ABSTRACT
Designing mechatronic systems is known to be both a very complex and tedious process. This complexity is due to the high number of system components, their multi-physical aspects, the couplings between different engineering domains and the interacting and/or conflicting design objectives. Due to this inherent complexity and the dynamic coupling between subsystems of mechatronic systems, a systematic and multi-objective design approach is needed to replace the traditionally used sequential design methods. The traditional approaches usually lead to functional but non optimal designs solutions. In this paper, and based on an integrated and concurrent design approach called “Design-for-Control” (DFC), a quadrotor UAV equipped with a stereo visual servoing system is used as a case study. After presenting the dynamics and the control model of the Quadrotor UAV and its visual servoing system, the design process has been performed in four iterations and as expected, the control performance of the system has been significantly improved after finishing the final design iteration.

Keywords: mechatronics; integrated design; concurrent; quadrotor; visual servoing.
1. INTRODUCTION

The domain of mechatronic systems deals with an interactive and synergistic application of mechanics, electronics, controls, and computer engineering in the integrated design and development of electromechanical devices. A multidisciplinary approach is ideally needed for the tasks of modelling, design, development, optimization and implementation of a mechatronic system.

Due to the large number of couplings and dynamic interdependencies occurring between elements and components, coming from different engineering domains with different physical natures; the design of mechatronic systems is considered to be a highly complex task on various levels [1–4]. Therefore, in order to achieve a better design process as well as a better final product more efficiently, these couplings need to be considered in the early stages of the design process [5–7]. The main difficulty in the process of designing mechatronic systems is that it requires a system perspective during all stages of the design process in such a way that system interactions can be considered at all times, as a comprehensive system modelling is required. This design process has been traditionally performed in a sequential manner where the design of the structure is carried out first and then the control system design is carried out. In such a sequential design process, once a mechatronic machine is developed, the mechanical structure can be hardly altered and all the mechanical parameters are therefore time-invariant.
A number of research efforts have demonstrated that compared to systems designed by a traditional sequential approach, designing the structure and control in a concurrent process, considerably improves the system performance and efficiency [8–13].

Integrated and concurrent design methodologies have been proposed over a number of works to optimally relate the mechanical and control components of mechatronic systems [14, 15]. In [14], various approaches towards design of control systems for mechatronic devices are explored to overcome the mechanical limitations. In [15], a concurrent structure-control redesign approach has been proposed to find the minimum positioning time of an underactuated robot manipulator, by considering a synergetic combination between the structural parameters and a specific control algorithm.

Toward the objective of optimal integrated design of mechatronic systems, several investigations have been done in the past decade. In [16], authors focused on the control system design for direct-drive manipulators performing high-speed trajectory control applications. They stated that the control algorithm could be simplified by using parallel drive mechanism in order to get invariant inertia and decoupled dynamics. First they introduced a concept for simplification and decoupling of system dynamics. Then, a simple procedure for control system design was presented. Although their method of design was case-specific. In [17], a method to reduce the control effort and increase the dynamic performance of an actively controlled space structure is presented. In [8] the control performance of a closed-chain machine has been improved by incorporating a PD control scheme along with a design approach of shaking force/shaking moment balancing. In all the above-mentioned design studies, the mechanical structure of the system is usually determined in advance without considering the future aspects of the controller design. Therefore a “perfect” control action may become far from practice, due to limitations imposed by the poorly designed mechanical structure.

A more general concept called Design for Control (DFC) was proposed in [18] where the design of the mechanical structure has been simplified as much as possible in such way that the dynamic modelling of the system is facilitated. Thus, a better overall control performance has been achieved. In DFC, the physical understanding of the overall system is fully explored with the aim of simplification of the controller design as well as the execution of the control algorithm with the least hardware-in-the-loop restrictions. In [12, 19, 20], three specific design methods for machine body were proposed for the DFC approach based on considering invariant potential energy, invariant generalized inertia and partially invariant generalized inertia in order to perform a re-design and simplify the system dynamics in just one iteration. In these studies, although the design integration takes place in a single step, less effort was focused on design of the control system. Furthermore, the optimality of the results for structure-control design is in doubt.

In this paper, the DFC is used to design a complex mechatronic system composed of a vision-guided UAV quadrotor. In terms of system dynamics, a quadrotor is an underactuated system with six degrees of freedom and four inputs which is inherently unstable and difficult to control. Thus, the design and control of this nonlinear system is a challenge from both practical and theoretical point of views [21–24]. Integrating the sensors, actuators and intelligence into a lightweight vertically flying system with a decent operation time is not a trivial task to achieve. Designing an autonomous quadrotor is a complex task since it requires dealing with numerous design parameters that are originated from various engineering disciplines and subsystems and more importantly they are closely interdependent. Taking a decision about all these parameters requires a clear integrated methodology. Moreover, in order to enable the system with autonomous capabilities, a visual feedback control strategy will be used which increases the parameters that need to be optimized hence increasing the overall complexity of the design task. The remaining of this paper is organized as follows; Section 2 recalls the dynamic model and formulations of a small quadrotor system. In Section 3, the control system design is presented. A formulation for the image-based stereo visual servoing system is also presented in this section. In Section 4, the DFC-based integrated design strategy is introduced while
in Section 5 this method is utilized to optimize the integrated design of the quadrotor system. This section also includes validations with computer simulations. Finally, the concluding remarks are discussed in Section 6.

