An Integrated Control Architecture for a Cloud-based Unmanned Aerial Vehicle System with Lossy Networks

An Integrated Control Architecture for a Cloud-based Unmanned Aerial Vehicle System with Lossy Networks

Weibin Gu[1], Carlos Perez-Montenegro[2], Elisa Capello[3], Alessandro Rizzo[2]

***Abstract*— Unmanned Aerial Vehicles (UAVs) have been successfully employed in cooperative tasks over recent years, particularly in the applications for smart cities. In such scenario, a networked control system (NCS) framework is usually adopted since measurement and actuation data are mainly transmitted through communication networks. This paper proposes an integrated control architecture for a Cloud-based fixed-wing UAV system, where the control logic resides in the Cloud and sensing and actuating signals are transmitted over a realistic wireless network. The proposed control strategy leverages model predictive control (MPC) and a specialized Kalman filter in combination with two *ad-hoc* buffers, which enables simultaneous compensation for measurement and control input packet dropouts. Simulations of a nonlinear aircraft model show the effectiveness and advantages of proposed integrated scheme over an existing linear quadratic (LQ)-based control strategy.**

I. INTRODUCTION

Recently, much attention has been devoted to the use of unmanned aerial vehicles (UAVs) in cooperative tasks. Thanks to the increasing sensing and communication capabilities and the reduced cost, UAVs are now ready for smart cities and Internet of Things (IoT) applications [1], [2].

In such a scenario, most of the navigation and control intelligence is still installed on-board the UAVs. There, a large amount of navigation and control data is required to be processed in real time, which is traditionally limited to the computational power of on-board autopilots. Besides, in the context of smart cities, the use of UAVs poses stringent requirements, due to the potential risks of hurting people and causing catastrophic damages. Consequently, control units for UAVs with high performance that guarantee safety and reliability are usually designed with complex control algorithms that generally include on-line optimization processes and/or with high sampling rate, which brings additional computational burden on-board. To overcome these issues, we envisage a Cloud-based control architecture, where most of the intelligence is located in the Cloud, which exchanges with UAVs a few data, mostly limited to sensing and actuation. It consists of an UAV front-end, the Cloud, which encompasses both high- and low-level control features, and a communication network. All the sensor data are transmitted to the Cloud by UAVs through the communication network and processed in the Cloud. Then, command signals are sent back to actuators to execute the planned control action, establishing, in fact, a networked control system (NCS).

The key features of the proposed Cloud-based control system are the reduction of the on-board computational power and the relaxation of the on-board hardware and software requirements. Placing low-level control (flight control) units along with high-level ones (including unmanned traffic management, path planning, situation awareness, etc.) into the Cloud provides convenience and flexibility to reconfigure flight control remotely using the data available in the Cloud, without the necessity of recalling UAVs for reprogramming. Such concept has already been used and validated in applications in different fields, such as the networked control of DC-motors [3], and seems to be very promising for multi-UAV systems. Despite the presence of a network unavoidably yields drawbacks, such as band-limited channels, sampling and delays, packet dropouts, which might lead to performance degradation or even cause instability in the closed-loop control systems [4].

Different control strategies and compensation methods have been proposed for NCS with lossy networks in recent years. To deal with sensor-to-controller dropouts, estimator models for missing measurements can be found, for example, customized Kalman filter [5], optimal estimator with deterministic filter gain [6], and linear minimum mean square error (LMMSE) estimator [7]. In different manners, compensation methods for controller-to-actuator data loss are zero-/hold-input strategy [8] and generalized hold-input strategy [9], which are mainly adopted in the Linear Quadratic Gaussian (LQG)-based control framework. Alternatively, predictive-input strategy is deemed to have better performance thanks to the use of a sequence of "future" control inputs as backup solution when control input packet is lost. Control scheme with such strategy is referred to as networked predictive control (a.k.a. sequence-based control, packetized predictive control) [7], [10]–[12], which usually incorporates model predictive control (MPC) due to its intrinsic prediction nature. Motivated by [5] and [7], our core idea is to design a controller, based on the theory of MPC, capable of handling input constraints. Meanwhile, a complete design of control system is addressed, not only an estimator as did in [7], and compensation of missing packets in both communication channels are included, as main difference with [10].

In this paper, a novel control strategy is proposed to deal with packet dropouts, induced by a wireless communication network for Cloud-based UAV control applications. The

[1]Weibin Gu is with the Department of Electrical and Computer Engineering, University of Denver, Colorado, USA weibin.gu@du.edu

[3]Elisa Capello is with the Department of Mechanical and Aerospace Engineering, Politecnico di Torino and with CNR-IEIIT, Torino, Italy elisa.capello@polito.it

[2]Carlos Perez-Montenegro and Alessandro Rizzo are with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy {carlos.perez, alessandro.rizzo}@polito.it

proposed control strategy is part of the low-level control unit devoted to a single UAV. Besides tracking a given flight trajectory autonomously, the controller should be capable of compensating for both measurement and control input packet dropouts induced by the network. The main novelties with respect to the state of the art are: (i) the ability of managing control-related constraints; (ii) the ability to treat packet dropouts both in the actuation and in the measurement channels of the control system; (iii) the ability to deal with partial-state feedback in the control loop; and (iv) the design and validation for a real-time complex system.

