ISA Transactions xxx (xxxx) xxx

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Practice article

Closed-loop nonlinear optimal control design for flapping-wing flying robot (1.6 m wingspan) in indoor confined space: Prototyping, modeling, simulation, and experiment☆

Saeed Rafee Nekoo [a,*], Anibal Ollero [a,b]

[a] The GRVC Robotics Lab., Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Seville, 41092, Spain
[b] FADA-CATEC, Centro Avanzado de Tecnologías Aeroespaciales, Seville, 41300, Spain

ARTICLE INFO

Article history:
Received 20 April 2023
Received in revised form 1 August 2023
Accepted 2 August 2023
Available online xxxx

Keywords:
Flapping-wing robot
Aerial robotics
SDRE
Nonlinear
Optimal
Closed-loop control
Flight control

ABSTRACT

The flapping-wing technology has emerged recently in the application of unmanned aerial robotics for autonomous flight, control, inspection, monitoring, and manipulation. Despite the advances in applications and outdoor manual flights (open-loop control), closed-loop control is yet to be investigated. This work presents a nonlinear optimal closed-loop control design via the state-dependent Riccati equation (SDRE) for a flapping-wing flying robot (FWFR). Considering that the dynamic modeling of the flapping-wing robot is complex, a proper model for the implementation of nonlinear control methods is demanded. This work proposes an alternative approach to deliver an equivalent dynamic for the translation of the system and a simplified model for orientation, to find equivalent dynamics for the whole system. The objective is to see the effect of flapping (periodic oscillation) on behavior through a simple model in simulation. Then the SDRE controller is applied to the derived model and implemented in simulations and experiments. The robot bird is a 1.6 m wingspan flapping-wing system (six-degree-of-freedom robot) with four actuators, three in the tail, and one as the flapping input. The underactuated system has been controlled successfully in position and orientation. The control loop is closed by the motion capture system in the indoor test bed where the experiments of flight have been successfully done.

1. Introduction

Flapping-wing flying robots have been increasingly used in research and autonomous robotics targeting the applications of monitoring and inspection [1,2], manipulation and sampling [3], an alternative way of flight with less noise in nature [4], etc. One application and challenge in FWFR is perching on a branch. Perching is needed to perform a post-perching activity such as sampling, manipulation, taking a picture, or other tasks. Perching on a branch was done using multi-rotor drones [5], fixed-wing unmanned aerial vehicles [6], flapping-wing robot [7], etc. The flight and perching by the ornithopter used a series of linear controllers to regulate the height of the robot close to the branch position [7]. The flight was also performed in an indoor testbed limited to a 15 (m) flight path. Linear controllers are preferable for control designs if they could do the task satisfactorily since they are fast in the computation of control loops. To move towards a model-based nonlinear control, an efficient representative model is demanded.

The objective of this work is to control a robot bird in an indoor test bed. The flapping-wing robots are suitable prototypes for outdoor flights since they need a large area to freely maneuver. Control design with repeatability is difficult in outdoor tests due to a lack of precision in the measurement of position feedback. The indoor test bed conversely has the possibility of measurement and repetition of the flights using Opti-Track system. The indoor flight zone also offers an area with the same conditions (without wind and disturbance). The flapping-wing robot possesses a six-degree-of-freedom (DoF) model with four actuators. The actuators interact in all DoFs of the system, i.e. the tail could change the pitch angle, consequently, it affects the height. The flapping itself can ascend or descend the robot. Flapping is the source of forward thrust as well if the robot has a close-to-zero pitch angle. It is possible to control all the six-DoF variables of the system in outdoor unlimited space, though in this study, in indoor experiments, it is not in the interest of the work

☆ Funding: This work was supported by the European Project GRIFFIN ERC Advanced Grant 2017, Action 788247.

* Corresponding author.
E-mail addresses: saerafee@yahoo.com (S.R. Nekoo), aollero@us.es (A. Ollero).

https://doi.org/10.1016/j.isatra.2023.08.001

Please cite this article as: S.R. Nekoo and A. Ollero, Closed-loop nonlinear optimal control design for flapping-wing flying robot (1.6 m wingspan) in indoor confined space: Prototyping, modeling, simulation, and experiment. ISA Transactions (2023), https://doi.org/10.1016/j.isatra.2023.08.001.

to consider longitudinal control since the flight zone is limited to 15 (m).

This research has been done within the framework of the GRIFFIN project, "General compliant aerial Robotic manipulation system Integrating Fixed and Flapping wings to INcrease range and safety", advanced grant ERC. The GRIFFIN project targets the use of aerial robotics, specifically flapping-wing technology for building prototypes and robots for the application of monitoring and inspection, manipulation, and lightweight object transportation. The early-stage robot birds performed outdoor manual flight [8], gliding model, and flight [9]. The closed-loop control of the FWFR was done with linear designs and led to the possibility of perching on a branch autonomously [7]. This work will continue the research in this line of investigation by developing a more sophisticated control based on the state-dependent Riccati equation. The contribution of this work with respect to the previous research is the implementation of nonlinear optimal control in closed-loop design and the presentation of an equivalent dynamic to imitate the flapping effect on the six-DoF model of the robot.

The modeling of flapping-wing robots is a harder challenge than the fixed-wing unmanned aerial platform due to the additional complexity through the flapping of wings which adds two more 3D objects to the main body of the robot and consequently the generated aerodynamics terms. This additional motion of the wings changes the dynamics from single-body to multibody system dynamics. Tu et al. investigated the flight control of flapping-wing robots with damaged wings [10]. The presented dynamics included the drag and damping coefficients to consider the effect of wings and their motion. Guzman et al. studied the dynamics and aerodynamic coefficients of three types of tails for FWFRs; using two servomotor actuators with different configurations [11]. Fan et al. presented deformable wings for FWFRs and studied the effect of flexibility in the control performance [12]. A simplified model for the flapping wing was reported and validated experimentally, presenting the drag and lift terms due to actuation [13]. Despite the various versions of the dynamics of the robot birds, model-based controllers have been rarely implemented on the experimental platforms. The Lagrange method will be used in this work to derive an equivalent dynamic for translation and a conventional one for orientation. The equivalent dynamic will present the effect of periodic flapping on the base of the robot similar to experimental data. The derivation of the dynamics can be also done using other techniques such as Kane's method [14,15], Newton–Euler approach [16,17], and Gibbs–Appell formulation [18–20]. Modeling the robotic systems using different dynamic methods is getting popular and embedding the recursive structure on them provides powerful and fast derivations for complex systems.

The focus of this work is on the control of the FWFR. The beginning era of flapping wings presented many outdoor flights using radio controller (RC) transmitters. The term FWFR or ornithopter robots refers to unmanned systems (from a robotics point of view), which tried to perform the autonomous flight. The concept of flapping dates back to the ancient Greek legend of Daedalus and Icarus, Leonardo da Vinci, and the early years of flapping wing systems started in 1870 by Gustav Trouvé [21]. So it is worthwhile echoing the phrase of Chronister that the word ornithopter was defined already and several platforms accomplished the flight way back in history [21]. Flying outdoors even recently needed a human pilot due to the inaccuracy of the positioning systems and inertial measurement units (IMUs); Pan et al. presented two prototypes with a wingspan of more than 2 m flying outdoor [1,22]. Lee et al. designed and experimented with a robust controller for attitude control of a robot bird subjected to wind disturbance [23]. The outdoor control was also reported using global positioning system (GPS) feedback for localizing and an IMU for orientation [24]. The most popular reported controller for the flight was proportional–integral–derivative (PID) design [25], applied on trajectory tracking of a circular path [26], flapping-wing miniature aerial vehicle [27], and other platforms; however, active disturbance rejection control [28], and adaptive control were also implemented on gliding mode of the FWFR [29].

Here in this work, the state-dependent Riccati equation controller is implemented on the attitude control of the flapping-wing flying robot. The application of the SDRE controller diverged in the early 90s in aerospace control [30,31], then exploring other fields such as robotics [32,33], control of underwater platforms [34,35], etc. The SDRE and terminal sliding mode controller were also simulated on a version of a flapping wing robot though the experimental implementation was not exercised [36]. The SDRE is an optimal nonlinear closed-loop approach that needs a model for designing the controller. The available models for the flapping-wing systems are too complex for implementation and delivering a solution to the Riccati equation as an online onboard implementation on the robot bird. Here the SDRE is implemented on an equivalent dynamic whose output is similar to the experimental data of the FWFR in many (regulation) point-to-point control records. The experimental flight data of the robot will be also compared with the simulation results.

Experimental implementation of nonlinear optimal controllers, specifically the state-dependent Riccati equation has been a challenge in practical works. From the early works in the SDRE experimental tests, one could refer to hardware-in-the-loop with two computers back in the late 90s [37], or control of an underactuated robot in 2002 [32]. In 2003, practical implementation of the SDRE was reported for unmanned helicopter control [38]. It was reported the Riccati was solved at each time-step onboard the flying system for tracking the commanded signal of the pilot (through RC transmitter) after takeoff. Here in this current work, pilot or RC transmitter is not used, and the control design is done on the onboard computer. The inverted pendulum was also an interesting experimental platform to show the capability of the SDRE controller [39]. The main challenge in the experimentation of the SDRE has been the solution to the Riccati at each time-step that led to several implementations of stationary setups [40]. Communication between the main processor and the robot onboard computer was an effective method to reduce the burden of the computational load from the experimental platform [41]. The time-step of simulation then increases in comparison with linear controllers; so far the reports showed an average of 0.05 to 0.085 (s) for rotary inverted pendulum [42], and variable-pitch pendulum [43,44]. A PID-SDRE control design was studied for orientation control of a quadrotor drone [45]. The summary of this short discussion on the experimental implementation of the SDRE indicates that the beginning of practice started with stationary setups, communication between the computer and the processor of the platform, and then the implementation and solution to Riccati was done at each time-step onboard. Now the trend is to implement the algorithm totally onboard the platform to remove the risk of wireless communication, especially on flying systems.

The contributions of this work are as follows:

1. The first major contribution is to implement the SDRE controller (theoretical and experimental study with onboard implementation) on the attitude control of the flapping-wing flying robot for the first time using an equivalent dynamic model. The linear control on E-Flap was implemented in [7], and here a nonlinear closed-loop controller upgraded the system. To the best knowledge of the authors, the SDRE has been implemented experimentally on FWFR for the first time, in this work.

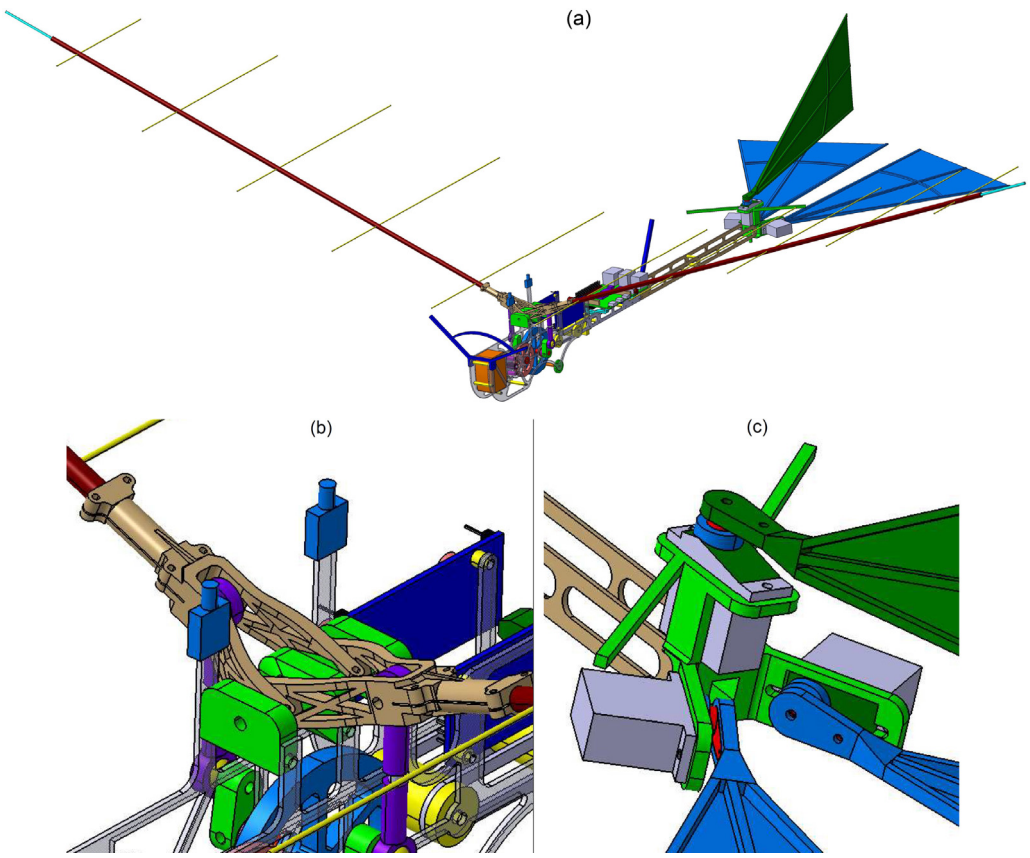

Fig. 1. (a): The CAD design of the flapping-wing flying robot in 3D view. (b): The zoomed view of the hinges, (c): and the tail motors. The elevators right and left are installed with an acute angle to avoid contact between the elevators in the actuation.

2. The second major contribution is the addition of Y-axis control to the height and orientation control, reported in previous works [7,25]. With this control design, the five-DoF of the system is controlled except for the X-axis which implies the necessity of forward flight to preserve the stability of the robot in regulation. Controlling the yaw angle of the robot results in an indirect control of the Y-axis; however, having control over that as an independent control variable would increase the possibility of more complex maneuvers.
3. The first minor contribution is the change in the design of the tail and adding one more actuator to have three servomotors for obtaining more control over the robot. It has been common to consider two servomotors for tail control [2,11,25,46,47]; and here for the first time a controlled flight is done with this specific tail design. The addition of actuators increases the weight and interaction of the control variables. A successful point-to-point control has been performed thanks to the new tail design using four actuators, flapping, rudder, and left/right elevators.
4. The second minor contribution is a technological one, using completely printed tails instead of conventional custom-made tails with fabrics and carbon fiber (CF) rods as ribs. This will facilitate a lot the fabrication process, symmetry in the bird, and repair of the system in case of a crash.

The paper structure: Section 2 presents the mechanical design procedure and mechatronics of the flapping-wing robot. Section 3 describes the modeling of the robot bird and the idea of the equivalent dynamics in the flapping as a periodic excitation of the base. The controller design and the structure of the state-dependent Riccati equation are reported in Section 4. Sections 5 and 6 present simulation, experimental flight data, and results respectively. Concluding remarks are reported in Section 7.

2. Mechanical and mechatronics design

2.1. Mechanical design

The objective of modification and development of the new prototypes is to deliver a more complete system and reach better performance while removing the drawbacks in previous versions — an iterative motion in the evolution of robot birds. This FWFR is built in the context of the GRIFFIN project,[1] as a new bird with the following changes with respect to E-Flap [25]: (1) increasing the wingspan from 1.5 m to 1.6 m to add lift force, (2) addition of one actuator to the tail to enhance maneuverability, (3) present a combination of "V" tail and "T" tail, please see Ref. [11], to achieve more control over the robot, (4) using the idea of printed tails instead of building the tails with fabrics and carbon fiber rods, (5) obtaining the Y-axis control and implementation of a nonlinear optimal controller, SDRE, onboard the robot.

