

# Quadcopter modeling and control using Controller Hardware-In-the-Loop

Lara Tavares de Oliveira  
Kaique Silveira Viana Costa  
Kenedy Matiasso Portella  
Aerospace Engineering course  
Universidade Federal de Santa Maria  
Santa Maria, Brazil

Lucas Vizzotto Belinaso  
Fernanda de Moraes Carnielutti  
Power Electronics and Control Group  
Universidade Federal de Santa Maria  
Santa Maria, Brazil

Deniel Desconzi Moraes  
DBServer Digital Services  
Porto Alegre, Brazil

**Abstract**—Quadcopters have many applications and an efficient controller is needed for reference tracking and to maintain vehicle stability. Quadcopter tests may require expensive laboratory setup to include certain conditions such as wind, with adequate control and monitoring. In this paper a benchmark model of quadcopter is implemented in Hardware-In-the-Loop, in order to test the quadcopter controller considering different conditions of wind. For experimental results, an LQT controller has been implemented in a DSP, while the quadcopter was implemented in the real-time simulator Typhoon HIL.

**Keywords** – Benchmark, C-HIL, Modeling, Quadcopter, Typhoon.

## I. INTRODUCTION

The use of quadcopters in civil and military applications has been increasing, due to its wide range of applications, simplicity of construction and low cost. In conjunction with sensors, they can be used in several applications, e.g., geographic mapping and disaster management, especially when these activities involve accessing hard to reach areas. However, due to its natural instability and vulnerability to wind gusts it needs an efficient stability augmentation system and control system capable of providing a satisfactory reference tracking even during critical conditions.

Testing a controller for a quadcopter requires, other than the physical quadcopter, the execution of various tests, turning this process exhaustive. In addition to that the generation of wind for testing the robustness of the controller can be risky and time-consuming. In real testing the vehicle is vulnerable to accidents and crashes, besides others problems that can complicate and affect the tests.

Hardware-In-the-Loop (HIL) is a technique for real-time simulation of dynamical systems. Controller-Hardware-In-the-Loop (C-HIL) has the controller implemented in the real hardware, a Digital Signal Processor, DSP, or microcontroller, while the dynamical system is simulated in real-time. In this way it is possible to estimate the behavior of a vehicle in different scenarios. Because of these advantages, the HIL technique have been used in a range of aerospace applications ([6], [7] and [8]).

There are works with the application of a quadcopter in HIL, [13] presents a HIL simulation setup for quadcopters, [14]

introduces a modeling technique tested in HIL, [15] develops a fuzzy controller for quadcopter tested in HIL and [16] presents a self repairing control of quadcopter tested in HIL. However none of this uses Typhoon HIL devices for real-time simulation of the quadcopter.

The HIL real-time simulation devices from the company Typhoon have been commonly applied for design, test and validation of power electronics control systems. The devices have been widely used in micro-grids, photovoltaic solar inverters and in aerospace systems such as satellite attitude control systems, [3].

Therefore, this paper presents the modeling of a benchmark quadcopter and the application of a Linear Quadratic Tracker, LQT, control with C-HIL testing. The controller is implemented in a DSP, and the quadcopter model in a real-time simulator of Typhoon HIL.

## II. BENCHMARK MODELING AND HIL IMPLEMENTATION

### A. Quadcopter benchmark model

The benchmark quadcopter chosen is the Gyro 200-ED, manufactured by the the company Gyrofly. This model was developed specially for research and, therefore, provide a wide bibliography for validation of results. The parameters of the aircraft are presented at Table I.

TABLE I  
PARAMETERS OF THE QUADCOPTER

Variable	Value	Unit
$m$	1.03	kg
$C_t$	$1.4341x10^{-5}$	$N/(rad/s)^2$
$C_d$	$2.4086x10^{-7}$	$N\cdot m/(rad/s)^2$
$d$	0.26	m
$I_x$	0.01683	$kg\cdot m^2$
$I_y$	0.01683	$kg\cdot m^2$
$I_z$	0.02834	$kg\cdot m^2$
$J$	$5x10^{-5}$	$kg\cdot m^2$

where  $m$  is the mass,  $C_t$  is the trust coefficient,  $C_d$  is the drag coefficient,  $d$  is the arm length,  $I_x$  is the moment of inertia about the  $x$  axis,  $I_y$  is the moment of inertia about the

$y$  axis,  $I_z$  is the moment of inertia about the  $z$  axis and  $J$  is the total moment of inertia of the rotors.