2. SYSTEM MODELLING AND FORMULATION

The design of quadrotor systems involves various engineering domains and their affecting factors e.g., aerodynamics, mechanics, control and intelligence should be included in the design and optimization process. The model of the quadrotor should consider all the important effects such as aerodynamic, inertial counter torques, friction, gyroscopic and gravitational effects. Therefore, in this paper, Euler–Lagrange formulation and DC motor equations were used to model the Quadrotor system. The dynamic model developed in this section is derived based on the following simplifying assumptions:

- The structure of the system is supposed to be rigid and symmetric.
- The thrust and drag affecting the system are proportional to the square of propellers speed [25].
- The origin of the body fixed frame and the centre of gravity (COG) are located at the same position.

Figure 1 illustrates the coordinate system for the quadrotor model in which \( W \) is the fixed world coordinate frame and \( B \) is the body fixed frame. The space orientation is also given by a rotation matrix \( \mathbf{R} \) from frame \( B \) to \( W \), where \( \mathbf{R} \in SO^3 \).

For any point expressed in the fixed world coordinate frame, we can write (with \( C: \cos, S: \sin \))

\[
\begin{align*}
    r_x &= (C\psi C\theta)x + (C\psi S\theta - S\psi C\phi)y + (C\psi S\phi C\theta + S\psi S\phi)z \\
    r_y &= (S\phi C\theta)x + (S\psi S\theta S\phi + C\psi C\phi)y + (S\psi S\theta C\phi - C\psi S\phi)z \\
    r_z &= (-S\theta)x + (C\theta S\phi)y + (C\theta C\phi)z
\end{align*}
\]

The corresponding velocities are obtained by differentiation of Eq. (1). The squared magnitude of the velocity for any point can be given by

\[
    v^2 = v_x^2 + v_y^2 + v_z^2
\]

From Eq. (2), and by assuming the inertia matrix to be diagonal, the kinetic energy expression can be written as follows:

\[
    T = \frac{1}{2}I_{xx}(\dot{\phi} - \psi S\phi)^2 + \frac{1}{2}I_{yy}(\dot{\theta} C\phi + \psi S\phi C\theta)^2 + \frac{1}{2}I_{zz}(\dot{\theta} S\phi - \psi C\phi)^2
\]
and for the potential energy $U$ and with regards to the fixed frame, we can write

$$U = \int x \, dm(x)(-gS_\theta) + \int y \, dm(y)(gS_\phi C_\theta) + \int z \, dm(z)(gC_\phi C_\theta) \quad (4)$$

Using the Lagrangian function and the derived formula for the equations of motion we have

$$L = T - U, \quad Q_i = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (5)$$

where $q_i$ are the generalized coordinates and $Q_i$ are the generalized forces. Moreover, the non-conservative torques acting on the system result, firstly from the action of the thrust difference of each pair. Thus

$$\tau_x = b_t L (\Omega_1^2 - \Omega_2^2)$$
$$\tau_y = b_t l (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2)$$
$$\tau_z = d (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \quad (6)$$

where the $\Omega_i$ are angular speed of $i$th propeller. From the gyroscopic effects resulting from the propellers rotation we have the following torques:

$$\tau'_x = J_r w_y (\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)$$
$$\tau'_y = J_r w_x (\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) \quad (7)$$

where $w_x, w_y$ are the vectors of body rotational speeds which are approximated by the derivatives of Euler angles. Consequently, The quadrotor dynamic model describing respectively the roll, pitch and yaw rotations contains of three terms which are the gyroscopic effect resulting from the thrust difference of each pair. The mechanical symmetry of the quadrotor allows one to neglect the inertia products and consider the use of a diagonal inertia matrix. This can be verified using a CAD model of the proposed design concept where the inertia products is almost thousand times lower than the inertia moments. Applying the small angles approximation, we obtain

$$\begin{align*}
\dot{\phi} &= \frac{J_r \theta (\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)}{I_{xx}} + \frac{J_r (I_{xy} - I_{x})}{I_{xx}} \dot{\psi} + \frac{b_t l (\Omega_2^2 - \Omega_1^2)}{I_{xx}} \\
\dot{\theta} &= -\frac{J_r \theta (\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)}{I_{yy}} + \frac{J_r (I_{xy} - I_{y})}{I_{yy}} \dot{\phi} + \frac{b_t l (\Omega_3^2 - \Omega_1^2)}{I_{yy}} \\
\dot{\psi} &= -\frac{d (\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2)}{I_{zz}} + \frac{J_r (I_{xz} - I_{z})}{I_{zz}} \dot{\phi} \dot{\theta} \quad (8)
\end{align*}$$

