

Matko Orsag, Marina Poropat, Stjepan Bogdan

Hybrid Fly-by-Wire Quadrotor Controller

UDK 629.735.4-52
681.516
IFAC 5.7.5

Original scientific paper

This article presents one solution to a quadrotor control problem that is based on a discrete automaton. This automaton combines classical PID and more sophisticated LQ controllers to create a hybrid control system. This closed loop control concept is expanded with an open loop controller that enables the aircraft to perform aggressive flying maneuvers. The combination of open and closed loop controllers builds a hybrid controller concept that allows directed and autonomous flying of the quadrotor aerial vehicle. Proposed control concept was tested on an elaborate mathematical model. The article discusses these test results and presents the means to develop such a controller.

Key words: Hybrid control, Rotorcraft, Unmanned aerial vehicle

Hibridni sustav daljinskog upravljanja letjelicom. U radu je opisan hibridni upravljački koncept bespilotne letjelice pogonjene s četiri rotora, koji objedinjuje klasične PID regulatore i naprednije LQ regulatore primjenom Mooreova automata. Takav je hibridni koncept upravljanja u zatvorenoj petlji nadograđen upravljanjem u otvorenoj petlji koje omogućuje ostvarenje agresivnih letačkih manevara. Osim toga, kombinacija upravljanja u otvorenoj i zatvorenoj petlji omogućuje i daljinsko upravljanje letjelicom zasnovano na vizualnoj povratnoj vezi. Predloženi sustav upravljanja letjelicom testiran je na iscrpnom matematičkom modelu letjelice.

Ključne riječi: bespilotna letjelica, hibridno upravljanje, rotokopter

1 INTRODUCTION

The focus of this article is an aircraft propelled with four rotors, called the quadrotor. The quadrotor is capable of vertical takeoff and landing, hovering and horizontal flight which makes it an ideal surveillance vehicle. It can be utilized in various civilian and military operations such as: search and rescue operations, land mine sweeping and traffic surveillance.

Quadrotor was among the first rotorcrafts ever built. The first successful quadrotor flight was recorded in 1921, when De Bothezat Quadrotor remained airborne for two minutes and 45 seconds. Later he perfected his design, which was then powered by 180-horse power engine and was capable of carrying 3 passengers on limited altitudes. Quadrotor rotorcrafts actually preceded the more common helicopters, but were later replaced by them because of very sophisticated control requirements [1]. At the moment, quadrotors are mostly designed as small or micro aircrafts capable of carrying only surveillance equipment. In the future, however, some designs, like Bell Boeing Quad TiltRotor, are being planned for heavy lift operations [2,3].

In the last couple of years, quadrotor aircrafts have been a subject of extensive research in the field of autonomous

control systems. This is mostly because of their small size, which prevents them to carry any passengers. Various control algorithms, both for stabilization and control, have been proposed. The authors in [4] synthesized and compared PID and LQ controllers used for stabilization of a similar aircraft. They have concluded that classical PID controllers achieve more robust results. In [5,6] "Backstepping" and "Sliding-mode" control techniques are compared. The research presented in [6] shows how PID controllers cannot be used as effective set point tracking controller. Fuzzy based controller is presented in [7]. This controller exhibits good tracking results for simple, predefined trajectories. Each of these control algorithms proved to be successful and energy efficient for a single flying maneuver (hovering, liftoff, horizontal flight, etc.).

This article describes a hybrid control concept that combines these various types of controllers to create an energy efficient control system. Furthermore, the proposed control concept aims to unify stabilization and control and create a flight control system (FCS) capable of directed and autonomous flying. The article further examines the behavior of quadrotor's propulsion system focusing on its limitations (i.e. saturation and dynamic capabilities). A lot of

previous research failed to address this practical problem but in case of a demanding flight trajectory, control signals could drive the propulsion system well within the region of saturation, thus causing undesired or unstable quadrotor behavior.

Fly by wire is a concept that was introduced in 1960s, when McDonnell Douglas, now The Boeing Company produced the F-15 Strike Eagle. This flight control system was intended to replace mechanical and hydraulic flight control systems that were standard at that point. Standard mechanical and hydraulic FCSs linked the control of the aircraft directly to the pilot [8]. In Electrical FCSs digital control signals are sent via electrical installations, hence the term by-wire. One advantage of such a system is a significant weight reduction. Because of fail safety reasons, multiple signal sources and several lanes of computing are necessary in a Fly-by-wire system. They provide enough redundancy to achieve the same reliability as earlier mechanical systems. The drawback of such an implementation is a significant increase in the cost of the aircraft.

After its initial development in aviation industry the fly-by-wire concept slowly migrated to ground vehicle industry. Today, modern cars adopted this concept, which was then renamed drive-by-wire. As far as autonomous vehicles (ground or air) are concerned, the term by-wire is slightly stretched because the signal for these vehicles travels by air. Nevertheless, we use this term to note that the system is controlled via electrical signals generated from pilot controls. How the signal travels from the user to the vehicle is of little concern to final implementation.

Although by-wire control is expensive, the cost benefit of such a system can be very high. The computers used for transmitting and receiving control signals can be used to optimize vehicle performance. We can assure system stability by preventing certain moves that could compromise vehicle operation. In aircrafts a fly by wire controller can: optimize lift/drag ratio, improve the angle of attack, increase the energy efficiency and extend aircraft's conventional flight envelope. In this article we will show how a fly-by-wire control system can reduce system nonlinearities and provide the pilot with a very simple set of controls.

2 THE PHYSICAL PRINCIPLES OF QUADROTOR

The basic quadrotor design consists of a control unit placed in the center and four rotors placed symmetrically around that center at a radial distance D . Because each rotor blade is displaced from the center of mass, it produces both thrust and torques with respect to aircraft's center of mass. This is shown in Fig. 1. Torques M_x and M_y of each rotor can be calculated simply by multiplying the thrust of each rotor with distance from the axis of rotation. The

torques on one side of the aircraft cancel out the torques on the other side. If all the rotors are spinning with the same angular velocity, the net torque and hence the angular acceleration equals zero.

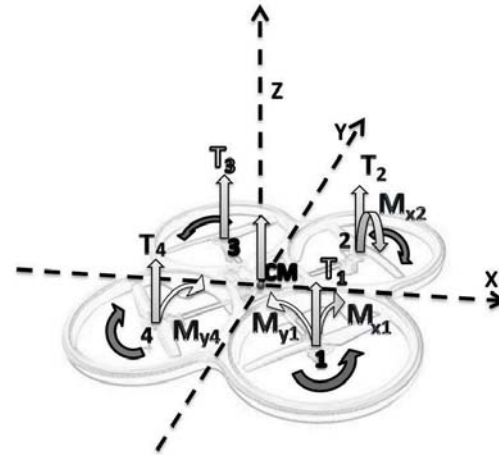


Fig. 1. Forces and torques produced in quadrotor's propulsion system

While rotating, the rotor blades produce aerodynamic torque M_z with respect to the yaw axis. The spinning direction of the rotor dictates the direction of the aerodynamic torque M_z . If all the rotors were spinning in the same direction, the quadrotor would continuously accelerate its yaw angle velocity, rendering the aircraft unstable. This is why rotors that are directly adjacent to each other rotate in different directions (i.e. rotors 1 and 3 rotate in one direction, and rotors 2 and 4 in opposite direction), thus cancelling out each other's torques. This modification achieves yaw angle stabilization.

