International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Quadrotor Model Predictive Flight Control System

Abubakar Surajo Imam[Ȧ*] and Robert Bicker[Ȧ]

[Ȧ]School of Mechanical and System Engineering, Newcastle University, United Kingdom

Accepted 20 February 2014, Available online 25 February 2014, **Vol.4, No.1 (February 2014)**

Abstract

This paper presents a model predictive control (MPC) scheme for the autonomous flight control system that permits the quadrotor to track predefined bounded position and heading reference trajectories. The longitudinal and lateral velocities of the quadrotor are produced from the pitch and roll tilts of the vehicle, and therefore, the design of the flight control system feedback loops was based on this fact. The structure of the feedback law is composed of inner and outer controllers, which are responsible for lateral and longitudinal control of the quadrotor, and each of the controller utilizes the decomposed control signals comprising two feedback loops, namely, roll/pitch and yaw/altitude. In order to include the position dynamics of the quadrotor in the vehicle linear model, the control system design begins with the tracking problem of a reference translational velocity and heading profile. A flight test was conducted to verify the performance and effectiveness of the developed control system, and satisfactory results were obtained.

Keywords: *Model Predictive Control, Autonomous Flight Control System, Inner and Outer Controller, Roll, Pitch, Yaw, Feedback Loops.*

1. Introduction

Research and development in rotorcraft unmanned aerial vehicle (UAV) has increased in the last ten years due to its capabilities of vertical take-off and hover, thus making it idle candidate for employment into various military and civilian field operations. The military applications of rotorcraft UAVs include reconnaissance, extension of line of communication, stores delivery to troops in forward positions, minefield detection when equipped with necessary sensors and detection of gaps for advancing troops to mention a few. On the other hand, the civilian uses of rotorcraft UAVs include border patrol, disaster management, and traffic control in congested cities. Other area applications are detection and disposal of Improvise explosive devices (IEDs), surveillance of key positions or plants, such as oil and gas pipelines, power plants and ports as well as search and rescue operation. However, in recent times, interest has shifted in the research community to small-scale rotorcraft UAVs, in which the choice of a quadrotor is more prominent (Haomiao *et al*., 2009; Kis and Lantos, 2012; Ryll *et al*., 2012; Bellens *et al*., 2012).

A quadrotor is a small-scale multi-rotor (four rotors) rotorcraft in which lift and thrust are produced by the angular speeds of the rotors. A pair-diagonal-propellers spins clockwise and the other spins counterclockwise, and the proportionate regulation of the rotors' angular speeds results in the control of the vehicle's lateral and longitudinal positions and velocities.

A robust flight control systems enables a quadrotor to navigate and accomplish the desired tasks in a mission. However, the design of a reliable flight control system for a quadrotor is challenging mainly due to the inherent complexity of the vehicle's dynamics which is non-linear, underactuated and multivariable. Numerous studies have proposed a number of approaches upon which a quadrotor flight control system can be designed. In (Wang *et al*., 2013), an autopilot system for a quadrotor was developed by utilizing a digital signal processor (DSP) as the on-board micro-control unit (MCU). The flight control algorithm of the vehicle was developed based on the inner loop and outer loop control method. Proportional Derivative (PD) controllers are proposed for the attitude dynamics (inner loop) and position dynamics (outer loop) respectively. Autonomous hovering control was achieved via the control of roll angle, pitch angle, yaw angle and altitude of the UAV. In another study (Jiang *et al*., 2013), the dynamics of a quadrotor near hovering mode were analyzed based on Newton-Euler equation using the PID control algorithm, experiments conducted to verify the efficiency of the designed controller showed outstanding performance. A no-linear model and attitude robust control of a quadrotor MIMO system was described in (Torres and Bolea, 2013), where the dynamic model of the vehicle was transformed into a LTI system and approximated by a SISO system. The SISO system was employed to design a H∞ robust flight controller; the effectiveness and performance of the proposed control approach have been proved through simulation and experiment.

*Corresponding author: **Abubakar Surajo Imam**

Similarly, in (Sampaio *et al*., 2013), a novel Software-in-the-Loop (SiL) solution to evaluate controllers of all classes on the stability of MAVs using Microsoft Flight Simulator (MSFS) was presented, where H∞ robust controller was implemented on Ascending Technologies Pelican Micro Aerial Vehicle using MSFS. This paper presents the presents a systematic procedure for the design of a quadrotor flight controller based on the linear dynamic representation of the vehicle using two control approaches, namely, proportional derivative integral (PID) and model predictive control (MPC). The purpose of the flight control system is to enable the quadrotor to track predefined reference trajectories.

2. Quadrotor Linear Model

The quadrotor's perturbed state vector from the hover reference flight is given by Equation (1), where at hover, the trim values of linear and angular velocities zeros.

$$\dot{v}_B = w_B = u_B = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

In the above equations, it is evident that when the quadrotor operates around hover, its state is equal to the perturbed state vector about the reference operating point. The linear model proposed here has been successfully adopted for control applications on numerous rotorcraft platforms (Guowei *et al*., 2011) and (Budiyonoa and Wibowob, 2007). However, the linear model demonstrated in (Mettler, 2003) provides a generalized and physically meaningful solution to developing practical linear models for small-scale rotorcraft vehicles, where the numeric values of the variables are determined through the system identification procedure described in (Imam and Bicker, 2014). The proposed quadrotor flight control system is intended to be capable of both semi-autonomous and fully autonomous flight missions. The classical feedback control method (i.e., PD, PI or PID control) is one of the most common control schemes of choice because of its simplicity in structure with fewer requirements on the accuracy of the vehicle dynamic model. Although, a number of studies have reported the implementation of flight control systems based on various advanced control techniques, such as the H infinity (H∞) Boukhnifer *et al.* (2012), neural network (Nakanishi and Inoue, 2003) and (Nakanishi *et al*., 2002), linear quadratic regulator (LQR) (Reyes-Valeria *et al*., 2013), gain scheduling Sadeghzadeh *et al.* (2012) and (Yusong *et al*., 2008), model predictive (Alexis *et al*., 2012) and (Alexis *et al*., 2011), backstepping (Zheng and Gao, 2011) and (Madani and Benallegue, 2006), and adaptive control approach (Fu-Hong and Jung-Shan, 2011)] to mention a few. However, many of the reported implementations neglected some of the important aspect hindering the vehicle's performance, such as the influence of wind velocity variation and effects of airframe/rotors interaction. In the design of the model predictive control scheme, this study adopts the quadrotor demonstrated in (Imam and Bicker, 2014), which parameterized dynamic model incorporating the effects of wind velocity variation and mechanical structure interaction was derived using comprehensive identification from frequency responses was reported in (Imam and Bicker, 2014).

2.1 Architecture of the flight control system

The fact that the longitudinal and lateral velocities of the quadrotor are produced from the pitch and roll tilts of the vehicle was considered in the design of the flight control system feedback loops; this makes the vehicle velocity proportional to the vehicle attitude (Mettler, 2003). The structure of the feedback law is composed of two main loops; the inner and outer loops as shown in Fig. 1. The inner loop regulates the dynamics of the quadrotor associated with its Euler angles ϕ, θ and ψ, and corresponding angular velocities p, q, and r, whilst the outer loop controls the positions and velocities of the quadrotor relative to the Earth-fixed reference frame (i.e., X, Y, and Z, and their respective velocities u, v, and w). Generally, the dynamics associated with the outer-loop layer are much slower compared to those of the inner loop.

