

Rapid Control Prototyping and PIL Co-Simulation of a Quadrotor UAV Based on NI myRIO-1900 Board

Soufiene Bouallègue

Research Laboratory in Automatic Control (L.A.R.A) of the National Engineering School of Tunis (ENIT), University of Tunis EL Manar.

High Institute of Industrial Systems of Gabès (ISSIG), Salaheddine EL AYOUBI Street, 6011 Gabès, Tunisia,

Rabii Fessi

Research Laboratory in Automatic Control (L.A.R.A) of the National Engineering School of Tunis (ENIT), University of Tunis EL Manar.

National Engineering School of Gabès (ENIG), University of Gabès,

Omar IBN ELKHATTAB Street, 6072 Gabès, Tunisia,

Abstract—In this paper, a new Computer Aided Design (CAD) methodology for the Processor-In-the-Loop (PIL) co-simulation and Rapid Control Prototyping (RCP) of a Quadrotor Vertical Take-Off and Landing (VTOL) type of Unmanned Aerial Vehicle (UAV) is proposed and successfully implemented around an embedded NI myRIO-1900 target and a host PC. The developed software (SW) and hardware (HW) prototyping platform is based on the Control Design and Simulation (CDSim) module of LabVIEW environment and an established Network Streams data communication protocol. A dynamical model of the Quadrotor UAV, which incorporates the dynamics of vertical and landing flights and aerodynamic forces, is obtained using the Newton-Euler formalism. PID and Model Predictive Control (MPC) approaches are chosen as examples for experiment prototyping. These control laws, as well as the dynamical model of the Quad, are implemented and deployed as separate LabVIEW Virtual Instruments (VI) on the myRIO-1900 target and the host PC, respectively. Several demonstrative co-simulation results, obtained for a 3D LabVIEW emulator of the Quadrotor, are presented and discussed in order to improve the effectiveness of the proposed Model Based Design (MBD) prototyping methodology

Keywords—Quadrotor UAV; modeling; Computer Aided Design; Model Based Design; Rapid Control Prototyping; PIL co-simulation; LabVIEW/CDSim; NI myRIO-1900; PID and MPC approaches

I. INTRODUCTION

The Unmanned Aerial Vehicles (UAV), particularly the Quadrotors ones [1]-[5], are flying robots without pilot which are able to conduct missions in autonomous or half-autonomous modes, also in hostile and disturbed environments. Among the tasks to be conducted with these robots are found military acknowledgment, monitoring missions and specially civilian missions such as the inspection of dams, border monitoring, the prevention of forest fires and so on [6]. This explains the interest shown by many researchers to study the flight dynamics and the advanced control laws real-world implementation of these kinds of vehicles.

The Quadrotor is a very promising concept of UAV aircrafts. It is a VTOL vehicle equipped with four rotors that are independently controlled [3]. The movement of such UAV results from changes in the speed of these rotors. The front and rear motors rotate counter-clockwise, while the left and right

motors rotate clockwise. During its flight, a Quadrotor is subjected to external forces like the gusts of wind, gravity, viscous friction and others such as the thrust of rotors and body and propellers drag forces. Moments generated by gyroscopic effects of motors are also noted [1,3]. So, this mechanical structure and the nonlinear and coupled flight dynamics increase the complexity of the Quadrotor. The problem of its flight control and prototyping, i.e. the dynamics stabilization and path tracking, becomes challenging and allowed to be a popular topic in the field of robotics research. Control laws must be designed to work in a real-world scenario and over different flight conditions. Among many control approaches proposed for the dynamics stabilization and path tracking of Quadrotors, we note mainly the PID and Linear Quadratic [7]-[10], Sliding Mode (SMC) [11]-[13], backstepping [14], neural network [15] and Model Predictive (MPC) [16]-[19] control methods.

From the real-world practical point of view, such flight control algorithms must be verified and well prototyped before their definitive implementation on the Quadrotor. Sophisticated and embedded SW/HW solutions for this design stage are usually needed and a powerful platform for achieving both the rapid prototyping and final real-world implementation is very required. This problem can be efficiently handled thanks to the MDB and CAD concepts, especially with the related Processor-In-the-Loop (PIL) and/or Hardware-In-the-Loop (HIL) co-simulation methods. In this framework, the recent increased processing power of the Reconfigurable Inputs and Outputs (RIO) targets of the National Instruments (NI) Company, particularly the embedded myRIO-1900 devices [20], makes these platforms well suited to perform the advanced processing tasks required by complex and hard applications such as the rapid control prototyping, the HIL co-simulation and the final real-world implementation of various flight embedded controllers for QTW UAVs. Based on its suitable architecture and powerful onboard devices such as the three-axis accelerometer, analog IO extensions and Wi-Fi module, the myRIO-1900 target is a promised embedded solution for both prototyping and final real-world implementation platform of various and sophisticated flight controllers of aerial robots. Its associated LabVIEW Real-Time (RT) software tool takes also advantage of deterministic execution and the highest degree of reliability. So, this paper deals with the design and development of a new MBD solution

for the PIL co-simulation and rapid control prototyping of a Quadrotor. Such a CAD methodology is built and successfully implemented around an embedded NI myRIO-1900 platform and a host PC. The software implementation is based on the Control Design and Simulation (CDSim) module of LabVIEW environment [21,22] and a set-up Network Streams-based data communication protocol. Both PID and MPC approaches are investigated as experiment examples in order to validate the proposed PIL co-simulation solution.