Yaw acceleration is induced by mismatching the balance in aerodynamic torques (i.e. by increasing the speed of rotors 1 and 3 and decreasing the speed of rotors 2 and 4). Mismatching the torque balance must be achieved without impacting the thrust balance. Therefore, the decreased thrust from one blade has to be compensated with an increase of speed and thrust in other blades. Either roll or pitch is achieved by increasing the speed of rotors on one side of the aircraft and decreasing on the other. For a pair of rotors rotating in the same direction, increasing thrust of one rotor implies decreasing thrust of the other. This is crucial for yaw angle stability.

When the rotors net thrust equals the aircraft's gravity force the quadrotor hovers in the air. Further increasing the net thrust accelerates the aircraft up in the air. Translational acceleration is achieved by maintaining a non-zero pitch or roll angle. Slightly pitching or rolling a quadrotor moves it in x or y direction. This happens because the

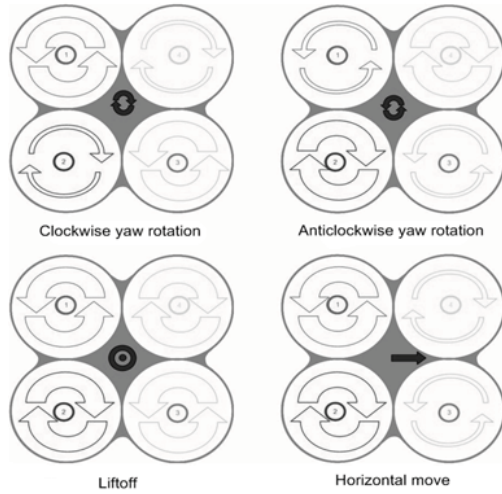


Fig. 2. Different combinations of varying the speeds of rotors can cause a quadrotor to move and rotate

overall thrust is projected in the tilted direction and thus accelerates the quadrotor in that direction. Different combinations of varying the speeds of rotors and the overall effect are shown in Fig. 2.

3 MATHEMATICAL MODEL OF THE QUADROTOR

Testing the proposed Hybrid controller requires an elaborate mathematical model. This model needs to simulate quadrotor behavior in climb, descent and forward flight. At the same time it has to be simple enough to provide fast simulation based experimenting with the proposed controller.

Mathematical model consists of body motion dynamics and propulsion system aerodynamics. Fore mentioned propulsion system aerodynamics includes: Momentum theory, Blade element theory, In Ground Effect, Vortex ring state and windmill break state. Details on this theories and effects can be found in [9], while the details on mathematical modeling of quadrotor's propulsion system can be found in [10]. Without going into details, this article brings the final equation of forces and torques that come from a single rotor.

$$\begin{aligned} T &= \frac{N\rho\bar{a}\bar{c}R^3\Omega^2}{4} C_T \\ \{H_x, H_y\} &= \frac{N\rho\bar{a}\bar{c}R^3\Omega^2}{4} \{C_{H_x}, C_{H_y}\} \\ M_z &= \frac{N\rho\bar{a}\bar{c}R^3\Omega^2}{4} C_{M_z} \\ \{M_x, M_y\} &= \frac{N\rho\bar{a}\bar{c}R^3\Omega^2}{4} \{C_{M_x}, C_{M_y}\} \end{aligned} \quad (1)$$

where N stands for the number of the blades, ρ is the air density, \bar{c} is the average cord length, R is the

blade radius and Ω is the rotor angular speed. The coefficients $\{C_T, C_{H_x}, C_{H_y}, C_{M_z}, C_{M_x}, C_{M_y}\}$ are complex functions of quadrotor's speed and aerodynamic drag/lift coefficients. The details are given in [9, 10].

Modeling of motion dynamics starts with Newton - Euler equations that combine translational and rotational dynamics of a rigid body. These equations formulate the relationship between torques and forces one side and aircraft orientation and position on the other side. The net torques and forces come from propulsion system aerodynamics (1) and net drag that the surrounding air exerts on quadrotors body. We mark the net forces and torques, from the aircraft frame of reference, with \vec{F} and \vec{M} respectively [9]. We write the corresponding Newton - Euler equations in vector form. The angles φ , ϑ and ψ are Euler angles in the fixed, earth perspective coordinate frame. Abbreviations cx and sx stand for $\cos(x)$ and $\sin(x)$, respectively. Rotations ω_x , ω_y and ω_z are observed in quadrotor coordinate system. I_{xx} , I_{yy} , I_{zz} are principal moments of inertia. To calculate corresponding Euler angles in earth reference system, one needs to multiply the second part of equation,

$$\begin{aligned} m_l \ddot{x} &= \left| \vec{F} \right| (c\varphi s\vartheta c\psi + s\varphi s\psi) \\ m_l \ddot{y} &= \left| \vec{F} \right| (s\varphi s\vartheta c\psi + c\varphi s\psi) \\ m_l \ddot{z} &= \left| \vec{F} \right| c\vartheta c\psi \end{aligned} \quad (2)$$

$$\begin{aligned} I_{xx} \frac{\delta\omega_x}{\delta t} - (I_{yy} - I_{zz}) \omega_z \omega_y &= \mathcal{M}_x \\ I_{yy} \frac{\delta\omega_y}{\delta t} - (I_{zz} - I_{xx}) \omega_z \omega_x &= \mathcal{M}_y \\ I_{zz} \frac{\delta\omega_z}{\delta t} - (I_{xx} - I_{yy}) \omega_x \omega_y &= \mathcal{M}_z \end{aligned}$$

with Jacobian matrix.

$$J = \begin{bmatrix} 1 & s_\psi t_\theta & c_\psi t_\theta \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi / c_\theta & c_\psi / c_\theta \end{bmatrix} \quad (3)$$

4 HYBRID CONTROLLER LAYOUT

When developing a controller for quadrotor, two main problems arise: First problem comes from the fact that quadrotor is a highly nonlinear system; Second problem is caused by limited power resources of the propulsion system. Nonlinearities come are caused by complex aerodynamic effects coupled with nonlinear nature of Newton - Euler equations. To avoid complex control solutions like neural networks, which could compensate for these nonlinearities, one needs to limit quadrotor's maneuvering capabilities. For these limited set of maneuvers we can synthesize simple PID controllers based on the linearized mathematical model of the aircraft [10].

As far as the power limitations are concerned the same principal applies. For instance, if we want to move the aircraft in horizontal direction we need to increase the amount

of blade rotation on one side and decrease on the other side of the aircraft. This imbalance will cause the quadrotor to tilt and move in one direction. If we were to direct the quadrotor to increase its altitude, then all four rotors would have to increase their rotation. It is obvious that one side of the quadrotor would need to double the amount of rotation to achieve simultaneous horizontal and vertical movement of the aircraft. Unfortunately, power limitations can block such an increase in blade rotation and a possible result could be quadrotor unstable behavior. By limiting quadrotor's maneuvering capabilities we assure that this situation never occurs, and that the aircraft is stable.

Apart from these inherent problems, we also need to keep in mind that a controller should be capable of both directed and autonomous flight. This implies that at some point a switch between these two states has to be initiated. Hence, a hybrid controller, which uses an automaton to switch between different states and has layer based architecture seems as a natural solution. The lowest layer in this design is assigned to quadrotor stabilization and angle control. The second layer is designed for height control. The last two layers, third and fourth, are Position control and Autonomous Navigation, respectively. The automaton's states switch different layers of control on and off, and thus by switching between the states automaton enables different level of control: directed control, position control or autonomous navigation. The overview of Hybrid controller layer based structure is shown in Fig. 3. The connection between different layers is marked with a corresponding arrow. This figure also shows how autonomous navigation layer receives operator's instructions via a joystick controller or a path planning panel.