The 3D printed tails are more accurate, it is easy to replace them and it increases the symmetry of the design. Making the tails by fabric requires cutting the fabric based on a pattern and use of glue which puts some inaccuracy in manufacturing. Both "V" and "T" tails in previous prototypes were built with two servo motors while here three servomotors are considered in the design.

The 3D CAD design of the FWFR is shown in Fig. 1. The design is done to consider two processes, CNC cutting of CF plates and 3D

[1] https://griffin-erc-advanced-grant.eu/

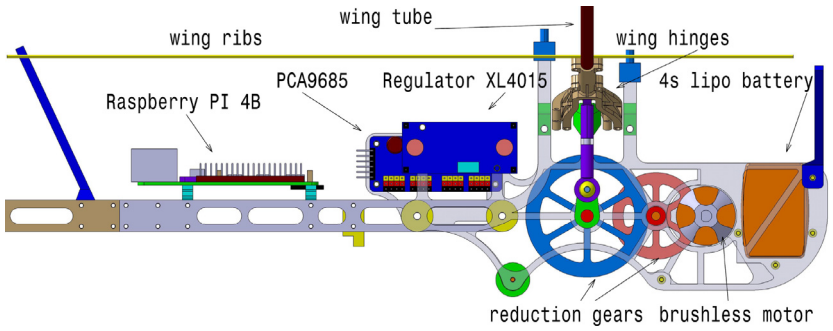

Fig. 2. The side view of the CAD design of the flapping-wing flying robot.

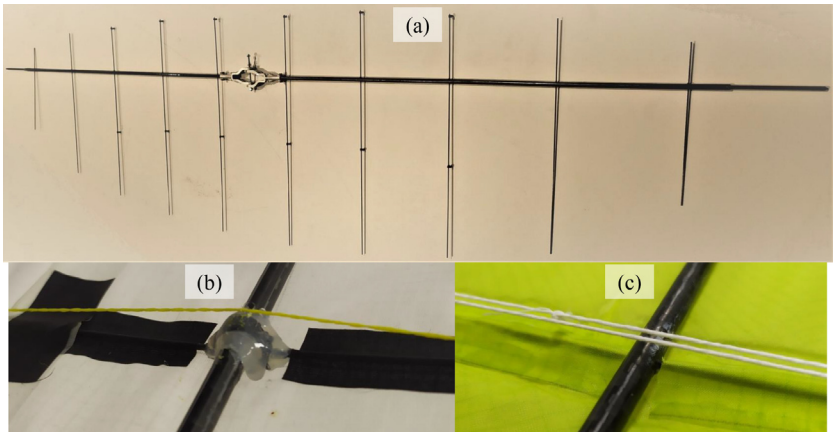

Fig. 3. (a): The wing structure of the robot, main tube, and the ribs, 1.6 (m) wingspan; (b): the old version using 3D printed part and glue; (c): the new version passing the ribs through the tube without using glue.

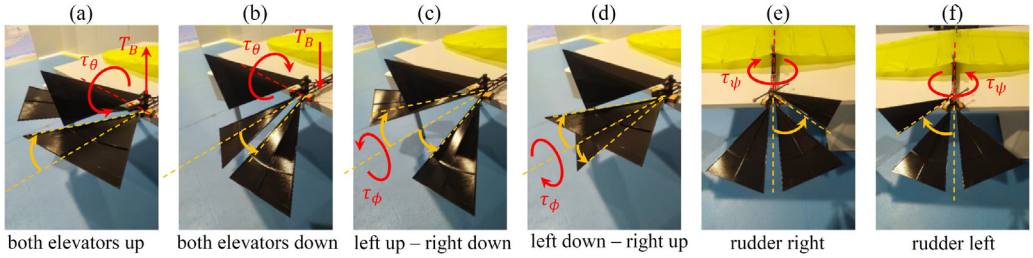

Fig. 4. The tail structure of the robot, definition of the motions and torque and thrust reactions; (a) shows both elevators up which results in negative pitch angle and consequently upward thrust force; (b) placed both elevators down that generates positive pitch angle and negative thrust force; (c) sets left elevator up and the right one down to produce negative roll angle; (d) sets the left elevator down and the right one up to generate positive roll angle; (e) places the rudder to right that provides negative yaw angle; (f) sets the rudder to left that provides positive yaw.

printing of parts with polylactide (PLA) material. The main plates for holding the motors and gears were made from CF plates of 1.5 (mm) thickness, presented in Fig. 2. The brushless DC motor is fixed on the left-side plate, gears were installed using bearings on both sides, and the battery is set between the two plates at the tip of the robot to balance the weight distribution and keep the center of gravity near the flapping point. The gear reduction ratio is $n = \frac{Z_2}{Z_1} \times \frac{Z_4}{Z_3} = 42$ where $Z_1 = 12, Z_2 = 72, Z_3 = 8, Z_4 = 56$ and it reduces the high-speed brushless DC motor to a proper zone for flapping frequency. The reduced rotary motion of the gears will be transformed into a reciprocating motion by a cam mechanism. The aluminum hinges hold the main wing tubes, with 6 (mm) and 4 (mm) outer/inner diameter made of CF. The ribs are 1.5 (mm) CF rods passing through the holes on the main 6 (mm) CF tubes, Fig. 3. In E-Flap version [25], the ribs passed through a 3D printed holder and the holder was installed (glued) on the main wing tube. This new design consideration reduces the 3D-printed holders and reduces the weight of the wings. Clearly, it has more accuracy and symmetry since gluing is removed from this part of the manufacturing process. The position of the robot is measured using the Opti-Track system. Seven emitting balls are installed on the robot bird to reflect infrared light to the Opti-Track cameras with high frequency. Two markers will be installed at the tip of the FWFR on the blue part antenna shape, see Fig. 1. Another pair will be installed above the hinges, one above the processor board, and the last two beside the servomotors. The tail of the robot includes three servomotors for two elevators and one rudder. The combination of the three actuators contributes to roll, pitch, and yaw (orientation) control. The elevators are also effective in thrust production. The motion is briefly explained as follows. Placement of both elevators up results in a negative pitch angle and consequently upward thrust force, Fig. 4-(a). Then this actuation increases the height of the system in flight (increasing positive Z-axis motion). The opposite

motion regulates the robot bird down and reduces the height in flight by placing both elevators down, Fig. 4-(b). Setting the left elevator up and the right one down will generate a negative roll angle and rotates the bird around X-axis, Fig. 4-(c). Setting the left elevator down and the right one up generates a positive roll angle, Fig. 4-(d). The rudder has a role in yaw rotation. Placing the rudder to the right provides a negative yaw angle and setting the rudder to the left provides positive yaw, Figs. 4-(e) and -(f). It will be shown that the rudder will contribute to the Y-axis control in the modeling section.

The last mechanical part is the linkage between the body and the tail which is constructed by a CF plate of 1.5 (mm) with 15 (mm) separation. In the whole design of the plates, extra material was removed by cutting different shapes and holes in them to reduce the mass as much as possible.

2.2. Mechatronics and electronics design

The motion of the flapping wing is generated by a brushless DC motor, Hacker A20-26M kv1130. It has a direct feed from the driver "Tmotor F 35 A 32 bit 3S" which can work with 3S to 5S lipo batteries. The schematic view of the electronics and actuators of the system are presented in Fig. 5. Here the power supply is chosen a 4S1P with 450 (mAh) and 14.8 (V) operating voltage. The input control signal to the DC motor is received from a driver board, PCA9685, which is capable of handling up to 16 pulse-width modulation (PWM) outputs. The operating voltage of this module is 5V and it communicates with the Raspberry PI 4B with serial peripheral interface (SPI) protocol. The power supply to the PCA9685 is provided by LM7805 conventional regulator, with 5V voltage and up to 1.5(A) current. This output current will be consumed by three servomotors for the tail of the FWFR, SH-0255MG type. These small servomotors can work with operative voltage $[4.8, 6.0]$ (V) with corresponding output torque $[3.1, 3.9]$ (kg.cm) for the input voltage range. The actuation time in the range of $[4.8, 6]$ (V) is $[0.13, 0.16]$ (s), with operating travel $\pi/2$ (rad) from $1000 \rightarrow 2000$ (μs). These servo motors do not provide position feedback for the control, and thanks to the external measurement system, Opti-Track, angular measurement of the tails is not necessary. The position and orientation of the robot bird will be obtained by the Opti-Track system with an accuracy of 1 to 2 (mm), depending on the calibration of the system.

The processor board of the flapping-wing robot is a Raspberry PI 4B which is a conventional small single-board computer. It will be powered up by a voltage regulator XL4015 which can work with input voltage $[8, 36]$ (V) and deliver the output voltage of $[1.25, 32]$ (V) with a current of up to 5(A). It is important to set the input power of the Raspberry PI at 5.2 (V) to avoid the problem of restarting and low-voltage warning. The processor communicates wireless with the motion capture system through wireless fidelity (WiFi) for receiving the position and orientation feedback of the robot in flight for employing in the closed-loop control loop. Therefore, independent feedback from the actuators is not necessary for the controller which is crucial since brushless DC and servomotors do not usually provide position feedback.

The motion capture system was installed in the testbed, confined space of $20 \times 15 \times 7$ (m), with 28 cameras that provided approximately 15 (m) diagonal flight path for the ornithopter. The communication with the Opti-Track system is done by the robot operating system (ROS), or to be more accurate, ROS1, to read the position and orientation of the FWFR body. Virtual-reality peripheral network (VRPN) interface is used for reading the data of the motion capture system. It will be read by a Python script code in the control loop by subscribing to the internet protocol address (IP) of the robot bird defined by Motive software.

2.3. Mass distribution and details

This section presents the mass breakdown of the components of the FWFR. The previous version of the robot bird weighed 500 (g), E-Flap prototype reported in [25]. The FWFR in this work is 567.3 (g) without the battery, shown in Fig. 6. Here we highlight the changes that led to the additional weight of this version. The processor of the E-Flap was Nano PI NEO which was 9.7 (g) changed to Raspberry PI 4B with the weight of 46 (g). The voltage regulators were installed on the main CF plate of the robot in E-Flap though here we used LM7805 and XL4015. The wingspan of the robot increased 10 (cm) and another servo has been added to the robot on the tail. These modifications were done to simplify the repair and manufacturing of the robot and add the payload by increasing the wingspan. The weight breakdown of the FWFR is presented in Fig. 7. The breakdown shows that the majority of the mass is devoted to the structure and mechanical parts, 42.5%, and in the second place, the wing with 20.9% share of the total weight. Using PWM module PCA9685 allows us to add servo inputs up to 16, which will be used for adding more servomotors for manipulation and leg of the robot birds in future works.

3. Modeling and equivalent dynamics

The dynamics modeling of the flapping-wing flying robot has the additional complexity of flapping with respect to airplane dynamics. A complete six-DoF modeling is considered for the translation and orientation of the robot bird. The dynamics of the airplanes have been well-developed in the literature; however, the current valuable dynamics of flapping-wing robots are too complex to be used in model-based control designs. One approach is numerical model development for flapping wing dynamics and computational fluid dynamics (CFD) [48]. The complexity of the flapping-wing model is rooted in the definition and computation of aerodynamics terms [49]. To present a model for the implementation of a nonlinear controller, a simplified approach is chosen.

In order to consider this approach, it should be noted that a flapping-wing robot is different from an airplane dynamics:

- A1. The ratio of the weight to the area of the wing is much lower than in an aircraft; in other words, an aircraft is heavier with respect to the surface of its wings.
- A2. The speed of the forward flight of an airplane is way higher than FWFR.
- A3. The FWFR has a periodic oscillation in the flight path synchronous with the frequency of the flapping.
- A4. The flight condition of this work is an indoor confined space, without wind and external disturbance.

Moreover, the following assumptions are considered to facilitate the modeling process:

- B1. The robot flies in the forward direction, $X(t)$, with forward velocity more than the minimum required speed to keep the stability of flight, $\dot{X}(t) \geq \dot{X}_{\min}$ [36].
- B2. There is always an initial speed for launching the robot which means an initial velocity $\dot{X}(0) \geq \dot{X}_{\min}$ [36].
- B3. The aerodynamics effect is neglected in the modeling due to low-speed flight conditions. This will reduce the precision of the modeling though it is intentionally done to simplify the control design.
- B4. The effect of the flapping is considered by equivalent dynamics and base excitation modeling.

The actuators of the FWFR are flapping wings and the motion of the tail (rudder and elevators). The main flapping provides the source of lift force to compensate for the gravity and the forward

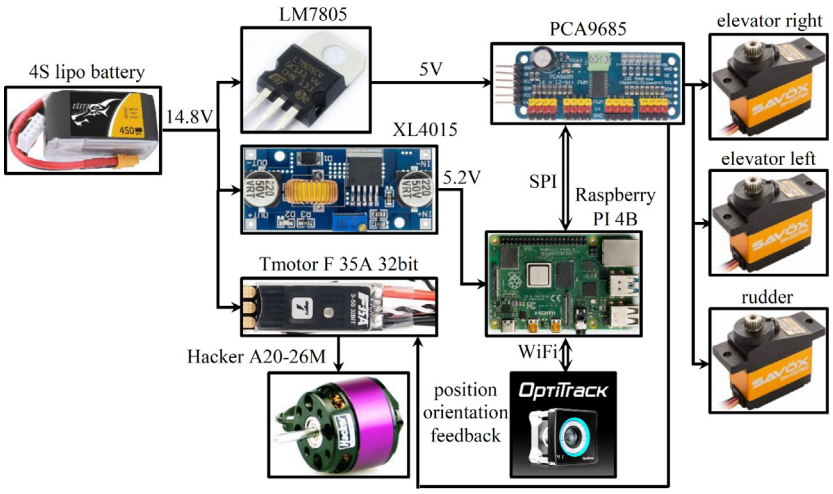

Fig. 5. The electronics and actuators block diagram of the FWFR.

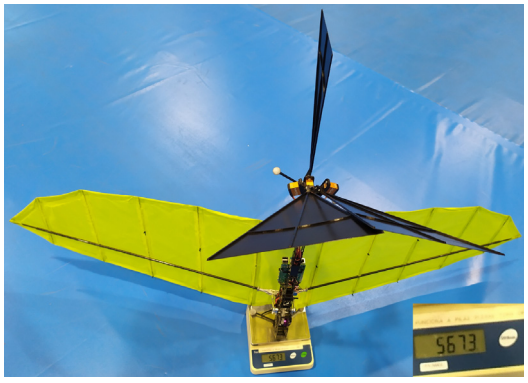

Fig. 6. The weight (mass) measurement of the FWFR.

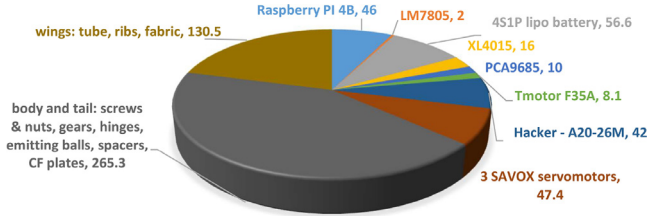

Fig. 7. The weight (mass) breakdown of the FWFR; the unit of measurement is (g) in the chart.