Figure 1 shows the coordinate frame of the quadcopter used to derive the model used for the simulation of the quadcopter in HIL. The model is described by Equation (1), from [11], [9] and [12].

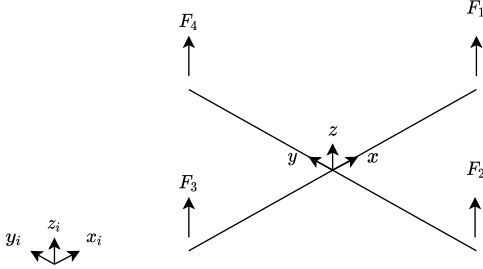


Fig. 1. Coordinate frame of the quadcopter

$$\dot{X} = \begin{cases} \ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_x - I_z}{I_x}\right) + \dot{\theta}\left(\frac{J}{I_x}\right)\Omega + \frac{d}{I_x}U_2 \\ \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) - \dot{\phi}\left(\frac{J}{I_y}\right)\Omega + \frac{d}{I_y}U_3 \\ \ddot{\psi} = \dot{\theta}\dot{\phi}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4 \\ \ddot{z} = g - (\cos\phi \cos\theta)\frac{1}{m}U_1 \\ \ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\frac{1}{m}U_1 \\ \ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\frac{1}{m}U_1 \end{cases} \quad (1)$$

In this equation the variable  $z$  is the vehicle altitude and the variables  $x$  and  $y$  define the position of the quadcopter. Also  $U_1$ ,  $U_2$ ,  $U_3$  and  $U_4$  are the control actions defined according the Equation (2).

$$U = \begin{cases} U_1 = C_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ U_2 = C_t(\omega_2^2 - \omega_4^2) \\ U_3 = C_t(\omega_1^2 - \omega_3^2) \\ U_4 = C_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (2)$$

Where  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$  and  $\omega_4$  are the angular velocities of the rotors. This angular velocities composed a disturbance  $\Omega$  shown in Equation (3).

$$\Omega = \omega_1 + \omega_3 - \omega_2 - \omega_4 \quad (3)$$

### B. Wind model

The wind models implemented in this work are the wind shear and a discrete wind gust model.

The discrete wind gust model implemented is of standard "1-cosine" shape as specified in [17]. It is applied to the axes individually as shown in Equation (4).

$$V_{w_i} = \begin{cases} 0, & x < 0 \\ \frac{V_{m_i}}{2} \left( \cos \frac{\pi x}{d_{m_i}} \right), & 0 \leq x \leq d_{m_i} \\ V_m, & x > d_{m_i} \end{cases} \quad (4)$$

In Equation (4),  $V_{w_i}$  is the resulting velocity of the wind in the axis,  $x$  is the distance traveled,  $V_{m_i}$  is the velocity of the gust and  $d_{m_i}$  is the length of the gust, where  $i = x, y, z$ .

The wind shear model implemented is specified in [17]. This model is valid for the altitude range of 3 ft to 1000 ft.

The magnitude of the wind shear,  $u_w$ , is given by the current altitude  $h$ , the wind speed at 20 ft,  $W_{20}$ , and the parameter  $z_0$ . This relation is specified by Equation (5).

$$u_w = W_{20} \frac{\ln \frac{h}{z_0}}{\ln \frac{20}{z_0}} \quad (5)$$

The parameter  $z_0$  is defined according to the flight phase. Here the adopted value is 0.15, corresponding to take off, approach and landing. This value was chosen because the quadcopter flies most of the time close to the ground.

### C. SCADA model

The Typhoon HIL SCADA is a graphical environment that makes the connection between the hardware and the real-time model. In this graphical environment widgets can be included in order to visualize the real-time response [2].

For monitoring the positions and attitude of the quadcopter during the simulations, five graphics are used. Three referring to the behavior of the positions  $x$ ,  $y$  and  $z$ , one for the angles  $\phi$ ,  $\theta$  and  $\psi$  and a third one for the control action.

For an easier insertion of wind disturbances in the simulation a check box is used. A graphic shows the wind being included in the simulation.