The proposed controller integrates an MPC-based controller with a customized Kalman filter for lossy networks (KFLN) [5], making up for partial state feedback and providing backup sequences for measurement packet dropouts. Moreover, two synchronized buffers, one on-board and one on-Cloud, are included to manage the backup control input sequence. We show that providing predicted backup series for missing control inputs yields better performance than retaining the last available control signal (hold-on strategy) or setting to zero the control signal (zero-input strategy) [5].

The remainder of this paper is organized as follows. Section II describes the mathematical model adopted for the fixed-wing UAV employed in the Cloud-based UAV system. Section III addresses the proposed integrated control framework, including the MPC controller, the KFLN and the two buffers. Simulation results with a fully nonlinear aircraft model are presented to assess the effectiveness of the resulting control architecture in Section IV. Finally, conclusions are drawn in Section V.

II. MATHEMATICAL MODEL OF THE FIXED-WING AIRCRAFT

Our control design starts from an open-source nonlinear model of a fixed-wing aircraft UltraStick-25e, provided by the UAV Laboratories of the University of Minnesota [13], [14], and proceeds with linearization process using MATLAB®&Simulink® dedicated tools. The nonlinear model is implemented for simulation purposes, as described in Section IV. Employed symbols and parameter values adopted in this paper are summarized in Table I. Table II summarizes all the trim conditions and resulting trimmed values from the linearization process.

For control purposes, we follow the common assumption that the linear model of the aircraft can be decoupled into longitudinal and lateral-directional motion dynamics [15]. The longitudinal state vector is $\mathbf{x}_{\mathrm{full-lon}} = (u, w, q, \theta)^T$ and the corresponding control input is $\mathbf{u}_{\mathrm{lon}} = \delta_e$. To further simplify the control process, only the short-period mode is used, which has a truncated state vector $\mathbf{x}_{\mathrm{lon}} = (w, q, \theta)^T$. The lateral state vector is $\mathbf{x}_{\mathrm{lat}} = (v, p, r, \phi)^T$ and the corresponding control input vector is $\mathbf{u}_{\mathrm{lat}} = (\delta_a, \delta_r)^T$. Indicating with $k \in \mathbb{Z}^+$ the discrete time index and using boldface and capital letters to refer to vectors and matrices, respectively, the discrete-time state-space linearized model of the aircraft is expressed as follows:

TABLE I

SYMBOLS OF THE MATHEMATICAL MODEL OF UAV

Name	Symbol	Unit
Roll Angle	ϕ	rad
Pitch Angle	θ	rad
Yaw Angle	ψ	rad
Angular Velocity in body frame	$(p, q, r)^T$	$\mathrm{rad\,s^{-1}}$
Linear Velocity in body frame	$(u, v, w)^T$	$\mathrm{m\,s^{-1}}$
Aircraft Mass	m	kg
Aircraft Inertia Matrix	J	$\mathrm{kg\,m^2}$
Airspeed	V_a	$\mathrm{m\,s^{-1}}$
Above Gound Level (AGL)	h	m
Flight Path Angle	γ	rad
Throttle	δ_{th}	nd
Elevator Deflection	δ_e	rad
Rudder Deflection	δ_r	rad
Aileron Deflection	δ_a	rad

TABLE II

TRIM CONDITIONS (1-3) AND TRIMMED VALUES (4-13)

ID	Variable	Value	Unit
1	V_a	17	$\mathrm{m\,s^{-1}}$
2	γ	0	rad
3	h	100	m
4	$(\phi, \theta, \psi)^T$	$(\text{-}1.7\text{e-}03, 5.4\text{e-}02, 2.7)^T$	rad
7	$(p, q, r)^T$	$(3.95\text{e-}08, \text{-}1.88\text{e-}07, 2.61\text{e-}09)^T$	$\mathrm{rad\,s^{-1}}$
10	δ_{th}	5.69e-01	nd
11	δ_e	-9.63e-02	rad
12	δ_r	3.2e-03	rad
13	δ_a	1.0e-02	rad

- Longitudinal motion:

$$\begin{aligned} \mathbf{x}_{\mathrm{lon}}(k+1) &= A_{\mathrm{lon}}\mathbf{x}_{\mathrm{lon}}(k) + B_{\mathrm{lon}}\mathbf{u}_{\mathrm{lon}}(k) \\ \mathbf{y}_{\mathrm{lon}}(k) &= C_{\mathrm{lon}}\mathbf{x}_{\mathrm{lon}}(k), \end{aligned} \tag{1}$$