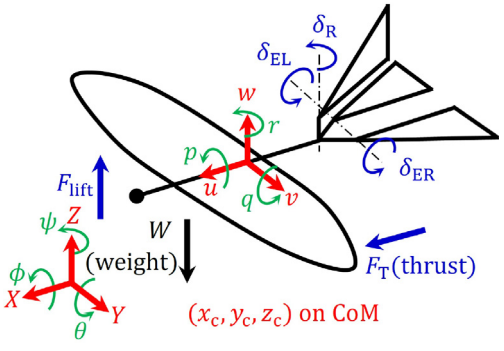

Fig. 8. The model of the flapping-wing robot, axes definition, and actuation.

thrust force. The two elevators are defined by $\{\delta_{ER}(t), \delta_{EL}(t)\}$ (rad) for right and left one respectively. The motion of the rudder is denoted by $\delta_R(t)$ (rad), presented in Fig. 8. The combination of the motion of the tail components generates three main torques along the local axis (placed at the center-of-mass (CoM) of the robot), presented in Fig. 4. The tail also contributes indirectly to the change in height. By changing the pitch angle, the flapping will move the robot higher/lower during the forward motion. It is also true that the rudder can indirectly change the position of the robot in the Y-axis.

The axes definition and the schematic of the robot are presented in Fig. 8. The global fixed coordinate is presented by $\{X, Y, Z\}$ and it measures the local (moving) coordinate by $\boldsymbol{\xi}_1(t) = [x_c(t), y_c(t), z_c(t)]^\top$ (m). The orientation of the local axis is measured by $\boldsymbol{\xi}_2(t) = [\phi(t), \theta(t), \psi(t)]^\top$ (rad) with respect to main fixed frame. The linear velocity of the moving axis is $\boldsymbol{\upsilon}_1(t) = [u(t), v(t), w(t)]^\top$ (m/s) and the angular velocity of that is $\boldsymbol{\upsilon}_2(t) = [p(t), q(t), r(t)]^\top$ (rad/s). The kinematic relationship between the global and local coordinates is defined by [50]:

$$\dot{\boldsymbol{\xi}}_1(t) = \mathbf{R}_b(\boldsymbol{\xi}_2(t))\boldsymbol{\upsilon}_1(t), \tag{1}$$

$$\dot{\boldsymbol{\xi}}_2(t) = \mathbf{T}(\boldsymbol{\xi}_2(t))\boldsymbol{\upsilon}_2(t), \tag{2}$$

where

$$\mathbf{R}_b(\boldsymbol{\xi}_2(t)) = \begin{bmatrix} c_\psi c_\theta & -c_\phi s_\psi + s_\phi s_\theta c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\psi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix},$$

$$\mathbf{T}(\boldsymbol{\xi}_2(t)) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}.$$

Neglecting the effect of aerodynamics, based on Assumption B3, the robot is considered a 3D flying object. The generalized coordinate vector for a flying object (the common form in the aerospace community) is selected as $\mathbf{q}(t) = \{x_c(t), y_c(t), z_c(t), \phi(t), \theta(t), \psi(t)\}$. Applying the Lagrange method, the equations of motions are derived as:

$$m_b \ddot{\boldsymbol{\xi}}_1(t) + m_b g \mathbf{e}_3 = \mathbf{R}_b(\boldsymbol{\xi}_2(t))\mathbf{F}_B(t), \tag{3}$$

$$\mathbf{J}(\boldsymbol{\xi}_2(t))\ddot{\boldsymbol{\xi}}_2(t) = \boldsymbol{\tau}_B(t) - \mathbf{c}(\boldsymbol{\xi}_2(t), \dot{\boldsymbol{\xi}}_2(t)), \tag{4}$$

where m_b(kg) is the mass of the base (total mass without the mass of wings; the weight of the wings will be added later in the base excitation model), $g = 9.81$ (m/s^2) is the gravity

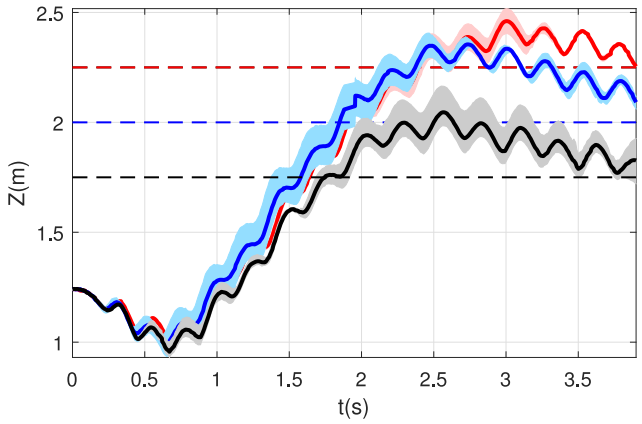

Fig. 9. A sample of flight data for height regulation of the FWFR, experimental data for three set-points [7]; bold lines show the average motion.

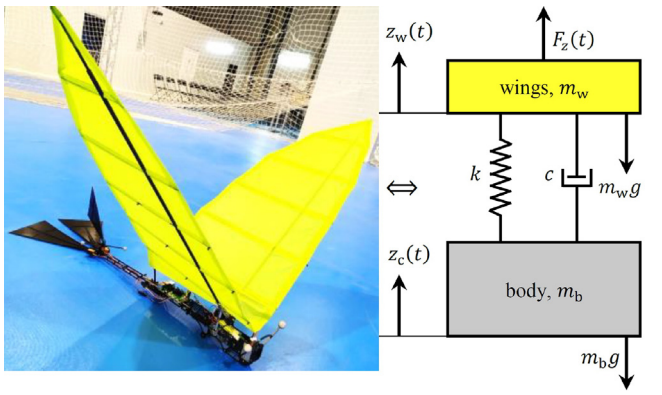

Fig. 10. The equivalent dynamic of flapping action as a base-excitation system for the FWFR.

constant, $\boldsymbol{\tau}_B(t) = [\tau_\phi(t), \tau_\theta(t), \tau_\psi(t)]^\top$(N.m) is the counteracting applied torque vector to the rotational dynamics, $\mathbf{F}_B(t) = [F_x(t), F_y(t), F_z(t)]^\top$(N) is the counteracting applied force vector to the translation dynamics, both set on the local axis and $\mathbf{e}_3 = [0, 0, 1]^\top$. $\mathbf{J}(\boldsymbol{\xi}_2(t)) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3\times3}$ is the inertia matrix of the rotational dynamics and $\mathbf{c}(\boldsymbol{\xi}_2(t), \dot{\boldsymbol{\xi}}_2(t)) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ includes the Coriolis and centrifugal terms.

Based on Assumption B1, the robot always flies at a forward speed in the X-axis and here the input control component (which would control the X position) of that is equal to zero, $F_x(t) = 0$. In reality, the frequency of the flapping in a long-term flight will define the forward speed of the X-axis motion; however, in this 15 (m) short distance (the diagonal length of indoor confined testbed), the forward speed was always observed from experimental data $u(t) = [2, 6]$ (m/s). Since the system is underactuated, the frequency of flapping was devoted to height control. For the sake of simplicity, a constant average speed of forward flight is considered for the modeling and simulation section. This average speed is based on the observation of experimental data from many recorded flights.

Observing the experimental data, especially the log record of the Z-axis, presented in Fig. 9, it can be concluded that the height component of the generalized coordinate $z_c(t)$ is subjected to a periodic and persistent oscillation, with the frequency of $[3.5, 4.5]$ (Hz) and amplitude of $[10, 15]$ (cm). It was already mentioned that the modeling of flapping action in the dynamic equation of the FWFR is difficult, and the oscillation in height control brings up the idea of modeling the system with equivalent dynamic and base excitation [51], Fig. 10. The motion of the wing plays the role of oscillatory part of the dynamics and the body of the robot bird receives the forced excitation.

Assumption 1. The frequency of the flapping is limited to $[3.5, 4.5]$ (Hz) which is sufficient to change the altitude of the flapping wing robot [52].

The pitch angle of the FWFR varies between $[0.2, 0.5]$ (rad) which is negligible though it is used to regulate the pitch angle slightly around an equilibrium point which provides flight stability for the bird. With a proper pitch angle around 0.4 (rad), the frequency of flapping ascends or descends the FWFR. The interactions of the motion of the elevators and flapping to height control are undeniable though devoting an independent role of height control to flapping simplifies the design.

Assumption 2. The robot flies with negligible change in the pitch angle, then the role of flapping frequency is dominant in the change of the altitude [52].

Considering Assumptions 1 and 2, the system can be modeled as an equivalent dynamic with base excitation in which the oscillation of the flapping directly disturbs z-axis and indirectly, other degrees of freedoms through the nonlinear coupling of dynamic. The free-body diagram of the simplified model is depicted in Fig. 10. Regarding the balance between the forces in the z-axis, the equation of motion can be derived, more details could be revisited in [53,54]. Equation of motion of free-body diagram of Fig. 10, as an independent one-DoF mass–spring–damper system, is presented by:

$$\begin{aligned} &m_b\ddot{z}_c(t) + c\dot{z}_c(t) + kz_c(t) + (m_b + m_w)g \\ &\quad = m_w\ddot{z}_w(t) + c\dot{z}_w(t) + kz_w(t) + F_z(t), \end{aligned} \tag{5}$$

where m_w is the mass of wings, c (N s/m) is the damping coefficient, k (N/m) is the elastic constant, $z_w(t) \in \mathbb{R}$, (m) is the base excitation motion. $m_b\ddot{z}_c(t)$ and $m_w\ddot{z}_w(t)$ are the effect of the acceleration on the mass of base and wings. In the free-body-diagram, $c(\dot{z}_c(t) - \dot{z}_w(t))$ is the damping force and $k(z_c(t) - z_w(t))$ the generated force due to elastic characteristics of the connection between the wings and the body. $(m_b+m_w)g$ is caused by the gravity and $F_z(t)$ is the lift force that compensates for the weight.

The idea of equivalent dynamics and base excitation is expressed for a one-DoF system, only z-axis; however, this periodic disturbance will be incorporated into six-DoF dynamics which later will show the effect of excitation on other degrees of freedom.

The base excitation is a periodic oscillation that can be presented by:

$$\begin{aligned} z_w(t) &= z_0 \sin(\omega(t)t), \\ \dot{z}_w(t) &= z_0\{\omega(t) + \dot{\omega}(t)t\} \cos(\omega(t)t), \\ \ddot{z}_w(t) &= z_0[\{2\dot{\omega}(t) + \ddot{\omega}(t)t\} \cos(\omega(t)t) - \{\omega(t) + \dot{\omega}(t)t\}^2 \sin(\omega(t)t)], \end{aligned} \tag{6}$$

in which z_0 (m) is the amplitude of the base excitation, and $\omega(t) \in \mathbb{R}$, (rad/s) is the flapping frequency. If one divides (5) by m_b and considers damping ratio and natural frequency as $\xi = \frac{c}{2m_b\omega_n}$ and $\omega_n = \sqrt{\frac{k}{m_b}}$ (rad/s), respectively, it can be rewritten as:

$$\begin{aligned} &\ddot{z}_c(t) + 2\xi\omega_n\dot{z}_c(t) + \omega_n^2 z_c(t) = \\ &\quad z_0([\alpha\{2\dot{\omega}(t) + \ddot{\omega}(t)t\} + 2\xi\omega_n\{\omega(t) + \dot{\omega}(t)t\}] \cos(\omega(t)t) + \\ &\quad [\omega_n^2 - \alpha\{\omega(t) + \dot{\omega}(t)t\}^2] \sin(\omega(t)t)) + \frac{F_z(t)}{m_b} - (1 + \alpha)g, \end{aligned} \tag{7}$$

which presents the standard form of second-order mass–spring–damper dynamic with forced input. Considering the effect of (5)

and (6) in the dynamic of six-DoF model, one can rewrite (3) in the form of:

$$m_b\ddot{\boldsymbol{\xi}}_1(t) + m_b g\mathbf{e}_3 = \mathbf{R}_b(\boldsymbol{\xi}_2(t))[\mathbf{F}_B(t) + \mathbf{E}_w(z_c(t), z_w(t), t)], \tag{8}$$

where

$$\mathbf{E}_w(z_c(t), z_w(t), t) = \begin{bmatrix} 0 \\ 0 \\ m_w\ddot{z}_w(t) + c\dot{z}_w(t) + kz_w(t) - c\dot{z}_c(t) - kz_c(t) \end{bmatrix}.$$

It should be noted that gravity and external force are already present in (8) and the effect of base excitation will apply only in the Z-axis direction. It is true that due to the interaction of the rotation matrix in the translation dynamics, a smaller scale of oscillation will appear in the other X- and Y-axis as well.

Selecting the state vector of the system as

$$\mathbf{x}(t) = [\boldsymbol{\xi}_1^\top(t), \boldsymbol{\xi}_2^\top(t), \boldsymbol{\upsilon}_1^\top(t), \boldsymbol{\upsilon}_2^\top(t)]^\top, \tag{9}$$

and derivative of $\mathbf{x}(t)$, result in the state-space representation of the system:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\boldsymbol{\xi}}_1(t) \\ \dot{\boldsymbol{\xi}}_2(t) \\ \dot{\boldsymbol{\upsilon}}_1(t) \\ \dot{\boldsymbol{\upsilon}}_2(t) \end{bmatrix}. \tag{10}$$

Computing the time derivative of (1) and (2), and extracting $\dot{\boldsymbol{\upsilon}}_1(t)$ and $\dot{\boldsymbol{\upsilon}}_2(t)$ provide:

$$\dot{\boldsymbol{\upsilon}}_1(t) = \mathbf{R}_b^{-1}(\boldsymbol{\xi}_2(t))\left(\ddot{\boldsymbol{\xi}}_1(t) - \dot{\mathbf{R}}_b(\boldsymbol{\xi}_2(t))\boldsymbol{\upsilon}_1(t)\right), \tag{11}$$

$$\dot{\boldsymbol{\upsilon}}_2(t) = \mathbf{T}^{-1}(\boldsymbol{\xi}_2(t))\left(\ddot{\boldsymbol{\xi}}_2(t) - \dot{\mathbf{T}}(\boldsymbol{\xi}_2(t))\boldsymbol{\upsilon}_2(t)\right). \tag{12}$$

Substituting (3),(4),(11), and (12) into (10) represents the state-space of the FWFR as (the argument t is not written for the brevity of the representation; the effect of base excitation is also considered in the state-space form):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}, \mathbf{u}) \begin{bmatrix} \mathbf{R}_b(\boldsymbol{\xi}_2)\boldsymbol{\upsilon}_1 \\ \mathbf{T}(\boldsymbol{\xi}_2)\boldsymbol{\upsilon}_2 \\ \mathbf{R}_b^{-1}(\boldsymbol{\xi}_2)\left(\frac{1}{m_b}\mathbf{R}_b(\boldsymbol{\xi}_2)\mathbf{E}_w(z_c, z_w) - m_b g\mathbf{e}_3 - \dot{\mathbf{R}}_b(\boldsymbol{\xi}_2)\boldsymbol{\upsilon}_1\right) \\ \mathbf{T}^{-1}(\boldsymbol{\xi}_2)\left(-\mathbf{J}^{-1}(\boldsymbol{\xi}_2)\mathbf{c}(\boldsymbol{\xi}_2, \dot{\boldsymbol{\xi}}_2) - \dot{\mathbf{T}}(\boldsymbol{\xi}_2)\boldsymbol{\upsilon}_2\right) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \frac{1}{m_b}\mathbf{F}_B \\ \mathbf{T}^{-1}(\boldsymbol{\xi}_2)\mathbf{J}^{-1}(\boldsymbol{\xi}_2)\boldsymbol{\tau}_B \end{bmatrix}. \tag{13}$$

4. Controller design: The state-dependent Riccati equation

4.1. Nonlinear sub-optimal control design for orientation dynamics

Consider a time-invariant nonlinear affine-in-control system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t), \tag{14}$$

where the input vector of the system is $\mathbf{u}(t) \in \mathbb{R}^m$, the state variables are collected in vector $\mathbf{x}(t) \in \mathbb{R}^n$, the state-dependent coefficient (SDC) parameterization of the system is represented by $\mathbf{A}(\mathbf{x}(t)) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ and $\mathbf{B}(\mathbf{x}(t)) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t)$ and $\mathbf{f}(\mathbf{x}(t)) = \mathbf{A}(\mathbf{x}(t))\mathbf{x}(t)$ in (14) are vector-valued smooth functions, piecewise continuous, that satisfies a local Lipschitz condition and:

$$\mathbf{f}(\mathbf{x}(t)) : \mathbb{R}^n \to \mathbb{R}^n,$$
$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n.$$

The objective of optimal control is to minimize the cost functional integral in an infinite-time horizon:

$$J(\cdot) = \frac{1}{2}\int_0^\infty \{\mathbf{u}^\top(t)\mathbf{R}(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{x}^\top(t)\mathbf{Q}(\mathbf{x}(t))\mathbf{x}(t)\}\,dt, \tag{15}$$

where $\mathbf{Q}(\mathbf{x}(t)) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ and $\mathbf{R}(\mathbf{x}(t)) : \mathbb{R}^n \to \mathbb{R}^{m\times m}$ are positive-semidefinite and positive-definite symmetric weighting matrices, respectively. More details on controllability and observability could be revisited in [55].