Using a Newton dynamics formulation we can also achieve

$$\begin{align*}
\ddot{x} &= \frac{U_1}{m} (S_\psi S_\phi + C_\psi S_\theta C_\phi) \\
\ddot{y} &= \frac{U_1}{m} (-C_\psi S_\phi + S_\psi S_\theta C_\phi) \\
\ddot{z} &= -g + \frac{U_1}{m} (C_\phi C_\theta) \quad (9)
\end{align*}$$
where $U_i$ are the system inputs as

$$
U_1 = b_1 \sum_{j=1}^{4} \Omega_i^2 \\
U_2 = b_1 (\Omega_4^2 - \Omega_2^2) \\
U_3 = b_1 (\Omega_3^2 - \Omega_1^2) \\
U_4 = d (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \\
\Omega = \Omega_1 + \Omega_3 - \Omega_2 - \Omega_4
$$

(10)

The rotors are considered to be driven by DC-motors with the following well established equations:

$$
\begin{align*}
L_m \frac{di}{dt} &= u - Ri - k_e w_m \\
J_m \frac{dw_m}{dt} &= \tau_m - \tau_d
\end{align*}
$$

(11)

where $u$ is the input voltage. Using a small motor with a very low inductance, the second order DC-motor dynamics may be approximated by the following equation

$$
J_m \frac{dw_m}{dt} = -\frac{k^2}{R} w_m - \tau_d + \frac{k_m}{R} u
$$

(12)

Now, by considering the propeller and the gearbox models, the above equation becomes

$$
\dot{w}_m = -\frac{1}{\eta} w_m - \frac{d}{\gamma R^3 J_t} w_m^2 + \frac{1}{k_m \eta} u
$$

(13)

where $\eta = R J_t / k^2_m$ is the motor time-constant. Now, by linearizing the above equation around an operation point $\dot{w}_0$ we achieve

$$
\dot{w}_m = -A w_m + B u + C
$$

(14)

where

$$
A = \left( \frac{1}{\eta} + \frac{2d w_0}{\gamma R^3 J_t} \right), \quad B = \left( \frac{1}{k_m \eta} \right), \quad C = \left( \frac{d w_0^2}{\gamma R^3 J_t} \right)
$$

(15)
3. CONTROLLER DESIGN

The control system of the proposed quadrotor UAV, consists of two components of motion control system and visual servoing (vision-based control) system. The cooperative configuration of these control systems is illustrated in a single control structure presented in Fig. 2.

3.1. Motion Control

In this paper a PID controller is proposed for position control of the quadrotor. The dynamic model of the system, contains two gyroscopic effects. The influence of these effects in the present case and by considering a near-hover situation is less important than the motor’s model. In order to design a PID controller for this system, one can neglect these gyroscopic effects and thus remove the cross couplings between body and propellers. The following equations have been derived by simplification of the system dynamics formulation:

\[
\ddot{\phi} = \frac{lU_2}{I_x}, \quad \ddot{\theta} = \frac{lU_3}{I_y}, \quad \dot{\psi} = \frac{U_4}{I_z}
\]

(16)

The transfer functions of quadrotor attitude plant (i.e., roll, pitch and yaw) can be obtained separately as follows:

\[
\frac{\phi(s)}{U_2(s)} = \frac{l}{I_x s^2}, \quad \frac{\theta(s)}{U_3(s)} = \frac{l}{I_y s^2}, \quad \frac{\psi(s)}{U_4(s)} = \frac{1}{I_z s^2}
\]

(17)

The error signals can be also introduced based on the difference between the current states and desired angles as

\[
e_{\phi} = \phi_d - \phi, \quad e_{\theta} = \theta_d - \theta, \quad e_{\psi} = \psi_d - \psi
\]

(18)

The system can also be described by using the motor inputs in Laplace domain, as

\[
\phi(s) = \frac{B^2 lb}{s^2(s+A)^2 I_x} (\Omega_2^2(s) - \Omega_1^2(s))
\]

(19)

\[
\theta(s) = \frac{B^2 lb}{s^2(s+A)^2 I_y} (\Omega_3^2(s) - \Omega_1^2(s))
\]

(20)

\[
\psi(s) = \frac{B^2 l d}{s^2(s+A)^2 I_z} (\Omega_2^2(s) + \Omega_3^2(s) - \Omega_1^2(s) - \Omega_2^2(s))
\]

(21)

where \(A\) and \(B\) are the coefficients of the linearized rotor dynamics from equation (14). The output of a PID controller, which is also the input to the system control plant (e.g., \(\phi(s)\)), in the time domain is as follows:

\[
u_i(t) = k_p e_i(t) + k_i \int e_i(t) dt + k_d \frac{d e_i(t)}{dt}
\]

(22)

The transfer function of a PID controller is found by taking the Laplace transform of the last equation

\[
\text{PID} = G_{\phi}(s) = \left( k_p + \frac{k_i}{s} + k_d s \right) = \frac{k_ds^2 + k_ps + k_i}{s}
\]

(23)

3.2. Visual Servoing Control

In general it can be stated that in an image-based visual servoing system, the goal of vision-based control scheme is to minimize the error defined as

\[
e(t) = s - s^*
\]

(24)

where \(s\) and \(s^*\) are the vectors of current and desired image features. These image features may include detected edges on the target object, colored points, image moments and etc. In our case, three colored
figures (red circle, blue triangle, green square) on the target object are considered as image features which can be extracted using a basic shape-color detection image processing algorithm.