- Lateral-directional motion:

$$\begin{aligned} \mathbf{x}_{\mathrm{lat}}(k+1) &= A_{\mathrm{lat}}\mathbf{x}_{\mathrm{lat}}(k) + B_{\mathrm{lat}}\mathbf{u}_{\mathrm{lat}}(k) \\ \mathbf{y}_{\mathrm{lat}}(k) &= C_{\mathrm{lat}}\mathbf{x}_{\mathrm{lat}}(k), \end{aligned} \tag{2}$$

where $A_{\mathrm{lon}} \in \mathbb{R}^{3\times3}$, $A_{\mathrm{lat}} \in \mathbb{R}^{4\times4}$ denote system matrices, $B_{\mathrm{lon}} \in \mathbb{R}^{3\times1}$, $B_{\mathrm{lat}} \in \mathbb{R}^{4\times2}$ input matrices and $C_{\mathrm{lon}} \in \mathbb{R}^{1\times3}$, $C_{\mathrm{lat}} \in \mathbb{R}^{1\times4}$ output matrices.

The control objective is to track reference signals θ_{ref} for longitudinal dynamics and ϕ_{ref} for lateral dynamics. Since, in practice, these two references are generated by altitude and heading angle tracking systems, a cascade control scheme is conventionally employed in aeronautical applications [15]. This scheme separates the control loop into an inner and an outer one, as detailed in the next section.

III. INTEGRATED CONTROL FRAMEWORK

A. System Design

As stated in the introduction, the architecture of the NCS adopted in this work corresponds to the low-level control unit for a single UAV. The following assumptions are considered.

- The communication network is packet-based and TCP-like, hence an acknowledgment signal (*ACK*) is available to the controller.
- Sensors, controller and actuators are synchronized and time-stamp information for each packet is available.

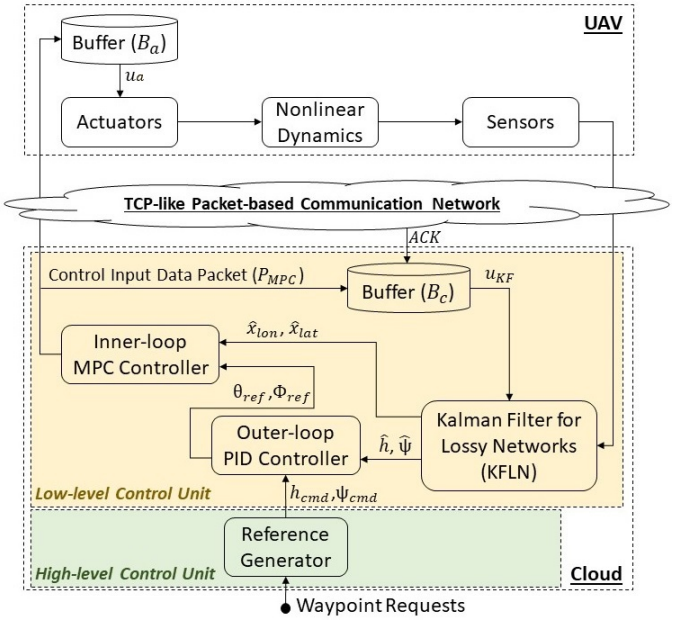

Fig. 1. The proposed control architecture

- Sensors are clock-driven while controller and actuators are event-driven.
- Packet dropouts are the main source of network malfunctioning and they may occur both in the sensor-to-controller and the controller-to-actuator channel.

The main objective of this work is to mitigate the effects of missing measurement and control input data packets caused by network malfunctioning, while guaranteeing an acceptable performance in autonomous trajectory tracking. Figure 1 illustrates the proposed control scheme. The KFLN receives available measurements and provides a full state estimate (denoted by the symbol " ˆ ") based on the status of the measurement packets transmitted by the sensors, the available control input, and the statistics of noise. The outer-loop controller is responsible for altitude and heading angle tracking, carried out by a proportional-integral-derivative (PID) control [16], [17]. The inner-loop controller adopts an MPC-based control strategy, which produces a control input sequence that will drive the inner-loop variables θ and ϕ as close as the references provided by the outer-loop PID controller. The buffer at actuators is designed to provide a backup control input sequence in case of packet dropouts due to network malfunctioning. The inputs of KFLN $\mathbf{u}_{\mathrm{KF}}$, based on *ACK* information, and the actuator inputs $\mathbf{u}_a$ are synchronized thanks to the introduction of the controller buffer $\mathbf{B}_{\mathrm{c}}$ (see Fig. 1). In the following, we address the design of the MPC controller, the buffers and the KFLN.