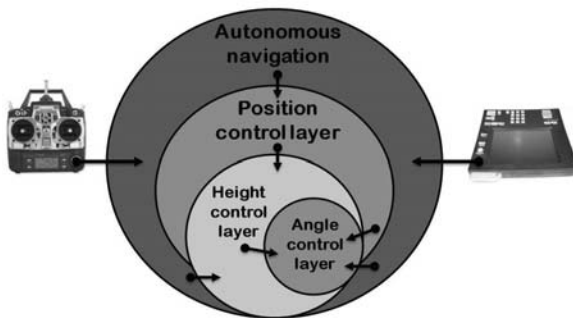


Fig. 3. Overview of Hybrid controller layers

4.1 Angle Control Layer

The angle control layer is in charge of aircraft stabilization. By keeping the roll and pitch angles at 0° this controller effectively stabilizes the aircraft in hover position. If we were to roll or pitch the aircraft by a certain angle

α , then it would move in y or x direction, respectively. It is obvious that this layer of control is crucial not only for quadrotor stabilization but also for its positioning.

For a symmetrical body ($I_{xx} = I_{yy}$) with small amounts of moments of inertia ($I_{xx}, I_{yy}, I_{zz} \ll$) we can neglect the terms $(I_{yy} - I_{zz})\omega_z\omega_y$ in (2). This simplification allows us to write a very simple equation for angle control:

$$I_{\alpha\alpha} \frac{\delta\omega_\alpha}{\delta t} = \ddot{\alpha} = \mathcal{M}_\alpha \quad (4)$$

This is of course, observed from quadrotor coordinate system. To calculate corresponding Euler angles, one needs to multiply (4) with Jacobian transformation matrix (2). This is why we further simplify the problem by considering only small angles, and thus the Jacobian transformation matrix (2) becomes the identity matrix I .

Fore made assumptions leave us with a second – order system with a constant propulsion system gain K_{Trq} (i.e. the propulsion system linearization $\mathcal{M}_\alpha = \alpha \cdot K_{Trq}$ is applied). For this system we derive a cascade controller with a set of gains K_ω and K_α that are tuned so that the rise time and the percentage overshoot of angle control equals $255ms$ and 12% respectively [10].

4.2 Height Control Layer

Height controller is the main controller in the proposed control system. It is designed as a three state automaton. This automaton is parallel to the main hybrid automaton which switches between the layers. It changes the parameters or switches the controller transfer function depending on the quadrotor's current situation. Fig. 4 shows the height control system. There are two main problems that have to be resolved in order to derive an effective height controller: The first problem is the constant interference of the gravity force and the second is the nonlinear aerodynamic drag on a moving quadrotor.

The proposed height controller comprises three states: change of altitude, the settling of altitude and hovering state. When a change of height reference is sensed the hybrid controller enters the change of altitude control state. During this state, the automaton switches the height controller to a P type control with a relatively large gain. As a result of this the propulsion system starts to operate at its limits, always switching either off or to the maximum voltage. This way we shock the aircraft into moving towards the new reference point. When the aircraft approaches the referenced height the next state is triggered.

During the settling state, the gain of the proportional controller is decreased and the integral component is added to the height controller. It is necessary to note that the integral component was dormant during the change of height. This way we avoided possible integrator windup during

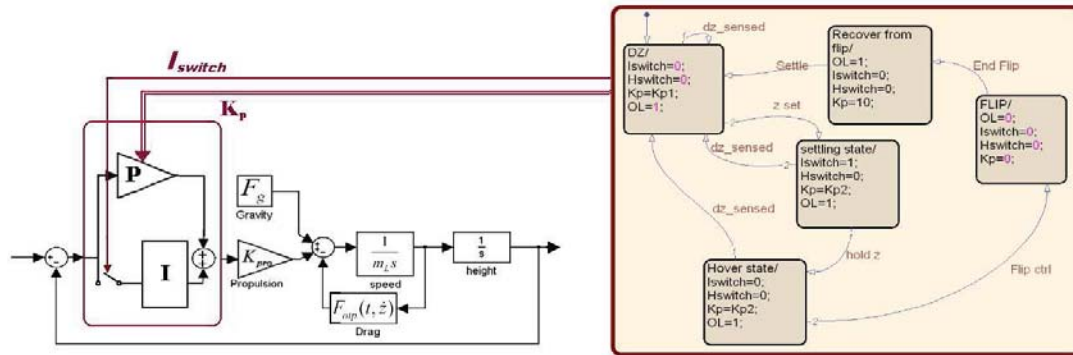


Fig. 4. Height controller composed of a variable PI controller operated with a three state automaton [10]

this transition state. We are able to turn on the integrator now because the aircraft is close to its final destination and the error signal is too small to cause integrator windup.

When the aircraft settles in referenced destination the hovering state of controller is triggered. During this state, the main hybrid controller enables the change of XY coordinates and aircraft orientation. Height controller has to use as little energy as possible to keep the aircraft in the desired height. This is achieved with a small proportional gain controller. Smaller gain decreases control system’s reaction to changes in quadrotor’s altitude caused by aircraft’s movement in horizontal direction. As a consequence, quadrotor’s propulsion system remains practically undisturbed, hence, enabling design of a control system that is less energy consuming. The entire height controller was parameterized using ITAE optimization criteria to achieve the desired behavior [10].

4.3 Position Control Layer

Combined with height control layer, position control layer yields a full 3D positioning control system. We start off with a small angle approximation of equation (1). Keeping in mind that ψ and ϑ are very small, and that $\varphi = 0$, we can write:

$$\begin{aligned} m_l \ddot{x} &\approx \left| \vec{\mathcal{F}} \right| \vartheta \\ m_l \ddot{y} &\approx \left| \vec{\mathcal{F}} \right| \psi \end{aligned} \quad (5)$$

Because we want to be able to control the aircraft’s yaw angle φ we cannot assume that it is always equal to zero. On the other hand, if we know the yaw angle of the aircraft, then we can rotate the earth coordinate system for that angle, thus effectively lining up the two coordinate systems. The overall result is that for the lined up coordinate systems $\varphi = 0$ and equation (5) holds. The hybrid controller automaton therefore needs to assure that the yaw angle is

aligned to the referenced value before or after, but never during the position control process.

Now as it is obvious from (5), the angle control layer becomes an open loop transfer function of the position control layer. We can write independent equations for X and Y position control which are basically the same (i.e. only the corresponding angle differs). During the change of position, the change of altitude is disabled and the quadrotor is hovering. Therefore, the net force of the quadrotor $\left| \vec{\mathcal{F}} \right|$ can be linearized for hover mode as a constant gain $K \left| \vec{\mathcal{F}} \right|$. The closed loop position control system based on the fore mentioned principals is shown in Fig. 5.

The proposed PDT1 controller has been parameterized using Root locus method so that the system is fast enough and has a 10% overshoot [10].

4.4 Navigation Layer

This layer enables an operator to control the aircraft. It allows three types of control: fly-by-wire, position or LQ trajectory control. The operator switches between these controls by switching the navigation layer state. Depending on its current operating state, the navigation layer switches the lower layers on or off and controls their operation.