## III. CONTROLLER DESIGN

### A. Physical limitations and control requirements

Saturation of angles are necessary to assure the stability of the quadcopter, [4]. It was ascertain, though preliminary simulations, that a saturation for the angles as  $\leq 28^\circ$  provides a satisfactory stability. The velocity saturation, was verified to be  $\leq 7m/s$ .

### B. Control structure

Observing the model of the quadcopter, a dependency between linear and angular variables is verified. The angular variables do not depend on the linear variables. However, the linear variables depend on the angular ones. It means that the control strategy can be express as two loops, where the inner loop (faster) contains the attitude control and the outer loop contains the position and altitude controls.

Another key relation between linear and angular variables is the conversion from linear references to angular references, to be used by the attitude controller. However to solve theses equations is necessary to provide a reference value of  $\psi$ , so that the set of equations become solvable.

Therefore the altitude control results in the first control action and the position control results in the angular references of  $\phi$  and  $\theta$ . The relations between the controls, references and control actions are presented in the Figure 2.

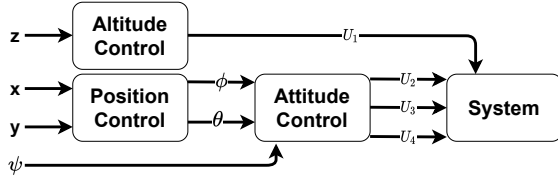


Fig. 2. Block diagram of variables

### C. Model linearization

As the chosen control law is linear it is necessary to linearize the quadcopter model. The hover condition is considered for the linearization. At this operating point the state variables are as in 6.

$$\begin{cases} \theta = \phi = \psi = \dot{\theta} = \dot{\phi} = \dot{\psi} = \ddot{\theta} = \ddot{\phi} = \ddot{\psi} = 0 \\ \dot{x} = \dot{y} = \dot{z} = \ddot{x} = \ddot{y} = \ddot{z} = 0 \end{cases} \quad (6)$$

Applying the conditions of the operating point in the dynamic of the quadcopter, the linear equations of motion are obtained as presented in [9]. The resulting equations are shown in 7.

$$\begin{aligned} \ddot{\phi} &= \frac{1}{I_x} U_2, & \ddot{\theta} &= \frac{1}{I_y} U_3, & \ddot{\psi} &= \frac{1}{I_z} U_4, & \ddot{x} &= u_x \frac{1}{m} U_1, \\ \ddot{y} &= u_y \frac{1}{m} U_1, & \ddot{z} &= u_z \end{aligned} \quad (7)$$

In the above equation  $u_x$  and  $u_y$  are the position references and  $u_z$  is the altitude reference. These values are function of the angles and the first control action, as described by Equation 8.

$$\begin{cases} u_x = \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ u_y = \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ u_z = g - (\cos \phi \cos \theta) \frac{1}{m} U_1 \end{cases} \quad (8)$$

The linear equations can be divided in three expressions, being attitude (Equation 9), position (Equation 10) and altitude (Equation 10).

$$\begin{aligned} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 & 0 \\ 1/I_x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1/I_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1/I_z \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \end{aligned} \quad (9)$$

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{mU_1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{mU_1} & 0 \end{bmatrix} \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} \end{aligned} \quad (10)$$

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [U_z] \quad (11)$$

### D. LQT controller

The chosen control is a Linear Quadratic Tracker, LQT, which is an extension of the Linear Quadratic Regulator, LQR. The LQT consist of an LQR controller with the action of a servomechanism, which ensures the tracking of the reference signal [1].

The LQT controller consists of the application of a gain matrix in the feedback loop, as done by the LQR controller. Another gain, associated to the integrator, is added to the error between the reference and the system output.

The gain matrix and the integrator gain are calculated by the Ricatti's equation. To do that it is necessary to have the augmented system matrices and the control design matrices  $\mathbf{Q}$  and  $\mathbf{R}$ .

The augmented matrices are defined by the Equation 12. Where  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are the state space system matrices.

$$\mathbf{A}_h = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_h = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \quad (12)$$

The  $\mathbf{Q}$  matrix is diagonal and has  $n \times n$  dimension, where  $n$  is the number of state variables. The greater an element of this matrix, the faster is the response of the state variable associated to that element.