Constructing the Hamiltonian function:

$$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) = \boldsymbol{\lambda}^\top(t)\{\mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t)\} + \frac{1}{2}[\mathbf{u}^\top(t)\mathbf{R}(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{x}^\top(t)\mathbf{Q}(\mathbf{x}(t))\mathbf{x}(t)],$$

and stationary condition of optimal control, $\frac{\partial\mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\lambda}(t))}{\partial\mathbf{u}(t)} = \mathbf{0}$, result in

$$\mathbf{u}(t) = -\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^\top(\mathbf{x}(t))\boldsymbol{\lambda}(t). \tag{16}$$

Defining the co-state vector as $\boldsymbol{\lambda}(t) = \mathbf{P}(\mathbf{x}(t))\mathbf{x}(t)$ and substituting that in (16), the SDRE control law is found [55]:

$$\mathbf{u}(t) = -\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^\top(\mathbf{x}(t))\mathbf{P}(\mathbf{x}(t))\mathbf{x}(t), \tag{17}$$

where $\mathbf{P}(\mathbf{x}(t)) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the solution to the SDRE and $\mathbf{K}(\mathbf{x}(t)) = \mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^\top(\mathbf{x}(t))\mathbf{P}(\mathbf{x}(t))$ is the sub-optimal gain of the controller.

The optimality condition $\frac{\partial\mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\lambda}(t))}{\partial\boldsymbol{\lambda}(t)} = \dot{\mathbf{x}}(t)$ confirms (14), and the optimality condition $\frac{\partial\mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\lambda}(t))}{\partial\mathbf{x}(t)} = -\dot{\boldsymbol{\lambda}}(t)$ provides:

$$\frac{1}{2}\mathbf{u}^\top\frac{\partial\mathbf{R}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{u} + \mathbf{Q}(\mathbf{x})\mathbf{x} + \frac{1}{2}\mathbf{x}^\top\frac{\partial\mathbf{Q}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{x} + \mathbf{A}^\top(\mathbf{x})\boldsymbol{\lambda} + \mathbf{x}^\top\left(\frac{\partial\mathbf{A}(\mathbf{x})}{\partial\mathbf{x}}\right)^\top\boldsymbol{\lambda} + \mathbf{u}^\top\left(\frac{\partial\mathbf{B}(\mathbf{x})}{\partial\mathbf{x}}\right)^\top\boldsymbol{\lambda} = -\dot{\mathbf{P}}(\mathbf{x})\mathbf{x} - \mathbf{P}(\mathbf{x})\dot{\mathbf{x}}. \tag{18}$$

Substituting control law (17), system (14) and co-state vector $\boldsymbol{\lambda}(t)$ into Eq. (18) and mathematical manipulation, result in:

$$\begin{aligned} &\frac{1}{2}\left(\frac{\partial\mathbf{R}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x}\right)^\top\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x} + \mathbf{Q}(\mathbf{x})\mathbf{x} + \\ &\frac{1}{2}\left(\frac{\partial\mathbf{Q}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{x}\right)^\top\mathbf{x} + \mathbf{A}^\top(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x} + \left(\frac{\partial\mathbf{A}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{x}\right)^\top\mathbf{P}(\mathbf{x})\mathbf{x} + \\ &\left(\frac{\partial\mathbf{B}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x}\right)^\top\mathbf{P}(\mathbf{x})\mathbf{x} + \\ &\dot{\mathbf{P}}(\mathbf{x})\mathbf{x} - \mathbf{P}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x} + \mathbf{P}(\mathbf{x})\mathbf{A}(\mathbf{x})\mathbf{x} = \mathbf{0}. \end{aligned} \tag{19}$$

Eq. (19) provides state-dependent Riccati equation:

$$\mathbf{A}^\top(\mathbf{x})\mathbf{P}(\mathbf{x}) - \mathbf{P}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x}) + \mathbf{Q}(\mathbf{x}) + \mathbf{P}(\mathbf{x})\mathbf{A}(\mathbf{x}) = \mathbf{0}, \tag{20}$$

and "the necessary condition for optimality" [56]:

$$\begin{aligned} &\frac{1}{2}\left(\frac{\partial\mathbf{R}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x}\right)^\top\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x}) + \\ &\frac{1}{2}\left(\frac{\partial\mathbf{Q}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{x}\right)^\top + \left(\frac{\partial\mathbf{A}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{x}\right)^\top\mathbf{P}(\mathbf{x}) + \\ &\left(\frac{\partial\mathbf{B}(\mathbf{x})}{\partial\mathbf{x}}\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^\top(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x}\right)^\top\mathbf{P}(\mathbf{x}) + \dot{\mathbf{P}}(\mathbf{x}) = \mathbf{0}, \end{aligned}$$

which collects the derivative terms of the result of Hamiltonian derivation $\frac{\partial\mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\lambda}(t))}{\partial\mathbf{x}(t)} = -\dot{\boldsymbol{\lambda}}(t)$.

Finite-time or infinite-time horizon control: The proposed controller for practical implementation is chosen infinite horizon SDRE. The infinite-time design and algebraic Riccati equation are based on $t_f \to \infty$ in the cost function integral (15); and results

in, $\dot{\mathbf{P}}(\mathbf{x}) = \mathbf{0}$. It is true that for a short limited time, finite-horizon SDRE could be used, though it needs a more complex solution to the differential Riccati equation. In this work, for the sake of obtaining a shorter sampling time, an infinite-time horizon version is used.

The SDRE (20) generates a solution to Riccati, a positive-definite sub-optimal matrix for the gain of the control law (17). The SDRE controller is used for the orientation control of the FWFR. So, if the orientation dynamic of the FWFR is collected into a subsystem, presented as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{T}(\xi_2)\upsilon_2 \\ \mathbf{T}^{-1}(\xi_2)\left(-\mathbf{J}^{-1}(\xi_2)\mathbf{c}(\xi_2,\dot{\xi}_2) - \dot{\mathbf{T}}(\xi_2)\upsilon_2\right) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{T}^{-1}(\xi_2)\mathbf{J}^{-1}(\xi_2)\tau_B \end{bmatrix},$$

hence, the SDC parameterization of the orientation dynamics is designed as:

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{T}(\xi_2) \\ \mathbf{0}_{3\times3} & -\mathbf{T}^{-1}(\xi_2)\mathbf{J}^{-1}(\xi_2)\mathbf{C}(\xi_2,\dot{\xi}_2)\mathbf{T}(\xi_2) - \mathbf{T}^{-1}(\xi_2)\dot{\mathbf{T}}(\xi_2) \end{bmatrix}, \tag{21}$$

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{T}^{-1}(\xi_2)\mathbf{J}^{-1}(\xi_2) \end{bmatrix}, \tag{22}$$

where $\mathbf{c}(\xi_2(t),\dot{\xi}_2(t)) = \mathbf{C}(\xi_2(t),\dot{\xi}_2(t))\dot{\xi}_2(t)$ in which $\mathbf{C}(\xi_2(t),\dot{\xi}_2(t)) : \mathbb{R}^3\times\mathbb{R}^3 \to \mathbb{R}^{3\times3}$. This process is also referred to as apparent linearization which is a factorization of the state vector $\mathbf{x}(t)$ from $\mathbf{f}(\mathbf{x}(t))$ and the input vector $\mathbf{u}(t)$ from $\mathbf{g}(\mathbf{x}(t),\mathbf{u}(t))$.

Remark 1. The Coriolis term $-\mathbf{T}^{-1}(\xi_2)\mathbf{J}^{-1}(\xi_2)\mathbf{C}(\xi_2,\dot{\xi}_2)\mathbf{T}(\xi_2)$ in SDC matrix (21), is multiplied by $\mathbf{T}(\xi_2)$ from right-hand side which is not presented in state-space form (13). If $\dot{\xi}_2$ is factored from Coriolis vector $\mathbf{c}(\xi_2,\dot{\xi}_2)$, it can be seen that $\mathbf{C}(\xi_2,\dot{\xi}_2)\dot{\xi}_2$ is not compatible with state-vector (9). So, the kinematic Eq. (2) is substituted in that to form $\mathbf{C}(\xi_2,\dot{\xi}_2)\mathbf{T}(\xi_2)\upsilon_2$.

Remark 2. The derivative of rotation matrices, $\dot{\mathbf{T}}(\xi_2(t))$ and $\dot{\mathbf{R}}_b(\xi_2(t))$ are numerically computed in the control loop of the simulations.

The control law of the orientation dynamic is presented as

$$\tau_B(t) = -\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^{\top}(\mathbf{x}(t))\mathbf{P}(\mathbf{x}(t))\mathbf{e}(t), \tag{23}$$

where

$$\mathbf{e}(t) = \begin{bmatrix} \phi(t)-\phi_{des} \\ \theta(t)-\theta_{des} \\ \psi(t)-\psi_{des} \\ \dot{\phi}(t)-\dot{\phi}_{des} \\ \dot{\theta}(t)-\dot{\theta}_{des} \\ \dot{\psi}(t)-\dot{\psi}_{des} \end{bmatrix}, \tag{24}$$

shifts the equilibrium point from zero to a constant value in the workspace of the system; subscript "des" stands for the desired value. The gain $\mathbf{P}(\mathbf{x}(t))$ in (23) is a solution to the SDRE (20), updated by SDC (21), (22), weighting matrices $\mathbf{R}(\mathbf{x}(t))$ and $\mathbf{Q}(\mathbf{x}(t))$ of appropriate dimension. The block diagram of the control design for simulation is presented in Fig. 11. It demonstrates the orientation control using the SDRE which is fed to the orientation dynamic and the PD and PI control of the translation dynamic which will be explained in Section 4.2.

4.2. Linear control design for translation dynamic

An average forward speed is considered for the X-axis motion in control design and simulation, based on the flight assumptions in Section 2. In the experiments, it happens in reality like that; in a short 15 (m) flight zone with an initial launching speed of 4 (m/s), the speed of the robot in X-axis reduces slightly from 4 (m/s) and stays around that value. Therefore $F_x(t) = 0$ is set to zero.

The error of the Z-axis is set as $e_z(t) = z_c(t) - z_{des}$ and the one for Y-axis is defined by $e_y(t) = y_c(t) - y_{des}$. A proportional-derivative (PD) controller is considered for lateral control:

$$F_y(t) = -K_P e_y(t) - K_D \dot{e}_y(t), \tag{25}$$

where $K_P \in \mathbb{R}^+$ and $K_D \in \mathbb{R}^+$ are proportional and derivative gains of PD control.

Considering Assumptions 1 and 2, $\omega_{min} \le \omega(t) \le \omega_{max}$ in which $\omega_{min} = 3.5$(Hz) is the minimum frequency of the flapping, $\omega_{max} = 4.5$(Hz) is the maximum one, and $\omega(t)$ is the flapping frequency which is one of the control inputs. This limited change in the frequency of the oscillation ascends and descends the FWFR which means the elastic term and gravity are compensated indirectly by input lift force produced by flapping [52]:

$$F_z(t) := m_b g \mathbf{R}_b^{-1}(\xi_2(t))\mathbf{e}_3 + kz_c(t) + \tau(t), \tag{26}$$

where

$$\tau(t) = \frac{(\omega(t)-\omega_{min})(p_{max}-p_{min})}{(\omega_{max}-\omega_{min})} + p_{min},$$

that maps $\omega(t)$ from range of $[21.9911, 28.2743]$ (rad/s) to the range of $p \in [-0.5, 0.5]$. A proportional–integral (PI) control finally regulates the frequency around a local equilibrium point for the flapping frequency $\omega_0 = 4$(Hz) $= 25.1327$ (rad/s) which keeps the robot flying steadily in the air:

$$\omega(t) = -K_{P,f} e_z(t) - K_{I,f}\int_0^t e_z(\sigma)\,d\sigma + \omega_0, \tag{27}$$

where $K_{P,f} \in \mathbb{R}^+$ and $K_{I,f} \in \mathbb{R}^+$ are proportional and integral gains of PI control.

A PD controller has been chosen for the lateral control, the servomotor of the rudder. Since the design is done for a short final time of 4 (s), simple control structures are favorable to simplify the tuning. The tuning of the PD control was done through a series of iterations. The height control was sensitive to derivative terms and presented acceptable results with PI control. One of the difficulties in the control design was the fact that within the 15 (m) flight distance, the robot will be still inside the transient zone of the control design, which will be discussed more in Section 5. The robustness of the controllers, SDRE, PD, and PI was not studied in this work and the SDRE is in nature a nonlinear optimal method. So, the advantage of this approach is in optimality rather than robustness.

The proposed control design is a combination of the SDRE for orientation and linear controllers for translation, Y- and Z-axis. Both controllers are popular methods, applied on a variety of systems, i.e. the application of SDRE could be found in [55]. This specific design showed a successful performance in the simulations which will be verified in the experiments as well.

5. Simulation results

The time of the simulation is considered 4 (s) similar to the real results, the initial condition is set as $\mathbf{x}(0) = [-4.7, 3.5, 1.8, 0, 0.15, -0.67, 4, \mathbf{0}_{1\times5}]^{\top}$ and the final condition is chosen as $\mathbf{x}(4) = [9, -3.5, 2, 0, -0.55, 0, \mathbf{0}_{1\times6}]^{\top}$. The initial and final conditions of the flight are defined based on the experimental data and the confined space of the Opti-Track testbed to compare the model with the real data and prototype. The forward velocity of the robot in the X-axis is also set as $u(t) = 3.6$ (m/s). The parameters of the robot are presented in Table 1.

