Resilient control of a small drone to man-in-the-middle cyberattacks

Alexandru Codrean
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
alexandru.codrean@aut.utcluj.ro

Octavian Ștefan
Department of Automation and Applied Informatics
Politehnica University Timișoara
Timișoara, Romania
octavian.stefan@aut.upt.ro

Ramona Roxana Pașcalău
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Pascalau.Mi.Ramona@student.utcluj.ro

Zsófia Lendek
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
zsofia.lendek@aut.utcluj.ro

***Abstract*—Cybersecurity is becoming a pressing issue in networked control systems and cyber-physical systems. The current paper proposes a resilient networked control methodology for small drones, in the case of man-in-the-middle cyberattacks. An adaptive sliding mode controller is used as a low level mitigation approach, together with redundant encrypted data for high level mitigation. This combination ensures both robustness and resilience for the networked control of a drone with limited computational resources. The whole approach is validated through experiments on a Parrot Mambo drone.**

***Index Terms*—networked control system, small drone, sliding mode control, cryptography, cyberattack.**

I. Introduction

Small unmanned aerial vehicles (sUAVs) are being used in a wide range of applications, from education to industry, in research, military and commercial applications [1]. Besides low level control of such drones, of increasing interest is also high level planning, coordination and cooperation with other drones or robots [2], [3]. This leads to different types of networked control structures, especially in the case of small drones with limited computational resources and limited autonomy. Networked control, despite the multiple advantages [4], also makes the system more exposed to different types of cyberattacks [5], [6]. Consequently, the topic of resilience to cyberattacks [3], with special application to drones [7], [8], has become a research topic of high interest.

The classical categories of cyberattacks are disclosure attacks (confidentiality), deception attacks (integrity), and disruption attacks (availability) [9]. Cyberattacks on drones can also be classified from other perspectives. From a control perspective, cyberattacks can be at sensor level, actuator level, process level, or controller level [5]. From a different point of view, closer to computer science, a cyberattack can be at the sensor level, hardware level, software level or communication level [7]. In this paper our focus is on deception attacks at the communication level, among which the man-in-the-middle attack (false data injection) is the most common [7], [5].

In order to increase the resilience of the system, there is an increasing need [3] to integrate high level cyberattack mitigation approaches from computer science with low level mitigation based on control theory. In the case of stealthy cyberattacks, where the attack can be easily confused with disturbances at the process level (e.g. wind, unmodelled dynamics), the need of integrating both approaches - low level and high level - becomes even more significant. By a stealthy attack we mean a certain type of man-in-the-middle (false-data injection) attack that cannot be detected from the input-output data at the process level [10].

In the current paper, we propose a new framework for resilient networked control of small drones with limited computational capabilities, consisting of the integration of both low level and high level cyberattack mitigation methods. Our starting point is the networked control structure for small drones from our previous work [11], where the network is only on the direct path, while the feedback path is closed with the help of measurements from a motion capture system. Thus, our focus is on stealthy man-in-the-middle cyberattacks on the commands sent to the drone. The interplay between a robust control method (low level) and an encryption based method (high level) provides a higher degree of resilience of the system to cyberattacks, while also taking into account the limited computational resources of a small drone. We show the effectiveness of our approach with real time experiments on a Parrot Mambo drone.

The structure of the paper is as follows: Section II provides the problem formulation and the initial networked control structure, Section III describes the methodology for resilient networked control, Section IV presents experimental results

A. Codrean was supported by project ARUT no. 3/1.07.2024, funded through the GNAC ARUT 2023 competition. O. Ștefan was supported by the project "Romanian Hub for Artificial Intelligence - HRIA", Smart Growth, Digitization and Financial Instruments Program, 2021-2027, MySMIS no. 334906.

on a Parrot Mambo drone, while the last section draws the conclusions.