In the case of a traditional proportional controller, the input to the robot controller $u_c$ is designed by letting

$$\dot{e} = -\lambda e$$

$$u_c = -\lambda J_e^+ e$$

(25)

where $J_e$ is the image interaction matrix which relates the time variation of error $e$ and the camera velocity and $J_e^+$ is the Moore–Penrose pseudo-inverse of the interaction matrix. $\lambda$ is the proportional gain for the visual controller. In the case of moving image features we have

$$u_c = J_e^+ \left( -\lambda e - \frac{\partial e}{\partial t} \right)$$

(26)

where the term $\left( \frac{\partial e}{\partial t} \right)$ represents the time variation of $e$ caused by the target motion which is considered to have a constant velocity. In our case we assume that the vision system is composed of a stereo vision system with two parallel cameras which are perpendicular to the baseline [26, 27]. The focal points of two cameras are apart at distance $b/2$ with respect to origin of sensor frame $C$ on the baseline which means the origin of the camera frame, is in the centre of these points. Focal distance of both cameras is $f$ so the image planes and corresponding frames for left and right cameras are located with the distance $f$ from the focal points and orthogonal to the optical axis. We assign $L$ and $R$ as the frames of the left and right images. Figure 3 illustrates the case where both cameras observe a 3D point $^P C P$. Using the image interaction matrices for the left and right cameras the stereo image interaction matrix, $J_{st}$, can be calculated as

$$J_{st} = \begin{pmatrix} J_l & ^1M_c \\ J_r & ^rM_c \end{pmatrix}$$

(27)

The image interaction matrix for each camera is calculated as

$$J_l = \begin{pmatrix} \frac{1}{Z} & 0 & \frac{x}{Z} & x_i y_i & -(1 + x_i^2) & y_i \\ 0 & \frac{1}{Z} & \frac{y}{Z} & 1 + X_i^2 & -x_i y_i & -x_i \end{pmatrix}$$

(28)

The stereo feature vector is defined as $s = [x_l, y_l, x_r, y_r]^T$ where $p_l = [x_l, y_l]^T, p_r = [x_r, y_r]^T$ are the normalized image coordinates of the 3D point, observed by the left and right cameras respectively. A perspective camera model can be used to project observed point into left and right image planes. Thus, the following equations hold for 3D coordinates of the observed point:

$$(X, Y, Z) = \left( \frac{b}{2} \frac{x_l + x_r}{x_l - x_r}, \frac{b}{x_l - x_r}, \frac{x_l}{x_l - x_r} \right)$$

(29)

4. INTEGRATED DESIGN STRATEGY

An engineering design process can be parametrically defined as a mapping from a requirement space consisting of behaviours to a structural parameter space. To gain insight into the design of a mechatronic system, Li et al. [18] suggested dividing the requirement space into two subspaces which represent (this formalism is adopted in this paper):

1. Real-time behaviours (RTBs) and
2. Non-realtime behaviours (non-RTBs)
Following this division of the requirements, system parameters in structural space can also be divided into two subspaces as follows:

1. Real-time (or controllable) parameters (RTPs) and
2. Non-real-time (or uncontrollable) parameters (non-RTPs)

From above, “real-time” means parameters, specifications, constraints and behaviours that may change with time after the machine is built. Controller gains, accuracy and speed are some examples of RTPs and RTBs. On the other hand, non-real-time parameters, constraints and specifications are the ones that can be hardly changed after the machine is built, because it would be costly to change them. Structural material, dimensions, weight, and workspace can be considered as non-RTPs and non-RTBs. Traditional methodologies for mechatronic systems design consisted of sequences of the real-time and non-real-time requirements rather than a concurrent design process. At the beginning of such a traditional design scenario, non-RTPs are designed based on the non-RTB specifications. This process itself includes designing the mechanical structure and then adding electrical components. The mechanical structure (e.g., configurations, dimensions, layout of actuators and sensors, etc.) is first determined based on the requirements in the non-RTB space (e.g., workspaces, maximum payloads, etc.). Subsequently, RTPs (e.g., controller algorithm and parameters, signal conditioning) are determined based on RTB specifications (e.g., desired trajectory, speed, stability, etc.) to control the already established structure. Due to recent advancements in control and computer engineering one may conclude that the design of the imperfections and inadequacies in structure and hardware of a mechatronic system can be compensated by some state-of-the-art control schemes. This thinking can be easily criticized because a perfect control response may be hardly achieved due to hardware limitations and dynamic interactions, regardless of the effort devoted to the design of the control system. Although it can be observed in several cases that the performance of a mechatronic system can be improved by using better control strategies, but reaching design process “optimality” is in serious doubt. In a concurrent model for mechatronic systems design (Fig. 4), both RTBs and non-RTBs should be considered simultaneously for realization of RTPs and non-RTPs. In a mechatronic system, the system performance, which is the RTBs and non-RTBs, explicitly relies on the design of its control algorithm and parameters (RTPs) and also the design of the mechanical structure (non-RTPs).
Let \( X_R \) and \( X_N \) be RTP and non-RTP design vectors. We also assume there exists \( n \) RTPs and \( m \) non-RTPs, that is \( X_R \subset \mathbb{R}^n \), and \( X_N \subset \mathbb{R}^m \), where the total number of design parameters is \( q = m + n \). Respectively, the determination of design parameters is subject to a set of constraints produced by the behaviour requirements. Thus, let \( Y_R \) and \( Y_N \) denote \( u \)-RTB and \( v \)-non-RTB requirements which sums to \( p = u + v \) as the total number of variables in the requirement space. Thus, \( Y_R \subset \mathbb{R}^u \), and \( Y_N \subset \mathbb{R}^v \). Assuming \( Y = [Y_R, Y_N] \), the performance error can be defined as \( E = Y - Y_d \) where \( Y_d \) is the vector of desired behaviours. Accordingly, let \( S_{\text{min}} \) and \( S_{\text{max}} \) denote the design requirements associated with a particular design problem, where “min” and “max” indicate the performance indices of the requirements to be minimized and maximized, respectively. Finally, let \( P \) denote the system actuation power. A mechatronic system design problem can be described using the following mathematical models for objectives and constraints [18]

\[
E = \min \sum_{i=1}^{p} \alpha_i |E_R|_i + \xi_i |E_N|_i 
\]

(30)

\[
P = \min \sum_{i=1}^{\text{dof}} \beta_i p_i 
\]

(31)

\[
S_{\text{min}} = \min \sum_{i=1}^{q_1} \lambda_i |S_{\text{min}}|_i 
\]

(32)

\[
S_{\text{max}} = \max \sum_{i=1}^{q_2} \rho_i |S_{\text{max}}|_i 
\]

(33)

\[
I = E + P + S_{\text{min}} - S_{\text{max}} 
\]

(34)

where \( \alpha_i, \xi_i, \beta_i, \lambda_i, \rho_i \) are weighting factors determined by the designer, \( p_i \) is the power generated by each actuator in the system and \( q_1 \) and \( q_2 \) are the number of the design parameters associated with the minimized and maximized requirements. To optimize the overall design performance, a performance index \( I \) is introduced to integrate all introduced individual objectives in one equation. The equality and inequality constraints can be respectively expressed by

\[
Y_R^E = f_R^E (X_R, X_N) 
\]

\[
Y_N^E = f_N^E (X_N) 
\]

(35)

\[
Y_{R,\text{low}} < f_R^I (X_R, X_N) < Y_{R,\text{up}} 
\]

\[
Y_{N,\text{low}} < f_N^I (X_N) < Y_{N,\text{up}} 
\]

(36)
where the superscript $I$ indicates the inequality constraints and the superscript $E$ indicates the equality constraints. From the above design constraints it can be observed that, for a mechatronic system, the system dynamic performance (RTBs or $Y_R$) depends on both the control parameters (RTPs or $X_R$) and the mechanical structure behaviours (non-RTPs or $Y_N$). As stated before, the essence of DFC method is to design the mechanical structure ($X_N$) in an effort to achieve a simple dynamic model for the ease of designing the control system ($X_R$), to ideally achieve an optimal system dynamic performance ($Y_R$). In a simulation-based iterative integrated design strategy, $X_N$ is first set for a mechanical structure based on the desired behaviours and requirements (associated with $Y_N$ directly yet $Y_R$ indirectly). This step can be expressed as

$$ Y_1 = f_1(X_N) \quad (37) $$

Then having $X_R$ determined, the dynamic performance $Y_2$ is obtained (based on $Y_R$ explicitly and $Y_N$ implicitly). This step can be expressed as

$$ Y_2 = f_2(Y_2, X_R) \quad (38) $$

Next, $Y_N$ will be configured by comparing the desired behaviours with the measured ones. If the result is not satisfactory, then $X_R$ is modified to improve the control performance. Thus, we have

$$ Y_3 = f_3(Y_2, X_N). \quad (39) $$

If the control performance $Y_3$ does not satisfy the requirements, $X_R$ is varied again to attain an improved performance. This step can be formulated as

$$ Y_4 = f_4(Y_3, X_R) \quad (40) $$

For an algorithmic implementation, the iterations can be formulated as

$$ \begin{align*}
Y_{2i-1} &= f_{2i-1}(Y_{2i-2}, X_N) \\
Y_{2i} &= f_{2i}(Y_{2i-1}, X_R)
\end{align*} \quad (i = 1, 2, \ldots, k) \quad (41) $$