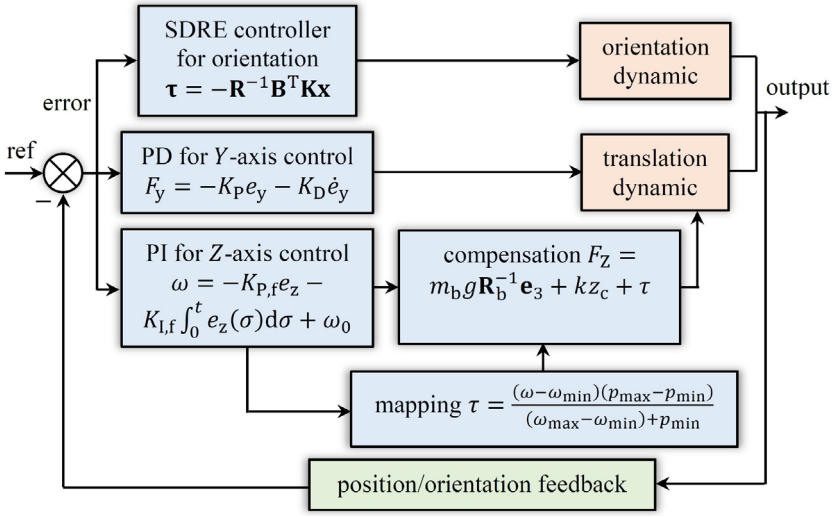

Fig. 11. The block diagram of the control design for the simulation section.

Table 1
The parameters of the flapping-wing flying robot.

Definition	Parameter	Value	Unit	Method
total mass	m	0.6239	kg	measured
body mass	m_{b}	0.4934	kg	measured
wings mass	m_{w}	0.1305	kg	measured
moment of inertia	I_{xx}	0.0124	kg m^2	computed
moment of inertia	I_{yy}	0.0136	kg m^2	computed
moment of inertia	I_{zz}	0.0136	kg m^2	computed
Elastic modulus CF	E	65×10^9	N/m^2	catalogue
gravity constant	g	9.81	m/s^2	–

The distance between the CoM of the bird and the effective lift force of the left or right wing is $L = 0.25$ (m), The moment of inertia of cross-sectional area of the wing tube is $I = 5.1051 \times 10^{-11}$. Considering the tube acting like a clamped-free beam connected to the joint, the equivalent elastic constant is defined by $k = 3EI/L^3 = 637.11$ (N/m). The natural frequency of the system is $\omega_{\mathrm{n}} = \sqrt{(k/m_{\mathrm{b}})} = 35.9343$ (rad/s) which results in damping coefficient $c = 0.3901$(N.s/m) where the damping ratio was set $\xi = 0.011$; the amplitude of the motion due to the flapping is also set $z_0 = 0.025$ (m). The damping ratio, natural frequency, and physical characteristics of the dynamic have been defined based on Eq. (7).

The control parameters are set as $\mathbf{Q} = I_{6\times6}$ and $\mathbf{R} = I_{3\times3}$ which control the orientation dynamic; $K_{\mathrm{P,f}} = 3$ and $K_{\mathrm{I,f}} = 0.6$ which control the height and finally, $K_{\mathrm{P}} = 0.85$ and $K_{\mathrm{D}} = 0.5$ which control the lateral motion of the robot.

Tuning: Identity matrices are possible choices for initial values for the gain of the SDRE controller; here the identity for **Q** and **R** were selected, which resulted in a proper orientation control. The selection of linear gains is more sensitive. For the tuning, we start by setting the proportional gain for $K_{\mathrm{P,f}}$ and K_{P} until the height and lateral component of the robot regulates with enough speed. At this stage, the error constraint is relaxed and the power of the signal is important. Then the next parameter for tuning of height and lateral controllers are $K_{\mathrm{I,f}}$ (integral gain) and K_{D} (derivative gain). They will be changed until the error is reduced to an allowable range for the control task.

The trajectory of the robot bird and its configuration are plotted in Fig. 12 which shows the flight from the initial condition to the final one with the error of 0.0854 (m) for both Y- and Z-axis. The height state variable, $z_{\mathrm{c}}(t)$, is illustrated in Fig. 13. It starts from the initial condition and regulates to the desired point

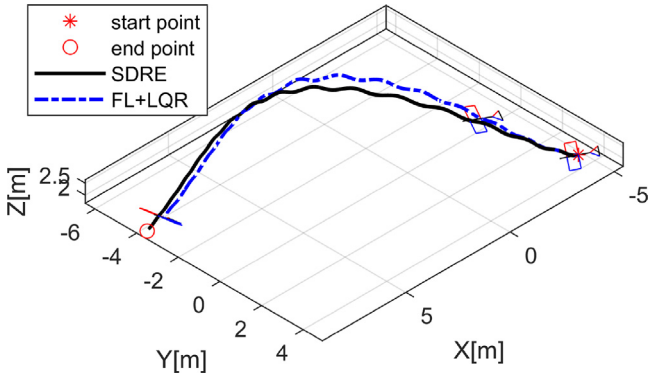

Fig. 12. The 3D trajectory of the FWFR and configuration of the robot in flight. The FL+LQR is the result of the comparison which will be presented at the end of Section 5.

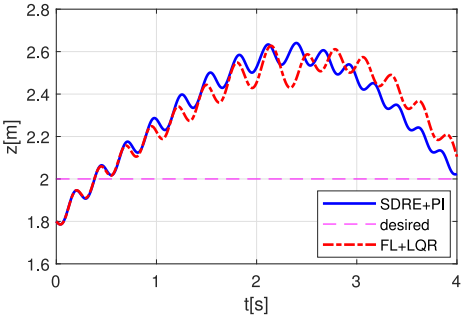

Fig. 13. The height state variable of the system in simulation.

with an overshoot of 60 (cm). Comparing the height variable with Fig. 9, it can be seen that the overshoot in real data is less and there is a kind of dive at the beginning of the flight. The reason for the dive is that the flapping in practice starts after the launching, so as not to disturb the initial condition of the bird on the launcher. Hence, there is a small dive at the beginning of the motion in experimental data. The amplitude of the oscillatory motion due to flapping is approximately 10 (cm), which can be observed in Fig. 12. The height control and frequency of the flapping are illustrated in Fig. 14, constrained by upper and lower bounds. The position and orientation state variables are also

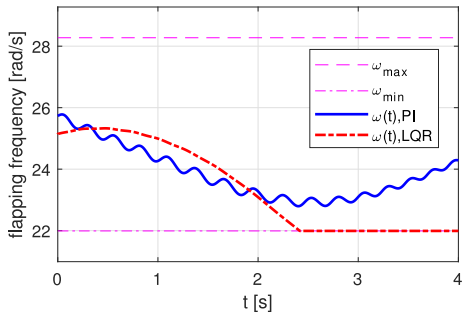

Fig. 14. The flapping frequency of the FWFR in simulation, controlled by PI. The LQR is the signal for the comparison at the end of Section 5.

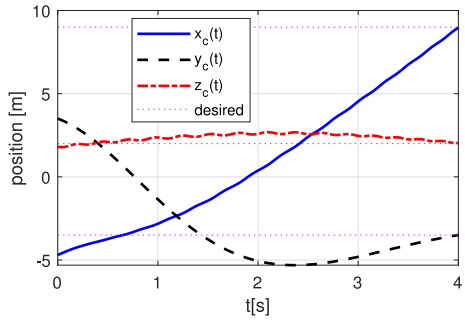

Fig. 15. The position states of the system in simulation.

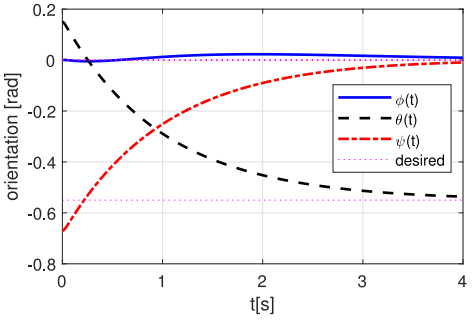

Fig. 16. The orientation state variables of the system in simulation.

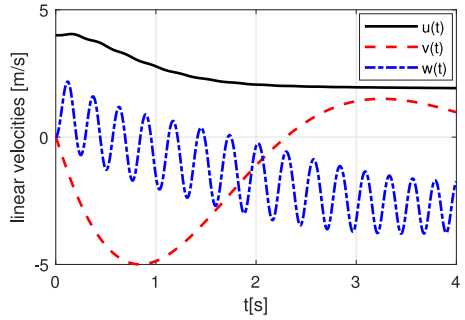

Fig. 17. The linear velocities (translation) of the FWFR in the simulation.

presented in Figs. 15 and 16, respectively. The linear velocities of the FWFR are illustrated in Fig. 17; the velocity of the X-axis started from the initial condition $\dot{x}_c(0) = 4$ (m/s) and adjust itself towards the desired speed $u(t) = 3.6$ (m/s). Note that there is also a transformation between $\dot{x}_c(t)$ and $u(t)$, see Eq. (1).

Comparing the motion of the FWFR in Fig. 9 with simulation results, a kind of dive at the beginning of the flight is observable.

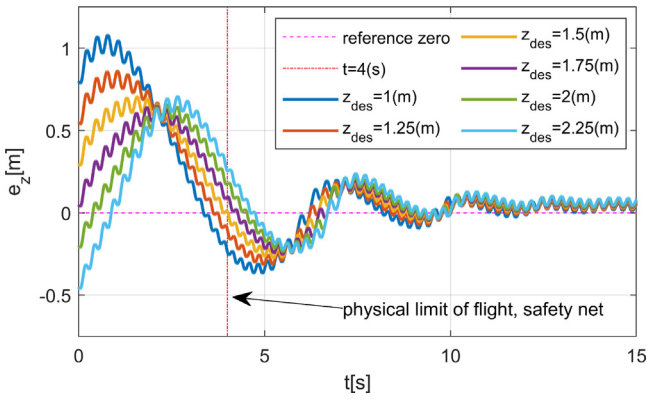

Fig. 18. The height error in regulation for different desired conditions; the control gains of the height are set $K_{P,f} = 3$ and $K_{I,f} = 0.225$.

The launching mechanism of the robot bird is a linear guide with a DC motor that provides the initial speed of the flight. At the end of the guide, the robot detaches from the launcher and starts flapping. Therefore, after a couple of flapping, the FWFR will recover itself from the dive and regulates towards the final height. The simulation does not reflect this dive since it assumes a perfect condition and flapping at the exact time of detachment from the launcher. A new adapter has been designed for the launcher to put the FWFR on top of the system (previously, it was on the side of the launcher) to reduce the dive at the beginning of the flight.

Control design in transient flight, forced by the confined limited space: The dynamics of the flapping-wing robot are very slow with respect to forward motion. It should be noted that the Opti-Track workspace of the GRVC laboratory[2] is relatively big though the 15 (m) diagonal flight zone is not enough to pass the transient dynamics. The error of the height control for the flight trajectories of the FWFR is shown in Fig. 18. It is clear that at 4 (s) which is our limited time for the corresponding limited space, performance of the all desired conditions is not the same; the controller works though the errors converge to zero around 15 (s) which is the steady-state condition.

This subsection was provided to show the problem of confined space for the flight which limited the control design. It is an undesirable condition for the simulation and experiment; however, the modeling and design were performed in a way to address at least a logical set of conditions with enough repetitions to show also the reliability of the design. More data on the flight experiments will be provided in Section 6. More details on transient flight data could be visited in Ref. [52], the interaction of the orientation dynamic with the equivalent vertical one exercised here in this work.

Fig. 18 shows that the control gains of the height result in different values for various desired conditions. The proposed gains provide enough accuracy for 2 (m) set-point, though, for the other desired heights, the errors are higher at 4 (s). That means the design for the 2 (m) is done in the transient flight zone. If one increases the time of the simulation, as it has been done in Fig. 18, after passing the transient, the errors for all set points converge towards zero. This undesirable situation in the simulation and experimental controller design in this work was the result of the confined space of the test bed. As a result, to change the settling point of the regulation, the control parameters can be enhanced though it will not be clear how that can affect the error at 4 (s) which is in the transient zone. So far the tuning has been

2 https://grvc.us.es/newweb/infraestructures/

done to achieve the best possible results in the simulation and experimentation. The repetition of the test in Section 6 shows the reliability of the controller for the defined set of initial and final conditions.

Comparison: A brief comparison has been done in this part to assess the performance of the SDRE (for orientation dynamic) and linear controllers (PD/PI for translation dynamic) with respect to two popular methods, feedback linearization (FL) and linear quadratic regulator (LQR) plus an integrator. The FL approach is nonlinear and LQR linear optimal, hence the comparison would be a fair one. The linear system matrices for the LQR have been defined as $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k/m_\mathrm{b} & -c/m_\mathrm{b} \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 0 \\ 1/m_\mathrm{b} \end{bmatrix}$. This controller is designed for the height regulation of the FWFR with tuning of the weighting matrices $\mathbf{Q}_{\mathrm{LQR}} = \mathrm{diag}(1, 0)$ and $R_{\mathrm{LQR}} = 0.06$ which results in the control law:

$$u_{\mathrm{LQR}}(t) = \omega(t) = -R_{\mathrm{LQR}}^{-1}\mathbf{B}_{\mathrm{LQR}}^{\top}\mathbf{P}_{\mathrm{LQR}}\begin{bmatrix} e_z(t) \\ \dot{e}_z(t) \end{bmatrix} - K_{\mathrm{I,LQR}}\int_0^t e_z(\sigma)\,\mathrm{d}\sigma + \omega_0,$$

in which $\mathbf{P}_{\mathrm{LQR}} = \begin{bmatrix} 0.6199 & 0.0004 \\ 0.0004 & 0.0005 \end{bmatrix}$ and $K_{\mathrm{I,LQR}} = 5$ and can be compared with (27).

The orientation control is defined by the FL law:

$$\boldsymbol{\tau}_{\mathrm{B,FL}}(t) = \mathbf{J}(\boldsymbol{\xi}_2(t))\mathbf{T}(\boldsymbol{\xi}_2(t))\mathbf{v}_{\mathrm{FL}}(t) + \mathbf{C}(\boldsymbol{\xi}_2(t), \dot{\boldsymbol{\xi}}_2(t))\mathbf{T}(\boldsymbol{\xi}_2(t))\boldsymbol{\upsilon}_2(t) + \mathbf{J}(\boldsymbol{\xi}_2(t))\mathbf{T}(\boldsymbol{\xi}_2(t))\dot{\mathbf{T}}(\boldsymbol{\xi}_2(t))\boldsymbol{\upsilon}_2(t),$$

where $\mathbf{v}_{\mathrm{FL}}(t)$ is a linear PD controller with identity matrix gains. Simulating the comparison code, the configuration of the FWFR and the trajectories are found, presented in Fig. 12. The error of the FL+LQR controller was found 0.8537 (m). The input of the LQR controller is presented in Fig. 14 which shows saturation of the flapping on the lower bound. The controller tried to reduce the speed of flapping to reduce the height which was the result of the high gain of the integrator, clearly without the high gain the error would be more. This analysis shows that linear controller without derivative term performs better for height regulation.

