat{\mathbf{u}}(k)\} \leftarrow EVENT\{X_p(k-1), \hat{\mathbf{x}}(k)\}$
3: Uplink delay compensation (in client)
       $\mathbf{u}^* \leftarrow \mathbf{u}(min(k, k-\tau_u+H-1)|k-\tau_u) \in U(k-\tau_u)$
4: UAV inner control loop (in client)
       $\mathbf{x}(k+1) \leftarrow f(\mathbf{x}(k+1), \mathbf{u}^*)$
5: Downlink delay compensator (in server)
       $\hat{\mathbf{x}}(k) \leftarrow A_d^{\tau_d}\mathbf{x}(k-\tau_d) + \sum_{i=1}^{\tau_d} A_d^{i-1}B_d\hat{\mathbf{u}}(k-i)$
6: Repeat until the UAV state meets the final desired state
```

5. SIMULATION RESULTS

We present several simulation results in this section to illustrate the effectiveness of the delay compensation algorithms and networked UAV control system. At first, we use the multi-rotor type UAV model. The control structure is designed as shown in the Fig. 1, and we assume that the data links between the server part and the client part have the following network delays:

$$P(X=\tau) = \frac{2^{\lfloor(\tau-0.4)/0.02\rceil}e^{-2}}{\lfloor(\tau-0.4)/0.02\rceil}, \quad \tau > 0.4 \tag{17}$$

$\lfloor\rceil$ means the nearest integer. The packet dropout rate is 0.1. It is modified from the Poisson distribution with the minimum latency of 0.4 seconds. This distribution reflects randomly occurring delay components in a network queue. A time interval between each event is 0.02 seconds. Fig. 2 shows the probability density function (PDF) and the cumulative distribution function (CDF) for the above delay function.

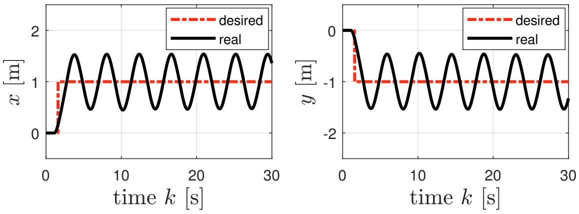


(a) Network delays PDF (b) Network delays CDF

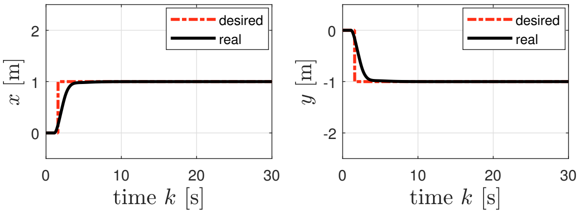
Fig. 2. The network delays of the data links between the server and the client

5.1 Simulation 1: Networked UAV control with delay compensation algorithms

In this simulation, we conduct the networked UAV position control in the presence of both network delays and packet dropout. It tests the effect of delay compensation, without event triggering. Fig. 3 compares two cases: (a) position control without delay compensation, (b) position control with the proposed delay compensation algorithm. Even though the network limitation deteriorates the UAV control, position control performance of (b) is satisfactory because of the proposed algorithm.

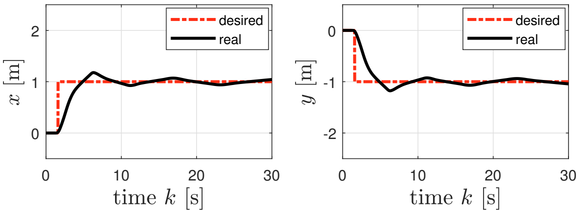


(a) without delay compensation, without event-triggered control

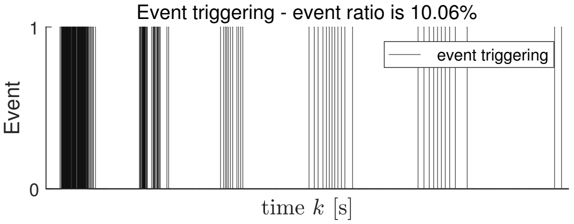


(b) with delay compensation, without event-triggered control

Fig. 3. Simulation 1: networked UAV control with delay compensation algorithms. The red dash line represents the desired position, the black solid line represents UAV position.



(a) with delay compensation, with event-triggered control



(b) with delay compensation, with event-triggered control

Fig. 4. Simulation 2: Networked UAV control with delay compensation algorithms and event-triggered control. The top figures show the UAV position control result, and the bottom figure shows the control event occurrence ratio.

5.2 Simulation 2: Networked UAV control with delay compensation and event-triggered control algorithms

The second simulation more realistically reflects the issues of the network. In other words, the event-triggered method to reduce the frequency of the UAV control signal is applied in consideration of the band limitation of the network. We set $\epsilon = 3.6$ and $\lambda = 1.5$. Compared with Fig. 3-(b), Fig. 4-(a) shows that the position control performance is degraded. However, the control signal generation rate can be significantly reduced as shown in Fig. 4-(b). In particular, when used with the delay compensation algorithm, the event generation rate can be reduced 10 times smaller without significantly deteriorating the control performance. It means that it can reduce the load on the network band and reduce the chance of jitter and packet dropout. Table 1 shows the comparison of the position control performance of simulation 1 and simulation 2, numerically showing the effect of the algorithms mentioned above.