B. Model Predictive Control

Since MPC is an effective control strategy especially for constrained control problems, a large amount of research has been put forward over the last decade making it a considerably mature technique. Maciejowski [18] proposes solutions to theoretical issues based on stability and feasibility, while Oettershagen [19] presents practical solutions to enable the applicability of MPC on systems with fast dynamics. In our design, an MPC-based estimator is implemented to make the controller insensitive to errors in the steady-state gain and unknown constant additive disturbances.

The MPC is formulated for the discrete time-invariant aircraft models given in (1) and (2). An incremental state space model is formulated as in (3) using the control increment $\Delta\mathbf{u}$ as system input to directly impose constraints on the deflection rate of control surfaces in the optimization. In the sequel, we present the MPC control formulation for the longitudinal motion control, hence, the subscript "lon" is dropped from matrices and vectors of Eq. (1) to enhance readability. The same strategy can be adopted for lateral motion control, which we assume as decoupled from the longitudinal one, as previously mentioned. Due to space constraint and to enhance readability, we omit the description of lateral control in this paper.

The dynamical model of the system is described as

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= M\bar{\mathbf{x}}(k) + N\Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= Q\bar{\mathbf{x}}(k), \end{aligned} \tag{3}$$

where

$$\begin{aligned} \bar{\mathbf{x}}(k+1) \triangleq \begin{pmatrix} \mathbf{x}(k+1) \\ \mathbf{u}(k) \end{pmatrix} \in \mathbb{R}^{4\times 1} \quad N = \begin{bmatrix} B \\ I \end{bmatrix} \in \mathbb{R}^{4\times 1} \\ M = \begin{bmatrix} A & B \\ \mathbf{0} & I \end{bmatrix} \in \mathbb{R}^{4\times 4} \qquad Q = \begin{bmatrix} C & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{1\times 4}, \end{aligned} \tag{4}$$

Manipulating the incremental equations yields the explicit prediction model employed in MPC

$$\mathbf{Y}(k) = F\hat{\mathbf{x}}(k) + H\Delta\mathbf{U}(k), \tag{5}$$

where

$$\begin{aligned} &\mathbf{Y}(k) \triangleq (\hat{\mathbf{y}}(k+1|k)\cdots\hat{\mathbf{y}}(k+N_{\mathrm{p}}|k))^T \in \mathbb{R}^{N_{\mathrm{p}}\times 1}, \\ &\Delta\mathbf{U}(k) \triangleq (\Delta\mathbf{u}(k)\cdots\Delta\mathbf{u}(k+N_{\mathrm{c}}))^T \in \mathbb{R}^{N_{\mathrm{c}}\times 1}, \\ &F = [QM\cdots QM^{N_{\mathrm{p}}}]^T \in \mathbb{R}^{N_{\mathrm{p}}\times 4}, \\ &H \in \mathbb{R}^{N_{\mathrm{p}}\times N_{\mathrm{c}}} \text{ is a lower triangular matrix, } H_{ij} = QM^{i-j}N. \end{aligned} \tag{6}$$

Parameters N_{p} and N_{c} represent the prediction and control horizon, respectively. Variable $\hat{\mathbf{y}}(m|n)$ denotes the estimate of $\mathbf{y}$ at time m, given observations time n included ($n < m$) and $\hat{\mathbf{x}} \in \mathbb{R}^{4\times 1}$ is the state estimate.

The prediction model described above generates biased predictions when model-plant mismatches occur, particularly if the steady-state gain of the model is inaccurate. To mitigate this issue, the traditional MPC is slightly modified by adding a disturbance estimate term $\mathbf{D}$ leading to an improved prediction model

$$\mathbf{Y}_{\mathrm{up}}(k) = F\hat{\mathbf{x}}(k) + H\Delta\mathbf{U}(k) + \mathbf{D}(k), \tag{7}$$

where

$$\begin{aligned} d(k) &= \mathbf{y}_{\mathrm{p}}(k) - \hat{\mathbf{y}}(k|k-1) \in \mathbb{R}^{1\times 1} \\ \mathbf{D}(k) &= \mathbf{1}d(k),\ \mathbf{1} \in \mathbb{R}^{N_{\mathrm{p}}\times 1}. \end{aligned} \tag{8}$$

The term $\mathbf{D}$ represents the discrepancy between the last plant output $\mathbf{y}_\mathrm{p}$ and the latest prediction model output, assuming that it will continue to act throughout the prediction horizon.