The  $\mathbf{R}$  matrix is also diagonal and has  $m \times m$  dimension, where  $m$  is the number of system inputs. The greater an element of this matrix, the lower is the value of the input associated to that element.

After solving the Ricatti's equation the augmented gain matrix is obtained, Equation 13. That matrix is composed by the feedback gain matrix  $\mathbf{K}$  and the integrator gain  $k_i$ .

$$\mathbf{K}_h = [ \mathbf{K} \quad k_i ] \quad (13)$$

The final structure of the control is shown in Figure 3. In this, besides the control, is also included an anti-windup filter associated to the angular saturation to be discussed in the next subsection.

### E. Designed control

The system matrices, for each dynamic, are discretized by the Zero Order Hold method. For the discretization process a sample time of 0,001 is stalled. That way the control frequency is considerably higher than the system frequency.

The parameters of the control design matrices are chosen so that the attitude control has a faster response, the altitude control has a intermediate response and the position control has the slower response. The control design matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , for attitude control are defined as in Equation 14 and 15, the associated gain matrix is shown in Equation 16

$$\mathbf{Q}_{atti} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$\mathbf{R}_{atti} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad (15)$$

$$\mathbf{K}_{atti} = \begin{bmatrix} 28.65 & 3.03 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28.65 & 3.03 & 0 & 0 \\ 0 & 0 & 0 & 0 & 29.76 & 3.25 \end{bmatrix} \quad (16)$$

The control design matrices for the altitude control are shown in Equation 17. The associated gain matrix and the integrator gains are shown in Equation 18.

$$\mathbf{Q}_{alt} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 100 \end{bmatrix}, \quad R_{alt} = 0.1 \quad (17)$$

$$\mathbf{K}_{alt} = [ 21.6732 \quad 7.2967 ], \quad k_{iz} = 31.5074 \quad (18)$$

Lastly the control design matrices  $\mathbf{Q}$  and  $\mathbf{R}$  are shown in Equations 19 and 20. The associated gain matrix is given by 21 and the interrogator's gains for positions  $x$  and  $y$  are shown in Equation 22

$$\mathbf{Q}_{pos} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$$\mathbf{R}_{pos} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad (20)$$

$$\mathbf{K}_{pos} = \begin{bmatrix} 1.8185 & 1.1659 & 0 & 0 \\ 0 & 0 & 1.8185 & 1.1659 \end{bmatrix} \quad (21)$$

$$k_{ix} = 0.9943, \quad k_{iy} = 0.9943 \quad (22)$$

## F. Digital Signal Processor implementation

For the real-time control simulation of a quadcopter is necessary to implement the control in a Digital Signal Processor, DSP. This implementation was made by a Texas Instruments's microcontroller TMS320F28335, by the compiler debugger of Code Composer Studio.

The reception and conversion of the analog signals is done by 12 ADC, set as simultaneous sampling. One for each state variable. That way the signal of all the state variables is received by the controller.

The control frequency was set as  $1kHz$ , as in [5]. That way one PWM with the cited frequency is used for the control interruptions. Other four PWMs were used for the four control outputs, with frequency of  $10kHz$  in order to avoid timer overflow.

## IV. RESULTS

The setup used for the C-HIL implementation is shown in Figure 4. The quadcopter model is defined in Schematic Editor of Typhoon HIL and the DSP is configured by the compiler debugger of Code Composer Studio. With the model loaded in HIL SCADA and the DSP compiler debugger the interaction with the simulation parameters are made.

Many simulations can be made, with or without wind and varying the magnitude and axes of action of the wind. Here three simulations are demonstrated.

The quadcopter is first positioned in high altitude, 8 m, and a unit step is applied to the references for the  $x$  and  $y$  positions. This simulation is done without the presence of wind, with tailwind and with wind in all the axes of the quadcopter.

### A. Nominal operation

The results obtained for the simulation at nominal operation, without wind, are shown in Figure 5. In this simulation the  $x$  position converged to the reference, as expected, the  $y$  position also converged but has permanent error of about 20%. The response of altitude for the constant reference of 8 m presents an error of 0.75% and also has oscillation with amplitude of about 0.01.

In this simulation the three angles have oscillations, the angle  $\psi$  has an amplitude of about 2 degrees, the angle  $\theta$  has an amplitude of 4.5 degree and the  $\phi$  has an amplitude of 5 degrees. At the time of the beginning of the motion of the positions,  $\theta$  reaches a peak close to 7 degrees and  $\phi$  has a minimum value of  $-7$  degrees.