Fig. 5 shows the analysis of control performance and event occurrence ratio according to event function parameters. As the decay rate λ increases, the event occurrence ratio also increases, and as the tolerance ϵ increases, the event occurrence ratio decreases. The event occurrence ratio and RMSE of the trajectory are inversely related as expected.

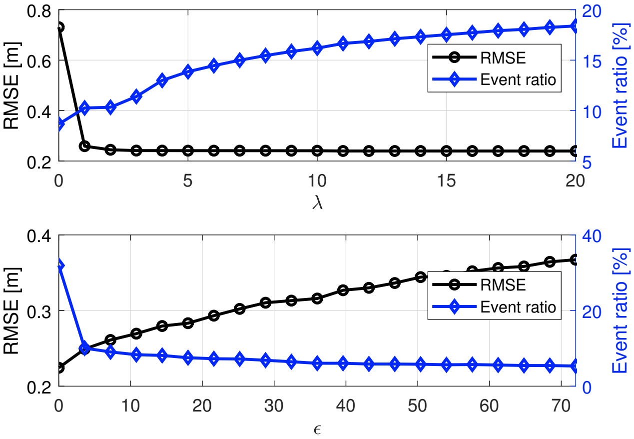


Fig. 5. Analysis of control performance (RMSE) and event occurrence ratio according to the event function parameters (λ and ϵ in (16))

Table 1. RMSE to the desired trajectory

	Sim 1-(a)	Sim 1-(b)	Sim 2
x RMSE	0.3724	0.1068	0.1762
y RMSE	0.3719	0.1057	0.1761
xy RMSE	0.5263	0.1502	0.2492

6. CONCLUSION

This paper proposes an event-based networked UAV control to cope with the network issues. We design a control structure to compensate the network delays and packet dropout. To avoid the congestion of the network, we utilize the event-based control scheme with hierarchical MPC. The effectiveness of networked UAV control system and event-triggered control was validated by simulation.

REFERENCES

Cortes, P., Rodriguez, J., Silva, C., and Flores, A. (2012). Delay compensation in model predictive current control of a three-phase inverter. *IEEE Transactions on Industrial Electronics*, 59(2), 1323–1325.

Dimarogonas, D.V., Frazzoli, E., and Johansson, K.H. (2012). Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5), 1291–1297.

Durand, S., Dumon, J., Marchand, N., and Guerrero-Castellanos, J.F. (2014). Event-based control for embedded and networked system application to a mini quadrotor helicopter using motion capture. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, 1188–1195. IEEE.

Eqtami, A., Dimarogonas, D.V., and Kyriakopoulos, K.J. (2010). Event-triggered control for discrete-time systems. In *American Control Conference (ACC), 2010*, 4719–4724. IEEE.

Heemels, W., Johansson, K.H., and Tabuada, P. (2012). An introduction to event-triggered and self-triggered control. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 3270–3285. IEEE.

Jang, D., Yoo, J., and Kim, H.J. (2018). Tracking control of a multirotor uav in a network environment with time-varying delay. In *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, 612–617. IEEE.

Kamel, M., Stastny, T., Alexis, K., and Siegwart, R. (2017). Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. In *Robot Operating System (ROS)*, 3–39. Springer.

Liu, G.P. (2008). Analysis and design of networked predictive control systems. In *Networked Control Systems*, 95–119. Springer.

Mattingley, J. and Boyd, S. (2012). CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering*, 13(1), 1–27.

Miskowicz, M. (2018). *Event-Based Control and Signal Processing*. Chapman and Hall/CRC.

Montestruque, L.A. and Antsaklis, P.J. (2003). On the model-based control of networked systems. *Automatica*, 39(10), 1837–1843.

Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., and Rinner, B. (2010). Networked uavs as aerial sensor network for disaster management applications. *e & i Elektrotechnik und Informationstechnik*, 127(3), 56–63.

Seiler, P. and Sengupta, R. (2001). Analysis of communication losses in vehicle control problems. In *Proceedings of the American Control Conference*, volume 2, 1491–1496.

Tang, P.L. and de Silva, C.W. (2006). Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. *IEEE Transactions on Control Systems Technology*, 14(4), 707–718.

Tipsuwan, Y. and Chow, M.Y. (2003). Control methodologies in networked control systems. *Control engineering practice*, 11(10), 1099–1111.

Vega-Alonzo, A., Guerrero-Castellanos, J., Durand, S., Marchand, N., Maya-Rueda, S., and Mino-Aguilar, G. (2016). Event-based control strategy for consensus of a group of vtol-uavs. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on*, 1–8. IEEE.

Yoo, J. and Johansson, K.H. (2017). Learning communication delay patterns for remotely controlled uav networks. *IFAC-PapersOnLine*, 50(1), 13216–13221.

Zhang, W.A. and Yu, L. (2008). Modelling and control of networked control systems with both network-induced delay and packet-dropout. *Automatica*, 44(12), 3206–3210.

Zuo, T., Min, H., Zhang, T., and Zhang, X. (2017). The self-tuning networked control system with online delay prediction. *Transactions of the Institute of Measurement and Control*, 39(9), 1365–1373.