Robustness analysis: The SDRE is sensitive to model uncertainty and disturbance without a mechanism to compensate for it. The robust characteristics can be added through augmentation of other tools such as sliding mode control [57], iterative learning control [44] or H_∞ design [58], with the SDRE. However, reasonable robustness exists in the design, provided by the nonlinearity and closed-loop feedback which can be checked with a simulation. The damping ratio ξ is a critical parameter which defines the damping coefficient $c = 2m_\mathrm{b}\omega_\mathrm{n}\xi$(N.s/m). The damping coefficient changes the characteristics of the vertical dynamic and the interaction between the translation and orientation dynamics affects the performance of the entire system. A series of different values are defined for the damping ratio, $\xi = \{0.001, 0.005, 0.011, 0.05, 0.1\}$, to check the robustness of the system and the effect of that on the final error of the regulation. The result of the simulation is presented in Fig. 19 which shows different trajectories caused by parameter changes. The error of system is found $E = \{0.8073, 0.3561, 0.0854, 1.0980, 1.2634\}$ (m). The results show that without a mechanism to compensate for the uncertainty and parameter changes, the performance will drop although the system regulates towards the final desired coordinate. The worst-case error is still acceptable with an 8% error of the travel distance.

6. Experimental results

The position $\{x_\mathrm{c}(t), y_\mathrm{c}(t), z_\mathrm{c}(t)\}$ and orientation $\{\phi(t), \theta(t), \psi(t)\}$ feedback of the FWFR are provided by a motion capture system which is read by ROS1 and the virtual reality peripheral network. A ROS node broadcasts the IP and the name of "FWFR", and a subscriber will read the position and orientation of the defined body through "geometry messages".

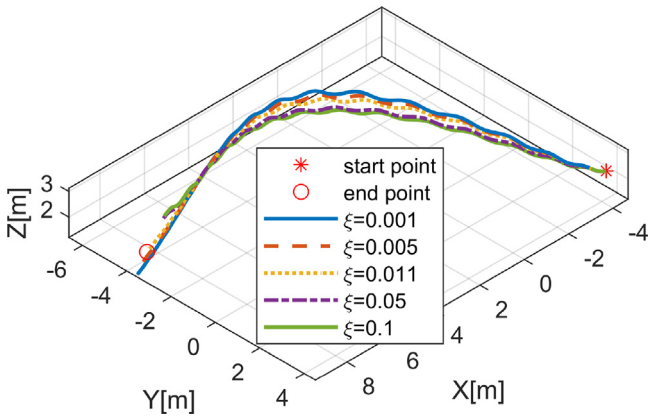

Fig. 19. The robustness analysis of the system.

The MATLAB script codes of the simulation section are transformed into a Python3 script file for experimental implementation. The first part of the control loop is to transform the quaternion orientation data of the ROS node to Euler angles for updating the SDC matrices. To simplify the computation load of the onboard computer, derivative term $\dot{\mathbf{T}}(\boldsymbol{\xi}_2(t)) \approx \mathbf{0}$, then the following matrices are updated at each time-step of the control loop:

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{T}(\boldsymbol{\xi}_2) \\ \mathbf{0}_{3\times3} & -\mathbf{T}^{-1}(\boldsymbol{\xi}_2)\mathbf{J}^{-1}(\boldsymbol{\xi}_2)\mathbf{C}(\boldsymbol{\xi}_2, \dot{\boldsymbol{\xi}}_2)\mathbf{T}(\boldsymbol{\xi}_2) \end{bmatrix},$$

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{T}^{-1}(\boldsymbol{\xi}_2)\mathbf{J}^{-1}(\boldsymbol{\xi}_2) \end{bmatrix}.$$

The function "care" from the control library of Python (control - PyPI) is used to find the gain of the optimal control $\mathbf{P}(\mathbf{x})$ to use in the control law (23). The input torque has a nonlinear-complex relation with the tail actuators and is presented by:

$$\boldsymbol{\tau}_\mathrm{B}(t) = \begin{bmatrix} \tau_\phi(t) \\ \tau_\theta(t) \\ \tau_\psi(t) \end{bmatrix} = \mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \boldsymbol{\delta}(t)), \tag{28}$$

where $\boldsymbol{\delta}(t) = [\delta_\mathrm{R}(t), \delta_{\mathrm{ER}}(t), \delta_{\mathrm{EL}}(t)]^\top$ (rad), please see Fig. 8.

Remark 3. The vector-valued function $\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \boldsymbol{\delta}(t))$ is unknown, and derivation of that requires aerodynamics computation and computational fluid dynamics analysis for verification/validation. Considering input nonlinearity in $\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \boldsymbol{\delta}(t))$ imposes a non-affine control problem which is undesirable for experimentation at this stage since we do not have feedback from the inputs. To remove this complexity from the implementation, a linearization approach is used which was presented for the simulation of six-DoF FWFR using terminal sliding mode control design [36].

$\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \boldsymbol{\delta}(t))$ is unknown and could be nonlinear in state $\mathbf{q}(t), \dot{\mathbf{q}}(t)$ and input $\boldsymbol{\delta}(t)$. The objective is to find a constant matrix coefficient multiplied by the inputs, $\mathbf{M}_\mathrm{x}\boldsymbol{\delta}(t)$, to replace $\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \boldsymbol{\delta}(t))$. As a result, Taylor series expansion with respect to $\boldsymbol{\delta}(t)$ is done around equilibrium point $\mathbf{q}(t) = \dot{\mathbf{q}}(t) = \mathbf{0}$, and only partial derivatives of the rudder, elevator of right and left will appear in the expansion. A linearization will be performed around the equilibrium point of the tail to change the non-affine control problem to an affine simplified version. The Taylor series

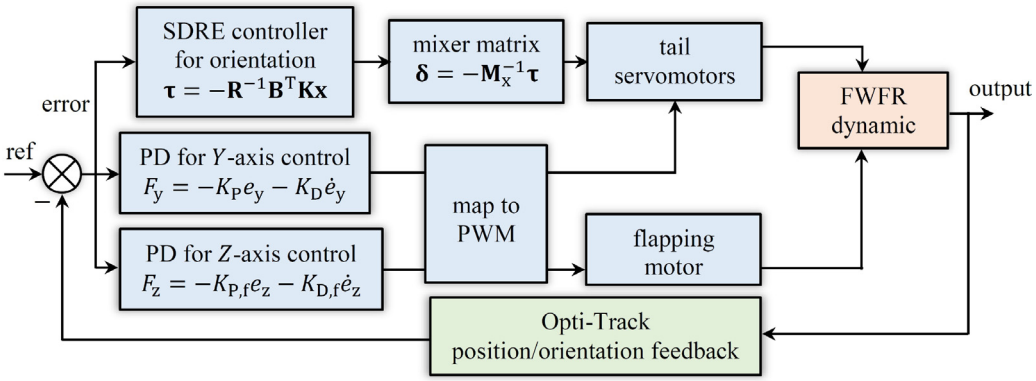

Fig. 20. The control block diagram of the experiment.

expansion of (28) results in [36]:

$$\boldsymbol{\tau}_{\mathrm{B}}(t) = [\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{0})]_{\mathbf{q}(t)=\dot{\mathbf{q}}(t)=\mathbf{0}} + \left[\frac{\partial \mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{0})}{\partial \boldsymbol{\delta}(t)}\right]_{\mathbf{q}(t)=\dot{\mathbf{q}}(t)=\mathbf{0}} \boldsymbol{\delta}(t) + \frac{1}{2}\left[\frac{\partial^2 \mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{0})}{\partial \boldsymbol{\delta}^2(t)}\right]_{\mathbf{q}(t)=\dot{\mathbf{q}}(t)=\mathbf{0}} \boldsymbol{\delta}^2(t) + \cdots$$

All the components of $\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \boldsymbol{\delta}(t))$ have at least one term of $\boldsymbol{\delta}(t)$; therefore, setting $\boldsymbol{\delta}(t) = \mathbf{0}$ results in $\mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{0}) = \mathbf{0}$. Neglecting the higher terms of Taylor expansion, a linearized relation can be presented as:

$$\boldsymbol{\tau}_{\mathrm{B}}(t) = \underbrace{\begin{bmatrix} \frac{\partial \tau_\phi(t)}{\partial \delta_{\mathrm{R}}(t)} & \frac{\partial \tau_\phi(t)}{\partial \delta_{\mathrm{ER}}(t)} & \frac{\partial \tau_\phi(t)}{\partial \delta_{\mathrm{EL}}(t)} \\ \frac{\partial \tau_\theta(t)}{\partial \delta_{\mathrm{R}}(t)} & \frac{\partial \tau_\theta(t)}{\partial \delta_{\mathrm{ER}}(t)} & \frac{\partial \tau_\theta(t)}{\partial \delta_{\mathrm{EL}}(t)} \\ \frac{\partial \tau_\psi(t)}{\partial \delta_{\mathrm{R}}(t)} & \frac{\partial \tau_\psi(t)}{\partial \delta_{\mathrm{ER}}(t)} & \frac{\partial \tau_\psi(t)}{\partial \delta_{\mathrm{EL}}(t)} \end{bmatrix}}_{\mathbf{M}_{\mathrm{x}}} \boldsymbol{\delta}(t). \tag{29}$$

The constant matrix $\mathbf{M}_{\mathrm{x}} = \left[\frac{\partial \mathbf{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \mathbf{0})}{\partial \boldsymbol{\delta}(t)}\right]_{\mathbf{q}(t)=\dot{\mathbf{q}}(t)=\mathbf{0}}$ is called a mixer matrix that shows the linearized relationship between the input torque and the rudder and elevators angles. Observing the geometry of the FWFR in Fig. 8 and also the tail motion in Fig. 4, some of the components of (29) can be set zero:

$$\mathbf{M}_{\mathrm{x}} = \begin{bmatrix} 0 & \frac{\partial \tau_\phi(t)}{\partial \delta_{\mathrm{ER}}(t)} & \frac{\partial \tau_\phi(t)}{\partial \delta_{\mathrm{EL}}(t)} \\ 0 & \frac{\partial \tau_\theta(t)}{\partial \delta_{\mathrm{ER}}(t)} & \frac{\partial \tau_\theta(t)}{\partial \delta_{\mathrm{EL}}(t)} \\ \frac{\partial \tau_\psi(t)}{\partial \delta_{\mathrm{R}}(t)} & 0 & 0 \end{bmatrix},$$

and finally, the rudder and elevators angles can be found by:

$$\boldsymbol{\delta}(t) = \mathbf{M}_{\mathrm{x}}^{-1}\boldsymbol{\tau}_{\mathrm{B}}(t).$$

There are several methods to find the mixer matrix such as mathematical modeling, system identification, or definition of that experimentally. In this work, the third method is used since we have access to the experimental platform and we directly find $\mathbf{M}_{\mathrm{x}}^{-1}$. By defining the matrices and control law in the code, and changing the orientation of the FWFR, a mapping will be obtained that changed the scale of the torque to the scale of the angles (or in other words the PWM signal of the servos). This action will be tuned in a couple of flights in the experiments.

The other library needed for Python code is "Adafruit_PCA9685" to connect the Raspberry PI processor to the PWM module. Then the desired angles will be set to the servomotors by "pwm.set_pwm" command. Similar to Section 5, linear controllers will be assigned for the flapping control, commanded as a PWM signal to the brushless DC motor, and the PD control of the Y-axis is augmented to the rudder PWM with corresponding $K_{\mathrm{P}} = 67$ and $K_{\mathrm{D}} = 72$ gains.

The weighting matrices of the nonlinear optimal controller are set identity similar to simulations, $\mathbf{Q} = I_{6\times6}$ and $\mathbf{R} = I_{3\times3}$. That will generate a nonlinear gain $\mathbf{K}(\mathbf{x}(t)) = \mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^{\top}(\mathbf{x}(t))\mathbf{P}(\mathbf{x}(t))$, multiplied by orientation error vector (24). The control input $\boldsymbol{\tau}_{\mathrm{B}}(t) = -\mathbf{K}(\mathbf{x}(t))\mathbf{e}(t)$ provides a stable negative-feedback control law in torque scale, (N.m) unit. To find the mapping between the torque $\boldsymbol{\tau}_{\mathrm{B}}(t)$ and PWM commands (angles of rudder and elevators) to servomotors $\boldsymbol{\delta}(t)$, the effective components of mixer is found and the signs of the components are set based on the direction of the servomotor actuation which results in $\mathbf{M}_{\mathrm{x}}^{-1} = \begin{bmatrix} 0 & 0 & a \\ -b & c & 0 \\ -d & -e & 0 \end{bmatrix}$ where a, b, c, d, e are positive constants, i.e. a must be positive since the stable negative feedback in $\boldsymbol{\tau}_{\mathrm{B}}(t)$ is already compatible with the direction of the servomotor of the rudder. That means the torque direction based on Fig. 8 is similar to the motion of the servomotor of the rudder in the experimental platform. A similar definition of the motion of the servomotors must be done to check the signs of the components of $\mathbf{M}_{\mathrm{x}}^{-1}$. The next step is to find the values a, b, c, d, e experimentally. Setting an initial value and moving the robot inside the OptiTrack testbed (manually, without flapping) shows if the values are sufficient to actuate the servomotors rapidly enough. The last phase is fine-tuning through a series of flight experiments which results in:

$$\mathbf{M}_{\mathrm{x}}^{-1} = \begin{bmatrix} 0 & 0 & 300 \\ -150 & 150 & 0 \\ -150 & -150 & 0 \end{bmatrix}.$$

The gain 150 is for elevators right and left which are symmetrically designed, therefore the negative sign for one of them corrects the direction of actuation for both elevators. The gain of 300 is for the rudder that must be faster and high gain does not damage the servomotor since it is not working under the effect of gravity. The values of the inverse of the mixer matrix map the torque to PWM and increasing or decreasing it significantly leads to input signals outside the minimum and maximum bounds of the PWM signals.

The height control simulation worked properly with PI control; however, in the experiment, the results were better with PD control, with corresponding gains, $K_{\mathrm{P,f}} = 250$ and $K_{\mathrm{D,f}} = 100$. The block diagram of the experiment control is presented in Fig. 20. The experiment block diagram is a little bit different from the simulation one. In the simulation, the control signals of the systems are force and torque though in the experiment the final input signals are PWMs. Therefore mapping block and mixer matrix are introduced in the experiment section. The dynamic of the FWFR is also unified and the Y-axis signal is controlled by the rudder as well. With the same initial and final conditions in

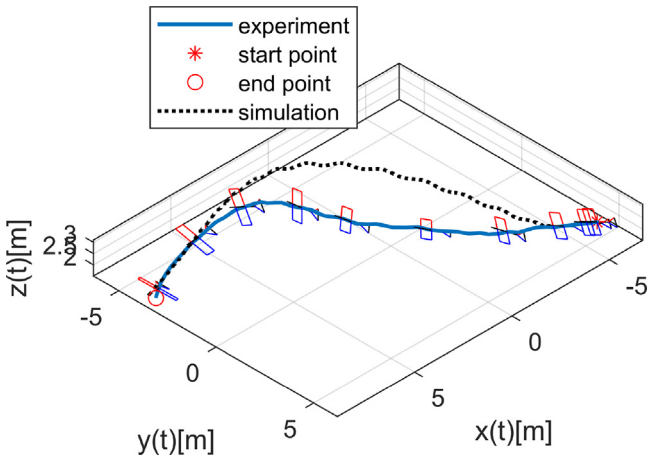

Fig. 21. The 3D trajectory of the FWFR and configuration of the robot in flight; experimental results and comparison with simulation.