II. Networked control structure and problem formulation

We consider the networked control structure for a small drone from Figure 1. The control structure consists of an inner loop, running onboard the drone, and an outer loop implemented on a remote computer. Splitting the control structure into an inner loop and outer loop relies on the assumption that the inner loop is much faster than the outer loop. The attitude (pitch, roll, yaw) and altitude (height Z) control is handled by an inner control loop. The tracking for the X and Y positions is done by the outer loop controller, using the measurements from motion capture cameras (e.g. OptiTrack cameras). It is important to note that the network is only on the direct path, and the feedback is directly connected to the remote computer.

On one hand, there are several benefits for adopting such a control structure for small drones: i) it alleviates the drone hardware limitations by moving a part of the control algorithm on a remote computer; ii) permits more accurate position measurements using motion capture cameras, instead of onboard measurements and estimations (e.g using optical flow); iii) transmission delays and packet loss due to wireless transmissions can be minimized by sending smaller data packets one way. On the other hand, the control system becomes vulnerable to different types of cyberattacks, like man-in-the-middle or denial-of-service [7], [5]. However, we do not consider here the case of severe cyberattacks, which can lead even to complete interruption of the network communication channel, because in such cases the best solution is a local back-up control which ensures a safe drone landing.

In particular, in this work, our focus is on man-in-the-middle stealthy attacks. Through stealthy attack we mean the case when the attacker tries to modify the control commands sent to the drone without being detected, for as long as possible. It is important to take into account that, for a man-in-the-middle attack, we can not influence how the attacker alters the data packets sent through the network, and if a countermeasure is detected, the attacker could switch to a major cyberattack, which can lead to total corruption of the data packets (equivalent to temporary communications loss forcing the drone to enter a safe mode). Since the attacker only has access to the data sent through the network on the direct path (see also Figure 1), he cannot actually notice the effects of the attack on the drone (e.g. in terms of drone position etc.). Moreover, he cannot detect any countermeasure besides possible modifications on the data packets sent on this path. Thus, the countermeasure that we adopt needs to also be stealthy with respect to the data sent over the network.

Thus, the problem that we solve is the following: develop a methodology which combines both low-level and high-level cyberattack mitigation, in order to ensure the resilience of the overall system.

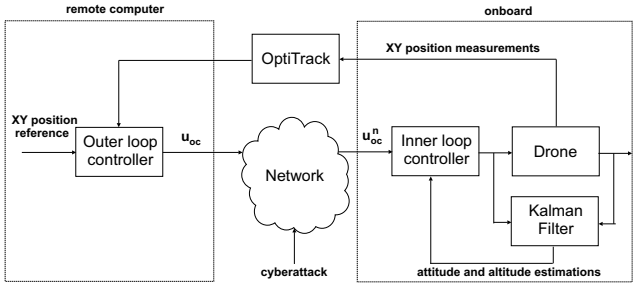

Fig. 1: Networked control structure for a small drone.

III. Framework for resilience to cyberattacks

This section describes the proposed framework that ensures resilience to cyberattacks, for the networked control structure for a small drone - see Figure 2. The framework consists in using a combination of a low-level cyberattack mitigation method (inspired from control theory) and a high-level cyberattack mitigation method (inspired from information technology). We consider man-in-the-middle attack types, which can inject false data $\mathbf{d}(t)$ on the direct path, acting like an additive input disturbance $\mathbf{u}^n_{oc}(t) = \mathbf{u}_{oc}(t) + \mathbf{d}(t)$, where $\mathbf{u}_{oc}(t)$ is the command send from the remote computer and $\mathbf{u}^n_{oc}(t)$ is the command received at the drone level. For low-level mitigation, an adaptive sliding mode controller is considered, which we design to reject matched input disturbance while ensuring reference tracking, and it also provides an estimation of the disturbance, thus triggering an alarm when it detects the onset of an cyberattack. High level mitigation implies the transmission of additional redundant encrypted data, which can be used at the receiving end (at the drone level), to recover the signal corrupted due to the cyberattack. It is important to send redundant information in a stealthy mode in respect with the attacker, as it needs to pass unaltered.

The components of this setup are detailed in the next subsections, with the last subsection suming up the methodology.