The remainder of this paper is organized as follows. Section 2 deals with the modeling of the Quadrotor for the vertical and landing flight dynamics using the Newton-Euler formalism. In Section 3, the developed NI myRIO-1900 based platform for the PIL co-simulation and control rapid prototyping of the rotorcraft is presented. All related software and hardware tools of such a CAD prototyping methodology are described. Section 4 is dedicated to the implementation and hardware co-simulation of the PID and MPC approaches, chosen as experiment examples for the Quadrotor attitude and altitude control prototyping. Several demonstrative results as well as our own developed LabVIEW graphical interfaces for rapid flight control prototyping are presented and discussed. Finally, concluding remarks end the paper in Section 5.

II. MODELING OF THE QUADROTOR UAV

A. System description and aerodynamic forces

The studied Quadrotor is detailed with their body-frame $\mathbf{R}_B(O, x, y, z)$ and earth one $\mathbf{R}_E(o, e_x, e_y, e_z)$ respectively, as shown in Fig. 1. Let denote by m the total mass of the Quadrotor, g the acceleration of the gravity and l the distance from the center of each rotor to the Center of Gravity (COG). The orientation of the Quadrotor is given by the rotation matrix $\mathcal{R}: \mathbf{R}_E \rightarrow \mathbf{R}_B$ which depends on the three Euler angles (ϕ, θ, ψ) and defined by the following equation:

$$\mathcal{R}(\phi, \theta, \psi) = \begin{bmatrix} c\psi c\theta & s\psi c\theta & c\phi s\theta c\psi - s\psi c\theta & c\phi s\theta s\psi + s\psi s\theta \\ s\psi c\theta & c\psi c\theta & c\phi s\theta s\psi - s\psi c\theta & c\phi s\theta c\psi + s\psi s\theta \\ -s\theta & c\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (1)$$

where $c(\cdot) = \cos(\cdot)$ and $s(\cdot) = \sin(\cdot)$.

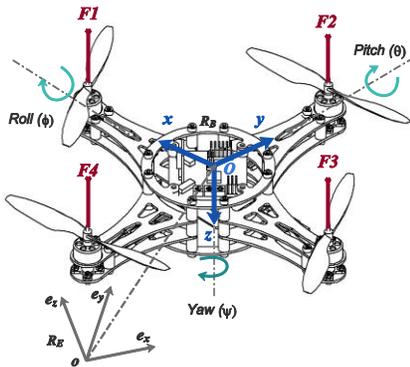


Fig. 1. Mechanical structure of the Quadrotor rotorcraft

Let consider the following model partitions of the Quadrotor naturally into translational and rotational coordinates:

$$\boldsymbol{\xi} = (x, y, z)^T \in \mathbb{R}^3, \quad \boldsymbol{\eta} = (\phi, \theta, \psi)^T \in \mathbb{R}^3 \quad (2)$$

where $\boldsymbol{\xi} = (x, y, z)^T$ denotes the position vector of the COG of the Quadrotor relative to its fixed earth-frame, $\boldsymbol{\eta} = (\phi, \theta, \psi)^T$ denotes the attitude of the Quadrotor given by the Euler angles for rolling $\phi \in [-\pi/2, \pi/2]$, pitching $\theta \in [-\pi/2, \pi/2]$ and yawing $\psi \in [-\pi, \pi]$ motions.

Let a vector $\mathbf{v} = (u, v, w)^T$ denote the linear velocity of the UAV in the earth-frame \mathbf{R}_E , while the vector $\boldsymbol{\mathcal{G}} = (p, q, r)^T$ represents its angular velocity in \mathbf{R}_B frame. The kinematic equations of rotational and translational movements are obtained, respectively, as follows [12,13]:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (3)$$

$$\mathbf{v}_e = \mathcal{R}(\phi, \theta, \psi) \mathbf{v}_B \quad (4)$$

where \mathbf{v}_e and \mathbf{v}_B are linear velocities of the mass center expressed in the earth-frame and body-frame, respectively.

Each motor of the Quadrotor produces the force F_i which is proportional to the square of the angular speed. The trust force generated by the i^{th} rotor of Quadrotor is given by:

$$F_i = \frac{1}{2} \rho \Lambda C_T r^2 \Omega_i^2 = b \Omega_i^2 \quad (5)$$

where ρ is the air density, r and Λ are the radius and the section of the propeller respectively, C_T is the aerodynamic thrust coefficient.

The aerodynamic drag torque, caused by the drag force at the propeller of the i^{th} rotor and opposed the motor torque, is defined as follows:

$$\delta_i = \frac{1}{2} \rho \Lambda C_D r^2 \Omega_i^2 = d \Omega_i^2 \quad (6)$$

where C_D is the aerodynamic drag coefficient.

The pitch torque is a function of the difference $(F_3 - F_1)$, the roll torque is proportional to the term $(F_4 - F_2)$ and the yaw one is the sum of all reactions torques generated by the four rotors and due to the shaft acceleration and propeller drag. All these pitching, rolling and yawing torques are defined respectively as follows:

$$\Gamma_\theta = l(F_3 - F_1) \quad (7)$$

$$\Gamma_\phi = l(F_4 - F_2) \quad (8)$$

$$\Gamma_\psi = \lambda(F_1 - F_2 + F_3 - F_4) \quad (9)$$

where λ is a proportional coefficient.

Two gyroscopic effects torques, due to the motion of the propellers and the Quadrotor body, are additively provided. These moments are given respectively by:

$$M_{gp} = \sum_{i=1}^4 \Omega \wedge (0, 0, J_r (-1)^{i+1} \Omega_i)^T \quad (10)$$

$$M_{gb} = \Omega \wedge J \Omega \quad (11)$$

where Ω is the vector of angular velocities in the fixed earth-frame and $J = \text{diag}(I_{xx}, I_{yy}, I_{zz})$ is the inertia matrix of the Quadrotor, J_r denotes the z-axis inertia of the propellers rotors.