The control actions  $U_2$  and  $U_3$  remains between 3 N and  $-3$  N for most of the simulation, while  $U_4$  has little variation around 0 N. The control action  $U_1$  has a mean value of 10.1 as expected for the linear model.

### B. Performance under wind disturbance

As mentioned, the simulation was made for two types of wind, tailwind and with wind in all axes. In those, the magnitudes of the wind are defined as 2.5 m/s for the  $x$  axis and 2 m/s for the axes  $y$  and  $z$ , the gust length is set as 80 m for all the axes.

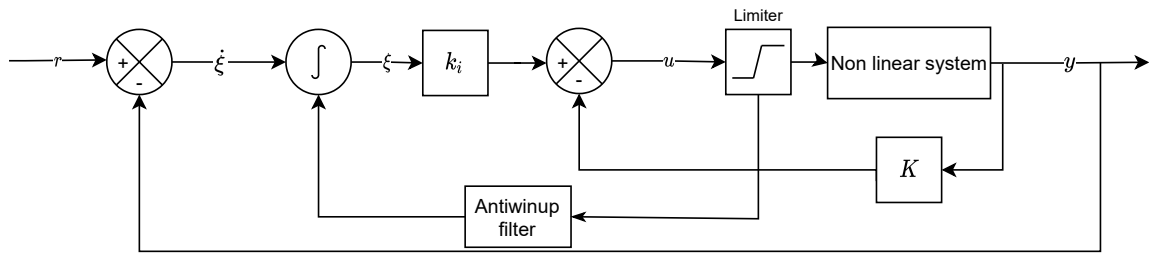


Fig. 3. Scheme of general state space controller with servomechanism

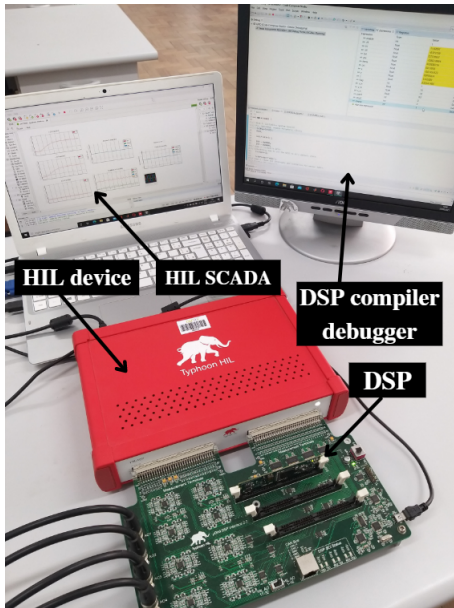


Fig. 4. C-HIL setup

The response of the simulation with tailwind is shown in Figure 6. The response of  $x$  and  $y$  positions are similar to the simulation without wind. However the altitude response presents an oscillation of higher amplitude, increasing the error. The minimum value of the angle  $\phi$  differs from the previous simulation, in this it has a lower value, of about  $-11$  degrees.

For the simulation with wind in all the axes the responses are shown in Figure 7. For this simulation the  $x$  position and the altitude responses are similar as for the simulation with tailwind. However the  $y$  position presents an oscillation of lower amplitude than the other two simulations. The only angle with significant difference from the other simulations is  $\theta$  that has a peak of about 10 degrees when the movement of the positions starts.

## V. CONCLUSION

This paper presents the technique of C-HIL, applied to the control of a quadcopter, to be used for easy, safe and unexpensive testing of control laws under different conditions of wind. This was obtained using the Typhoon HIL technology

alongside with a benchmark model of a quadcopter. Through the SCADA panel and the compiler debugger of Code Composer Studio changes can be made in the wind parameter and references in real time.

The functionality of this system is demonstrated by application of the LQT control law. The results represent the diverse number of simulations that can be done with the current system.

For future work a wind model of continuous turbulence can be added for a more complete model. Besides that other disturbances can be included for simulations, as variation of mass.