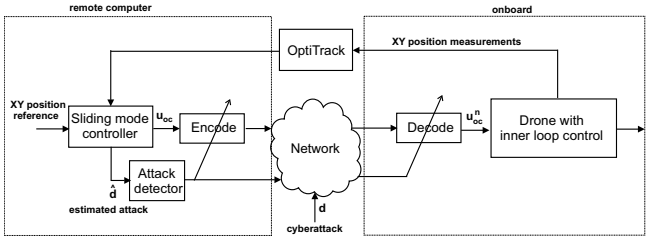

Fig. 2: Resilient networked control structure for a small drone.

A. Drone model and onboard inner loop control

The nonlinear model of the drone can be written in the general form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, with the the state vector $\mathbf{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ and the input vector is $\mathbf{u} = [U_{coll}, U_\phi, U_\theta, U_\psi]^T$ (see [11] for the detailed equations of the nonlinear model). $\{x, y, z\}$ are the positions in the world frame, while $\{\phi, \theta, \psi\}$ are the roll, pitch, and yaw angles. The control inputs are the torques on the

three rotational axes U_ϕ, U_θ, U_ψ, and the collective force is U_{coll}. Let the equilibrium point in hovering mode be $\mathbf{x}^e = [x^e, y^e, z^e, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, with inputs $\mathbf{u}^e = [U^e_{coll}, 0, 0, 0]^T$. The linearized model is

$$\begin{cases} \ddot{x} = \theta g \\ \ddot{y} = -\phi g \\ \ddot{z} = \dfrac{\Delta U_{coll}}{m} \end{cases} \qquad \begin{cases} \ddot{\phi} = \dfrac{U_\phi}{I_x} \\ \ddot{\theta} = \dfrac{U_\theta}{I_y} \\ \ddot{\psi} = \dfrac{U_\psi}{I_z} \end{cases} \tag{1}$$

where $\Delta U_{coll} = U_{coll} - m\,g$. The linearized model can also be written as

$$\dot{\mathbf{x}}_l = A\mathbf{x}_l + B\mathbf{u}_l \tag{2}$$

where $\mathbf{x}_l = \mathbf{x} - \mathbf{x}^e$ and $\mathbf{u}_l = \mathbf{u} - \mathbf{u}^e$, and the expressions for the matrices A and B are omitted due to space restrictions.

We further define the state vector for the inner loop $\mathbf{x}_i = [z, \phi, \theta, \psi, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$. We consider that an LQR controller along with a steady-state Kalman filter is already designed for stabilization and tracking, as in [11].

Remark 1: Although the altitude control is considered here as part of the inner loop, this does not limit the overall approach to tracking in the horizontal plane because the altitude reference z_r can still be sent remotely. Alternatively, the states z and $\dot{z}$ can easily be moved to the outer control loop without altering the overall framework.

B. Low level resilient control

For the outer control loop of the structure from Figure 2, we consider ideally that the inner control loop is faster than the outer loop control, i.e. $\phi \to \phi_r$, $\theta \to \theta_r$, $\psi \to \psi_r = 0$ and $z \to z_r = z^e$. Consequently, the outer loop can be modeled in a simplified manner, by using the first two equations from (1):

$$\dot{\mathbf{x}}_o = A_o\mathbf{x}_o + B_o\mathbf{u}_{oc} \tag{3}$$

where

$$\mathbf{x}_o = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, A_o = \begin{bmatrix} 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix}, B_o = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & g \\ -g & 0 \end{bmatrix}, \mathbf{u}_{oc} = \begin{bmatrix} \phi_r \\ \theta_r \end{bmatrix}.$$

At this point, we consider that there are no delays or packet loss (i.e. $\mathbf{u}_{oc} = \mathbf{u}^n_{oc}$). Since our outer loop model assumes that the movement on the X and Y axes are decoupled, we will restrict our attention only to the X axis movement (the discussion is the same for the Y axis). The model for just the X axis movement is:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= k_p(u_x(t) + d_x(t)) \end{aligned} \tag{4}$$

where $x_1 = x$, $x_2 = \dot{x}$, $u_x = \theta_r$, $k_p = g$. The added disturbance d_x represents the false data injection caused by the cyberattack.