Fly by wire control. When in Fly by wire control state, the navigation layer behaves similar to the position control layer. The operator becomes a pilot during this control state and controls the velocity and the direction of the quadrotor by moving a joystick. Because the joystick sends speed control inputs, a speed controller must be designed. The velocity open loop transfer function is based on equation (5), but this time the control loop has only one astatism. This allows us to choose a simpler P controller. Like the PDT1 controller for the position layer, the P controller was also tuned using Root locus method to achieve

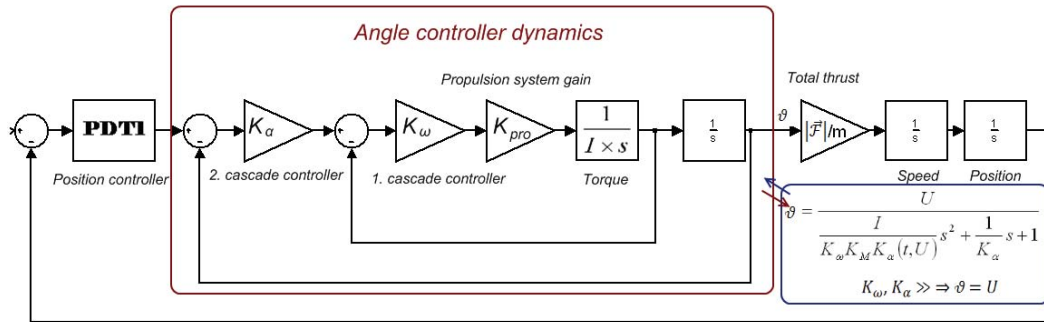


Fig. 5. Cascade control system for position control comprised of angle control system, motion dynamics and a PDTI controller [10]

quick response with no overshoot. When the pilot moves the joystick, the magnitude of the move is translated into speed reference for the fly-by-wire control system.

Due to the linearization and power constraints, the change of altitude cannot be initiated during the XY positioning of the quadrotor. The navigation layer needs to delay the height references sent to height control layer during the positioning of the quadrotor. Height is operated via throttle. The pilot pushes the throttle in order to move the quadrotor to a higher altitude, but the quadrotor can respond only when the joystick is at rest. While the joystick is at rest, the throttle signal is sent to the height control layer, which executes the change of altitude.

Position control. The position control state (Fig. 6) is a hierarchical automaton which controls the operation of lower layer control systems, specifically the position control layer and height control layer. In this mode of operation, the operator is controlling the quadrotor by setting the desired position and orientation. When the desired position is set, the quadrotor is initiated to execute the necessary changes. The position control automaton divides every operator request into three simple maneuvers:

1. From every desired move the controller takes the change in height and executes it with the highest priority.
2. When the change of height in the desired move is achieved, the controller moves to the next state. In this state it enables the change of XY position of the quadrotor aircraft.
3. At the end, the quadrotor control system enters the third state. This state is dedicated to the change of quadrotor’s yaw angle which actually directs the quadrotor. Because of the rotation matrix coupling it

is very important that during the change of XY position this angel remains constant. This allows us to develop a very simple decoupling based on the previous quadrotor orientation [10].

Navigation. The primary function of the navigation layer is to guide the quadrotor through the designated search patterns. The operator designs these patterns by defining waypoints. These waypoints are then interconnected with a series of splines that form a corresponding search pattern trajectory. These is achieved with a modified Ho - Cook algorithm [11]. The trajectory is then supplied to the linear quadratic tracker (LQT) controller which calculates the necessary control actions.

Trajectory planning for unmanned aircraft vehicles is an active field of research. Trajectory planning can be on-line or offline. With online trajectory planning, the aircraft’s control system detects obstacles, both moving and

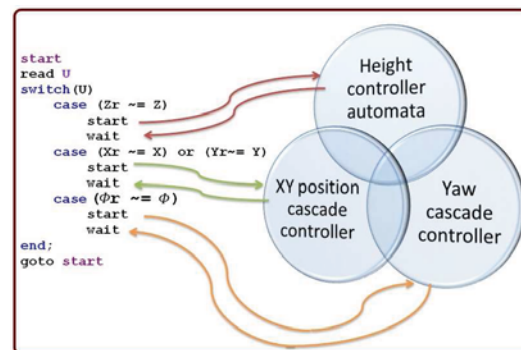


Fig. 6. Executive controller of the proposed hybrid system which contains a three state automaton controlled with simple transition logic [10]

non moving, and adjusts its trajectory to avoid these obstacles. This type of trajectory planning is necessary in cluttered environments [12]. When, on the other hand, the quadrotor is deployed to a search and rescue or exploration missions it hovers above the ground. The hover altitude can be chosen so that the quadrotor avoids both ground obstacles (ie. trees, buildings and ground vehicles) and air traffic (ie. birds or other aircraft vehicles). This allows us to use offline trajectory planning, where the aircrafts trajectory is planned in advance and dynamic changes to the surrounding environment are neglected [13].

Different types of offline trajectory planning algorithms, mainly for fixed wing aircrafts, have been proposed [12,13,14]. One significant difference between a fixed wing aircrafts and a rotorcrafts is a substantially higher maneuvering capability of rotorcrafts. Generally speaking, to move a fixed wing aircraft one needs to change its heading direction first. For a rotorcraft, the positioning can be done independent from its heading direction. This makes a Ho - Cook interpolation method that is frequently being used in robotics a good choice for planning a trajectory for a quadrotor.

Given a set of n waypoints, we construct an interpolation trajectory comprised of splines $S_i(t)$, where $i = 1, \dots, n-1$; $t = [0, \Delta t_i]$. In order to assure that the quadrotor flies smoothly through the planed trajectory, we need to make splines S_i , C^2 continuous. Moreover, we need to take into account the starting and final settling positions, at which the quadrotor needs to be fully stopped. This means, that the first and second derivatives of the starting and final trajectory position need to be equal to zero [11]. The fore mentioned conditions are written in the following equations:

$$\begin{aligned} \ddot{S}_1(0) &= \dot{S}_1(0) = 0 \\ \ddot{S}_{n-1}(\Delta t_{n-1}) &= \dot{S}_{n-1}(\Delta t_{n-1}) = 0 \\ S_i(\Delta t_i) &= S_{i+1}(0) \\ \dot{S}_i(\Delta t_i) &= \dot{S}_{i+1}(0) \\ \ddot{S}_i(\Delta t_i) &= \ddot{S}_{i+1}(0) \end{aligned} \quad (6)$$

C^2 continuity and standstill conditions can be satisfied using cubic polynomial splines for $i = 2, \dots, n-2$, and fourth degree polynomial splines for $i = 1, n$. We can optimize the trajectory characteristics by varying time intervals Δt_i . One way of setting time intervals is taking into account the length of each chord (i.e. $|X_i - X_{i-1}|$). The longer the chord the longer the time interval Δt_i . [15]

A modified Taylor method is used to achieve good approximation results [11]. When a given set of waypoints is interconnected with splines, the operator is prompted to inspect the given trajectory. If the trajectory deviates from the desired pattern additional waypoints are added to the trajectory. These new waypoints are positioned at half a distance between each old waypoint.

LQT control. When the navigation planner designs the search pattern, the LQT controller is prompted to calculate the necessary control signals. The LQT controller uses a simplified linearized quadrotor model to calculate the appropriate control sequence for the given trajectory. During the execution of LQT control, the angle and height control layers are also active. These layers are used to stabilize the quadrotor at the desired altitude. They also provide control inputs for the LQT controller, thus allowing it to execute 2D trajectory. During trajectory execution, the height controller holds the quadrotor in the previously set altitude.