The design procedure iterates until a final design on $X_R$ is found that enables the system to achieve a satisfactory performance. When an analytical system dynamic model is obtainable, the iterative design process described before can be carried out via a simulation process. $X_N$ can be further changed towards various directions along the solution-search path. It is quite possible to find a solution to the optimal design problem with the fewest constraints. Having the dynamics model, $X_N$ can be varied until a simpler dynamic model can be achieved which results in facilitating the procedure of the control system design.

5. DFC-BASED DESIGN OPTIMIZATION

Using the Design-for-Control (DFC) approach, an integrated design of a vision-guided quadrotor UAV is detailed in this section. Before starting the design process, we need to define parameters and behaviours. The first column of Table 1 classifies all the RTPs and non-RTPs, as the design parameters in the process of designing a vision-guided quadrotor drone with a PID attitude control system. After identifying all the parameters and behaviours, the integrated design approach can be divided in 4 iterations as follows.

5.1. Iteration 1: Design $X_N$ Based on non-RTBs, $Y_N$

The first step is to determine $X_N$, the mechanical structure parameters, so that the specified non-RTBs, $Y_N$, are satisfied. As the first requirement and based on a series of commercial benchmarks, the quadrotor is subjected to the following physical constraints

$$ 0.2 \leq l \leq 0.4 \text{ (m)} \quad (42) $$
Table 1. DFC-based design results for a vision-guided quadrotor system.

<table>
<thead>
<tr>
<th>Non-RTPs, $X_N$ Descriptions</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
<th>Iteration 3</th>
<th>Iteration 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l$ Arm length (m)</td>
<td>0.25 / 0.28</td>
<td>/</td>
<td>0.28 / 0.28</td>
<td>/</td>
</tr>
<tr>
<td>$m$ Total mass (kg)</td>
<td>0.65 / 0.72</td>
<td>/</td>
<td>0.72 / 0.72</td>
<td>/</td>
</tr>
<tr>
<td>$I_{xx}$ Inertia moments on x (kg·m²)</td>
<td>0.009 / 0.0076</td>
<td>/</td>
<td>0.0076 / 0.0076</td>
<td>/</td>
</tr>
<tr>
<td>$I_{yy}$ Inertia moments on y (kg·m²)</td>
<td>0.008 / 0.0076</td>
<td>/</td>
<td>0.0076 / 0.0076</td>
<td>/</td>
</tr>
<tr>
<td>$I_{zz}$ Inertia moments on z (kg·m²)</td>
<td>0.017 / 0.0152</td>
<td>/</td>
<td>0.0152 / 0.0152</td>
<td>/</td>
</tr>
<tr>
<td>$b$ Distance between cameras (m)</td>
<td>0.15 / 0.1</td>
<td>/</td>
<td>0.1 / 0.1</td>
<td>/</td>
</tr>
</tbody>
</table>

RTPs, $X_R$

| $k_p$ Proportional control gain | / | 1.5 | / | 1.3 |
| $k_i$ Integral control gain    | / | 1.0 | / | 0.8 |
| $k_d$ Derivative control gain  | / | 0.6 | / | 0.4 |
| $\lambda$ Proportional gain in visual servoing | / | 0.5 | / | 0.35 |

\[
m_r \geq 0.4 \text{ (kg)}
\]

\[
0.006 \leq I_{xx}, I_{yy} \leq 0.01 \text{ (kg·m}^2\text{)}
\]

\[
0.01 \leq I_{zz} \leq 0.03 \text{ (kg·m}^2\text{)}
\]

where the inertia moments can be calculated from a simple physical model of the quadrotor where it consists of two rods as the arms, one disk at centre and four concentrated mass at the end of each arm. One of the major physical limitations of a quadrotor is the propeller’s rotational speed which is constrained by the motor saturation speed. This saturation speed of the propellers should be approximately 41% higher than the hovering speed [28]. The propeller’s rotational speed in hovering condition can be found by solving Eqs (8–10) for equilibrium point

\[
\Omega_H = \left( \frac{mg}{4b_t} \right)^{1/2}
\]

Thus, having the condition of $\Omega_i \leq 350$ (rad/s) based on some frequently used brushless motors and also the propellers’ thrust factor of $b_t = 3.15E - 5$, we can achieve an allowable total mass and payload capacity:

\[
m \leq \frac{4b_t \Omega_i^{max}}{g(1.41)^2} = 0.791 \text{ (kg)}
\]

Having the aforementioned non-RTB constraints the first set of non-RTPs can be calculated as the starting point of the optimization problem. The design result of $X_N$ is given in the first column of Table 1.