Because we are interested here in the tracking problem, we make a change of variables in terms of the tracking errors $x_{e1} = x_1 - x_{r1}$ and $x_{e2} = x_2 - x_{r2}$, where x_{r1} represents the reference for the X axis, with x_{r2} being the velocity reference (i.e. $\dot{x}_{r1} = x_{r2}$). Model (4) becomes

$$\begin{aligned} \dot{x}_{e1}(t) &= x_{e2}(t) \\ \dot{x}_{e2}(t) &= k_p u_x(t) - \dot{x}_{r2}(t) + k_p d_x(t). \end{aligned} \tag{5}$$

We further assume that the disturbance and it's derivative are bounded, with $|d_x(t)| < d_x^{m0}$ and $|\dot{d}_x(t)| < d_x^{m1}$, but with the *bounds d_x^{m0} and d_x^{m1} unknown*.

Remark 2: The boundness assumption on the disturbance and its derivative is very reasonable in practice, because the command input to the drone usually includes a saturation element, and the bandwith of the control signal is also limited.

Consider the control law

$$u_x(t) = u_b(t) + u_{sc}(t), \tag{6}$$

where $u_b(t)$ is a linear control term determined based on the equivalent control concept [12], while $u_{sc}(t)$ is a discontinous nonlinear control term. We also introduce the sliding variable

$$\sigma(t) = x_{e2}(t) + \lambda_b x_{e1}(t), \tag{7}$$

where $\lambda_b > 0$ is a tuning parameter. By imposing the condition that $\dot{\sigma}(t) = 0$ for $d_x(t) = 0$ and $u_{sc}(t) = 0$, we can determine $u_b(t)$ as

$$u_b(t) = -\frac{\lambda_b}{k_p}x_{e2}(t) + \frac{1}{k_p}\dot{x}_{r2}(t). \tag{8}$$

In the general case with a non-zero disturbance $d_x(t)$, we have the dynamics of the sliding variable $\dot{\sigma}(t) = k_p(d_x(t) + u_{sc}(t))$. For driving both σ and $\dot{\sigma}$ to zero in finite-time, we will adopt the adaptive super-twisting control law proposed in [13]:

$$u_{sc}(t) = -\lambda|\sigma(t)|^{1/2}sign(\sigma(t)) + z(t) \tag{9}$$

$$\dot{z}(t) = -k(t)sign(\sigma(t)), \tag{10}$$

for which convergence to the sliding surface is ensured if λ is sufficiently large and $k(t) > d_x^{m1}$. As in [13], a dual layer adaptation scheme is adopted for the sliding gain $k(t)$, such that we get the lowest gain possible that can ensure sliding, depending on the disturbance variation, and at the same time we avoid chattering. The first adaptation layer is

$$\dot{k}(t) = -\rho(t)sign(\delta(t)), \tag{11}$$

where δ is an error term, defined as $\delta(t) = k(t) - \frac{1}{\alpha}|\hat{u}_{eq}| - \epsilon$, with the parameters $0 < \alpha < 1$ and $\epsilon > 0$ as safety margins. In sliding mode, δ is forced sufficiently close to zero. The estimate of the (unknown part of the) equivalent control $\hat{u}_{eq}$ can be determined by using a low pass filter [13]:

$$\dot{\hat{u}}_{eq}(t) = \frac{1}{T_f}(k(t)sign(\sigma(t)) - \hat{u}_{eq}(t)), \tag{12}$$

where $T_f > 0$ is a sufficiently small time constant.

The second adaptation layer involves the varying gain $\rho(t)$:

$$\rho(t) = r_0 + r(t), \tag{13}$$

$$\dot{r}(t) = \begin{cases} \gamma|\delta(t)| & \text{if } |\delta(t)| > \delta_0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where the parameter $\gamma > 0$ must be chosen sufficiently large, $r_0 > 0$ represents a small parameter, and δ_0 can be interpreted as a dead zone, and needs to be larger than noise or computational error.