Based on the unbiased prediction model (7) and taking into account actuator constraints, the control sequence is derived by solving the following optimization problem:

$$\begin{aligned} \min_{\Delta\mathbf{U}(k)} \quad & \|\mathbf{Y}_\mathrm{up}(k) - \mathbf{Y}_\mathrm{ref}(k)\|^2_{Q_y} + \|\Delta\mathbf{U}(k)\|^2_{R_{\Delta\mathbf{u}}} \\ \text{s.t.} \quad & \text{(7) with system dynamics (1) or (2)} \\ & \Delta\mathbf{u}_\mathrm{min} \le \Delta\mathbf{U}(k) \le \Delta\mathbf{u}_\mathrm{max} \\ & \mathbf{u}_\mathrm{min} \le \mathbf{u}(k) \le \mathbf{u}_\mathrm{max}, \end{aligned} \tag{9}$$

where $\mathbf{Y}_\mathrm{ref} \in \mathbb{R}^{N_\mathrm{p}\times 1}$ is the reference signal vector, $Q_y, R_{\Delta u} \in \mathbb{R}^{N_\mathrm{p}\times N_\mathrm{p}}$ are positive-definite penalty matrices on tracking error and control increments, $\|x\|^2_A$ denotes a quadratic form $\mathbf{x}^T A\mathbf{x}$ of an arbitrary vector $\mathbf{x}$ and matrix A, $(\Delta\mathbf{u}_\mathrm{min}, \Delta\mathbf{u}_\mathrm{max}, \mathbf{u}_\mathrm{min}, \mathbf{u}_\mathrm{max})$ are constraints on deflection rate and deflection of control surfaces. Having quadratic cost function and linear inequalities, optimization problem (9) belongs to Quadratic Programming (QP) which can be solved by several available numerical solvers. The computational complexity is not directly evaluated here since this MPC controller will be implemented on the Cloud.

C. Buffers

There are two buffers deployed in the integrated control architecture. The actuator buffer is located before the actuator block and is denoted as $\mathbf{B}_\mathrm{a}$ in Fig. 1. This is used to provide backup control input when packet dropout occurs in the controller-to-actuator channel. Each time the controller solves the MPC optimization problem (9), it sends a data packet $\mathbf{P}_\mathrm{MPC}$, containing both control input sequence and corresponding timestamps through the communication channel. Each time the buffer receives a data packet, it compares the timestamp of the incoming packet with that of the stored packet and keeps only the latest one. The controller buffer, as $\mathbf{B}_\mathrm{c}$ in Fig. 1, informs KFLN of the actual control input applied to actuators, either samples produced in real time by the controller, or samples surrogated by buffer $\mathbf{B}_\mathrm{a}$. Such a selection is operated by monitoring the *ACK* signal available in the TCP protocol.

Algorithm 1 describes the detailed working principle designed for the controller buffer, which has a similar structure with actuator buffer except the update mechanism. We denote as $\mathbf{P}_\mathrm{MPC}$ and $\mathbf{B}_\mathrm{c}$ the data packets generated by the MPC controller and those stored in the buffer, respectively. A record-like structure is envisaged for $\mathbf{P}_\mathrm{MPC}$, such as $(\mathbf{P}_\mathrm{MPC}.\mathbf{u}, \mathbf{P}_\mathrm{MPC}.T)$, where $\mathbf{u}$ is a field to store future control input sequence and T to store timestamps. The two buffers have the same length equal to the prediction horizon (i.e., $N_\mathrm{B} = N_\mathrm{p}$), and i denotes the buffer index. We also denote as $\mathbf{u}_\mathrm{KF}$ the actual control input applied to the KFLN.

The lengths of buffers are normally related to the reliability of communication networks. Experimental trials are required to find out the maximum length of consecutive packet dropouts of the employed network. Then, an appropriate length of prediction horizon should be chosen to simultaneously cope with the possible consecutive number of data loss while maintaining satisfactory nominal performance of control problem (9). Hence, a simple rule for the design of the buffer length is

$$N_\mathrm{B} = N_\mathrm{cpd} \le N_\mathrm{p}, \tag{10}$$

where N_B is the minimum length required for the buffer, N_cpd the maximum number of consecutive packet dropouts and N_p the prediction horizon.

Algorithm 1: Controller Buffer (on-Cloud)

Input: $\mathbf{P}_\mathrm{MPC}$, *ACK*
Output: $\mathbf{u}_\mathrm{KF}$

```
while Simulation runs do
    if ACK = 1 then
        // Successful transmission
        i ← 1, B_c.u ← P_MPC.u, B_c.T ← P_MPC.T
        u_KF ← B_c.u(i), i ← i + 1
    else
        // Packet dropout
        if i ≤ N_B then
            u_KF ← B_c.u(i), i ← i + 1
        else
            u_KF ← B_c.u(N_B)
        end
    end
end
```

D. Kalman Filter for Lossy Networks

The proposed KFLN is similar to the filter proposed in [5], but it plays two roles. First, as every Kalman filter, it reduces the effect of measurement and process noises. Second, it provides estimates for all the variables required in the proposed control algorithms, even if measurement data packets are lost. Different from [5] where no actuator buffer is used, the control input packet is synchronized with the control input stored at the actuator in our design. This guarantees prediction consistency and enables the use of predictive-input control strategy. The KFLN is formulated in discrete time for the longitudinal motion, with the definition of the innovation and correction steps. Similarly to the controller design, the KFLN for the lateral motion is equivalent to the one presented here and is omitted.