Again, as in a position control layer, the navigation layer includes the angle control layer. Here we expand our assumptions from the position control layer and assume that the angle control layer is fast enough so that its dynamics can be neglected. This way, the nonlinear model of the quadrotor orientation can be replaced with a simple gain (Fig. 5). Because the quadrotor height control layer is active, the net force of the quadrotor $|\vec{F}|$ can be linearized for hover mode as a constant gain $K_{|\vec{F}|}$. This simplification of the nonlinear quadrotor model yields an open loop transfer function for the LQT controller. Because the quadrotor is symmetric with respect to X and Y position control the same transfer functions applies to both position control loops.

$$G_{Ox} = G_{Oy} = \frac{K_{|\vec{F}|}}{s^2} \quad (7)$$

The transfer function (7) is a simple double integrator for which we can write the following state space representation:

$$\begin{aligned} A &= \begin{bmatrix} A_x & 0 \\ 0 & A_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ B &= \begin{bmatrix} B_x & 0 \\ 0 & B_y \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ T_u & 0 \\ 0 & 0 \\ 0 & T_u \end{bmatrix} \\ C &= \begin{bmatrix} C_x & 0 \\ 0 & C_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \quad (8)$$

The LQT control scheme is obtained by minimizing the associated *quadratic performance index* [16,17]. For a linear, fully observable system we can write the following tracking problem performance index (9):

$$J = \frac{1}{2} \left\{ \mathbf{e}(t_f)^T \cdot \mathbf{P} \cdot \mathbf{e}(t_f) + \int_{t_0}^{t_f} \left[\mathbf{e}(t)^T \cdot \mathbf{Q} \cdot \mathbf{e}(t) + \mathbf{u}(t)^T \cdot \mathbf{R} \cdot \mathbf{u}(t) \right] dt \right\} \quad (9)$$

The performance index (9) is calculated for a time interval $t \in [t_0, t_f]$ by multiplying tracking error $\mathbf{e}(t)$, control signal $\mathbf{u}(t)$ and final position error $\mathbf{e}(t_f)$ with corresponding weighting matrices \mathbf{Q} , \mathbf{R} and \mathbf{P} . The tracking error $\mathbf{e}(t)$ and control signal $\mathbf{u}(t)$ are written in vector form. The control output and the quadrotor behavior are tuned by varying matrices \mathbf{Q} , \mathbf{R} , \mathbf{P} . The matrices are chosen to be symmetric and positive-definite. The magnitude of the \mathbf{Q} matrix minimizes the error of LQT, the \mathbf{R} matrix minimizes its energy consumption and the \mathbf{P} matrix minimizes the final position error. A first choice for the matrices \mathbf{Q} and \mathbf{R} in is given by the Bryson's rule:

$$\mathbf{Q}_{ii} = \frac{1}{\text{maximum acceptable value of state } i} \quad (10)$$

$$\mathbf{R}_{ii} = \frac{1}{\text{maximum size of control signal } i}$$

Bryson's rule is often a starting point for trial-and-error iterative design procedure aimed at obtaining desirable properties for the closed-loop system [16]. One of these procedures uses a Bryson's rule as a starting point, and iterates the matrices using the following rule:

$$\mathbf{Q}_{ii}^{NEW} = \mathbf{Q}_{ii}^{OLD} \left| \frac{\text{max. measured value of state } i}{\text{max. allowed value of state } i} \right|$$

$$\mathbf{R}_{ii}^{NEW} = \mathbf{R}_{ii}^{OLD} \left| \frac{\text{max. meas. size of control signal } i}{\text{max. size of control signal } i} \right| \quad (11)$$

New values for the control input are calculated with every iteration. The simulation is performed with the calculated control inputs and maximum measured values for errors of states and control signals are measured. Using (11), new values for \mathbf{Q} , \mathbf{R} , \mathbf{P} matrices are set. The procedure ends when all of the control signals and state values fall beneath the maximum allowed values [16].

4.5 The looping

In the previous sections we analyzed nonaggressive flying techniques. In this chapter, however, we present a control algorithm for an aggressive maneuver. This is more or less a standard aerobatic maneuver called the looping. During the looping, the aircraft lifts up in the air, very quickly and starts to spin around X or Y axis of the aircraft's reference frame. After the whole 360 degrees spin is executed, the aircraft returns to its previous state. The control algorithm is divided into six steps, for which we derived a corresponding six state automaton. The states are:

1. *The dormant state*, in which the flip controller waits for instructions from the user. Upon receiving a looping command from the superior controller the automaton moves to the second state.
2. *The fast liftoff state* triggers a request for maximal rotor spin. This causes the aircraft to liftoff with maximum climb speed. This state also switches off the height controller, causing the aircraft to fly in an open loop control. When vertical speed of the aircraft reaches $2.6m/s$ the controller switches to the next state.
3. *The spin state* turns off the rotors one and two (Fig. 1), thus making the quadrotor to spin around Y axis. This state is active for a certain amount of time τ_1 . During that time the aircraft rotation accelerates to a speed Ω_{flip} . Assuming that the air resistance is much smaller than the momentum induced by rotors, we can easily calculate the time τ_1 . The next trigger arises upon the expiration of τ_1 and the controller switches to the fourth state.
4. *The spinning state* shuts down the propulsion system. When the aircraft makes the whole 360° loop the sensors trigger the next state.
5. *The counter spin state* turns on rotors one and two. This is very important because we need to stop the aircraft from spinning once it rotated the whole 360° . This state also needs to be active for the time τ_1 .
6. *The control activation state* returns the control of the quadrotor to the main hybrid controller. During this state, the main controller stabilizes the aircraft and returns it to the initial position [10].

5 SIMULATIONS

To evaluate the proposed hybrid controller we carried out a series of simulation studies with a nonlinear mathematical model of the quadrotor. This model describes the quadrotor behavior in climb, descent and forward flight and includes various realistic aerodynamic effects (i.e. Ground effect, Windmill break state, Aerodynamic drag). In this article we present the results of Position and LQT control simulation studies. The Position and LQT control operate on the highest level of the proposed controller. Due to the layer interconnection, they affect lower level control systems. Thus by examining the behavior of the highest level of control we can evaluate the behavior of lower layer control systems.

5.1 Position control

The position control is a state in autonomous navigation layer. It directs the position control layer. The position

control layer is designed to execute control signals sent from the navigation layer. One representative case of position control is presented in this chapter. The event time table of the control signals for the proposed maneuver is given in Table 1. A 3D representation of a part of this sequence is given in Fig. 7. For better clarity, the change of altitude during this sequence is omitted in a 3D representation.

The quadrotor starts its flight at one meter above the ground and then lifts off to the requested altitude. The next control request moves it in y – axis direction. Following that is the move in all three directions including the change of orientation after which we request the quadrotor to settle at 5 meters above the ground. According to these control signals the quadrotor executes a flight trajectory given in Fig. 8. This figure shows the change of quadrotor’s position and orientation in 3D space.

The sequence in Fig. 8 ends with a looping maneuver. This figure shows how quadrotor is lifted up in the air and then rotated around y axis. Naturally there is a slight disturbance in quadrotor x position when the quadrotor pitches. When the quadrotor looping maneuver is finished, the control sequence, described in IVE, stops further rotation and triggers the position control on. Until the control system recovers from this aggressive maneuver it experiences relatively big position errors, nevertheless it finally brings the quadrotor to its previous and desired altitude and coordinates.

Table 1. The control signal timetable

Time[s]	0	45	70	105	130
X axis [m]	0	0	5	0	0
Y axis [m]	0	5	0	0	0
Z axis [m]	5	5	5.5	5	5
Yaw [rad]	0	$\pi/2$	$\pi/3$	0	0
Flip	no	no	no	no	yes

The control sequence that starts at 70 s gives the best insight into position control behavior (Fig. 9). Starting at 70 s, the position control layer prompts the height control layer to move the aircraft at the desired altitude. Fig. 9 shows how the height controller’s *bang-bang state*, which operates at the beginning of the height transition, saturates the control signals. During this stage, a change of any other quadrotor coordinate has to be blocked, otherwise, because of control signal saturation, the aircraft would become unstable. When the aircraft moves close to the requested altitude, the height controller switches to *PI control state* and the control signal falls beneath saturation limits. After the altitude is settled, the *hover state* of height controller is triggered. This enables the change of quadrotor X and Y positions.