The Quadrotor is controlled by independently varying the speed of the four rotors. Hence, these control inputs are defined as follows:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} F \\ \Gamma_\phi \\ \Gamma_\theta \\ \Gamma_\psi \end{pmatrix} = \begin{pmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{pmatrix} \begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} \quad (12)$$

where $\Omega_{1,2,3,4}$ are the angular speeds of the four rotors, respectively.

From Eq. (12), it can be observed that the input u_1 denotes the total thrust force on the Quadrotor body in the z-axis, the inputs u_2 and u_3 represent the roll and pitch torques, respectively. The input u_4 represents a yawing torque.

B. Modeling with Newton-Euler Formalism

While using the Newton-Euler formalism for modeling, the Newton's laws lead to the following motion equations of the Quadrotor:

$$\begin{cases} m\ddot{\xi} = F_{th} + F_d + F_g \\ J\dot{\Omega} = M - M_{gp} - M_{gb} - M_a \end{cases} \quad (13)$$

where $F_{th} = \mathcal{R}(\phi, \theta, \psi) \left(0, 0, \sum_{i=1}^4 F_i \right)^T$ denotes the total thrust force of the four rotors, $F_d = \text{diag}(\kappa_1, \kappa_2, \kappa_3) \mathbf{v}^T$ is the air drag force which resists to the Quadrotor motion,

$F_g = (0, 0, mg)^T$ is the gravity force, $M = (\Gamma_\phi, \Gamma_\theta, \Gamma_\psi)^T$ represents the total rolling, pitching and yawing torques, M_{gp} and M_{gb} are the gyroscopic torques and $M_a = \text{diag}(\kappa_4, \kappa_5, \kappa_6) \mathcal{G}^T$ is the torque resulting from aerodynamic frictions.

Substituting the position vector and the forces expressions into the Eq. (13), we have the following translational dynamics of the Quadrotor:

$$\begin{cases} \ddot{x} = \frac{1}{m} (c\phi c\psi s\theta + s\phi s\psi) u_1 - \frac{\kappa_1}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (c\phi s\psi s\theta - s\phi c\psi) u_1 - \frac{\kappa_2}{m} \dot{y} \\ \ddot{z} = \frac{1}{m} c\phi c\theta u_1 - g - \frac{\kappa_3}{m} \dot{z} \end{cases} \quad (14)$$

From the second part of Eq. (13), and while substituting each moment by its expression, we deduce the following rotational dynamics of the rotorcraft:

$$\begin{cases} \ddot{\phi} = qr \frac{I_y - I_z}{I_x} - \frac{J_r}{I_x} \Omega_r q - \frac{\kappa_4}{I_x} p + \frac{1}{I_x} u_2 \\ \ddot{\theta} = pr \frac{I_z - I_x}{I_y} + \frac{J_r}{I_y} \Omega_r p - \frac{\kappa_5}{I_y} q + \frac{1}{I_y} u_3 \\ \ddot{\psi} = pq \frac{I_x - I_y}{I_z} - \frac{\kappa_6}{I_z} r + \frac{1}{I_z} u_4 \end{cases} \quad (15)$$

According to the above established Eq. (14) and Eq. (15), $X = (\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z})^T \in \mathbb{R}^{12}$ is retained as the state vector of the Quadrotor nonlinear model. Note that $\kappa_{1,2,\dots,6}$ are the aerodynamic friction and translational drag coefficients, $\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$ is the overall residual rotor angular velocity.

III. PROPOSED CAD METHODOLOGY FOR RAPID CONTROL PROTOTYPING

A. Rapid control prototyping platform

An advanced MBD platform for control algorithms verification and prototyping must make easy the practical implementation with the same used hardware target and software tools. In this paper, the hardware setup of the proposed MBD methodology for rapid prototyping and PIL co-simulation of QTW UAVs is depicted in Fig. 2.

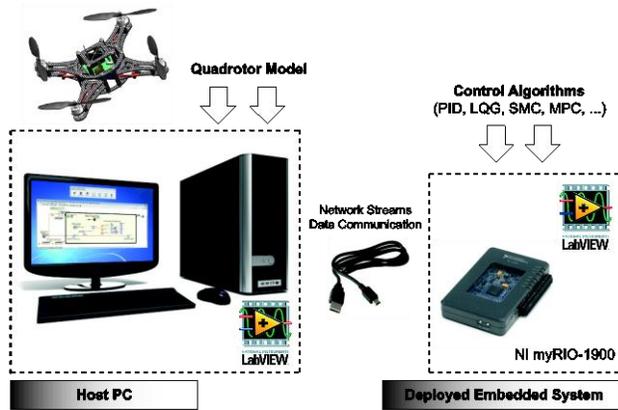


Fig. 2. Synoptic scheme of the NI myRIO-1900 based PIL co-simulation platform

The developed SW/HW solution is based mainly on the use of an embedded myRIO-1900 platform from National Instruments Company. Well-suitable for complex processing and real-time computing, this NI portable and RIO target is associated to a host PC with LabVIEW CDSim, myRIO 2015 and Robotics environments. After prototyping phase, the myRIO-1900 device operates autonomously to execute a LabVIEW control project which is deployed on its RT dual-core processor. Thanks to its powerful tools, LabVIEW software environment simplifies construction and prototyping of designed control systems and provides the ability to implement a variety of control algorithms.