- Innovation Step:

$$\begin{aligned} \hat{\mathbf{x}}_\mathrm{KF}(k+1|k) &= A_\mathrm{KF}\hat{\mathbf{x}}_\mathrm{KF}(k|k) + B_\mathrm{KF}\mathbf{u}_\mathrm{KF}(k) \\ P(k+1|k) &= A_\mathrm{KF}P(k|k)A^T_\mathrm{KF} + Q_N, \end{aligned} \tag{11}$$

- Correction Step:

$$\begin{aligned} K(k+1) &= P(k+1|k)C^T_\mathrm{KF}(C_\mathrm{KF}P(k+1|k)C^T_\mathrm{KF} + R_N)^{-1} \\ \hat{\mathbf{x}}_\mathrm{KF}(k+1|k+1) &= \hat{\mathbf{x}}_\mathrm{KF}(k+1|k) + \mu(k+1)K(k+1)(\\ &\qquad \mathbf{y}_\mathrm{KF}(k+1) - C_\mathrm{KF}\hat{\mathbf{x}}_\mathrm{KF}(k+1|k)) \\ P(k+1|k+1) &= (I - \mu(k+1)K(k+1)C_\mathrm{KF})P(k+1|k), \end{aligned} \tag{12}$$

where $\hat{\mathbf{x}}_{\mathrm{KF}} = (w, q, \theta, h)^T$, $\mathbf{y}_{\mathrm{KF}} = (q, \theta, h)^T$, $\mathbf{u}_{\mathrm{KF}} = \delta_e$, A_{KF}, B_{KF}, C_{KF} are system, input and output matrices for Kalman formulation derived from linearization. Q_N, R_N are measurement and process noise covariance matrices. Finally, $P(k+1|k), P(k+1|k+1)$ are a-priori and a-posteriori estimate covariance matrices. Variable $\mu(k+1)$ in (12), i.e., sensor-to-controller transmission status, is a Bernoulli random variable that is designed to actively ignore the values of missing measurements during correction. This variable makes the estimates less sensitive to measurement packet dropouts. It is computed by checking the checksum of measurement packets and this random variable is set to one if the corresponding packet is successfully received, zero otherwise. A major improvement with respect to [5] is in the design of the control input applied in (11).

IV. SIMULATION RESULTS

A. Simulation Setup

We assess the control system performance through simulations, using the full nonlinear model of the aircraft to simulate the plant to be controlled. Here, also the lateral motion control is implemented, even though its description is omitted for the sake of brevity.

The discrete-time linearized longitudinal dynamics (1) for control design has a sample time $T_{\mathrm{s}} = 0.02\,\mathrm{s}$ with numerical state-space matrices selected in accordance with data in Table II. The state-space matrices used in the Kalman formulation (11), (12) are given by

$$A_{\mathrm{KF}} = \begin{bmatrix} 8.408\text{e-}01 & 2.471\text{e-}01 & -9.755\text{e-}03 & 0 \\ -1.165\text{e-}01 & 7.111\text{e-}01 & 6.698\text{e-}04 & 0 \\ -1.265\text{e-}03 & 1.702\text{e-}02 & 1 & 0 \\ 1.493\text{e-}04 & -3.057\text{e-}03 & -3.400\text{e-}01 & 1 \end{bmatrix}$$
$$B_{\mathrm{KF}} = \begin{bmatrix} -4.088\text{e-}01 \\ -2.272 \\ -2.402\text{e-}02 \\ 2.797\text{e-}03 \end{bmatrix} \qquad C_{\mathrm{KF}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \tag{13}$$

while the state-space matrices used in the MPC formulation (3) are $A_{\mathrm{lon}} = A_{\mathrm{KF}}(1:3, 1:3), B_{\mathrm{lon}} = B_{\mathrm{KF}}(1:3), C_{\mathrm{lon}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$.

Design parameters for MPC, KFLN and PID are the following: (i) Prediction horizon $N_{\mathrm{p}} = 10$ and control horizon $N_{\mathrm{c}} = 1$; (ii) Manipulated variables rate weight $R_{\Delta u} = 0.1$ and output variables weight $Q_y = 0.08$; (iii) Process noise covariance matrix $R_N = diag(5\text{e-}07, 5\text{e-}07, 5\text{e-}07, 1\text{e-}02)$ and measurement noise covariance matrix $Q_N = diag(5\text{e-}07, 5\text{e-}07, 1\text{e-}02)$; and (iv) Discrete PID gains $K_{\mathrm{P}} = 0.3, K_{\mathrm{I}} = 0.001$. The actuator constraints are defined in terms of deflection angle, $\delta_e = [-25, 25]$ deg and of actuation rate, $\dot{\delta}_e = [-150, 150]$ deg/s.