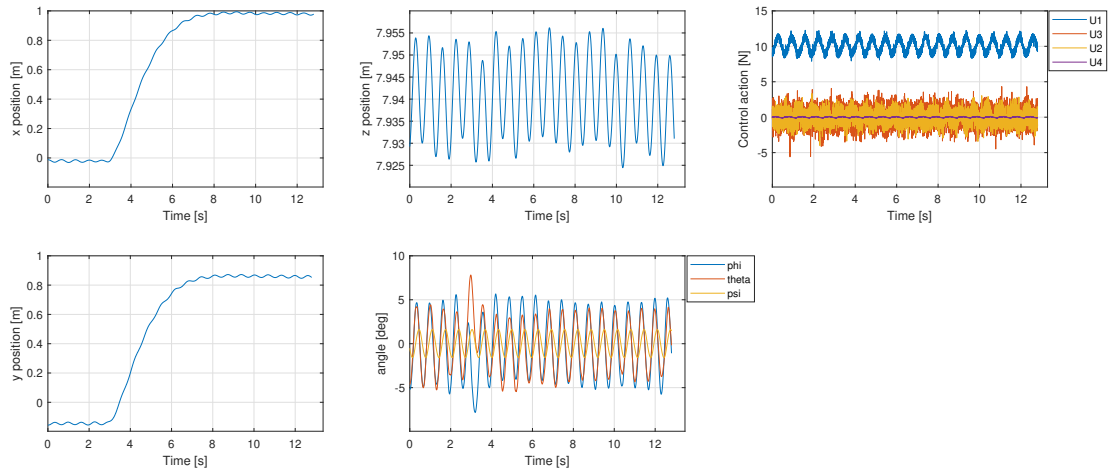


Fig. 5. Simulation at high altitude-step in x and y without wind

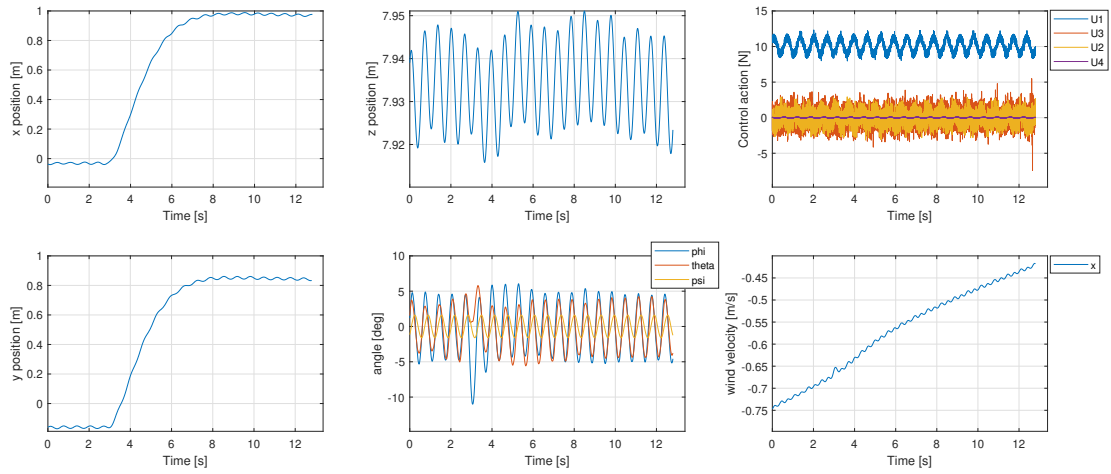


Fig. 6. Simulation at high altitude-step in x and y with tailwind

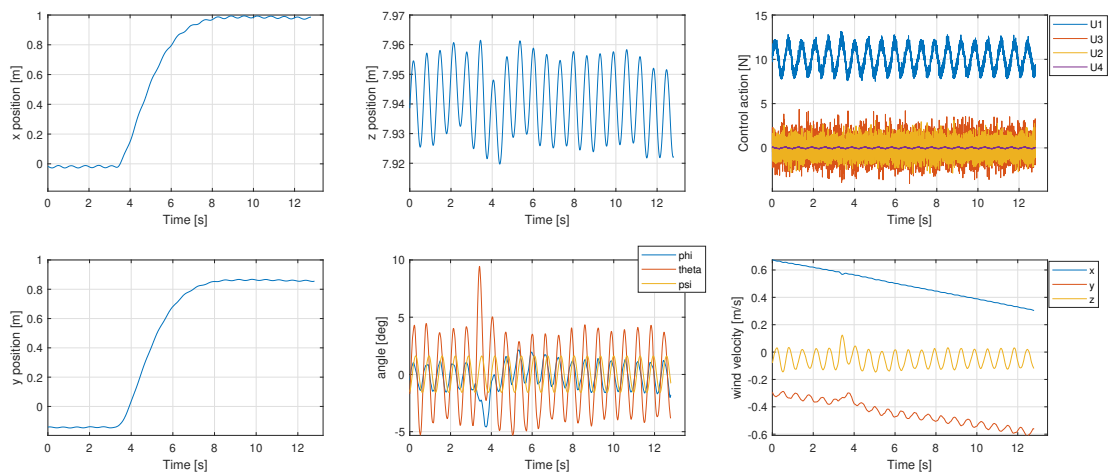


Fig. 7. Simulation at high altitude-step in x and y with wind in all axes

## REFERENCES

- [1] K., Ogata, and Y. Yanjuan. *Modern control engineering*. Vol. 4. India: Prentice hall, 2002.
- [2] Typhoon, H. I. L. Inc. *Typhoon Schematic Library User Guide*, www.typhoon-hil.com", 2016.
- [3] Gaber, Khaled and El Mashade, Mohamed B and Aziz, Ghada A Abdel. *Hardware-in-the-loop real-time validation of micro-satellite attitude control*. Computers and Electrical Engineering, v. 85, p. 106679, 2020
- [4] POYI, Gwangtim Timothy. *A novel approach to the control of quad-rotor helicopters using fuzzy-neural networks*. University of Derby, 2014.
- [5] Pounds, Paul, Robert Mahony, and Peter Corke. *System identification and control of an aerobot drive system*. 2007 Information, Decision and Control. IEEE, 2007.
- [6] Varshney, Vitul, Melvin Wilson, and Sakthivel Sivaraman. *PID based stabilization of gesture controlled drones using HIL simulation*. International Journal of Engineering and Advanced Technology (IJEAT), v. 3, n. 6, p. 222-225, 2014.
- [7] Chudy, Peter, Petr Dittrich, and Pawel Rzcudlo. *HIL simulation of a light aircraft flight control system*. 2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC). IEEE, 2012.
- [8] Ma, Ou, Angel Flores-Abad, and Toralf Boge. *Use of industrial robots for hardware-in-the-loop simulation of satellite rendezvous and docking*. Acta Astronautica, v. 81, n. 1, p. 335-347, 2012.