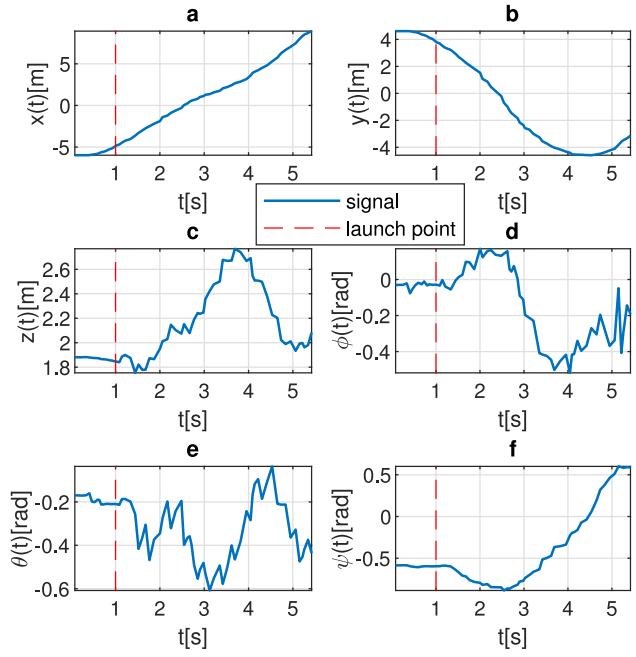

Fig. 22. The generalized coordinates of the flapping-wing robot in the experiment.

Section 5, the experiments were performed. The trajectory of the FWFR and the one in simulations are compared in Fig. 21.

The simulation flight data has more overshoot in the Z-axis in comparison with the experimental result. The error of the simulation was gained 0.0854 (m) which is 0.55% in a 15.38 (m) flight path. The error in Z-axis and Y-axis in experiment were obtained 0.083 (m) and 0.37 (m) which results in 0.53% and 2.4% precision in regulation. The mechanism for switching off the robot is that when the $x_c(t)$ variable passes the longitudinal limit, the robot stops flapping and lands on the safety net of the Opti-Track testbed. The orientation variables were also regulated to desired values with an error of $e_\phi(t_f) = 0.19$ (rad), $e_\theta(t_f) = 0.43$ (rad) and $e_\psi(t_f) = 0.03$ (rad). The objective was to regulate the FWFR with less position error rather than orientation, so the tuning was done to minimize the height and lateral error. Therefore the orientation error is bigger than the translation one. The position and orientation states were shown in Fig. 22. The

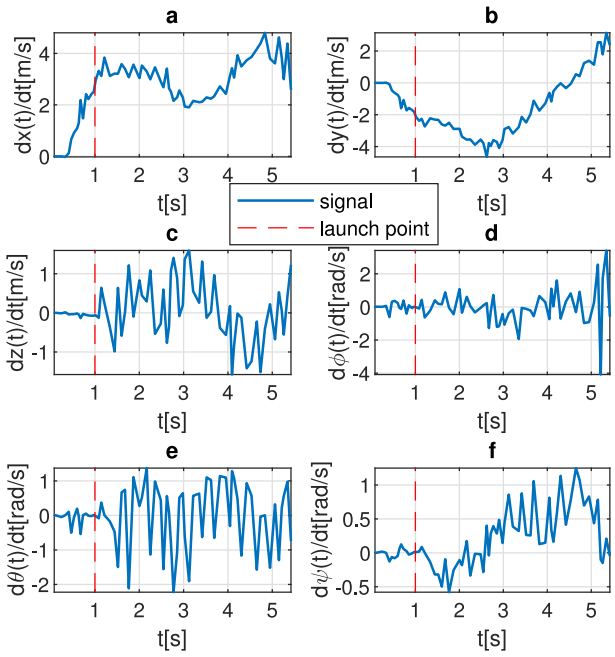

Fig. 23. The velocity of the generalized coordinates of the flapping-wing robot in the experiment.

velocity states were illustrated in Fig. 23. The velocity variables are close to zero at the end of the experiment which shows they are getting close to passing the transient state of the flight.

The PWM range of the PCA9685 is usually between PWM $\in$ [150, 600], which depends on the type and frequency range of the servomotors. The PWM input to flapping actuation is illustrated in Fig. 24-(a). The maximum and minimum bounds of flapping, through the brushless DC motor, is $\text{PWM}_{\text{flapping}} \in [465, 415]$, defined experimentally to regulate the bird in a safe flapping zone without damaging the robot; one example is breaking the hinges due to fast flapping. The PD control of the flapping input works around a local equilibrium $\text{PWM}_{\text{flapping},0} = 425$. The range for the rudder is $\text{PWM}_{\text{rudder}} \in [495, 235]$ with equilibrium position at $\text{PWM}_{\text{rudder},0} = 365$. The input bounds for right and left elevators are $\text{PWM}_{\text{elevator,right}} \in [495, 350]$ and $\text{PWM}_{\text{elevator,left}} \in [470, 325]$ with zero position $\text{PWM}_{\text{elevator,right},0} = 395$ and $\text{PWM}_{\text{elevator,left},0} = 425$, respectively. The elevators have a different range of PWM signals since they are installed in a mirror configuration (opposite direction). The input signal to the rudder is shown in Fig. 24-(b), starting from saturation to generate maximum input to the bird to correct the ψ orientation and reduce the Y-axis error. The elevator signals were presented in Fig. 24-(c) and -(d), controlling the orientation of the robot in ϕ and θ angles using the state-dependent Riccati equation. The sampling time of the control loop is plotted in Fig. 25, with an average value of 0.08 (s). The sampling time with a simple PD control is quite short, less than 0.005 (s) though adding the nonlinear optimal control to the control loop, specifically using the "care" command, increased the sampling time significantly. Considering this sampling time and the fact that the dynamic of the flapping-wing robot is very slow, the system successfully regulated to the desired condition. A snapshot of the flight is presented in Fig. 26; the corresponding video of the flight is available as supplementary material of the paper.

More tests have been performed to show the repeatability of the controller, design, and implementation of the approach, presented in Fig. 27. The error details of the FWFR in 11 trials

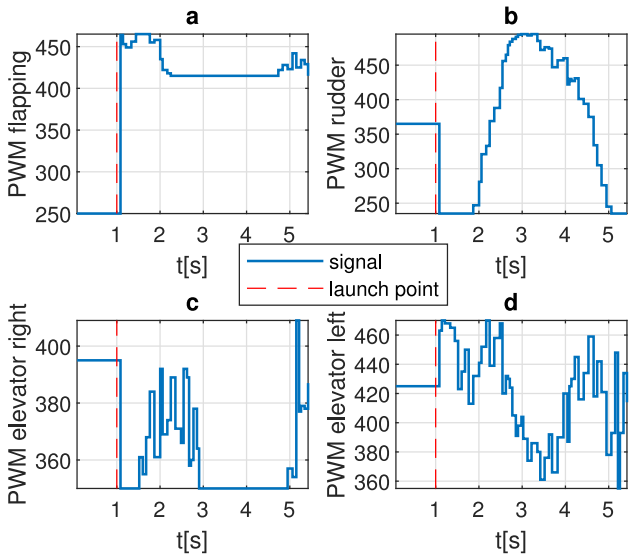

Fig. 24. The input PWM to servomotors and brushless DC of the FWFR in the flight experiment.

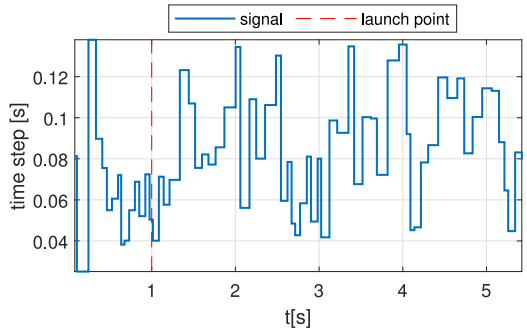

Fig. 25. The sampling time of the flight experiment.

of flights are presented in Table 2. The behavior of the robot is similar in flights, see Fig. 27-(c) which shows the 3D trajectories of the system. The Y-axis control just finished the first overshoot of the transient flight near the end of the trajectories, Fig. 27-(a). The height control is presented in Fig. 27-(b); regulating towards 2 (m) desired height. Based on the discussion on the transient flight in Section 5, it is clear that the flight zone for the flapping-wing robot must be bigger than the available Opti-Track system. The best error in Y-axis was gained 0.0493 (m) and for Z-axis 0.0434 (m), the overall error also was gained 0.0657 (m). The worst overall error was obtained 0.7417 (m). More statistics of the data are provided in Table 2. The median error of flight was found 0.4137 (m).

Error analysis: Now is the time to answer this critical question; if the FWFR could perch on a branch in Ref. [7] with the presented error of 15 (cm), why one should go under the machinery of the SDRE, the complexity of nonlinear control and the risk of bigger sampling time for obtaining average 40 (cm) overall error in X- and Y-axis? It must be noted that the previous flights in perching were done in a diagonal straight line trying to regulate the height towards the branch's vertical position. The linear controller was active in the Z-axis, pitch angle, and yaw angle. By controlling the yaw angle, the deviation from the straight line was minimized. So the presented error was only reported in the Z-axis. Here in this work, the DoF of the tail increased, the control in Y-axis and roll angle were added and the error report is done in both X- and

Table 2
The error details of the 11 trial flights for the FWFR.

No.	Y-axis error	Z-axis error	Error
1	0.5010	−0.5469	0.7417
2	−0.3500	−0.2167	0.4117
3	−0.4215	−0.4640	0.6268
4	−0.4616	−0.1843	0.4970
5	0.6335	0.1049	0.6421
6	−0.1098	−0.4529	0.4660
7	−0.0493	0.0434	0.0657
8	−0.1080	−0.3993	0.4137
9	−0.1080	−0.0502	0.1191
10	0.3284	0.0499	0.3322
11	0.3630	0.0831	0.3724
best	0.0493	0.0434	0.0657
worst	0.6335	−0.5469	0.7417
median	−0.1080	−0.1843	0.4137
mean	0.0198	−0.1848	0.4262
Std. Dev.	0.3793	0.2468	0.2072

Y-axis. Commanding the robot bird in an aggressive maneuver and regulating both directions is more challenging than moving in a diagonal linear line. The second point of improvement in this work is repeatability. The success rate of the perching (regulation with 15 (cm) error) was reported as 66% [7]. Conducting the perching experiment was more challenging since deviation from that 15 (cm) error led to an impact on the branch though in this work the bird swiftly lands on the safety net which provided the possibility of performing more experiments without crash or damaging the bird. More videos for the flight experiment can be seen in the supplementary video file of this paper. Increasing the wingspan of the robot also resulted in a slightly lower frequency of flapping in the experiment which is also useful for the endurance of the wing hinges against fatigue. The precision of the Opti-Track system is between 1 to 2 (mm) and it provokes an explanation on the obtained mean position error of 0.4 (m). The control of the robot is done in the transient flight zone (physical limit of the test bed), where small changes in the initial condition might be considered as uncertainty to the system, battery voltage, speed of the launcher, initial position of the wings, flexibility of the tails and servomotor precision can be a part of the reason for the mean error and the uncertainty in the measurement (a relatively high standard deviation), which is acceptable and is 2% of travel distance.

Stability and repeatability: The FWFR weighs 623.9 (g) and the aerodynamic surface of the robot in comparison with the weight is relatively large. This means in outdoor flights with a little wind disturbance, the flight can be a challenge. However, favorable wind can provide good conditions for gliding and save significant energy for the battery without flapping. From a control prospect, the existence of the wind imposes disturbance to the predefined model in the nonlinear controller. The flights of this work were performed in an indoor test bed without disturbance and with steady and similar conditions for gaining repetitions. The stability and control characteristics (stability derivatives) of the flying robot must be evaluated from data obtained during the flight test program. Since several conditions and flight phases must be considered, we do not discuss this in this work. Therefore the SDRE was the controller to gain optimality in the energy usage and position error. Without the disturbance and uncertainty in the modeling, the stability of the system is checked (from the control perspective) through the stability of the controller, by defining a positive definite Lyapunov function $V(\mathbf{x}(t)) = \mathbf{x}^{\top}(t)\mathbf{P}(\mathbf{x}(t))\mathbf{x}(t)$ where $V(\mathbf{x}(t)) > \mathbf{0}$ at $\mathbf{x}(t) \neq \mathbf{0}$ and $V(\mathbf{x}(t)) = \mathbf{0}$ at $\mathbf{x}(t) = \mathbf{0}$. And then finding the negative definite condition of the derivative $\dot{V}(\mathbf{x}(t)) < \mathbf{0}$ at $\mathbf{x}(t) \neq \mathbf{0}$. Computing the derivative of the Lyapunov candidate and replacing the control

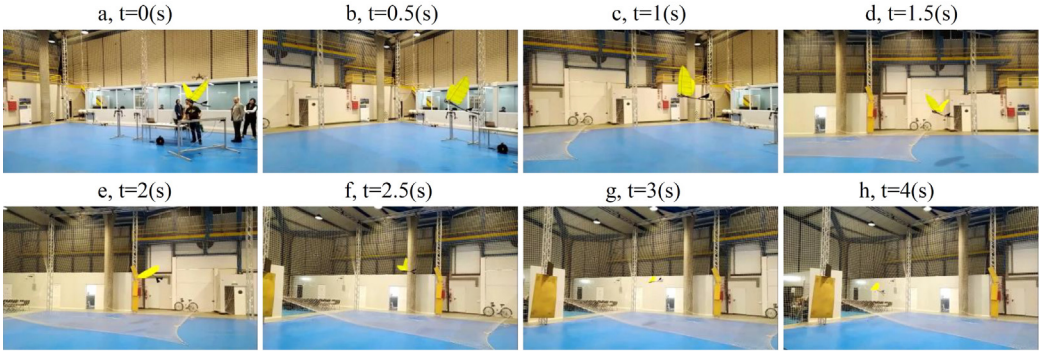

Fig. 26. The snapshot of the flight experiment; the corresponding video of the flight is available as supplementary material of the paper.

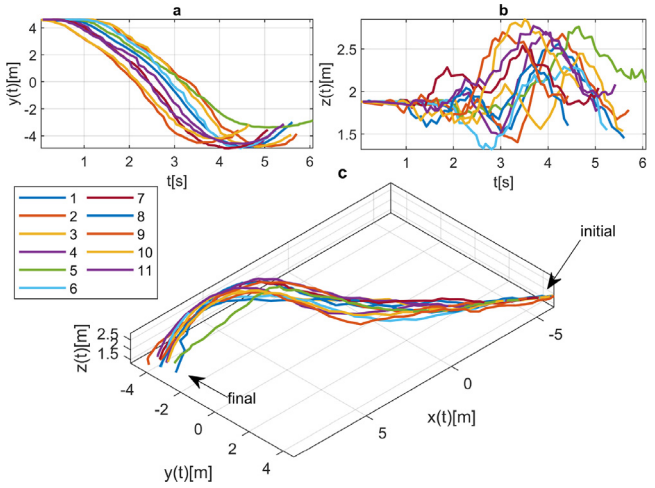

Fig. 27. The flight data record of 11 trials to show the repeatability of the control and implementation. More flights were performed and presented in the video presentation of the experiment, as supplementary material for the paper.

law and system equation result in $\dot{V}(\mathbf{x}(t)) = -\mathbf{x}^{\top}(t)[\mathbf{Q}(\mathbf{x}(t)) + \mathbf{P}(\mathbf{x}(t))\mathbf{B}(\mathbf{x}(t))\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^{\top}(\mathbf{x}(t))\mathbf{P}(\mathbf{x}(t))]\mathbf{x}(t)$ which is negative definite [59,60]. The same indoor condition provided the possibility of the repetitions of the experiments for more than 10 flights with statistical analysis, presented in Table 2.