Finally, the whole control scheme that encompass the interaction between equations (6)-(14) can be represented through the block diagram from Figure 3.

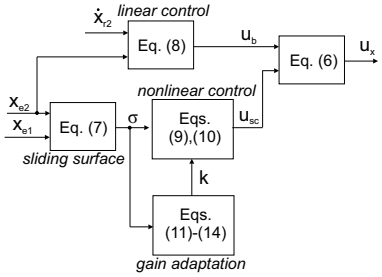

Fig. 3: Adaptive sliding mode control scheme

C. High level resilience

Recall the control structure from Figure 2. The presence of a cyberattack is detected by the attack detector block, with the help of the estimated disturbance from the sliding mode controller ($\hat{d} = \hat{u}_{eq}$ [13]). The attack detector signals an attack when the estimated input disturbance increases over a threshold Tr, for a certain time interval Δ (i.e. $\hat{d} > Tr \quad \text{for} \quad t \in [t_a, t_a + \Delta]$, where t_a is the initial time moment when the threshold has been surpassed). Based on this signal, the control system switches between countermeasure and normal modes of operation.

Remark 3: It is important to mention that because of the reduced computational possibilities of the drone's embedded computer, in normal mode it is unfeasible to continuously use encryption, since this would sacrifice computational resources for other tasks, like image processing and transmission.

The countermeasure that we propose to ensure high level resilience implies the transmission of both unencrypted and encrypted redundant data. The high level mitigation approach is represented in Figure 2 by the Encode and Decode blocks.

In the *countermeasure mode* (i.e. when an attack is signaled by the attack detector), the objective is to ensure data security, specifically data integrity and confidentiality. First the Encode block uses a message integrity code (MIC) to compute a checksum remotely and encrypts it together with the signals data $\mathbf{u}_{oc}$, using a symmetric key algorithm. We adopted the Advanced Encryption Standard (AES) algorithm [9], with a block size of 128 bits, as it ensures a good tradeoff between security performance and computational burden. The secret key is considered apriori installed on both ends (remote computer and the drone), in order to avoid the computational cost of a key exchange protocol based on an asymmetric key algorithm. The resulting cipher text is added to the packet payload, alongside the unencrypted values of the signals $\mathbf{u}_{oc}$, and sent through the network. To maintain similar packet structure, in order to make the countermeasure harder to detect by the attacker, in *normal mode* random padding data of the same size is used instead of the cipher text.

At the drone level, in countermeasure mode, the Decode block decrypts the signal data and validates the checksum (verify data integrity). If the checksum is incorrect, we discard the packet data, and use previous values. In the case of consecutive packets discarded over a certain threshold N_f, we consider that a major cyberattack is present and switch the drone to the fail-safe protocol. In normal operation mode the padding data is simply ignored.

D. Resilient networked control methodology

The inner loop control, the sliding mode controller, and the high level cyberattack mitigation method previously discussed, can all be embedded into a resilient networked control methodology, in accordance with Figure 2. The methodology is synthetically presented as a nine step algorithm - see Algorithm 1.

First we need to define a *fail safe mode*, where a protocol ensures autonomous flight to a predetermined location, or simply temporary hovering and landing [8]. The fail safe mode becomes active in case of a major cyberattack, or other possible emergencies. At each sample time, the sliding mode controller computes new commands to ensure robust reference tracking, and provides an estimate of the matched input disturbances, which are to detect the onset of a cyberattack. In case of a man-in-the-middle attack, this is the first level of resilience, with the controller rejecting the disturbance at the price of higher gains. If the attack persists, this state is undesirable because the overall control performances are affected by the high gains of the controller, and sensitivity to other types of disturbance like wind can increase. Therefore, when the estimated disturbance is larger than a threshold the attack is signaled, the higher level resilience method becomes activated, and starts to encrypt and decrypt data, along with a checksum validation. When the checksum becomes invalid, it means that the attacker has stepped up to a major cyberattack, corrupting all data that is transmitted on the direct path. If this happens for a number N_f of consecutive samples (thus excluding random network interference) the fail safe mode is activated.