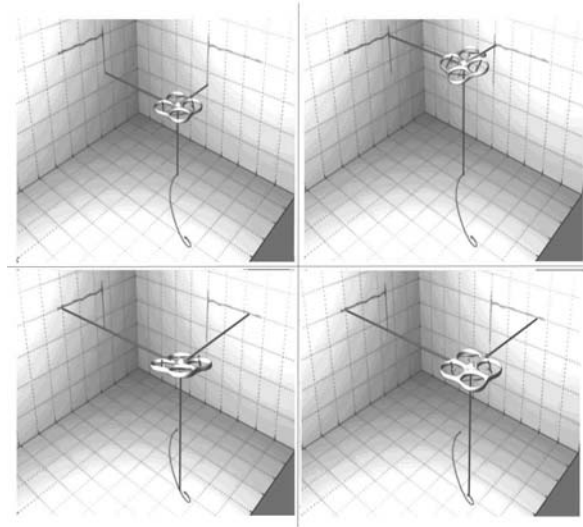


Fig. 7. Screenshots from a 3D virtualization of 3rd maneuver in Table 1. Starting with a liftoff on upper left screenshot. Upper right picture shows the start of horizontal movement. Lower left figure shows how the quadrotor turns to slow down, and finally stabilizes at the last screenshot

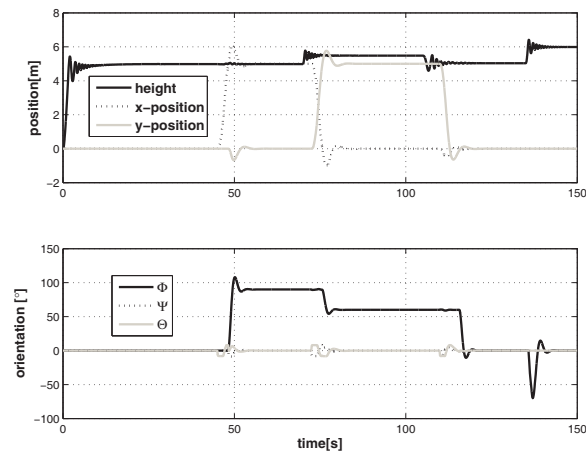


Fig. 8. Quadrotor’s flight trajectory as a result of control signals in Table 1

If we closely examine this sequence, we see that even though the request was set for all three coordinates, X and Y do not change before the quadrotor starts settling in the desired altitude. The position controller is triggered when control signals leave the saturation level. This can be seen as a control impulse in Fig. 9. This impulse moves the quadrotor in the desired xy direction. During this time, the change of orientation is still prohibited by the position

controller automaton (Fig. 6). After the quadrotor reaches the desired x and y coordinates the change of orientation (yaw) is triggered. The change of yaw was triggered even though the aircraft did not actually settle in the desired coordinates. This is acceptable, because, as can be seen on Fig. 9, there is no risk of saturating control signals during this time interval.

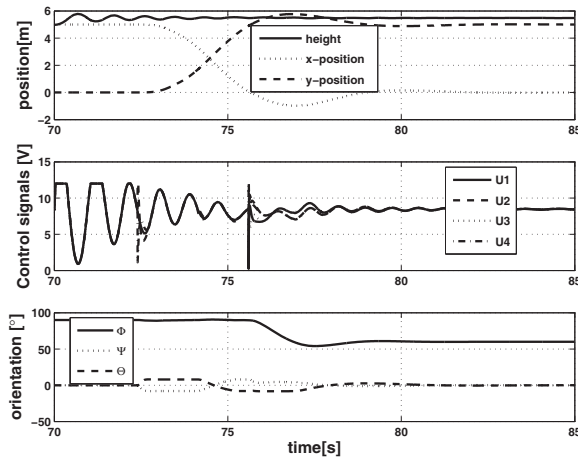


Fig. 9. Quadrotor's position, control and orientation signals for the third control sequence at 70 s

5.2 LQT controller

In this simulation example we show the tracking abilities of the LQT controller for a quadratic search pattern trajectory. In the quadratic search pattern trajectory, the aircraft moves in x direction first, then it moves in y direction, and finally returns to the previous x position. First we present the derivation of the LQT controller and afterwards we show the simulation results.

The derivation of the LQT control starts with optimizing the matrices \mathbf{Q} , \mathbf{R} , \mathbf{P} using Bryson's method. The maximum allowed values for the quadrotor maneuvers are given in Table 2. The angle controllers limit the angle inputs within $\pm 8^\circ$ in order to keep the quadrotor in the desired linearized area of behavior. The speed limitations come from aircraft's dynamics. Table 3 shows the final values of matrices \mathbf{Q} , \mathbf{R} , \mathbf{P} , obtained by using the described optimizing procedure (10).

Table 2. Maximum allowed values of quadrotor states and control signals

	Max.Position	Max.Speed	Max.Angle (input)
X axis	5.25 m	2 m/s	8°
Y axis	5.25 m	2 m/s	8°

Table 3. The optimized values of matrices \mathbf{Q} , \mathbf{R} , \mathbf{P}

\mathbf{Q}	$\begin{bmatrix} 0.6569 & 0 \\ 0 & 0.6569 \end{bmatrix}$
\mathbf{R}	$\begin{bmatrix} 0.0328 & 0 \\ 0 & 0.0318 \end{bmatrix}$
\mathbf{P}	$\begin{bmatrix} 0.0328 & 0 \\ 0 & 0.0318 \end{bmatrix}$

We assume that the linear connection between waypoints creates the ideal trajectory which LQT needs to track. Furthermore, we require that the aircraft needs to pass through the inputted waypoints. In the following figures we compare the Position control state, the LQT controller without using the offline trajectory planning algorithm (LQT*) and the LQT controller with offline trajectory planning algorithm.

In Figures 10, 11 and 12 we compare the different trajectories of each control method. The LQT* displays the ability to track the desired trajectory with the smallest error. Unfortunately, the LQT* tends to miss the desired waypoints. Fig. 9 shows that LQT* never reaches the second x coordinate. If we regard the waypoints as checkpoints, as it was previously mentioned, then this behavior of LQT* is not acceptable.

One way we can make the quadrotor reach the desired waypoints is by using the position controller. In comparison with LQT* trajectory it is obvious that Position control cannot follow the ideal trajectory. On the other hand, unlike the LQT*, the position controller flies the quadrotor onto the desired waypoints. If we want to achieve both of this conditions (i.e. good trajectory tracking and reaching waypoints), then we need to utilize the offline trajectory planning algorithm.

The simulation results shows how the offline trajectory planning algorithm creates a simplified trajectory, one that the quadrotor LQT controller can successfully follow and reach the desired waypoints at the same time. This planned trajectory differs from the ideal trajectory. From Fig. 13 it is obvious that this control method has the highest deviations from the ideal trajectory. These deviations come solely from the simplified trajectory deviations because the LQT controller accurately tracks the simplified trajectory. This is important if we consider that, during the trajectory planning, the operator is made aware of the simplified trajectory. This allows him to make the necessary changes and plan the desired trajectory. One more benefit of the

LQT with trajectory planning is that it reaches every way-point. This happens because the planning algorithm acts as a look - ahead tool which prepares the quadrotor for a sharp turn. In effect the trajectory planning algorithm successfully replaces these sharp turns with a curved trajectory that the LQT is able to make.