5.2. Iteration 2: Design $X_R$ Based on RTBs, $Y_R$

Once the initial design of the mechanical structure is completed, the motion controller and visual servoing system must be designed carefully such that the required dynamic and visual performances are satisfied. Thus, the design objective is to minimize the performance index over the entire range of motion

\[
I_R = E_R^Q + E_R^C + P
\]

\[
E_R^Q = \min \left( \alpha_1 \int_0^{t_f} \left( (X(t) - X_d)^2 + (Y(t) - Y_d)^2 + (Z(t) - Z_d)^2 \right) dt \\
+ \alpha_2 \int_0^{t_f} \left( (\dot{X}(t) - \dot{X}_d)^2 + (\dot{Y}(t) - \dot{Y}_d)^2 + (\dot{Z}(t) - \dot{Z}_d)^2 \right) dt \right)
\]

\[ E_R^C = \min(\alpha_3 \int_0^{t_f} (s(t) - s^*)^2 dt) \]  
(50)

\[ P = \min(\beta \int_0^{t_f} |T(t)| dt) \]  
(51)

where \( E_R^Q \) is the minimum performance error for position and velocity tracking and \( E_R^C \) is the minimum performance error for the visual servoing system. \( P \) signifies the driving torque generated by the motion control, and \( \alpha, \beta \) are the weighting factors to be determined. Accordingly, and again based on frequently used commercial benchmark parameter, the following RTB constraints (control inputs) are imposed on the controller design:

\[
0 \leq \sum T_i \leq 2 mg \quad (52)
\]

\[
|\phi| \leq 0.6 \text{ rad} \quad (53)
\]

\[
|\theta| \leq 0.6 \text{ rad} \quad (54)
\]

\[
0 \leq \psi \leq 0.01 \text{ rad} \quad (55)
\]

For translational speed and descend rate we also have

\[
|\dot{z}| \leq 5 \text{ ms}^{-1} \quad (56)
\]

\[
|\dot{x}| \leq 10 \text{ ms}^{-1} \quad (57)
\]

\[
|\dot{y}| \leq 10 \text{ ms}^{-1} \quad (58)
\]

The target object which is being tracked by the vision system is moving along a circle path on \( x-y \) plane with the radius of 4 meters and the quadrotor is required to follow the target with the height of 2 meters with respect to target. The target object is travelling with the speed of 10 m/s along the circular path and the quadrotor is not allowed to have a translational speed more than the object. In order to simplify the problem no minimum time-trajectory is given. The control design problem is solved using MATLAB optimization toolbox. To ensure each performance characteristic (i.e., \( E_R^Q, E_R^C \) and \( P \)) contributes properly to the performance index in an equivalent magnitude, the weighting factors are selected to be \( \alpha_1 = 1.0, \alpha_2 = 0.1, \alpha_3 = 0.5, \beta = 0.005 \). The design result of \( X_R \) is given in the second column of Table 1. The simulation model built in SIMULINK to reflect the design process results is shown in Fig. 5.

Using the control gains as a result from the aforementioned optimization solution, the tracking performances for both motion and vision-based control are comparatively displayed in Figs. 6 and 7 and as it can be observed some undesired performance appears in the position tracking and the visual features error are also not quite satisfactory. Hence an extra iteration is needed.

5.3. Iteration 3: Redesign \( X_N \) to Improve non-RTBs, \( Y_N \)

In the third iteration the non-RTPs, \( X_N \), will be modified with the aim of simplifying the system dynamic model so that the controller design on \( X_R \) can be facilitated. The longitudinal dynamics of a quadrotor system can be considered as the dominant dynamics of the vehicle. Around hovering position, the motion is largely decoupled in each axis. If the geometry of the system can be considered as symmetric, the important attitude dynamics can be described by a single equation. The natural stability of these dynamics is important to be analysed to provide insight into the best airframe geometry for controllability of the system [29].

From the basic dynamic equations for a quadcopter with translational and rotational motion in only \( x \) and \( \theta \), and equal speeds on all rotors, the stability derivative equation is [29]

\[
\begin{bmatrix}
-ms + \frac{\partial s}{\partial x} - \frac{\partial s}{\partial \theta} s - mg \\
-\frac{\partial \theta}{\partial x} - I_{yy} s^2 + \frac{\partial \theta}{\partial \theta} s
\end{bmatrix}
\begin{bmatrix}
\dot{x} \\
\dot{\theta}
\end{bmatrix} = 0
\]  
(59)
where $s$ is the Laplace transform of the differential operator. These stem from the stability constraints in the hovering position. Applying the Routh–Hurwitz stability criterion, the first column of the stability parameters table is needed to be all strictly positive. Thus

$$
\left( \frac{1}{m} \frac{\partial x}{\partial x} + \frac{1}{I_{yy}} \frac{\partial \theta}{\partial \theta} \right) > 0 \quad (60)
$$

$$
\frac{g}{I_{yy}} \frac{\partial \theta}{\partial x} > 0 \quad (61)
$$

Furthermore, the dynamic model can be finally simplified as

$$
\begin{align*}
I_{xx} \dot{\phi} &= \theta \psi (I_{yy} - I_{zz}) \\
I_{yy} \dot{\theta} &= \phi \psi (I_{zz} - I_{xx}) \\
I_{zz} \dot{\psi} &= \phi \theta (I_{xx} - I_{yy})
\end{align*} \quad (62)
$$

The redesigned values of $X_N$ are now given in the third column of Table 1.