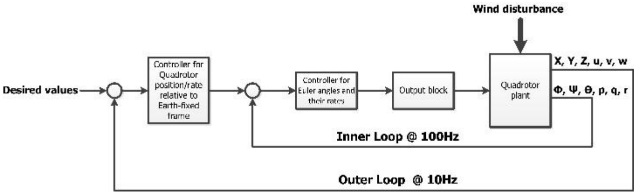


Fig. 1 Architecture of the flight control system.

It has been stated previously that the purpose of the flight control system design is to permit the quadrotor to track predefined bounded position and heading reference trajectories. However, in order to include the quadrotor position dynamics in the linear model, the control system design begins with the tracking problem of a reference translational velocity and heading profile. This is achieved by integration of the position tracking with the control problem. The initial output vector of interest of the quadrotor is:

$$\dot{y} = \begin{bmatrix} u & v & w & \psi \end{bmatrix}^T \tag{2}$$

The first design task is for the quadrotor to track the reference output:

$$\dot{y}_r = \begin{bmatrix} u_r & v_r & w_r & \psi_r \end{bmatrix}^T \tag{3}$$

The tracking problem requires the determination of the control signal *U(t)* as a function of the state variables of the vector *x(t)* and the reference output *y_r(t),* with its higher derivatives, such that:

$$\lim_{t \to \infty} \left\| y(t) - y_r(t) \right\| = 0 \tag{4}$$

While the quadrotor state *x(t)* and its control input *U(t)* remain bounded for any bounded reference output *y_r(t).* Although, typical for the tracking control problem is that not all the quadrotor states can be measured, therefore, only a subset of the state variables can be used by the controller for feedback purposes. In this case, only the vehicle motion state variables can be directly measured. Therefore, it is assumed that the following measurement vector is available for the vehicle:

$$Cx = \begin{bmatrix} u & v & w & p & q & r & \theta & \phi & \psi \end{bmatrix}^T \tag{5}$$

The tracking problem with output feedback for a linear system can be resolved using two approaches namely (i) tracking with integral control and (ii) tracking via the use of an internal model. In the internal model approach, the reference output signal is generated by a fixed reference dynamic system driven by a bounded input referred to as internal model, the structure of which is used by the controller to yield a dynamic feedback scheme. Typical application of such control design is met when the reference output is fixed or sinusoidal at a constant frequency (Khalil, 2002). This approach is robust but complex to implement. Details about the internal model approach can be found in (Isidori *et al*., 2003), (Xingyong *et al*., 2012), (Mohammadpour *et al*., 2013).

The use of integral control for the tracking problem results in the design of a robust dynamic feedback controller capable of providing a reliable and consistent solution when the desired output has constant values over time. However, in the case of a time varying output profile, the integral control design requires determination of a steady state response $x_{ss}(t)$ and a steady state control input $U_{ss}(t)$, such that when $y(t)$ tends to $y_r(t)$, the following holds:

$$\dot{x}_{ss} = Ax_{ss} + BU_{ss} \tag{6}$$

Determining x_{ss} and U_{ss} is a difficult task, which renders the integral control design impractical for the tracking problem of a time varying output. Further information about the integral control of linear systems can be found in (Yi-Rui and Yangmin, 2012), (Franklin *et al*., 2002).

To overcome the difficulties posed by the above standard methodologies, a simple tracking design approach is adopted, which is well suited to the problem under consideration. The first task involves determining a desired state vector x_d consisting of the components of the reference output vector y_r and their higher derivatives. Let $x_e = x_a - x_d$ represent the state error between the actual helicopter state and its desired value. The desired vector x_d should be chosen such that the following is satisfied:

$$\lim_{t\to\infty} \|e(t)\| = 0 \tag{7}$$

$$\lim_{t\to\infty} \|y(t) - y_r(t)\| = 0 \tag{8}$$

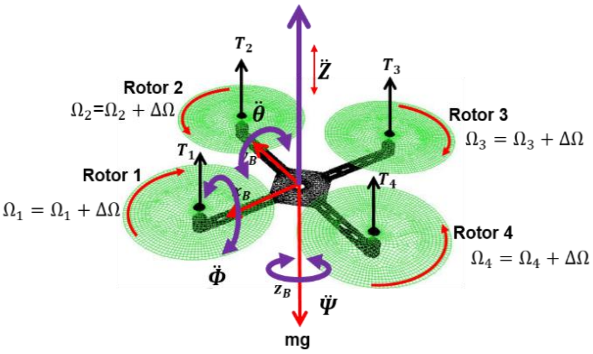
The proposed controller design provides a recursive methodology for the derivation of a desired state vector x_d and a desired control input U_d that satisfies Equations 6.6 and 6.7 as well as:

$$\dot{x}_d = Ax_d + BU_d \tag{9}$$

The role of the desired state vector x_d and the control input U_d is identical to that of the steady state vector x_{ss} and the input vector *Uss* required for the integral control methodology. Detail of this approach can be found in (Raptis and Valvanis, 2011).

2.2 Quadrotor Control Signal

A quadrotor is controlled by independently varying the angular speed of the four rotors driven by brushless DC electric motors. Each rotor produces a thrust and a torque whose combination generates the vehicle's control input. The control input comprises main thrust (u_1) roll moment (u_2) pitch moment (u_3) and yaw moment (u_4) acting on the vehicle as shown in Fig. 2 and Equations (10) – (13). In addition to manipulating pitch and roll moments, the control commands u_2 and u_3 also provides the means of achieving translation motions along the *x* and *y* axes. The throttle command u_1 controls the magnitude of the thrust of the four rotors producing the necessary lift force, while the yaw control command u_4 controls the heading of the vehicle. From the foregoing, the ideal solution is for each control command to be as independent as possible from one another. This implies designing four independent SISO feedback loops (such as PID) for each control input.



T_2
$\ddot{Z}$
T_3
Rotor 2
Ω_2=$\Omega_2 + \Delta\Omega$
$\ddot{\theta}$
Rotor 3
$\Omega_3 = \Omega_3 + \Delta\Omega$
T_1
y_B
Rotor 1
x_B
T_4
$\Omega_1 = \Omega_1 + \Delta\Omega$
Rotor 4
$\Omega_4 = \Omega_4 + \Delta\Omega$
$\ddot{\Phi}$
z_B
$\ddot{\Psi}$
mg

Fig. 2 The quadrotor's control command.

$$u_1 = T_1 + T_2 + T_3 + T_4 \tag{10}$$

$$u_2 = l(T_4 - T_2) \tag{11}$$

$$u_3 = l(T_1 - T_3) \tag{12}$$

$$u_4 = (\tau_2 + \tau_4 - \tau_1 - \tau_3) \tag{13}$$

However, since the system is a highly coupled nonlinear system, this approach cannot guarantee an efficient and stable solution.

2.3 Quadrotor Control Signal Decomposition

To overcome the problem stated above, the quadrotor dynamics can be separated in two interconnected subsystems to facilitate the design of a multiple-input-multiple-output (MIMO) MPC feedback loops. The use of MPC to control small-scale aerial vehicles has been reported in numerous studies (Alexis *et al*., 2011; Aswani, A. Bouffard and Tomlin, 2012; Alexis *et al*., 2010). In the

MPC design approach, the first subsystem accounts for the longitudinal and lateral motion, while the second subsystem represents the coupled yaw and vertical motion dynamics as described in the following subsections.