B. NI myRIO-1900 board based implementation

The main component of the proposed PIL platform for rapid prototyping of the QTV vehicle is the embedded and reconfigurable NI myRIO-1900 board, depicted in Fig. 3. Featuring the NI industry-standard reconfigurable I/O technology, this hardware target is the enclosed version of myRIO platforms which presents three I/O connectors, wireless capabilities and a dual-core ARM RT Cortex-A9 processor with 667 MHz frequency speed [20]. The NI myRIO-1900 platform is also equipped with 256 MB of nonvolatile memory and 512 MB of DDR3 RAM memory. Running a real-time OS as well as a customizable FPGA circuit with the Xilinx SoC Zynq-710 architecture, this embedded target provides differential and single-ended (referenced 0-5V and $\pm 10V$) 10 Analog Inputs (AI) and 6 Analog Outputs (AO), 40 general-purpose Digital I/O lines (DIO), with 3.3V output, 3.3 V/5 V-compatible, audio, and power output in a compact embedded device.

The NI myRIO-1900 platform contains an onboard three-axis accelerometer with a range of $\pm 8g$, a resolution of 12 bits and a sample rate of 800 S/s. This integrated device samples each axis continuously and updates a readable register with the result and remains very suitable for the UAV control prototyping and implementation framework. Connected to a host computer over USB and wireless 802.11b, g, n possibilities [20], the myRIO-1900 is equipped with a box Wi-Fi module which can be used for remote control of such UAV aircrafts. Adaptable for LabVIEW programming level and thanks to its onboard devices (PWM, SPI, I2C, encoder, etc.),

its reduced physical dimensions and low weight, the myRIO-1900 provides an affordable tool that helps to design and prototype advanced flight control algorithms and real-world design projects of UAV guidance.



Fig. 3. Embedded NI myRIO-1900 Board

C. LabVIEW CDSim and myRIO software tools

The LabVIEW CDSim, as a sophisticated LabVIEW add-on from NI Company [21], leads to identify and simulate online and offline dynamic systems, analyze open-loop model behavior, design closed-loop controllers and estimators, and deploy digital control systems implementation on NI hardware platforms. As depicted in Fig. 4, the CDSim tool provides various libraries for control design and simulation of dynamic systems.

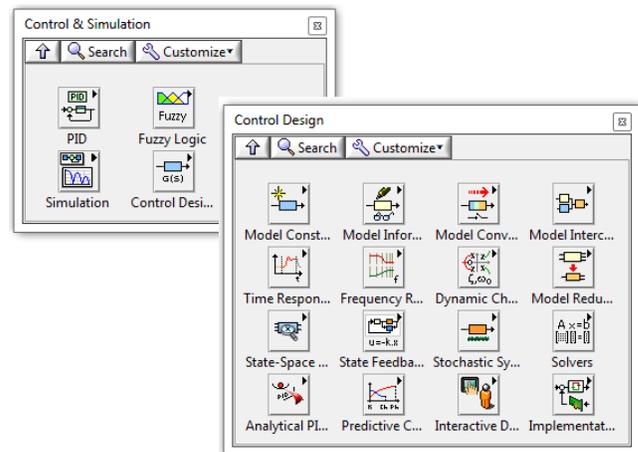


Fig. 4. Control Design & Simulation palettes of the LabVIEW/CDSim module

When integrated with others NI built-in tools, such as LabVIEW MathScript RT module, the CDSim module performs textual mathematics and algorithm design in LabVIEW using the *.m file based syntax. Furthermore, the usability of LabVIEW CDSim module can be also expanded with LabVIEW System Identification toolkit to find empirical models from real plant stimulus-response information, with LabVIEW PID and Fuzzy Logic toolkit for design and tuning various PID and fuzzy control structures, and with LabVIEW Statechart module for event-based control prototyping [21]. Once the control algorithms are designed, it is easy to deploy

dynamic systems to real-time hardware targets without the need to generate code by using the LabVIEW Real-Time and LabVIEW FPGA modules. The LabVIEW myRIO toolkit can be instead used while working with NI myRIO platforms as shown in Fig. 5.

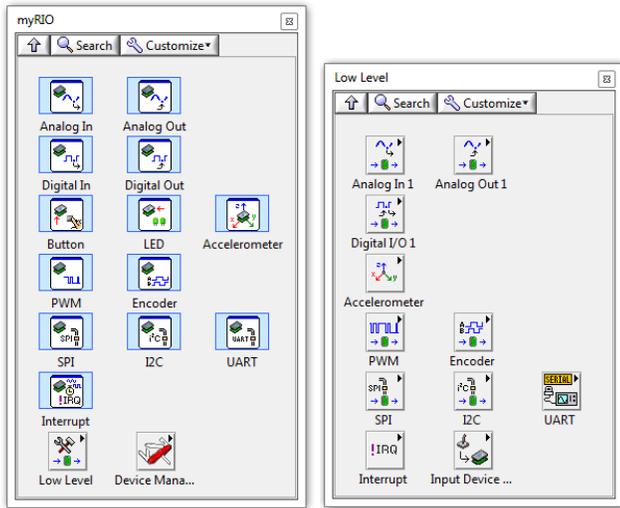


Fig. 5. NI myRIO low level VIs and advanced IO palettes

IV. HARDWARE IMPLEMENTATION AND CO-SIMULATION RESULTS

In order to show the effectiveness of the proposed PIL co-simulation methodology, PID and MPC approaches are investigated for prototyping tests. For the discrete-time control laws implementation, a sampling time equal to 0.01 sec is used. The physical parameters of Table I are used for controller’s synthesis and PIL co-simulation stages. The setpoints for the attitude and position dynamics control are applied by external potentiometers connected to the appropriate analog inputs of the myRIO-1900 board.