The tracking accuracy has repercussions to the energy consumptions. Greater accuracy and speed basically cause bigger energy consumptions. In search and rescue or minesweeping operations the terrain inspection is often very slow and time consuming. This means that we do not need to force the aircraft to move quickly. On the other hand, we want to stay airborne for as long as we can and therefore we need to economize the energy consumptions. In simple LQ control, the energy consumption requirements are met with tuning the R matrix but this inevitably causes the problems in tracking capabilities of the controller. If the trajectory is dynamically complicated, energy consumption limits the tracking capabilities of the controller. We use the trajectory planning algorithm to ease the dynamics of the ideal trajectory and thus economize energy consumptions and achieve the desired tracking.

The control signals are presented in figures 13 and 14. The results in these figures show how Position control has the highest energy consumptions. Figures 10 and 11 show that this type of control causes the highest speeds of quadrotor. The position control uses the bang-bang control to move the quadrotor, and this saturates the control signals and moves the quadrotor very quickly.

The LQT* control, on the other hand, shows much better power saving capabilities. The control signals enter saturation limits only at the beginning of the maneuver. This happens because the ideal trajectory has no dynamic restrictions. The LQT with trajectory planning displays even better results. The control signal of LQT never saturates and has the lowest overall values compared to other control methods.

5.3 The effect of imbalanced rotor characteristics

Static propulsion characteristic of a rotor is a ratio of thrust per voltage. For an ideally tuned quadrotor, the static propulsion characteristic of each rotor is the same. In reality though, the quadrotor is never ideally tuned and therefore a certain difference in thrust per voltage ratio of each rotor is expected. This can lead to errors in trajectory tracking and in some cases it can trigger unstable behavior of the quadrotor.

In these experiments, we have varied the static propulsion characteristics of rotors 1 and 2. (Fig. 1) The results of these simulation experiments are presented in Fig. 15. From these results we can conclude that varying static propulsion characteristics of rotors 1 and 2 by a same

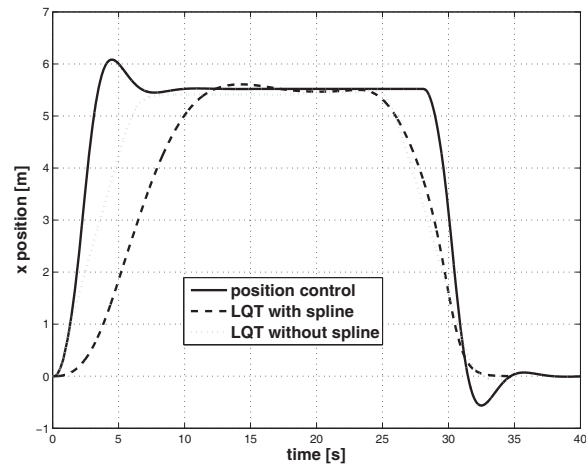


Fig. 10. The x position trajectory for different types of tracking

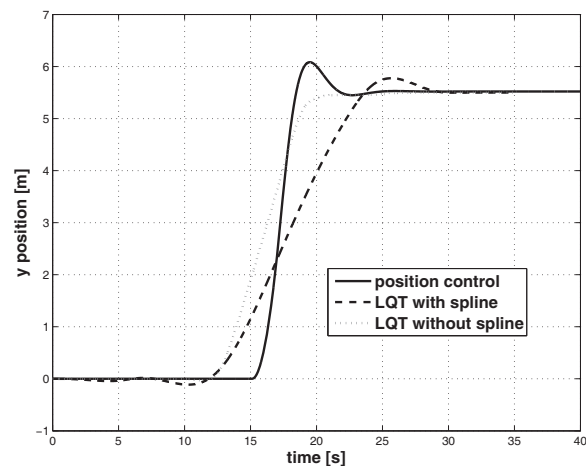


Fig. 11. The y position trajectory for different types of tracking

amount affects the x position tracking ability of the hybrid controller. On the other hand, y position tracking is not affected with these variations.

A simple review of quadrotor's physical principals explains this phenomenon. The rotors 1 and 2 are placed on the opposite sides of quadrotor's frame of reference x axis. Therefore, the variation of thrust in rotor 1 is canceled by the same variation of thrust in rotor 2. This causes the torque variations ΔM_{x1} and ΔM_{x2} to cancel out each other. With respect to y axis of quadrotor's frame of reference, these rotors are placed on the same side. This means that with the same voltage applied to all the rotors, the total torque $M_y = \sum_{i=1}^4 M_{yi}$ is equal to $2\Delta M_{y2} = 2\Delta M_{y1}$

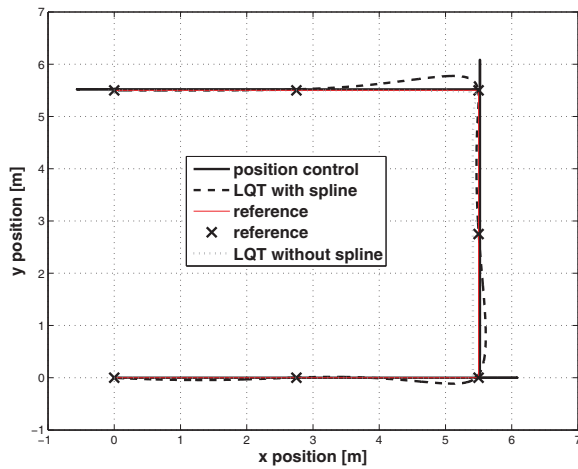


Fig. 12. Quadrotor trajectory in XY plane for different types of tracking

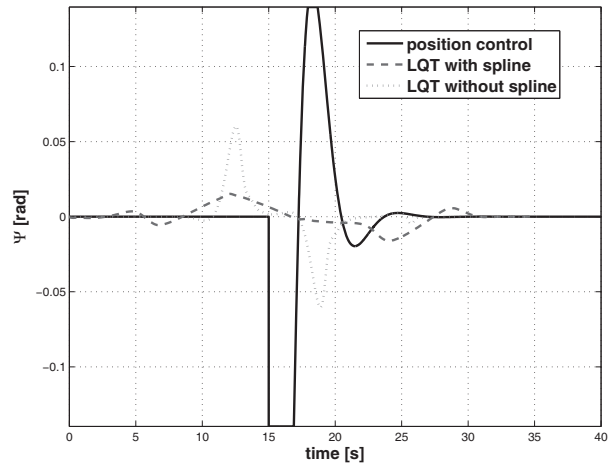


Fig. 14. Control signal for y position for different types of tracking

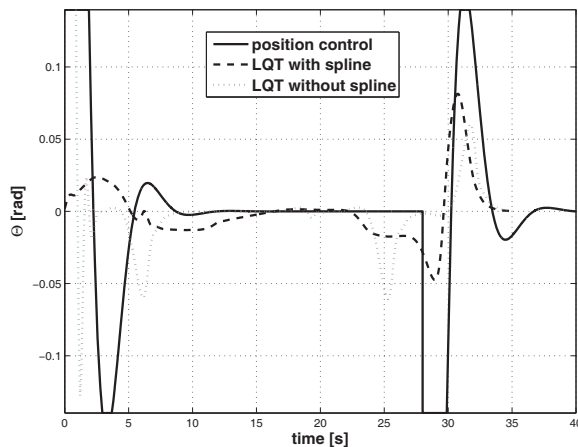


Fig. 13. Control signal for x position for different types of tracking

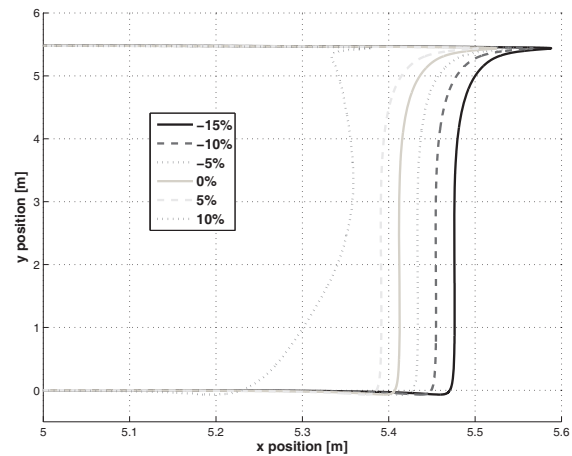


Fig. 15. Quadrotor trajectory in XY plane for different rotor characteristics

and not zero. This causes a constant interference in y position control system and thus creates an error in y position tracking.