2.2 The Lateral-Longitudinal Subsystem

The lateral-longitudinal subsystem is given by:

$$\dot{x}_{23} = A_{23}x_{23} + B_{23}U_{23} \tag{14}$$

Where

$$\dot{x}_{23} = [u \quad v \quad \theta \quad \phi \quad p \quad q]^T \tag{15}$$

And

$$U_{23} = [u_2 \quad u_3]^T \tag{16}$$

$$A_{23} = \begin{bmatrix} 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & g & 0 & 0 \\ 0 & 0 & Y_\theta & 0 & 1 & 0 \\ 0 & 0 & 0 & Y_\phi & 0 & 1 \\ 0 & 0 & L_\theta & L_\phi & 0 & 0 \\ 0 & 0 & N_\theta & N_\phi & 0 & 0 \end{bmatrix} \tag{17}$$

$$B_{23} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ L_{u_2} & L_{u_3} \\ N_{u_2} & N_{u_3} \end{bmatrix} \tag{18}$$

2.4 The Yaw-Vertical Subsystem

The yaw-vertical subsystem is given by:

$$\dot{x}_{14} = A_{14}x_{14} + B_{14}U_{14} \tag{19}$$

Where

$$\dot{x}_{14} = [w \quad \psi \quad r]^T \tag{20}$$

$$U_{14} = [u_1 \quad u_4]^T \tag{21}$$

$$A_{14} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & L_\psi & L_{u_4} \\ 0 & N_\psi & -g \end{bmatrix} \tag{22}$$

$$B_{14} = \begin{bmatrix} 0 & 0 \\ L_{u_1} & L_{u_4} \\ N_{u_1} & N_{u_4} \end{bmatrix} \tag{23}$$

By this decomposition, the design of the flight control system is reduced structurally to two separate feedback loops for each of two subsystems. This approach results in a mathematically consistent and systematic methodology for the development of a robust and efficient multivariable flight control system for a quadrotor.

3. Position and Heading Tracking

A requirement of the quadrotor flight control system design is to track a predefined position trajectory of the Earth-fixed frame expressed by the reference vector:

$$P_r^E = [P_{rx}^E \quad P_{ry}^E \quad P_{rz}^E]^T \tag{24}$$

The coordinate vector denotes the quadrotor's position expressed in the body-fixed frame is given by:

$$P_r^E = [P_x^B \quad P_y^B \quad P_y^B]^T \tag{25}$$

The position error expressed in the body-fixed frame is given by:

$$e_p^B = P^B - P_r^B \tag{26}$$

The position error dynamics are derived using the properties of the rotation matrix **R**, described in (Bouabdallah, 2007). The rotation matrix is used for mapping coordinate vectors from the body-fixed frame to the Earth-frame. The following expresses the position error in the body-fixed frame:

$$e_p^B = P^B - P_r^B = R^T P^E - R^T P_r^E \tag{27}$$

Finally, given a reference position vector in Equation (13) with respect to the Earth-fixed frame, the desired velocities values $v_d^B = [u_d \quad v_d \quad w_d]^T$ in the body-fixed frame are given by:

$$v_d^B = R^T \dot{P}_r^B \tag{28}$$

$$v_d^B = v_r^B \tag{29}$$

4. Control System Design

This section describes the multiple-input-multiple-output (MIMO) control of the quadrotor using the model predictive control (MPC) scheme, which is a model-based multivariable control technique suitable for systems with input-output constraints. The MPC technique is widely used to deal with large multivariable constrained control problems on systems and processes. The main aim of MPC is to minimize a performance criterion in the future that would possibly be subject to constraints on the manipulated inputs and outputs, where the future behaviour is computed according to a model of the plant.

4.1 Description of the Model Predictive Control Scheme

The MPC uses the mathematical expressions of a system model to predict its behaviour. The prediction is used to optimize the process over a defined time period. An MPC controller can operate according to the following algorithm: (i) Development of a system model, (ii) At time t, previous system inputs and outputs are used, along with the system model, to predict future system outputs u(f) over a prediction horizon, (iii) The control signals that produce the most desired behaviour are selected, (iv) The control signal is implemented over a predefined time interval and (v) Time advances to the next interval, and the procedure is repeated from step 2.

Figure 3 depicts the basic structure of an MPC. The model takes data from past inputs and outputs; it combines the data with the predicted future inputs to give the predicted output for the time step. This predicted output is combined with the reference trajectory, giving the predicted future errors of the system. These errors are fed into an optimizer, which enforces the constraints of the system (for instance, ensuring that a pitch angle for the quadrotor is not greater than the defined maximum) on the predicted outputs and minimizes the operating cost function. This gives the predicted future inputs, which are fed back into the main model, restarting the cycle.

The general design objective of a model predictive control for the quadrotor in this study is to compute a trajectory of a future manipulated variable *u* to optimize the future behaviour of the vehicle output *y*. The optimization is performed within a limited time window by giving the vehicle's information at the start of the time window.

Assuming a flight mission is defined where it is required to maintain the quadrotor at hover, at a defined altitude for a defined period, with the flight mission divided into vehicle warm-up, ascend to hover height, maintaining the hover height for the defined period and landing. Completing this mission will be a function of factors, such as ascend rate, hover height, vehicle's maximum attitude angles during the hover period, and the hover time. These are the manipulated variables for the controller. The limitations associated with the mission include the presence of wind disturbance and available battery power.

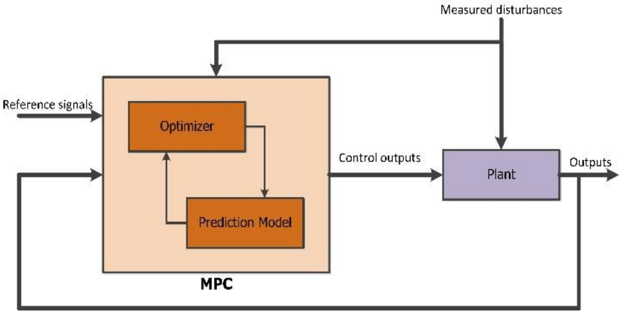


Fig. 3 MPC basic structure.

These are the hard and soft constraints in the planning. This information is sufficient for the design of an MPC for the quadrotor. Suppose the whole mission is expected to last for five minutes, then the controller design task is determined as a function of the manipulated variables for the mission period. If the rule is that always, the MPC algorithm plans the activities for accomplishing the mission for the next sixty seconds, but only implements the plan for the first second. This planning activity is repeated every second until the mission is completed. The MPC algorithm considers the mission limitations (as constraints), and find the best way to achieve the goal. At every second, the control algorithm will review the mission and redesign the remaining tasks until the mission is accomplished.

5. MPC Implementation

The design of the flight control system utilizes the decomposed control signals described by Equations (14-23) and is composed of two feedback loops, namely, roll/pitch and yaw/altitude for the inner and outer controllers. In the case of the outer controller (Fig. 4), the roll/pitch feedback loop compensates for the discrepancies between the vehicle actual roll and pitch attitudes obtained from the on-board IMU with the referenced attitudes obtained through the manual source (a joystick) or the on-board autonomous controller.