5.4. Iteration 4: Redesign $X_R$ Based on the Modified Non-RTBs, $Y_N$

After redesigning the non-RTPs, $X_N$, the visual servoing and motion control algorithms are again applied for the path and trajectory tracking of the target object. In this iteration, the design objective, constraints,
Fig. 6. The step-response of the attitude and altitude control systems based on the final system-level optimization results.
Fig. 7. Position tracking performances based on results from (a) Iteration 2 and (b) Iteration 4, and a comparative graph of paths for a complete motion.

and variables are the same as those in Iteration 2. The design result of control gains, $X_R$, is given in the fourth column of Table 1, which is the same as the control gains used in Iteration 3. The step response graphs based on the provided optimization solutions after the second and fourth iterations for altitude and attitude control systems are shown in Fig. 6. A comparative table is also presented (Table 2) describing the performances of the altitude and attitude control systems. The new visual tracking performances are also displayed in Figs. 7 and 8. Compared with the results of Iteration 2, it can be observed that the position tracking performance has been enhanced and the performance with regards to visual features errors has also shown better convergence characteristics. From the simulation measurements for Iterations 2 and 4, the convergence time was reduced from 3.7 sec to 1.35 sec, while the maximum tracking error decreased from 346 to 288 pixels. Less oscillations were also reported for the tracking performance of Iteration 4.

It can be observed that after only four iterations, the obtained design variables are quite satisfactory and elevate the performance of the proposed system. Which in our case, and considering the assumptions made, shows that the DFC does help in terms of integrated design of a complex mechatronic system. However, the complexity of our systems lies within its behaviour and not the number of components, as in our opinion the results obtained here will only hold for systems with small number of components and consequently design
Fig. 8. Visual feature errors from (a) Iteration 2 and (b) Iteration 4.

Table 2. The step-response characteristics of the attitude and altitude control systems based on the results from Iteration 2 and Iteration 4.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Max. Overshoot</th>
<th>Rise-time (sec)</th>
<th>Settling-time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration #2</td>
<td>$\phi (s), \theta (s)$</td>
<td>17%</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>$\psi (s)$</td>
<td>18%</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>$Z (s)$</td>
<td>18%</td>
<td>0.1</td>
</tr>
<tr>
<td>Iteration #4</td>
<td>$\phi (s), \theta (s)$</td>
<td>7%</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>$\psi (s)$</td>
<td>7%</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>$Z (s)$</td>
<td>18%</td>
<td>0.1</td>
</tr>
</tbody>
</table>

variables and parameters. Using the DFC method for more complex mechanisms in terms of behaviour and number of component as well as complex control systems would be very hard to implement in the way the DFC methodology is built. The difficulty stems from the fact that it requires for the designer to set a too large of a number of constants and this will definitely cause the design process to need much more iterations, not taking into accounts the new constraints which will be introduced to the optimization process. This will call for some additional efforts to establish guidelines for choosing those constants and more importantly, a faster and even more “integrated” approach, as future efforts.

6. CONCLUSIONS

In this paper, the problem of integrated and concurrent design of a vision-guided quadrotor UAV has been studied using the Design-for-Control methodology that has never been applied to a system with complex dynamics and numerous subsystems. We used the DFC considering the design of a mechatronic system as a mapping from a requirement space to a structure space. The mechatronic design concept is therefore interpreted as an integrated design framework that considers both real-time and non-real-time requirements simultaneously and configures both real-time and non-real-time parameters (design variables) concurrently.

Having discussed the design approach, the concurrent design of both mechanical and control structures of a vision-guided quadrotor system has been accomplished in an iterative manner and after finalizing the last iterations, desired performances with regards to both control systems, i.e., motion control and visual
servoing, have been achieved. However, for systems with larger number of components and more complex control systems, additional efforts to establish guidelines for choosing the design optimization constants, hence a more “integrated” approach is still required. This approach would need to lighten the burden of the designer in terms of choosing a too large number of parameters and weights, as these decisions can vary from one designer to another. Furthermore, a need to consider the complexity of the process itself (e.g., number of design loop-backs) needs to be considered in the optimisation process. Although, since the design process described is still in a preliminary stage, the results have not been applied to a real-world experimental setup, but the simulation process is quite accurate and closely mimics the required behaviours of a quadrotor system. These simulation results are right now further used in a more detailed design process in our research group.

REFERENCES