B. Simulation Results and Discussion

A reference trajectory is provided to the aircraft for all the simulations, starting from a straight and leveled flight at $100\,\mathrm{m}$, then giving a doublet altitude command of $10\,\mathrm{m}$, and finally maintaining altitude at $100\,\mathrm{m}$. No noise and disturbances are considered.

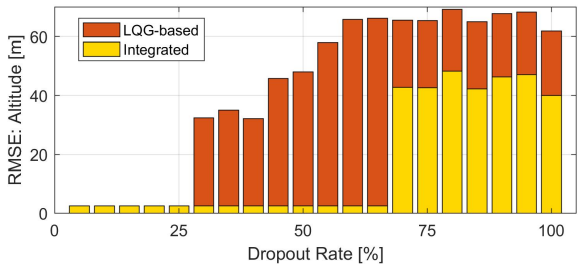

Fig. 2. Stability threshold of LQG-based and integrated control with respect to packet dropout rates.

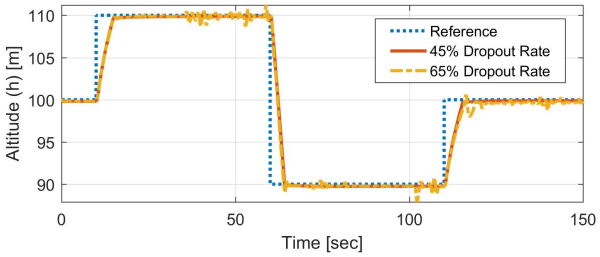

Fig. 3. Altitude tracking under 45% and 65% packet dropouts.

A comparison is firstly made with an LQG-based controller [5], relying on a similar structure of Kalman filter for lossy networks and hold-input strategy for packet dropouts. Figure 2 illustrates the stability threshold of the LQG-based and our proposed integrated controller with respect to packet dropout rates. It can be observed that our controller allows a wider operational range in terms of network conditions. Specifically, our system is stable even with 65% dropout rate (by noticing a considerably small root mean square error (RMSE) on the altitude), while the LQG-based one becomes unstable with 30% dropout rate. This is largely due to the constraint handling ability of MPC, which avoids actuator commands to easily saturate. In addition, the proposed predictive-input strategy provides more effective backup control inputs compared with either zero- or hold-input strategies, especially in cases where consecutive packet dropouts are a common issue as dropout rate increases. The tracking performance of the integrated controller with 65% dropout rate regardless of disturbances and uncertainties is shown in Fig. 3. It is remarkable that the increasing dropout rate will introduce more oscillations to the transient and steady-state behavior of the aircraft.

To further determine a more realistic stability threshold for the proposed control architecture, we take into account parametric uncertainties and variations in the initial conditions. The following simulation set offers a robust stability analysis executed through Monte Carlo simulations with 100 independent runs for several dropout rates, considering the subsequent conditions. Aircraft mass and inertia matrix are considered with a uniform distribution with 15% and the initial conditions are assigned in terms of altitude and pitch angle with a uniform distribution with 5%.

The upper graph in Fig. 4 illustrates the average tracking

trajectories of Monte Carlo simulations for three different dropout rates, with corresponding RMSE indicated in the bottom-left bar chart. The dropout rate of 65% is not included due to the occurrence of instabilities throughout the simulations, in spite of the fact that it is the stability threshold shown in Fig. 2. The bottom-right bar chart illustrates the number of unstable simulation in Monte Carlo runs for 50% (2 times), 55% (15), 60% (35) cases. Our experimental campaign lets us conclude that the stability threshold illustrated in Fig. 2 provides only a reference criteria for flight conditions without disturbances and uncertainties, whereas the real stability threshold for dropout rate should be selected in a more conservative fashion toward safe flight operations. In our case, 45% dropout rate (not shown in Fig. 4) guarantees that no instability occurs in simulations, under the selected perturbations.

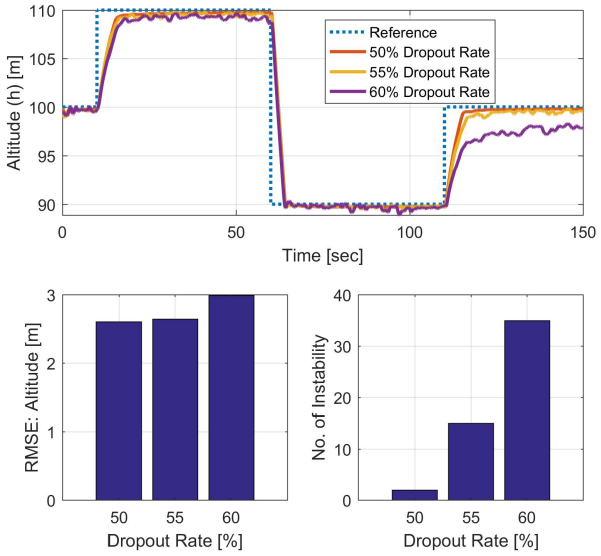

Fig. 4. 100 runs of Monte Carlo simulations for different dropout rates (50%, 55%, 60%). **Top**: Average tracking trajectory of Monte Carlo simulation results. **Bottom-left**: Altitude RMSE. **Bottom-right**: Number of instability occurrence in 100 runs.