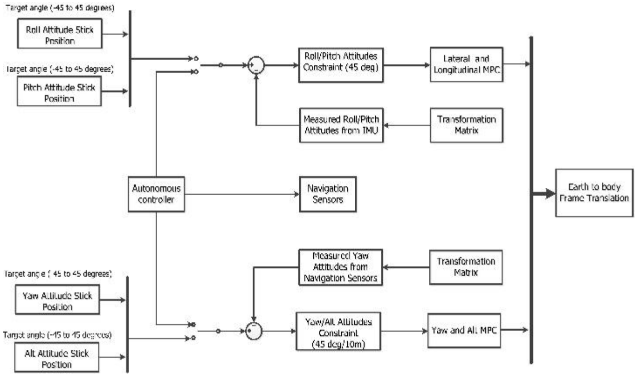

Target angle (-45 to 45 degrees)
Roll Attitude Stick Position
Target angle (-45 to 45 degrees)
Pitch Attitude Stick Position
Roll/Pitch Attitudes Constraint (45 deg)
Lateral and Longitudinal MPC
Measured Roll/Pitch Attitudes from IMU
Transformation Matrix
Autonomous controller
Navigation Sensors
Earth to body Frame Translation
Measured Yaw Attitudes from Navigation Sensors
Transformation Matrix
Target angle (-45 to 45 degrees)
Yaw Attitude Stick Position
Target angle (-45 to 45 degrees)
Alt Attitude Stick Position
Yaw/Alt Attitudes Constraint (45 deg/10m)
Yaw and Alt MPC

Fig. 4 MPC outer controller feedback loops.

The yaw/altitude feedback loop compensates for the vehicle yaw attitude and altitude obtained from the on-board navigation sensors. The two cases involve transformation of the measured quantities from the body-fixed reference frame to the Earth-fixed reference frame. The output of the outer controller then becomes the input of the inner controller. Since the design of the control system is based on the parameterized model of the quadrotor demonstrated in (Imam and Bicker, 2014), where the values of the matrices A, B and C were found through system identification procedure, inserting these values into Equations (14) and (15) yield:

$$A_{23} = \begin{bmatrix} 0 & 0 & -9.81 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.81 & 0 & 0 \\ 0 & 0 & -2.759 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2.452 & 0 & 1 \\ 0 & 0 & -4.407e+4 & -3.871e+6 & 0 & 0 \\ 0 & 0 & -1.308e-8 & 1.616e-4 & 0 & 0 \end{bmatrix} \qquad (30)$$

$$B_{23} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 3.594e+4 & 1.952e+4 \\ 1.327e+4 & 1.347e+4 \end{bmatrix} \tag{31}$$

Similar to the outer controller feedback loops, the inner controller feedback loops (Fig. 5), compensate for the attitude and altitude rates of the vehicle by comparing the measured rates obtained from the navigation sensors with the commanded inputs from the outer controller. The comparison happens after the necessary transformation of the measured quantities between the two frames of reference.

This is based on the MPC control technique comprising two outer and two inner controllers. The outer and inner controllers regulate the vehicle's position and position rate respectively. The four controllers include outer and inner roll/pitch attitude controllers and outer and inner yaw/altitude controllers as detailed in the following subsections.

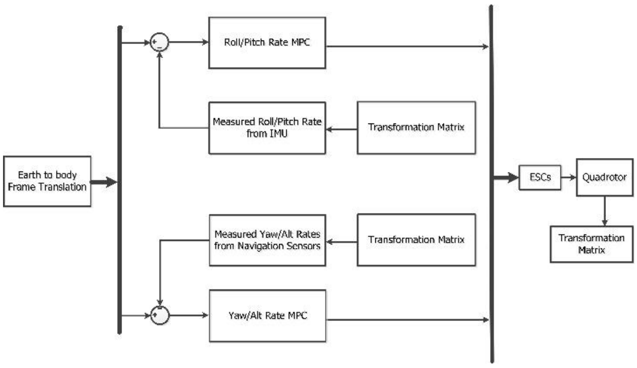


Fig. 5 MPC inner controller feedback loops.

6. Arduino-Simulink Blockset

The Arduino-Simulink (APM2) blockset was developed at Embry-Riddle Aeronautical University (Embry-Riddle University, 2014) to facilitate the development of Matlab/Simulink-based small-scale aerial vehicles Guidance, Navigation and Control (GNC) system. The code generated from the GNC design can be ported directly to the flight control board, thus allowing easy development of small-scale UAVs. The blockset (Hartley, 2012), uses the Run-On-Target-Hardware feature available in Matlab (from version 2012a upward). The feature allows auto-generation of code from a Simulink model to hardware targets. Hardware targets supported by this feature include PIC, ARM and Atmega family of microcontrollers. The hardware target (on-board flight controller board) used in this study was the Arduino-based microcontroller board, which integrates a sensor suite with an Atmega 2560 processor. The Simulink model to be developed for the quadrotor will be embedded directly to the on-board flight controller board. The model includes the necessary codes to read the sensors, read the RC PWM signals from the RC transmitter, output PWM signals to control the rotors of the quadrotor, output data to a telemetry system, and record data to a flash memory chip.

6.1 Flight Controller Realization

As mentioned earlier, the vehicle's control signal comprises four PWM signals, which can be from the on-board autonomous controller or via a wireless joystick. Therefore, the vehicle response to the input reference commands can be simulated by channeling the input reference commands via the RC Read block with its outputs connected to the RC Write block as shown in the Simulink model depicted in Fig. 6. However, embedding the model to the vehicle's on-board flight control board, would process the input commands and output the equivalent PWM throttle commands to the vehicle's rotors, thus operating the quadrotor in manual mode (i.e., without the controller’s action).

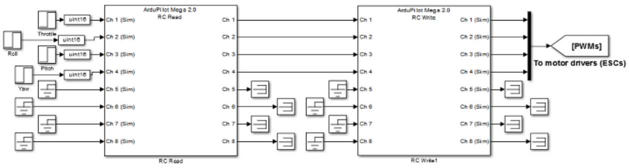


Fig. 6 Simulink model of the vehicle manual control

The MPC scheme comprises a total of four control feedback loops; two outer and two inner controllers, which regulate the vehicle's positions, velocities and their rates respectively. The four controller feedback loops include outer and inner roll/pitch attitude controllers and outer and inner yaw/altitude controllers as detailed in the following subsections.

6.2 Outer Roll/Pitch attitude MPC

The schematic shown in Fig. 7 depicts the MPC controller which regulates the vehicle's coupled angular positions relative to the *x* and *y* axes (roll and pitch angles), consisting of two inputs reference signals.

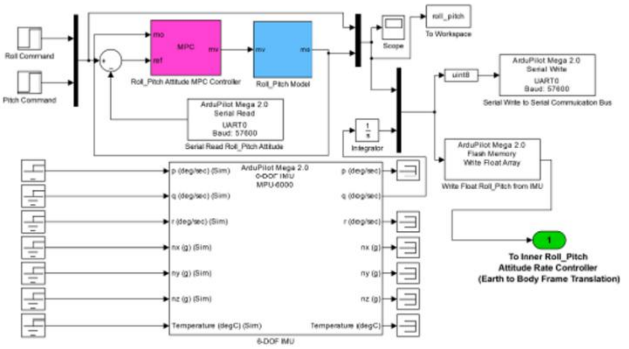

Roll Command
Pitch Command
mo
ref
MPC
mv
Roll_Pitch Attitude MPC Controller
mv
mo
Roll_Pitch Model
ArduPilot Mega 2.0 Serial Read UART0 Baud: 57600
Serial Read Roll_Pitch Attitude
Scope
roll_pitch
To Workspace
uint8
ArduPilot Mega 2.0 Serial Write UART0 Baud: 57600
Serial Write to Serial Commuication Bus
1/s
Integrator
ArduPilot Mega 2.0 Flash Memory Write Float Array
Write Float Roll_Pitch from IMU
ArduPilot Mega 2.0 6-DOF IMU MPU-6000
p (deg/sec) (Sim)
q (deg/sec) (Sim)
r (deg/sec) (Sim)
nx (g) (Sim)
ny (g) (Sim)
nz (g) (Sim)
Temperature (degC) (Sim)
p (deg/sec)
q (deg/sec)
r (deg/sec)
nx (g)
ny (g)
nz (g)
Temperature (degC)
6-DOF IMU
1
To Inner Roll_Pitch Attitude Rate Controller (Earth to Body Frame Translation)

Fig. 7 Simulink model for the outer roll/pitch MPC.