Remark 4: The reference position can be further generated according to a path planning algorithm that takes into account possible cyberattacks or disturbances, by considering a safe region around the drone or different types of constraints (e.g. using reference governor [14]).

IV. Experiments

This section presents real-time experiments on a Parrot Mambo drone (Figure 4) connected to a remote computer, and using an OptiTrack motion capture system, in accordance with the network control structure from Figure 2. The Parrot Mambo drone is a small quadcopter of 0.18×0.18 meters, weights 0.063 kg, and has four propellers actuated by DC motors. The drone has several onboard sensors: accelerometer, gyroscope, ultrasound sensor, vertical camera, barometer, and

Algorithm 1 Resilient networked control methodology

while *non fail safe mode* **do**
  Step 1: Read new XY reference values
  Step 2: Sliding mode controller computes new commands and disturbance estimation
  if *normal mode* **then**
    Step 3: Construct and send packets with payload and padding data
    Step 4: Apply received signals to the inner control loop
  else *//attack detected*
    Step 5: Compute checksum, encrypt data, construct and send packets
    Step 6: Decrypt received data and verify checksum
    if checksum valid **then**
      Step 7: Apply decrypted signals to the inner control loop
    else if *non major attack* **then**
      Step 8: Apply values from previous sample time to the inner control loop
    else *//major attack*
      Step 9: Activate fail safe mode
    end if
  end if
end while

a temperature sensor. The online communication with the drone is done wireless, using the UDP transport protocol, and a firmware specifically developed for Matlab [15].

For the inner control loop, an LQR controller and a linear steady-state Kalman filter was designed as in [11]. For the outer control loop, the gravitational acceleration parameter was taken as $g = 9.81\ m/s^2$. The following parameters were adopted for the sliding mode controller: $\lambda_b = 4.4$, $\lambda = 0.14$, $\alpha = 0.9$, $\epsilon = 0.1$, $T_f = 0.01$, $r_0 = 0.0004$, $\delta_0 = 0.1$, $\gamma = 0.6$. For further reducing chattering at high gains, the $sign(\sigma)$ function was approximated as $\frac{\sigma}{|\sigma|+0.1}$. Due to limitations of the communication capabilities of the drone [11], the sampling period for the outer loop was chosen $T_s = 0.03$ s. Also, the commands are saturated at ± 0.5 rad. All this limitations mean that we will actually have a pseudo-sliding mode regime, instead of the ideal sliding mode discussed Section III.

The high level mitigation method consists of encrypting redundant data on the remote computer, then decrypting it at the drone level and finally verifying the checksum. This method is activated when an attack is detected. In our experiments we adopted a fixed threshold of $Tr = 0.15$ rad, while for the time interval Δ, we considered different values in the range of $(2, 10)$ s. The threshold is needed in order to distinguish between false data injection and other input disturbances (e.g. wind, unmodelled dynamics), while the timing may take into account factors like false positives, computational availability, or transient regimes of the drone (e.g. dwell time). The fail safe mode is activated when more than 10 packets are lost or discarded simultaneously ($N_f = 10$), and leads to a drone safe landing.

We further consider two man-in-the-middle attack scenarios.

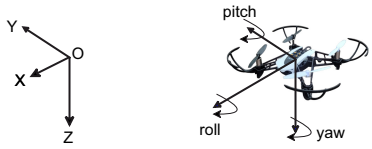

Fig. 4: Parrot Mambo mini-drone.