V. CONCLUSIONS

In this paper, an integrated control architecture for NCS with lossy networks is proposed with a combination of model predictive control, a Kalman filter for lossy network, along with two buffers. The architecture is applied to the Cloud-based control of a fixed-wing UAV. Simulation results have shown advantages over an existing LQG-based strategy by revealing higher stability threshold with respect to increasing packet dropout rates. Robust stability analysis has also been performed to analyze the effect of parameter uncertainties of the aircraft and variations in initial conditions through Monte Carlo simulations for several different dropout rates. Specifically, 45% dropout rate is found to be a conservative, realistic stability threshold in our case.

VI. ACKNOWLEDGEMENT

This work has been partially supported by Siebel Energy Institute, USA, and Compagnia di San Paolo, Italy.

REFERENCES

[1] E. Vattapparamban, I. Guvenc, A. I. Yurekli, K. Akkaya, and S. Uluagac, "Drones for smart cities: Issues in cybersecurity, privacy, and public safety," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 216–221, IEEE, sep 2016.

[2] A. Fotouhi, M. Ding, and M. Hassan, "Understanding autonomous drone maneuverability for Internet of Things applications," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, IEEE, jun 2017.

[3] D. Wu, X.-M. Sun, W. Wang, and P. Shi, "Robust predictive control for networked control and application to dc-motor control," *IET Control Theory & Applications*, vol. 8, no. 14, pp. 1312–1320, 2014.

[4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proceedings of the IEEE*, vol. 95, pp. 138–162, jan 2007.

[5] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of Control and Estimation Over Lossy Networks," *Proceedings of the IEEE*, vol. 95, pp. 163–187, jan 2007.

[6] A. I. Maass, F. J. Vargas, and E. I. Silva, "Optimal control over multiple erasure channels using a data dropout compensation scheme," *Automatica*, vol. 68, pp. 155–161, 2016.

[7] F. J. Vargas, F. Cid, and A. I. Maass, "Optimal estimation in feedback control loops with packet dropouts compensation strategies," *2017 25th Mediterranean Conference on Control and Automation (MED)*, pp. 993–998, 2017.

[8] L. Schenato, "To zero or to hold control inputs with lossy links?," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1093–1099, 2009.

[9] M. Moayedi, Y. Foo, and Y. Soh, "Networked LQG control over unreliable channels," *International Journal of Robust and Nonlinear Control*, vol. 23, pp. 167–189, jan 2013.

[10] D. E. Quevedo and D. Nesic, "Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts," *IEEE Transactions on Automatic Control*, vol. 56, pp. 370–375, 2011.

[11] G. Pin and T. Parisini, "Networked Predictive Control of Uncertain Constrained Nonlinear Systems: Recursive Feasibility and Input-to-State Stability Analysis," *IEEE Transactions on Automatic Control*, vol. 56, pp. 72–87, jan 2011.

[12] J. D. Fischer, A. Hekler, M. Dolgov, and U. D. Hanebeck, "Optimal sequence-based lqg control over tcp-like networks subject to random transmission delays and packet losses," *2013 American Control Conference*, pp. 1543–1549, 2013.

[13] A. M. Murch, Y. C. Paw, R. Pandita, Z. Li, and G. J. Balas, "A low cost small UAV flight research facility," *Advances in Aerospace Guidance, Navigation and Control*, pp. 29–40, 2011.

[14] "Open-source Matlab/Simulink files." `http://www.uav.aem.umn.edu/browser/reliabilitysimulation`. UAV Laboratories, University of Minnesota.

[15] B. Etkin, *Dynamics of Flight: Stability and Control*. Wiley, 1959.

[16] M. Mammarella and E. Capello, "A robust mpc-based autopilot for mini uavs," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1227–1235, June 2018.

[17] E. Capello, G. Guglieri, F. Quagliotti, and D. Sartori, "Design and Validation of an L1 Adaptive Controller for Mini-UAV Autopilot," *Journal of Intelligent & Robotic Systems*, vol. 69, pp. 109–118, jan 2013.

[18] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.

[19] P. Oettershagen, A. Melzer, S. Leutenegger, K. Alexis, and R. Siegwart, "Explicit model predictive control and L<inf>1</inf>-navigation strategies for fixed-wing UAV path tracking," *22nd Mediterranean Conference on Control and Automation*, pp. 1159–1165, 2014.