The reference signals are the roll and pitch commands (PWMs in milliseconds), which vary from 0-*100%* and centred at *50%* (neutral position). A PWM duty cycle of *50-100%* results in positive roll/pitch angle and *50-0%* represent a negative roll/pitch angle. The measured roll and pitch rates from IMU are integrated to get the respective positions. These quantities are then stored to the embedded controller's Flash Rom and placed on the serial communication bus. The controller regulates the vehicle's roll and pitch angles by comparing the requested angular positions with the measured angular positions and taking into consideration the coupling between the two axes. However, a constraint has been place to limit the vehicle's rolling and pitching to ±*45* degrees. The output of this controller goes to the inner roll/pitch attitude controller.

6.3 Inner Roll/Pitch attitude MPC

The Simulink model of the inner roll/pitch attitude controller or roll and pitch rates attitude MPC (Fig. 8) is similar to the roll/pitch angular position controller except this regulates the vehicle's roll and pitch rates instead of the roll/pitch angles. The IMU measured roll and pitch rates are stored to the embedded controller's Flash ROM and placed on the serial communication bus with no transformation. The controller regulates the vehicle's roll and pitch angular rates by comparing the reference inputs from the outer roll/pitch angular positions controller with the measured roll and pitch rates. The output of this controller drives the appropriate DC motors on the vehicle via the ECSs.

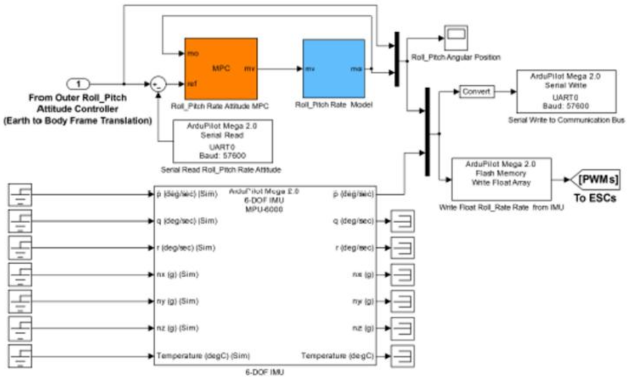

From Outer Roll_Pitch Attitude Controller (Earth to Body Frame Translation)
MPC
Roll_Pitch Rate Attitude MPC
Roll_Pitch Rate Model
Roll_Pitch Angular Position
ArduPilot Mega 2.0 Serial Write UART0 Baud: 57600
Serial Write to Communication Bus
Convert
ArduPilot Mega 2.0 Serial Read UART0 Baud: 57600
Serial Read Roll_Pitch Rate Attitude
ArduPilot Mega 2.0 Flash Memory Write Float Array
Write Float Roll_Rate Rate from IMU
[PWMs]
To ESCs
ArduPilot Mega 2.0 6-DOF IMU MPU-6000
6-DOF IMU

Fig. 8 Simulink for the inner the outer roll/pitch MPC

6.4 Outer altitude/yaw attitude MPC

Figure 9 depicts the Simulink model of the outer altitude/yaw attitude MPC controller, which regulates the vehicle's translational and angular positions along and relative to the z-axis respectively. It consists of two input command signal channels and a sensor suite which comprises 3-axis magnetometer, IMU, GPS, barometer and pitot probe. The reference signals channel are the altitude and yaw commands. On the altitude channel, *0%* command represents zero PWM duty cycle and *100%* represents a full PWM duty cycle. The measured yaw angular rate from the IMU is integrated to get the yaw angular position, this quantity together with the measured altitude from the navigation sensors are stored to the embedded controller's Flash Rom and placed on the serial communication bus. The controller regulates the vehicle's altitude and yaw angle by comparing the requested translational and yaw attitude positions with the measured ones, fully taking into consideration the coupling between the two axes. However, the vehicle is constrained to yaw angle of ±*45* and an altitude of 10m

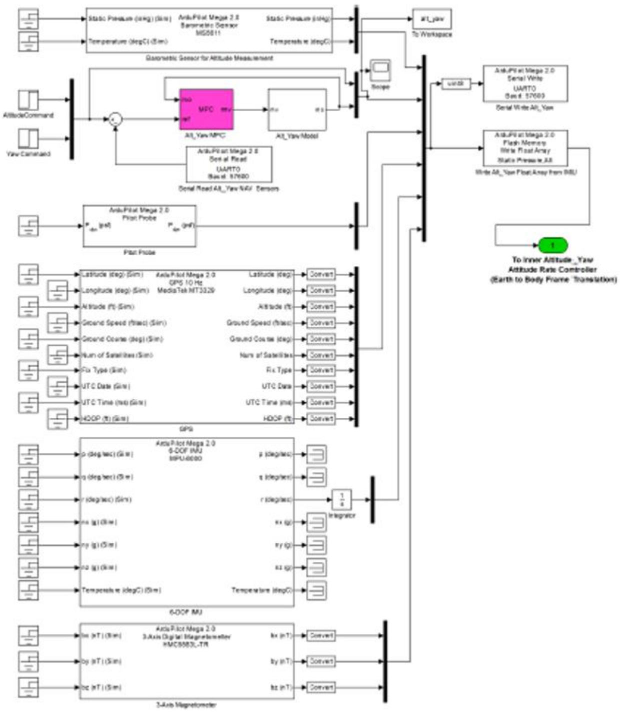

Barometric Sensor for Altitude Measurement
To Workspace
Scope
AltitudeCommand
Yaw Command
MPC
Alt_Yaw MPC
Alt_Yaw Model
Serial Read Alt_Yaw NAV Sensors
Pitot Probe
GPS
6-DOF IMU
3-Axis Magnetometer
To Inner Altitude_Yaw Attitude Rate Controller (Earth to Body Frame Translation)

Fig. 9 Simulink model for the outer altitude/yaw attitude MPC.

6.5 Inner Altitude/Yaw Attitude MPC

The Simulink model of the inner altitude/yaw attitude MPC is depicted in Fig. 10, which controls the vehicle's rate of climb and yaw angular rate.