TABLE I. QUADROTOR VEHICLE MODEL PARAMETERS

Parameters	Values
Mass m	0.650 (kg)
Rotor distance to COG l	0.23 (m)
Lift coefficient b	2.9e-05 (N/rad/s)
Drag propellers coefficient d	3.23e-07 (Nm/rad/s)
Body inertias I_{xx} , I_{yy} and I_{zz}	0.0075, 0.0075, 0.013 (kg.m ²)
Propellers inertia along z-axis J_r	6.00e-05 (kg.m ²)
Translational drag coefficients κ_1 , κ_2 and κ_3	5.57e-04, 5.57e-04, 6.35e-04 (N/rad/s)
Aerodynamic friction coefficients κ_4 , κ_5 and κ_6	5.57e-04, 5.57e-04, 6.35e-04 (N/m/s)
Acceleration of the gravity g	9.81 (m/s ²)

A. PID control prototyping

A typical structure of PID controller, given by Eq. (16), is used for the stabilization of the altitude and attitude dynamics of the QTW [23]:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (16)$$

The error signal $e(t)$ is used to generate the proportional, integral and derivative actions of the designed controller. The empirical Ziegler-Nichols based method is firstly adopted to compute the appropriate values of the PID parameters [23]. Then, such parameters can be tuned further thanks to the developed PIL co-simulation interface. Four PID control laws, for each altitude z , roll ϕ , pitch θ and yaw ψ dynamics, will then generated as control inputs for the Quadrotor according to Eq. (12).

For the PIL and HIL applications, communication between VIs and exchange data between them is always a critical part of a LabVIEW RT project. Such applications typically function as “data servers” and operate with the principle of the LabVIEW Server/Client architecture [22]. In such architecture, a host PC acts as a data “Server” in order to provide information to the myRIO NI-1900 platform as the “Client”. For such a communication protocol building and management, different data communication mechanisms, such as Network-Published Shared Variables, Raw Ethernet (TCP/UDP) and Network Streams have been proposed [22].

In our PIL co-simulation case, we need sending measures of Quadrotor states from the host PC VI and receive such data in the controller VI, deployed on the myRIO target, to provide in turn control signals and setpoints to the Quadrotor. Thus, we choose the Network Streams approach that best meets our mentioned needs. For setting-up a network stream for the proposed PIL solution, the LabVIEW diagrams of Fig. 6 and Fig. 7 are implemented on the host PC (Quadrotor model) and myRIO deployed embedded target (PID controller), respectively. We develop later the LabVIEW diagram for the MPC approach. Two control & simulation loops from the LabVIEW CDSim module are used to build separately the Quadrotor and control algorithms models. Each model can be then implemented inside these given loops.

The implemented LabVIEW front panels for the PID controller prototyping and the dynamic model of the Quadrotor are shown in Fig. 8 and Fig. 9, respectively. Such implementations are made separately on two VIs which are physically deployed on the host PC and the embedded myRIO-1900 platform, respectively. This proposed HW/SW solution leads to the hardware co-simulation results of Fig. 10 and Fig. 11 for the attitude and altitude dynamics control prototyping, respectively. These demonstrative curves show the effectiveness of such an advanced PIL solution based on the embedded NI myRIO-1900 platform.

Later, we give in Fig. 17 some flight illustrations in the 3D frame, obtained for the MPC based control prototyping of the emulated Quadrotor rotorcraft.

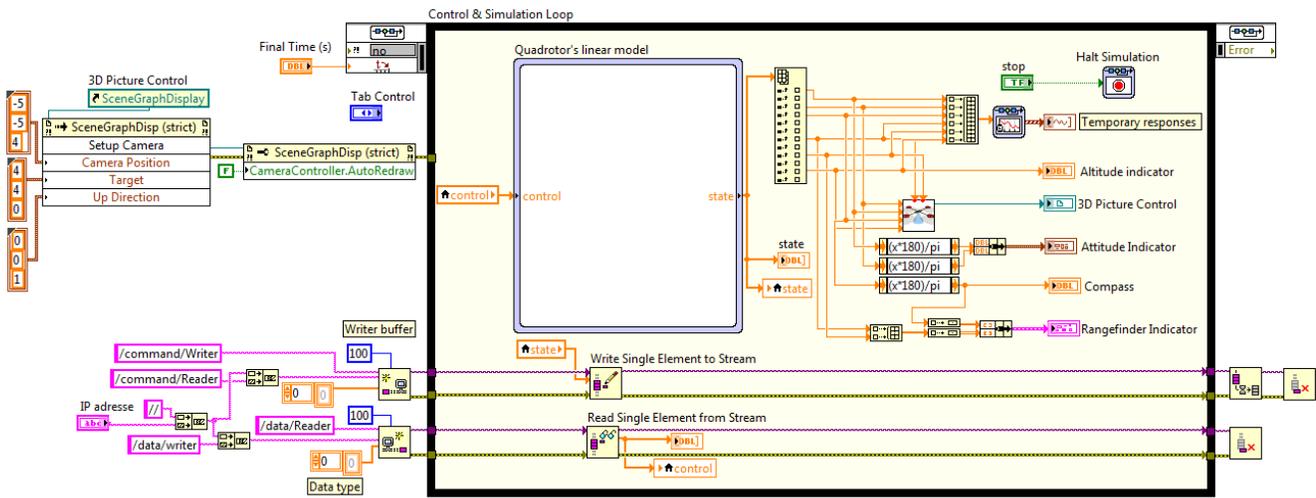


Fig. 6. LabVIEW diagram for the Network Streams “Server” based implementation of the Quadrotor model