In the first scenario, the attackers injects a step-like disturbance signal $d_x(t) = f(t - 36.5)$, where f is the unit step function $f(t) = \{0 \quad \text{if} \quad t < 0, \quad 1 \quad \text{if} \quad t \geq 0\}$. Although this is not very stealthy, the purpose of this scenario is to illustrate the different roles of the lower level and higher level mitigation methods. For the same purpose, we adopted a relatively large timing of $\Delta = 10$ s for the Attack Detector. Figure 5 shows the reference tracking results on the X and Z axis (the drone moves in the XOZ plane), while Figure 6 shows the evolution of the sliding variable σ and the adaptive sliding mode gain k. The onset of the attack is indicated with a vertical red dashed line. It can be noticed that the sliding mode controller rejects the attack in a few seconds after the onset of the attack, by increasing the gain k. The onset of the higher mitigation - involving encryption/decryption and checksum verification - is indicated by the vertical dashed green line. This cancels the disturbance d_x, leading to a transient tracking error that quickly goes to zero, while the sliding gain k also decreases.

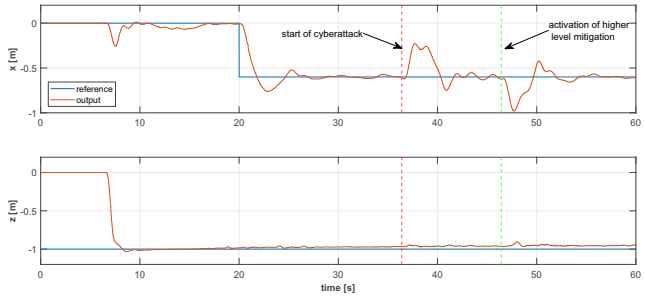

Fig. 5: XZ reference tracking for cyberattack scenario 1.

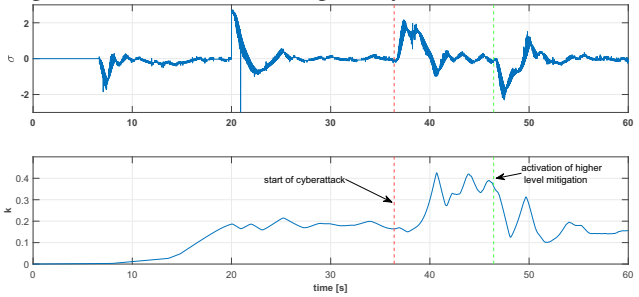

Fig. 6: Sliding variable σ and adaptive gain k for cyberattack scenario 1.

The second scenario is more stealthy, involving a ramp-like disturbance signal $d_x(t) = 0.06(t - 39.7)f(t - 39.7)$, while for the Attack Detector we adopted a timing of 2.5 s. The reference tracking results on the X and Z axis are shown in Figure 7, while the sliding variable σ and the adaptive gain k in Figure 8. The onset of the attack is indicated with a vertical

red dashed line. Due to the gradual increase of the disturbance, the sliding mode controller has more time to increase the gain k, and thus manages to reject the attack better than in the previous scenario. Once the high level mitigation is activated - indicated by the vertical dashed green line - the disturbance is again canceled and the gain k decreases to its previous level before the attack.

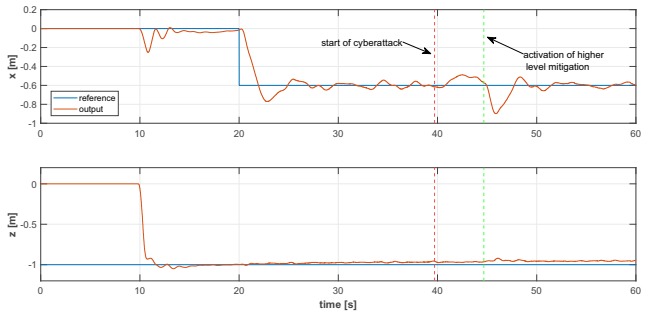

Fig. 7: XZ reference tracking for cyberattack scenario 2.