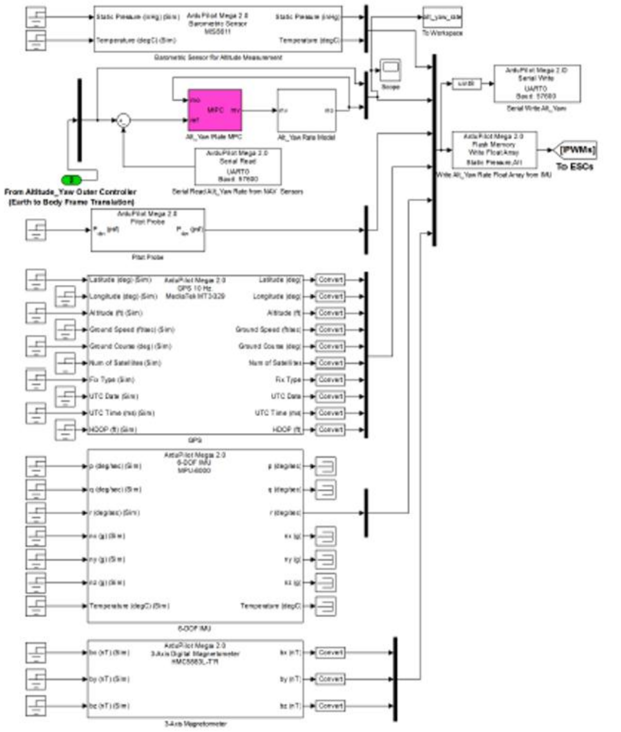

Barometric Sensor for Altitude Measurement
To Workspace
Scope
MPC
Alt_Yaw Rate MPC
Alt_Yaw Rate Model
From Altitude_Yaw Outer Controller (Earth to Body Frame Translation)
[PWMs]
To ESCs
Pitot Probe
GPS
6-DOF IMU
3-Axis Magnetometer

Fig. 10 Simulink model for the inner altitude/yaw attitude MPC.

It comprises the coupled throttle and yaw input command channels, IMU, magnetometer, pitot probe, sonar and barometric pressure sensor. The measured height of the vehicle from the altitude sensors together with the measured yaw rate from the navigation sensors are stored to the embedded controller's Flash Rom and placed on the serial communication bus. The controller regulates the vehicle's ascend/descend rates and yaw angular rate by comparing the reference inputs from the outer altitude/yaw attitude controller with the measured quantities. The output of this controller drives the appropriate DC motors on the vehicle via the ECSs.

6.6 Controller Response to Step Input

A step input function has been used to evaluate the performance of the MPC-based quadrotor flight control system. Figure 11 depicts the response of the altitude and yaw controllers to the step input, which yields a small transient overshoot on the altitude channel with a smaller rise time, a negligible steady state error and short settling time (of about *4.5s*). The response of the yaw MPC controller to a step input function yields a rise time of about *0.4s* with no overshoot.

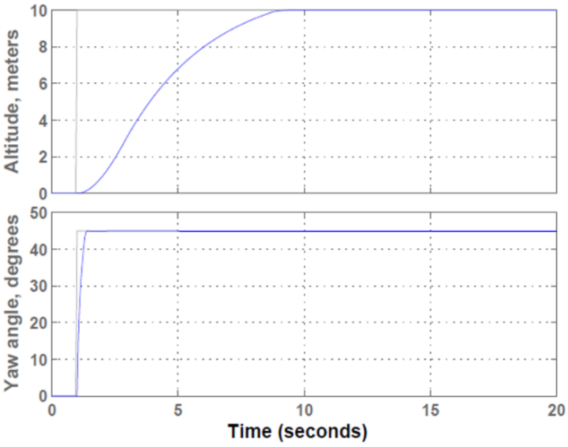


Fig. 11 Response of the altitude/yaw MPC to step input.

However, a step response to the roll/pitch MPC (Fig. 12) shows no overshoot, zero steady-state error, and faster system response on both channels.

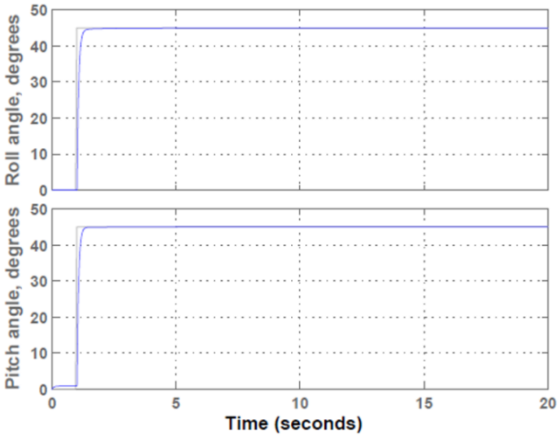


Fig. 12 Response of the roll/pitch MPC to step input.

7. Implementation and Evaluation

An experiment was conducted involving a real flight test using the quadrotor shown in Fig. 13 to evaluate the performance of the flight algorithm developed in this study within the vehicle’s operational envelope. The test involves generation of appropriate flight trajectories for two flight manoeuvres namely forward flight and hover turn. Each flight manoeuvre is defined in terms of a clear objective, full description and performance requirement, where performance requirements were categorized into two qualitative levels, i.e., the desired level (satisfactory) and the adequate level (barely acceptable).

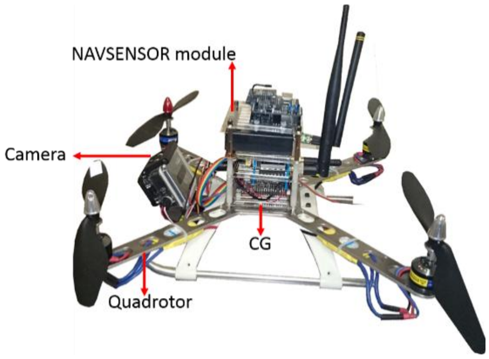


Fig. 13 Experiment platform.

7.1 Forward Flight Manoeuvre

The forward flight is a flight operation with moderate aggressiveness. The objective of this test is mainly to examine the vehicle in four aspects, which include:

(i) Testing handling quality and control performance of pitch and heave axes in the transient process,
(ii) Verifying for the existence of any undesirable coupling between the longitudinal and lateral directions,
(iii) Obtaining the working performance in the condition with the predefined maximum forward speed, and lastly,
(iv) Verifying the ability to re-establish automatic hover after the forward flight.

This manoeuvre will cause the front of the vehicle to pitch down which allows it to move horizontally with a resultant increase in airspeed and loss of altitude. Therefore, it is very important to ensure that current altitude is maintained. The manoeuvre starts from a hover condition at an altitude with good eyesight (ranging from *8-10 m*). The flight path is generally a straight line, with its destination point *150 m* away from the starting position. The overall procedure consists of the following three stages: (i) performing an accelerating departure until the predefined top forward speed (5 m/s) is achieved in 30 s, (ii) maintaining the forward flight at at top speed of speed 5 m/s and (iii) aborting the forward flight and decelerating to hover at the destination point for 10 s. Both the acceleration and deceleration manoeuvres should be accomplished smoothly, overshoot is not allowed. Table 1 depicts the performance requirements for this manoeuvre.

Table 1 Forward flight performance requirements

Specification	Desired level	Adequate level
Completion time	≤ 30 s	30-40 s
Altitude error	≤ 2 m	2-4 m
Heading maintaining range	$\leq 4^0$	$4\text{-}6^0$

7.2 Hover Turn Manoeuvre

The objective of this manoeuvre is to verify the existence of any undesirable handling qualities or inter-axis coupling in the hovering turn and to test the precision of heading maintenance after recovering from a hovering turn. The manoeuvre starts with the quadrotor at a hover state, and is required to complete a 90-degree heading turn either clockwise or counter-clockwise and re-achieve a stable hover at the same point where it started. Table 1 depicts the performance requirements for this manoeuvre.

Table 2 Hover turn performance requirements

Specification	Desired level	Adequate level
Lateral position error	≤ 1.5 m	1.5-3 m
Longitudinal position error	≤ 1.5 m	1.5-3 m
Heading error	$\leq 5^0$	$5\text{-}10^0$
Completion time	≤ 2 s	2-3 s

8. Flight Test Results

This section presents the result of the flight test results based on the flight scheduling defined in the previous section. The test was conducted outdoors in the situation with strong wind gusts (about 4 to 5 m/s in the horizontal plane, roughly measured with an anemometer), this coincides with the maximum wind velocity upon which the flight control system design was based.