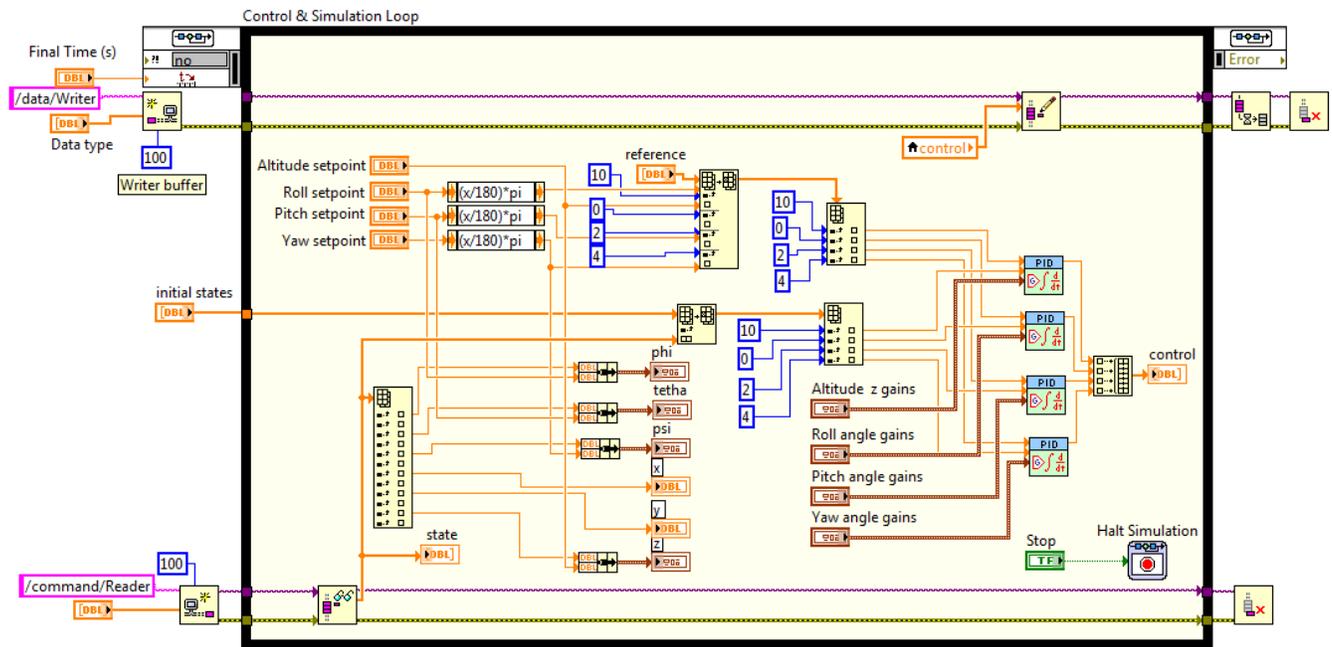


Fig. 7. LabVIEW diagram for the Network Streams “Client” based implementation of the PID controller algorithm

B. MPC control prototyping

This section is devoted to the rapid MPC prototyping of the Quadrotor while using the same PIL co-simulation platform. The main elements of the discrete-time MPC are the plant input, the controlled output and the reference trajectory which are denoted by $u \in \mathbb{R}$, $y \in \mathbb{R}$, and $r \in \mathbb{R}$, respectively. The plant model determines the predicted plant outputs on the prediction horizon N_p . The optimization algorithm is aimed at determining the control sequence given by $\{u(k-1+i), i=1, 2, \dots, N_c\}$ for the control horizon N_c . Only the first element $u^*(k)$ of the optimized control sequence is

applied to the plant and the control input is updated at each sampling instant.

At every sampling time and for a specified prediction and control horizons, the MPC controller, designed and implemented on the LabVIEW/CDSim environment, attempts to minimize the following cost function [24,25,21]:

$$J_{MPC}(k) = \sum_{i=1}^{N_p} \hat{e}(k+i|k)^T Q \hat{e}(k+i|k) + \sum_{i=1}^{N_c} [\Delta u^T(k+i|k) R \Delta u(k+i|k)] \tag{17}$$