The same effect, but on opposite axis, can be expected if rotors 2 and 3 static propulsion characteristic is varied. This means that different variations on all the rotors cause an error in overall tracking ability of this control system. Stronger variations of rotors propulsion characteristics can render the aircraft unstable.

6 CONCLUSION

This article introduces a layer based, hybrid quadrotor controller that provides a stable fly - by - wire and trajectory tracking control of the aircraft. In order to cope

with nonlinear effects and limits of the propulsion system, the highest layer of the proposed controller contains an automaton with predefined states. Each request regarding the change in position and/or orientation of the quadrotor is divided in discrete maneuvers that are executed separately, with a predefined sequence. Low level linear controllers are based on a cascade scheme, containing two loops: inner loop for angles and outer loop for 3D coordinates. They provide a stable and fast response to the control references sent from higher levels of control.

Simulation results, presented in the article confirmed effectiveness of the proposed control scheme. The proposed controller efficiently tracks the precompiled search pattern trajectories, utilizing very small amounts of energy. The

proposed interpolation technique creates well balanced trajectories that enable the control system to drive the quadrotor through the predefined waypoints.

Ongoing research is devoted to implementation of the proposed hybrid controller on a quadrotor laboratory setup based on an embedded microcontrol system. Further development of this hybrid control system includes developing a robust and adaptive angle control system. This type of control would stabilize the quadrotor behavior with respect to variations in static propulsion characteristics of rotors. Adaptive control could also be applied to height control layer, thus making it less affected by certain aerodynamic effects. (i.e. In Ground Effect, Descent aerodynamic effects).

REFERENCES

- [1] Gessow and G. C. Meyers, Jr., *Aerodynamics of the helicopter*. New York, USA: Frederick Ungar Publishing Co., 1967
- [2] Anderson, S. B. (March), "Historical Overview of V/STOL Aircraft Technology", NASA Technical Memorandum 81280
- [3] "US Army looking at three configuration concepts for large cargo rotorcraft". Flight International, October 9, 2007
- [4] S. Bouabdallah and R. Siegwart, "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor," in *Proceedings book of the IEEE International Conference on Robotics and Automation*, (Barcelona, Spain) pp.2247– 2252, 2005.
- [5] S. Bouabdallah, A. Noth and R. Siegwart, "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor", in *Proceedings book of the International Conference on Intelligent Robots and Systems*, (Sendai, Japan), 2004.
- [6] P. Adigbli, C. Grand, J. B. Mouret and S. Doncieux, "Non-linear Attitude and Position Control of a Micro Quadrotor using Sliding Mode and Backstepping Techniques", World Academy of Science, Engineering and Technology 26 2007
- [7] M. Varga, S. Bogdan, "Fuzzy – Lyapunov Based Quadrotor Controller Design", in *Proceedings book of the European Control Conference*, (Budapest, Hungary), 2009,
- [8] L. K. Leppin, *The Conversion of a General Motors Cadillac SRX to Drive-By-Wire Status*, Master Thesis, Virginia Polytechnic Institute and State University, 2005
- [9] Bramwell, A. R. S., *Bramwell's Helicopter Dynamics* :American Institute of Aeronautics & Ast; 2nd edition, 2001
- [10] M. Orsag, S. Bogdan, "Hybrid control of quadrotor", in *Proceedings book of the 17th Mediterranean Conference on Control and Automation*, (Thessaloniki, Greece), 2009
- [11] P. G. Ranky, C. Y. Ho, "Robot modeling – Control and Application with software", IFS (Publications) Ltd., UK, 1985.
- [12] K. Yang, S. Sukkarieh, "Real – time continuous curvature path planning of UAVS in cluttered environments", in *Proceedings book of the 5th International Symposium on Mechatronics and its Applications* (Amman, Jordan), 2008
- [13] T. G. McGee and J. K. Hedrick, "Path Planning and Control for Multiple Point Surveillance by an Unmanned Aircraft in Wind (Draft)", in *Proceedings book of American Control Conference*, 2006
- [14] S. Sharma, E. A. Kulczycki, A. Elfes, "Trajectory Generation and Path Planning for Autonomous Aerobots" in *Proceedings book of International Conference on Robotics and Automation*, (Roma, Italy), 2007
- [15] L. Piegl, W. Tiller, *The NURBS book*, (New York, USA): Springer - Verlag, 2nd ed., pp. 364 - 365, 1997
- [16] D. S. Naidu, *Optimal control systems*, (Boca Raton, USA): CRC Press, Inc., 2002
- [17] F. L. Lewis, V. L. Syrmos, *Optimal control*, 2nd ed. (New York, USA): John Wiley & Sons, Inc., 1995



Matko Orsag received his B.S.E.E. in 2008 at the University of Zagreb, Croatia. He is currently employed at the Faculty of Electrical Engineering and Computing, University of Zagreb, as a research assistant and a full-time Ph.D. student funded by the Ministry of Science, Education and Sports, Republic of Croatia. His main areas of interest are autonomous systems, robotics and intelligent control systems. During his undergraduate years he received several student awards, as well as a 2008. bronze plaque "Josip Lončar" for overall outstanding academic achievement. He also received the University of Zagreb Rector Award for the best student projects in 2006. He is a coauthor of several papers presented at international conferences and published in proceedings.



Marina Poropat works at "Tehnozavod-Marušić, Security & Communication Systems", as a project manager. She got her degree at the Faculty of Electrical Engineering (2003-2009, course: Automatics) where she wrote her thesis on the theme of "LQR control algorithm on a four-rotor aircraft". During her student days she was a member of BEST, an international student organization where she developed her soft skills and especially her interest in marketing and design. Her major roles include coordinating different projects, both local and international, as well as being member of the board of the Local BEST Group Zagreb. Born in a bilingual surrounding her mother tongue are both Croatian and Italian, she speaks English fluently, German and Japanese passively.



Stjepan Bogdan received his Ph.D.E.E. in 1999, M.S.E.E. in 1993 and B.S.E.E. in 1990 at the University of Zagreb, Croatia. Currently he is an associate professor at the Faculty of Electrical Engineering and Computing, University of Zagreb. His main areas of interest are discrete event systems, intelligent control systems, and autonomous systems. He is a coauthor of three books and numerous papers published in journals and proceedings. He serves as associate editor of IEEE Transactions of Automation Science and Engineering, Journal of Intelligent and Robotic Systems, Transactions of the Institute of Measurement & Control and Journal of Control Theory and Applications.

AUTHORS' ADDRESSES

Matko Orsag, M.Sc.

Prof. Stjepan Bogdan, Ph.D.

Faculty of Electrical Engineering and Computing,

University of Zagreb,

Unska 3, HR-10000 Zagreb, Croatia

emails :matko.orsag@fer.hr, stjepan.bogdan@fer.hr

Marina Poropat, M.Sc.

Tehnozavod Marušić d.o.o.,

XIII Podbrežje 26, HR-10000 Zagreb, Croatia

email: marina.poropat@tehnouzavod.hr

Received: 2009-11-16

Accepted: 2010-01-24