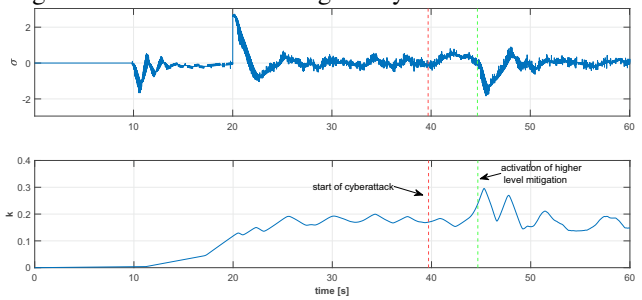

Fig. 8: Sliding variable σ and adaptive gain k for cyberattack scenario 1.

In both cyberattack scenarios it can be noticed that our proposed approach manages to ensure the resilience of the system, with the drone returning to its normal operating mode once the attack has been mitigated.

V. Conclusions

This paper has presented a resilient networked control methodology for small drones, considering man-in-the-middle cyberattacks. The framework consists in combining a low level attack mitigation method (sliding mode controller) with a high level mitigation method (redundant encrypted data), which enhances the robustness and resilience of the networked control system, while taking into account the computational limitations of the drone. Experimental results on a Parrot Mambo drone illustrate the effectiveness of our approach. As future work, we plan to further improve the performance of the networked control structure by ensuring smooth transitions between the activation and deactivation of the higher level mitigation method.

References

[1] J. Marshall, W. Sun, A. L'Afflitto, "A survey of guidance, navigation, and control systems for autonomous multi-rotor small unmanned aerial systems, Annual Reviews in Control," vol. 52, 2021 ,pp. 390–427.

[2] S.A. Mohsan, N.Q. Othman, Y. Li, M.H. Alsharif, M.A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," Intelligent Service Robotics, vol. 16, 2023, pp. 109–137.

[3] A.M. Annaswamy, K.H. Johansson, G. Pappas, Eds., Control for Societal-scale Challenges: Road Map 2030, IEEE Control Systems Society Publication, 2023.

[4] X.M. Zhang, Q.L. Han, X. Ge, D. Ding, L. Ding, D. Yue, C. Peng, "Networked control systems: A survey of trends and techniques," IEEE/CAA Journal of Automatica Sinica, vol. 7, 2020 , pp. 1–17.

[5] S.M. Dibaji, M. Pirani, D.B. Flamholz, A.M. Annaswamy, K.H. Johansson, A. Chakrabortty, "A systems and control perspective of CPS security," Annual Reviews in Control, vol. 47, 2019, pp. 394–411.

[6] Y. Zacchia Lun, A. D'Innocenzo, F. Smarra, I. Malavolta, M.D. Di Benedetto, "State of the art of cyber-physical systems security: An automatic control perspective," Journal of Systems and Software, vol. 149, 2019, pp. 174–216.

[7] Y. Mekdad, A. Aris, L. Babun, A. El Fergougui, M. Conti, R. Lazzeretti, A. S. Uluagac, "A survey on security and privacy issues of UAVs, Computer Networks," vol. 224, 2023, 109626.

[8] P. -Y. Kong, "A Survey of Cyberattack Countermeasures for Unmanned Aerial Vehicles," IEEE Access, vol. 9, 2021, pp. 148244-148263.

[9] M. Bishop, Computer Security Art and Science, Pearson, London, UK, 2019.

[10] A. Teixeira, I. Shames, H. Sandberg and K. H. Johansson, "Revealing stealthy attacks in control systems," 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 2012, pp. 1806-1813.

[11] A. Codrean, A. Kovács, O. Stefan and Z. Lendek, "Networked control of a Parrot Mambo drone," IEEE 33rd International Symposium on Industrial Electronics (ISIE), Ulsan, Republic of Korea, 2024, pp. 1-7.

[12] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, Sliding Mode Control and Observation, Birkhäuser, Basel, 2015.

[13] C. Edwards, and Y. B. Shtessel, "Adaptive continuous higher order sliding mode control," Automatica, vol. 65, pp. 183–190, 2016.

[14] E. Garone, S. Di Cairano, I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," Automatica, vol. 72, 2017, pp. 306-328.

[15] Matlab, Simulink Support Package for Parrot Minidrones, 2022.