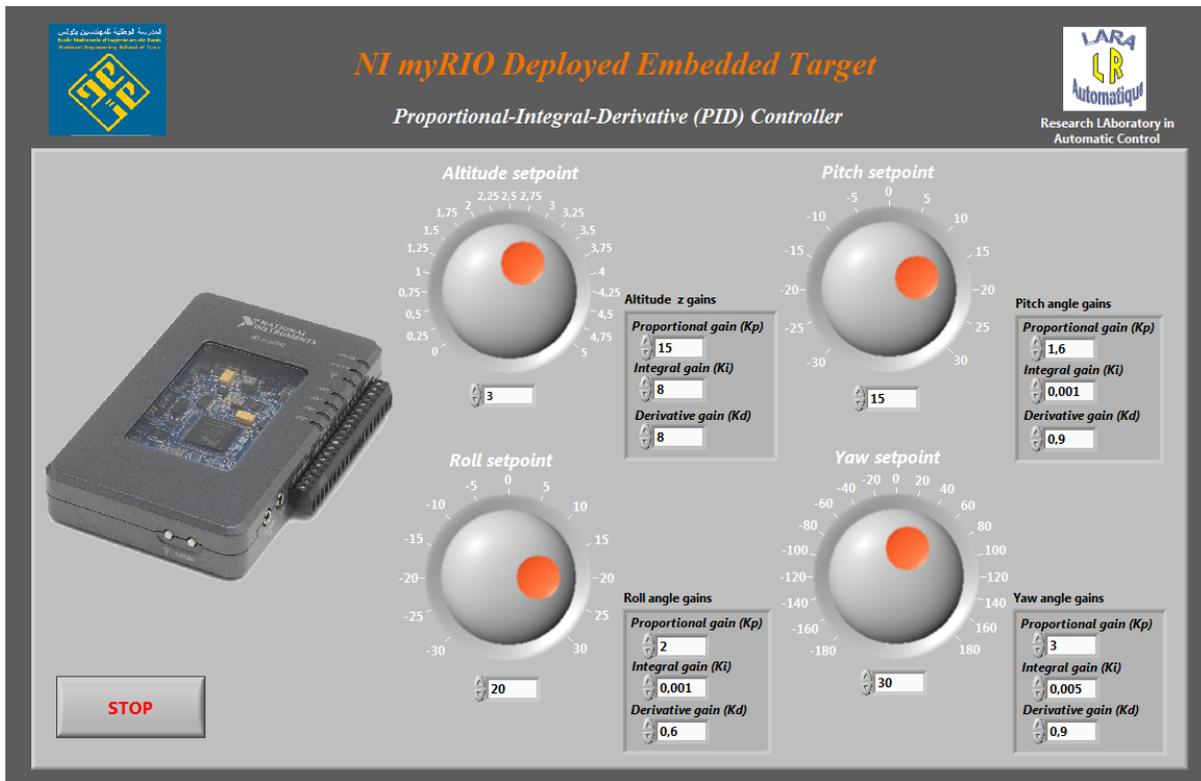


Fig. 8. LabVIEW Front Panel for the rapid PID control prototyping: VI deployed on the myRIO-1900 target

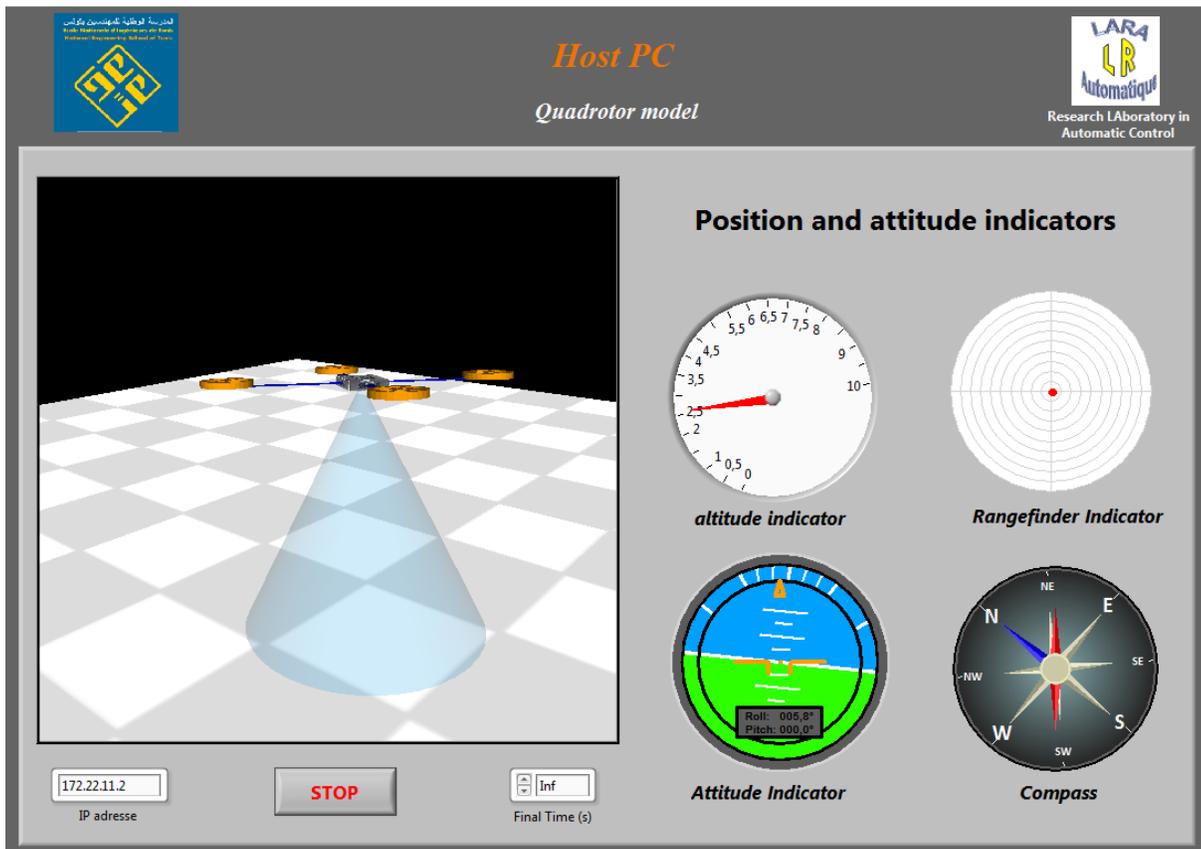


Fig. 9. LabVIEW Front Panel for the Quadrotor dynamic model implementation: VI deployed on the host PC