7. Conclusions

This work presented a theoretical and experimental study on the closed-loop control of a flapping-wing flying robot using a nonlinear optimal method, a state-dependent Riccati equation for orientation dynamic. The translation control was done through PD/PI control and the idea of equivalent dynamics for replicating the oscillation of the robot caused by flapping wings. The equivalent dynamic resulted in similar behavior of the robot with a simple and basic mathematical model to avoid complex machinery of aerodynamics. The mechanical design of the FWFR has been updated to increase the number of actuators, and an additional motor in the tail increased the controllability of the system. This change provided the possibility of the Y-axis control in addition to other degrees of freedom. The current controller design can regulate the system in 5-DoF due to this new change. The attitude control was done through a nonlinear control, which is a contribution with respect to previous works [25]. An analysis of the transient flight and equivalent dynamics for presenting a model for the FWFR was presented. The simulation and experimental results were compared and met a satisfactory alignment. The repetition of flights revealed an average error of 40 (cm) which is 2% of the flight trajectory. The video demonstration of the flights can be found as supplementary material of this work in the online version of the paper. Occasional unstable response due to a fault in tuning or the mechanical parts of the robot might lead to a crash; an example can be seen in the supplementary video that shows the deviation of the FWFR to the left while the rudder tries to correct the position to the right. The reason for that failure was two broken ribs of the left wing which caused a lack of lift of the left wing.

Proposal for future works: The mechanical design of the robot could be refined to increase the mechanical resistance to possible impacts and to increase the performance of the flight. The mechanical enhancement is always limited by the weight of the components which imposes the system to be delicate. The flexibility of the wing and considering an elbow in a controlled manner also would help to enhance the lifting power and present a system closer to the real birds. From the control point-of-view, going towards six-DoF control in experimentation and real flight is a goal for future works.

Declaration of competing interest

The authors declare that there is no conflict of interest for this paper.

Acknowledgments

This work was supported by the European Project GRIFFIN ERC Advanced Grant 2017, Action 788247.

The authors would like to thank Mario Hernandez for his help and useful discussions on the printing tail idea and manufacturing of the robot; Álvaro Satué Crespo for his help in electronics, Ernesto Sanchez Laulhe in aerodynamics; Raul Tapia and Juan Pablo Rodríguez Gómez for helping with the Opti-Track; and GRVC group members.

The authors would like to thank the editors and reviewers of this paper as well.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.isatra.2023.08.001.

References

[1] Pan E, Liang X, Xu W. Development of vision stabilizing system for a large-scale flapping-wing robotic bird. IEEE Sens J 2020;20(14):8017–28.

[2] Wu X, He W, Wang Q, Meng T, He X, Fu Q. A long-endurance flapping-wing robot based on mass distribution and energy consume method. IEEE Trans Ind Electron 2022.

[3] Nekoo SR, Feliu-Talegon D, Tapia R, Satue AC, Martínez-de Dios JR, Ollero A. A 94.1 g scissors-type dual-arm cooperative manipulator for plant sampling by an ornithopter using a vision detection system. Robotica 2023;1–18.
[4] Debiasi M, Lu Z, Nguyen QV, Chan WL. Low-noise flapping wings with tensed membrane. AIAA J 2020;58(6):2388–97.
[5] Roderick WR, Cutkosky MR, Lentink D. Bird-inspired dynamic grasping and perching in arboreal environments. Science Robotics 2021;6(61):eabj7562.
[6] Stewart W, Ajanic E, Müller M, Floreano D. How to swoop and grasp like a bird with a passive claw for a high-speed grasping. IEEE/ASME Trans Mechatronics 2022;27(5):3527–35.
[7] Zufferey R, Tormo-Barbero J, Feliu-Talegón D, Nekoo SR, Acosta JÁ, Ollero A. How ornithopters can perch autonomously on a branch. Nature Commun 2022;13(1):7713.
[8] Eguíluz AG, Rodríguez-Gómez J, Paneque J, Grau P, de Dios JM, Ollero A. Towards flapping wing robot visual perception: Opportunities and challenges. In: 2019 Workshop on research, education and development of unmanned aerial systems. IEEE; 2019, p. 335–43.
[9] Lopez-Lopez R, Perez-Sanchez V, Ramon-Soria P, Martín-Alcántara A, Fernandez-Feria R, Arrue BC, et al. A linearized model for an ornithopter in gliding flight: Experiments and simulations. In: 2020 IEEE international conference on robotics and automation. IEEE; 2020, p. 7008–14.
[10] Tu Z, Fei F, Liu L, Zhou Y, Deng X. Flying with damaged wings: The effect on flight capacity and bio-inspired coping strategies of a flapping wing robot. IEEE Robot Autom Lett 2021;6(2):2114–21.
[11] Guzmán M, Páez CR, Maldonado FJ, Zufferey R, Tormo-Barbero J, Acosta JÁ, et al. Design and comparison of tails for bird-scale flapping-wing robots. In: 2021 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2021, p. 6358–65.
[12] Fan X, Breuer K, Vejdani H. Wing fold and twist greatly improves flight efficiency for bat-scale flapping wing robots. In: 2021 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2021, p. 7391–7.
[13] Sanchez-Laulhe E, Fernandez-Feria R, Ollero A. Simplified model for forward-flight transitions of a bio-inspired unmanned aerial vehicle. Aerospace 2022;9(10):617.
[14] Banerjee A. Flexible multibody dynamics: algorithms based on Kane's method. CRC Press; 2022.
[15] Liu J, Chen X, Huang H, Ji S, Tu Q. A simplified method to analyze dynamic response of VLFS based on the kane method. J Offshore Mech Arct Eng 2023;145(3):031701.
[16] Porez M, Boyer F, Belkhiri A. A hybrid dynamic model for bio-inspired soft robots—Application to a flapping-wing micro air vehicle. In: 2014 IEEE international conference on robotics and automation. ICRA, IEEE; 2014, p. 3556–63.
[17] Bena RM, Yang X, Calderon AA, Perez-Arancibia NO. High-performance six-DOF flight control of the bee++: An inclined-stroke-plane approach. IEEE Trans Robot 2023;39(2):1668–84.
[18] Zahedi A, Shafei A, Shamsi M. On the dynamics of multi-closed-chain robotic mechanisms. Int J Non-Linear Mech 2022;147:104241.
[19] Honein TE, O'Reilly OM. On the gibbs–appell equations for the dynamics of rigid bodies. J Appl Mech 2021;88:074501.
[20] Shafei A, Riahi M. The effects of mode shapes on the temporal response of flexible closed-loop linkages under the impulse excitation. Mech Syst Signal Process 2022;178:109256.
[21] Chronister N. The ornithopter design manual, fifth edition. 2008, Published by the Ornithopter Zone, www.ornithopter.org.
[22] Pan E, Xu H, Yuan H, Peng J, Xu W. HIT-hawk and HIT-phoenix: Two kinds of flapping-wing flying robotic birds with wingspans beyond 2 meters. Biomim Intell Robotics 2021;1:100002.
[23] Lee J, Ryu S, Kim HJ. Stable flight of a flapping-wing micro air vehicle under wind disturbance. IEEE Robot Autom Lett 2020;5(4):5685–92.
[24] Pan E, Liu J, Chen L, Xu W. The embedded on-board controller and ground monitoring system of a flapping-wing aerial vehicle. In: 2018 IEEE international conference on real-time computing and robotics. IEEE; 2018, p. 72–7.
[25] Zufferey R, Tormo-Barbero J, Guzmán MM, Maldonado FJ, Sanchez-Laulhe E, Grau P, et al. Design of the high-payload flapping wing robot e-flap. IEEE Robot Autom Lett 2021;6(2):3097–104.
[26] Wenfu X, Erzhen P, Juntao L, Yihong L, Han Y. Flight control of a large-scale flapping-wing flying robotic bird: System development and flight experiment. Chin J Aeronaut 2022;35(2):235–49.
[27] Roberts L, Bruck HA, Gupta SK. Autonomous loitering control for a flapping wing miniature aerial vehicle with independent wing control. In: International design engineering technical conferences and computers and information in engineering conference. 46360, American Society of Mechanical Engineers; 2014, V05AT08A013.
[28] Liang S, Song B, Xuan J, Li Y. Active disturbance rejection attitude control for the dove flapping wing micro air vehicle in intermittent flapping and gliding flight. Int J Micro Air Veh 2020;12:1756829320943085.
[29] Maldonado FJ, Acosta JÁ, Tormo-Barbero J, Grau P, Guzmán M, Ollero A. Adaptive nonlinear control for perching of a bioinspired ornithopter. In: 2020 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2020, p. 1385–90.
[30] Mracek C, Cloutier J, Cloutier J, Mracek C. Full envelope missile longitudinal autopilot design using the state-dependent riccati equation method. In: Guidance, navigation, and control conference. 1997, p. 3767.
[31] Mracek CP, Cloutier JR. Missile longitudinal autopilot design using the state-dependent riccati equation method. In: Proceedings of the international conference on nonlinear problems in aviation and aerospace. 1996, p. 387–96.
[32] Erdem EB, Alleyne AG. Experimental real-time SDRE control of an underactuated robot. In: Proceedings of the 40th IEEE conference on decision and control (Cat. No. 01CH37228), vol. 3. IEEE; 2001, p. 2986–91.
[33] Nekoo SR. Nonlinear closed loop optimal control: A modified state-dependent riccati equation. ISA Trans 2013;52(2):285–90.
[34] Naik MS, Singh SN. State-dependent riccati equation-based robust dive plane control of auv with control constraints. Ocean Eng 2007;34(11–12):1711–23.
[35] Gao D-x, Lin X-h, Wang Y, Yang Q. Application of robust optimal sliding mode control for underactuated AUV system with uncertainties. In: 2020 Chinese control and decision conference. IEEE; 2020, p. 245–50.
[36] Nekoo SR, Acosta J, Ollero A. Combination of terminal sliding mode and finite-time state-dependent riccati equation: Flapping-wing flying robot control. Proc Inst Mech Eng I 2023;237(5):870–87.
[37] Langson W, Alleyne A. A stability result with application to nonlinear regulation: Theory and experiments. In: Proceedings of the 1999 American control conference (Cat. No. 99CH36251), vol. 5. IEEE; 1999, p. 3051–6.
[38] Bogdanov A, Carlsson M, Harvey G, Hunt J, Kieburtz D, van der Merwe R, et al. State-dependent riccati equation control of a small unmanned helicopter. In: AIAA Guidance, navigation, and control conference and exhibit. 2003, p. 5672.
[39] Suzuki S, Furuta K, Sugiki A, Hatakeyama S. Nonlinear optimal internal forces control and application to swing-up and stabilization of pendulum. J Dyn Sys Meas Control 2004;126(3):568–73.
[40] Guzey UM, Copur EH, Ozcan S, Arican AC, Kocagil BM, Salamci MU. Experiment of sliding mode control with nonlinear sliding surface design for a 3-DOF helicopter model. In: 2019 XXVII International conference on information, communication and automation technologies. IEEE; 2019, p. 1–6.
[41] Korayem MH, Lademakhi NY. Integrated nonlinear suboptimal control-and-estimator based on the state-dependent differential riccati equation approach. Optim Control Appl Methods 2022.
[42] Nekoo SR. Digital implementation of a continuous-time nonlinear optimal controller: An experimental study with real-time computations. ISA Trans 2020;101:346–57.
[43] Nekoo SR, Acosta JÁ, Heredia G, Ollero A. A benchmark mechatronics platform to assess the inspection around pipes with variable pitch quadrotor for industrial sites. Mechatronics 2021;79:102641.
[44] Nekoo SR, Acosta JÁ, Heredia G, Ollero A. A PD-type state-dependent riccati equation with iterative learning augmentation for mechanical systems. IEEE/CAA J Autom Sin 2022;9(8):1499–511.
[45] Giernacki W, Stepien S, Chodnicki M, Wróblewska A. Hybrid quasi-optimal PID-SDRE quadrotor control. Energies 2022;15(12):4312.
[46] Jiang H, Zhou C, Xie P. Design and kinematic analysis of seagull inspired flapping wing robot. In: 2016 IEEE International Conference on Information and Automation. IEEE; 2016, p. 1382–6.
[47] Ruiz C, Acosta JÁ, Ollero A. Optimal elastic wing for flapping-wing robots through passive morphing. IEEE Robot Autom Lett 2022;8(2):608–15.
[48] Nakata T, Liu H, Bomphrey RJ. A CFD-informed quasi-steady model of flapping-wing aerodynamics. J. Fluid Mech 2015;783:323–43.
[49] Han J-S, Breitsamter C. Aerodynamic investigation on shifted-back vertical stroke plane of flapping wing in forward flight. Bioinspiration Biomim 2021;16(6):064001.
[50] Nekoo SR, Acosta J, Gomez-Tamm AE, Ollero A. Optimized thrust allocation of variable-pitch propellers quadrotor control: A comparative study on flip maneuver. In: 2019 Workshop on research, education and development of unmanned aerial systems. IEEE; 2019, p. 86–95.
[51] Nekoo SR, Ollero A. Equivalent vertical dynamics of flapping-wing flying robot in regulation control: Displacement transmissibility ratio. In: 2023 International conference on unmanned aircraft systems. IEEE; 2023, p. 1301–7.
[52] Nekoo SR, Ollero A. A proportional closed-loop control for equivalent vertical dynamics of flapping-wing flying robot. In: 2023 International conference on unmanned aircraft systems. IEEE; 2023, p. 1294–300.
[53] Kelly JM, Konstantinidis D. Mechanics of rubber bearings for seismic and vibration isolation. John Wiley & Sons; 2011.
[54] Sinha A. Vibration of mechanical systems. Cambridge University Press; 2010.
[55] Nekoo SR. Tutorial and review on the state-dependent riccati equation. J Appl Nonlinear Dyn 2019;8(2):109–66.

[56] Cloutier JR, D'Souza CN, Mracek CP. Nonlinear regulation and nonlinear H-infinity control via the state-dependent riccati equation technique: Part 1, theory. In: Proceedings of the International conference on nonlinear problems in aviation and aerospace. Embry Riddle University Press, Daytona Beach, Florida; 1996, p. 117–31.

[57] Pang H, Wang L. Global robust optimal sliding mode control for a class of affine nonlinear systems with uncertainties based on SDRE. In: 2009 Second International workshop on computer science and engineering. 2, IEEE; 2009, p. 276–80.

[58] Franzini G, Innocenti M. Nonlinear H-infinity control of relative motion in space via the state-dependent riccati equations. In: 2015 54th IEEE conference on decision and control. IEEE; 2015, p. 3409–14.

[59] Feydi A, Elloumi S, Jammazi C, Benhadj Braiek N. Decentralized finite-horizon suboptimal control for nonlinear interconnected dynamic systems using SDRE approach. Trans Inst Meas Control 2019;41(11): 3264–75.

[60] Nekoo SR. Output-and state-dependent riccati equation: An output feedback controller design. Aerosp Sci Technol 2022;126:107649.