- [9] Okyere, Emmanuel et al. *LQR controller design for quad-rotor helicopters*. The Journal of Engineering, v. 2019, n. 17, p. 4003-4007, 2019.
- [10] da Silva, André Luís, and José Jaime da Cruz. *Fuzzy adaptive extended Kalman filter for UAV INS/GPS data fusion*. Journal of the Brazilian Society of Mechanical Sciences and Engineering 38.6 2016: 1671-1688.
- [11] Voos, Holger. *Nonlinear state-dependent Riccati equation control of a quadrotor UAV*. 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control. IEEE, 2006.
- [12] Bouabdallah, Samir, and Roland Siegwart. *Backstepping and sliding-mode techniques applied to an indoor micro quadrotor*. Proceedings of the 2005 IEEE international conference on robotics and automation. IEEE, 2005.
- [13] Bayrakceken, M. Kemal, et al. *HIL simulation setup for attitude control of a quadrotor*. 2011 IEEE International Conference on Mechatronics. IEEE, 2011. p. 354-357
- [14] Gachadoit, N., et al. *Advanced modeling with a symbolic based approach application to the modeling, control design and real-time control and hil simulation of a quadrotor helicopter*. ERTS2 2010, Embedded Real Time Software & Systems. 2010.
- [15] Mehranpour, Mohammad Reza, et al. *A new fuzzy adaptive control for a Quadrotor flying robot*. 2013 13th Iranian Conference on Fuzzy Systems (IFSC). IEEE, 2013. p. 1-5.
- [16] Yang, Huiliao, Bin Jiang, and Ke Zhang. *Direct self-repairing control of the quadrotor helicopter based on adaptive sliding mode control technique*. Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference. IEEE, 2014. p. 1403-1408.
- [17] *Military specification MIL-F8785C*. 1969