8.1 Forward Flight Test Result

The purpose of this test was to evaluate the vehicle's forward flight capabilities in terms of the manoeuvre completion time, altitude error, heading maintaining range, longitudinal position maintenance range as defined in Table 1. The manoeuvre started with the quadrotor at home position (zero meter along y-axis), after 20 s, the vehicle was at 100 m away from the home position, and after 30 s, the vehicle was at 145 m (5 m less the target position) away from the starting position.

Table 3 Forward flight performance evaluation result

Specification	Desired level	Actual test
Completion time	≤ 35 s	28 s
Altitude error	≤ 2 m	1 m
Heading maintaining range	$\leq 4^0$	2.5^0

Satisfactory results were obtained for this manoeuvre; with the altitude error, less than 1 m, heading maintaining error less than 2.5 degree, and the manoeuvre was completed in 28 s as illustrated in Table 3 and Fig. 14. The plots in Fig. 14 show excellent tracking of the reference signal by the altitude, heading and longitudinal position controllers.

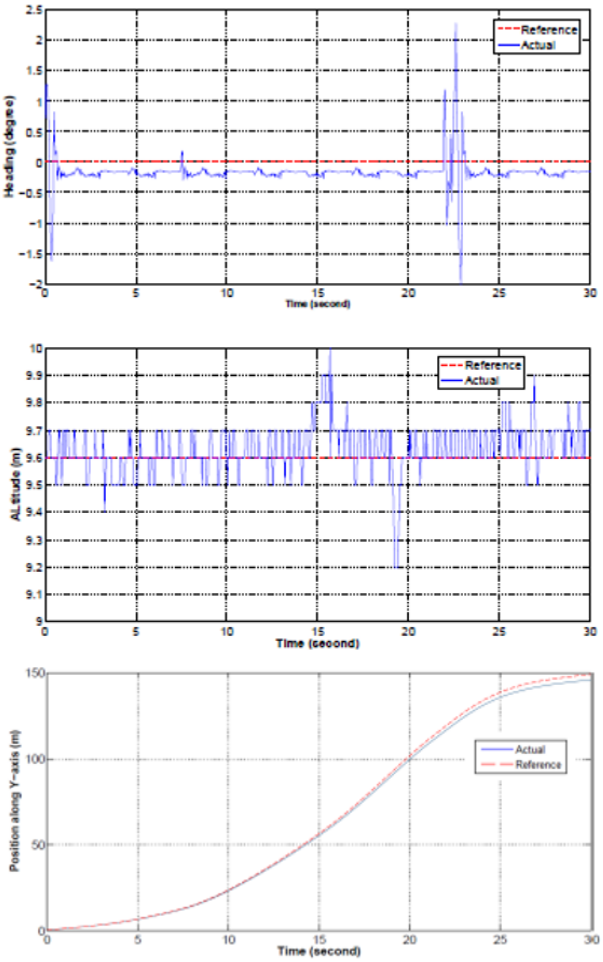


Fig. 14 Forward flight manoeuvre

8.2 Hover Turn Test Result

The purpose of this test was to evaluate the vehicle's capabilities in terms of the manoeuvre completion time, altitude error, heading maintaining range, longitudinal and lateral errors as defined in Table 2.

Table 4 Hover turn performance evaluation result

Specification	Desired level	Adequate level
Lateral position error	≤ 3 m	< 1.5 m
Longitudinal position error	≤ 3 m	< 1.5 m
Heading error	$\leq 10^0$	$< 3^0$
Completion time	≤ 3 s	2 s

The result of this manoeuvre is shown in Fig. 8.7, where at the beginning of the manoeuvre, the quadrotor was in hover flight condition at home position (lateral and longitudinal positions zero each) and having neutral heading (zero degree). The manoeuvre started with a forward flight along the longitudinal axis (y-axis) for 1 s, followed by a 90 degree heading turn for 0.3 s, and a forward flight for the remainder of the manoeuvre time. At the beginning of the 90-degree heading turn, the vehicle was at around 8 m along the longitudinal axis and zero meter on the lateral position. However, at the end of the 90 degree heading turn (i.e., at approximately 1.3 s), the vehicle maintained a 9 m position along the longitudinal axis, and moved from zero meter to 7 m position along the lateral axis for the remainder of the manoeuvre time. At the end of the manoeuvre, the results obtained were satisfactory (Table 4), with the heading maintaining error less than 3 degree, longitudinal and lateral positions errors both less than 1.5 m, and the manoeuvre was completed in 2 s. The plots in Fig. 15 show excellent tracking of the reference signal by the heading, lateral and longitudinal positions controllers.

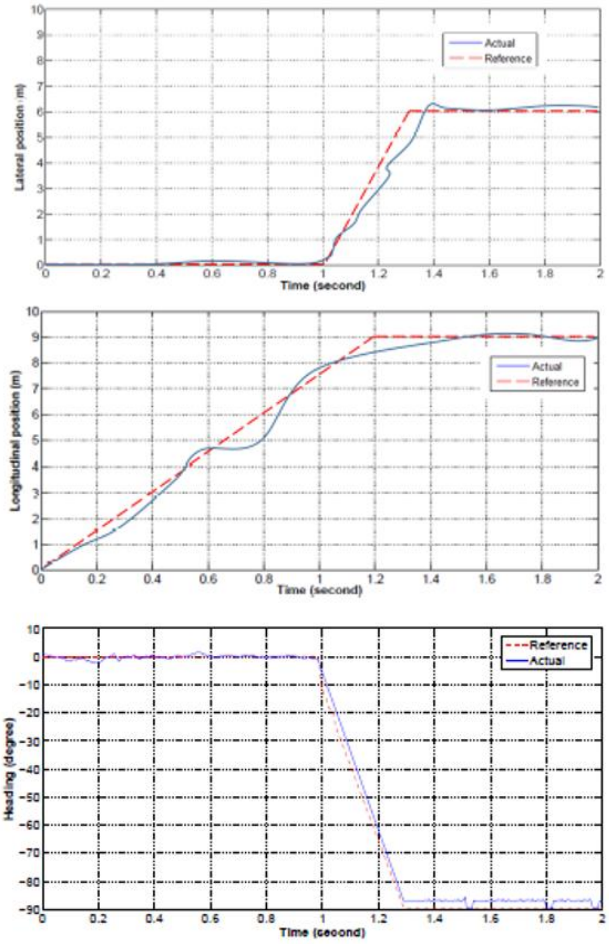


Fig. 15 Hover turn flight manoeuvre.

Conclusions

This paper has the design of the altitude and attitude regulating flight controllers for a quadrotor based on PID and model predictive control techniques using the Matlab/Simulink tools and Arduino-Simulink Blockset. The purpose of the flight control system design is to permit the quadrotor track predefined bounded position and heading reference trajectories. The performance of the controllers has been evaluated by simulation using a step input function. The responses of the PID controllers to a step input have zero steady-state error, relatively fast response but with some mild overshoots on the altitude and yaw attitude responses. Similarly, model predictive controllers' response to step input have no overshoot with nearly zero steady state error and relatively fast except for the altitude. Generally, the model predictive control scheme indicates better performance compared to the PID control action. On the overall, the simulation results show that the quadrotor can achieve altitude and attitude stabilization using either of the control techniques.

More so, the results of the flight test conducted to evaluate the performance of the quadrotor flight control system. The flight test was based on a defined flight schedule which comprised forward flight and hover turn, and. The results obtained from the test for each of the manoeuvre were satisfactory.

References

Alexis, K., Nikolakopoulos, G. & Tzes, A. (2010), Experimental model predictive attitude tracking control of a quadrotor helicopter subject to wind-gusts, in 'proceedings of the 18th Mediterranean Conference on Control Automation (MED), 2010', pp. 1461–1466.

Alexis, K., Nikolakopoulos, G. & Tzes, A. (2012), 'Model predictive quadrotor control: attitude, altitude and position experimental studies', Transactions on IET Control Theory Applications 6(12), 1812–1827.

Alexis, K., Papachristos, C., Nikolakopoulos, G. & Tzes, A. (2011), Model predictive quadrotor indoor position control, in 'proceedings of the 19th Mediterranean Conference on Control Automation (MED), 2011', pp. 1247–1252.

Aswani, A. Bouffard, P. & Tomlin, C. (2012), Extensions of learning-based model predictive control for real-time application to a quadrotor helicopter, in 'proceedings of the American Control Conference (ACC), 2012', pp. 4661–4666.

Bellens, S., Schutter, J. D. & Bruyninckx, H. (2012), A hybrid pose / wrench control framework for quadrotor helicopters, in 'proceedings of the IEEE International Conference on Robotics and Automation (ICRA), (Saint Paul, Minnesota, USA), pp. 2266-2274'.

Bouabdallah, S. (2007), Design and Control of Quadrotors with Application to Autonomous Flying, PhD thesis, Ecole Polytechnique Federal de Lausanne.

Boukhnifer, M., Chaibet, A. & Larouci, C. (2012), H-infinity robust control of 3-dof helicopter, in 'proceedings of the 9th International Multi-Conference on Systems, Signals and Devices (SSD), 2012', pp. 1–6.

Budiyonoa, A. & Wibowob, S. S. (2007), 'Optimal tracking controller design for a small scale helicopter', Journal of Bionic Engineering 4, 271–280.

Embry-Riddle University (2014), http://www.erau.edu/.

Franklin, G. F., Powell, J. D. & Emami, N. A. (2002), Feedback Control of Dynamic Systems Prentice Hall, New York.

Fu-Hong, J. & Jung-Shan, L. (2011), Nonlinear and adaptive control design of a helicopter in vertical flight, in 'proceedings of the 8th Asian Control Conference (ASCC), 2011', pp. 217–222.

Guowei, C., Wang, B., Chen, B. M. & Lee, T. H. (2011), Design and implementation of a flight control system for an unmanned rotorcraft using rpt control approach, in 'proceedings of the 30th Chinese Control Conference (CCC), 2011', pp. 6492–6497.

Haomiao, H., Hoffmann, G. M., Waslander, S. L. & Tomlin, C. J. (2009), Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering, in 'proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'.

Hatley, R. (2012), 'Apm2 simulink blockset', http://www.mathworks.co.uk/matlabcentral/fileexchange/apm2-simulink-blockset.

Imam, A. S. & Bicker, R. (2014a), 'Design and construction of a small-scale rotorcraft uav system', International Journal of

Engineering Science and Innovative Technology (IJESIT) 3, 96–109.

Imam, A. S. & Bicker, R. (2014b), ‘Quadrotor comprehensive identification from frequency responses’, International Journal of Scientific and Engineering Research (IJSER) 5, in Press.

Isidori, A., Marconi, L. & Serrani, A. (2003), Robust Autonomous Guidance, Springer, Berlin.

Jiang, J., Qi, J., Song, D. & Han, J. (2013), Control platform design and experiment of a quadrotor, in ‘proceedings of the 32nd Chinese Control Conference (CCC), 2013’, pp. 2974–2979.

Khalil, H. K. (2002), Nonlinear Systems, Prentice Hall, New York.

Kis, L. & Lantos, B. (2012), Time-delay extended state estimation and control of a quadrotor helicopter, in ‘proceedings of the 20th Mediterranean Conference on Control Automation (MED), 2012’, pp. 1560–1565.

Mettler, B. (2003), Identification Modeling and Characteristics of Miniature Rotorcraft, Kluwer Academic Publishers, Norwel.

Mohammadpour, J., Sun, J., Karnik, A. & Jankovic, M. (2013), Internal model control design for linear parameter varying systems, in ‘proceedings of the American Control Conference (ACC), 2013’, pp. 2409–2414.

Nakanishi, H., Hashimoto, H., Hosokawa, N., Sato, A. & Inoue, K. (2002), Autonomous flight control system for unmanned helicopter using neural networks, in ‘proceedings of the 41st SICE Annual Conference, 2001’, Vol. 2, pp. 777–782.

Nakanishi, H. & Inoue, K. (2003), Development of autonomous flight control system for unmanned helicopter by use of neural networks, in ‘proceedings of the International Joint Conference on Neural Networks, 2003’, Vol. 3, pp. 2400–2405.

Raptis, I. A. & Valvanis, K. P. (2011), Linear and Nonlinear Control of Small-Scale Unmanned Helicopters, Springer.

Reyes-Valeria, E., Enriquez-Caldera, R., Camacho-Lara, S. & Guichard, J. (2013), Lqr control for a quadrotor using unit quaternions: Modeling and simulation, in ‘proceedings of the International Conference on Electronics, Communications and Computing (CONIELECOMP), 2013’, pp. 172–178.

Ryll, M., Bulthoff, H. H. & Giordano, P. R. (2012), Modeling and control of a quadrotor uav with tilting propellers, in ‘proceedings of the IEEE International Conference on Robotics and Automation (ICRA), (Saint Paul, Minnesota, USA), pp. 4606-4613’.

Sadeghzadeh, I., Mehta, A., Chamseddine, A. & Youmin, Z. (2012), Active fault tolerant control of a quadrotor uav based on gainscheduled pid control, in ‘proceedings of the 25th IEEE Canadian Conference on Electrical Computer Engineering (CCECE), 2012’, pp. 1–4.

Sampaio, R. C. B., Becker, M., Siqueira, A. A. G., Freschi, L. W. & Montanher, M. P. (2013), Novel sil evaluation of an optimal h infinity controller on the stability of a mav in flight simulator, in ‘Aerospace Conference, 2013 IEEE’, pp. 1–8.

Torres, G. A. & Bolea, Y. (2013), Modeling and robust attitude control of a quadrotor system, in ‘proceedings of the 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2013’, pp. 7–12.

Wang, F., Xian, B., Huang, G. & Zhao, B. (2013), Autonomous hovering control for a quadrotor unmanned aerial vehicle, in ‘proceedings of the Control Conference (CCC), 2013’, pp. 620–625.

Xingyong, S., Yu, W. & Zongxuan, S. (2012), Robust stabilizer design for linear time varying internal model based control, in ‘proceedings of the American Control Conference (ACC), 2012’, pp. 6727–6732.

Yi-Rui, T. & Yangmin, L. (2012), Design of an optimal flight control system with integral augmented compensator for a nonlinear uav helicopter, in ‘proceedings of the 10th World Congress on Intelligent Control and Automation (WCICA), 2012’, pp. 3927– 3932.

Yusong, J., Xinmin, W. & Xiang, Y. (2008), Robust reliable gain scheduling control for helicopters, in ‘proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008’, pp. 1721–1724.

Zheng, F. & Gao, W. (2011a), Adaptive integral backstepping control of a microquadrotor, in ‘proceedings of the 2nd International Conference on Intelligent Control and Information Processing (ICICIP), 2011’, Vol. 2, pp. 910–915.