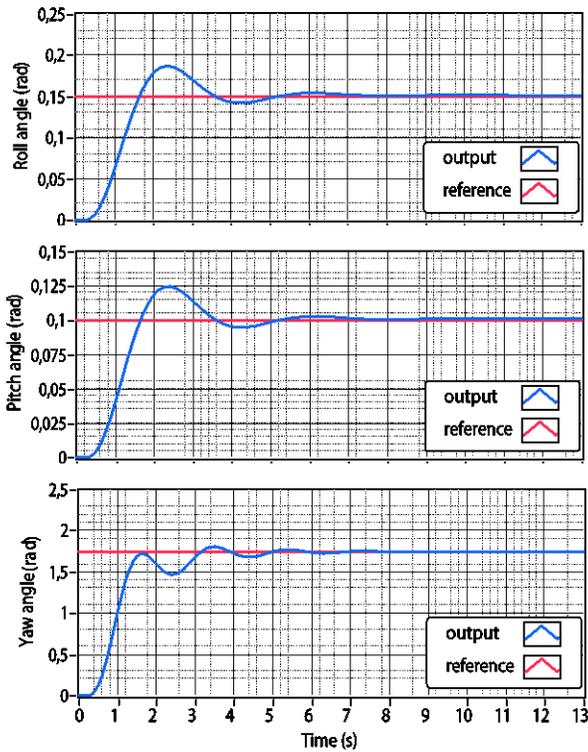


Fig. 10. PID prototyping results for the Quadrotor attitude

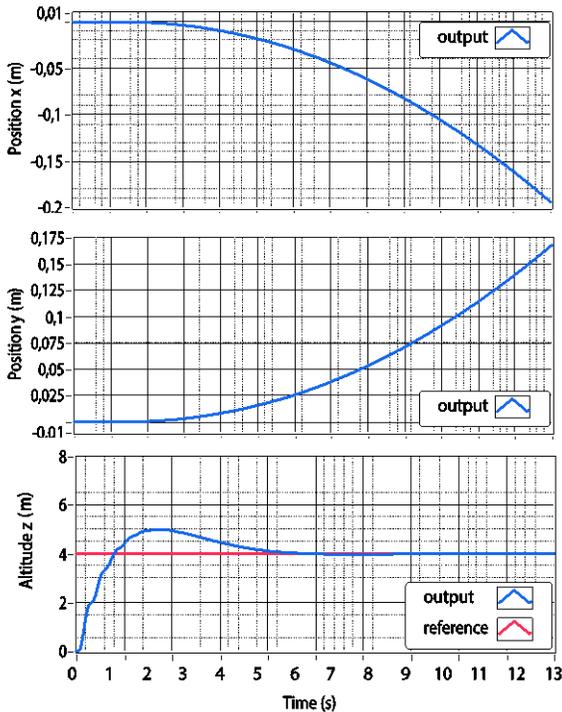


Fig. 11. PID prototyping results for the Quadrotor altitude

In Eq. (17), $\hat{e}(k+i|k) = \hat{y}(k+i|k) - r(k+i|k)$, $\hat{y}(k+i|k)$, $r(k+i|k)$ and $\Delta u(k+i|k)$ are the predicted

plant output, the output setpoint profile and the predicted increment of change in control action, respectively. The terms $Q = Q^T > 0$ and $R = R^T > 0$ are the weighting matrices. Minimizing the cost function (17) is usually subject to the operational constraints on the control action, its rate of change and plant output signals [21].

To create an MPC controller under the LabVIEW/CDSim environment, the CD Create MPC Controller VI of Fig. 12 is used. This VI bases the MPC controller on a state-space model of the controlled plant. The prediction and control horizons must be provided in the MPC Controller Parameters input of the CD Create MPC Controller VI. These predictive control parameters, as shown in Fig. 1, are fixed for the duration of the execution of the controller. MPC State Estimator Parameters of this VI specifies the parameters of the state estimator that the MPC Controller uses to estimate the states of the plant.

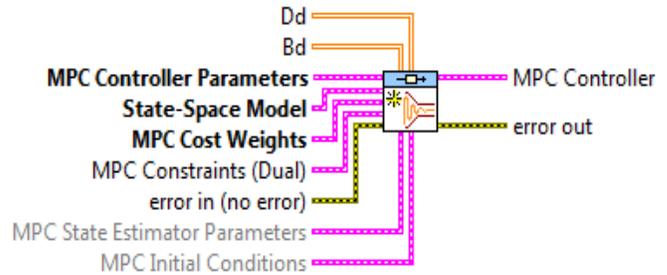


Fig. 12. LabVIEW CD Create MPC Controller VI

Furthermore, the created MPC controller is now implemented either in the co-simulation, or in a real-world scenario, while using the CD Implement MPC Controller VI of Fig. 13 within a timed or simulation loops.

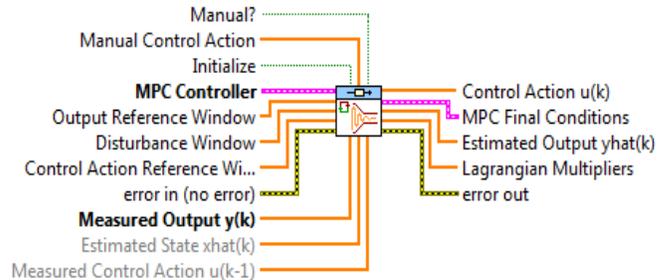


Fig. 13. LabVIEW CD Implement MPC Controller VI

After the MPC algorithm implementation for the studied Quadrotor, we give in Fig. 14 the developed LabVIEW diagram for the rapid MPC prototyping. The demonstrative co-simulation results are depicted in Fig. 15 and Fig. 16 for the position and attitude control prototyping, respectively. In order to show the controlled flight behavior of the co-simulated Quadrotor, a 3D prototype is built and animated under LabVIEW environment as depicted in Fig. 17. It can be observed that such a 3D prototype behaves properly in real-time according to the variation of the control setpoint inputs. This result improves further the effectiveness and the validity of our proposed HW/SW co-simulation solution.

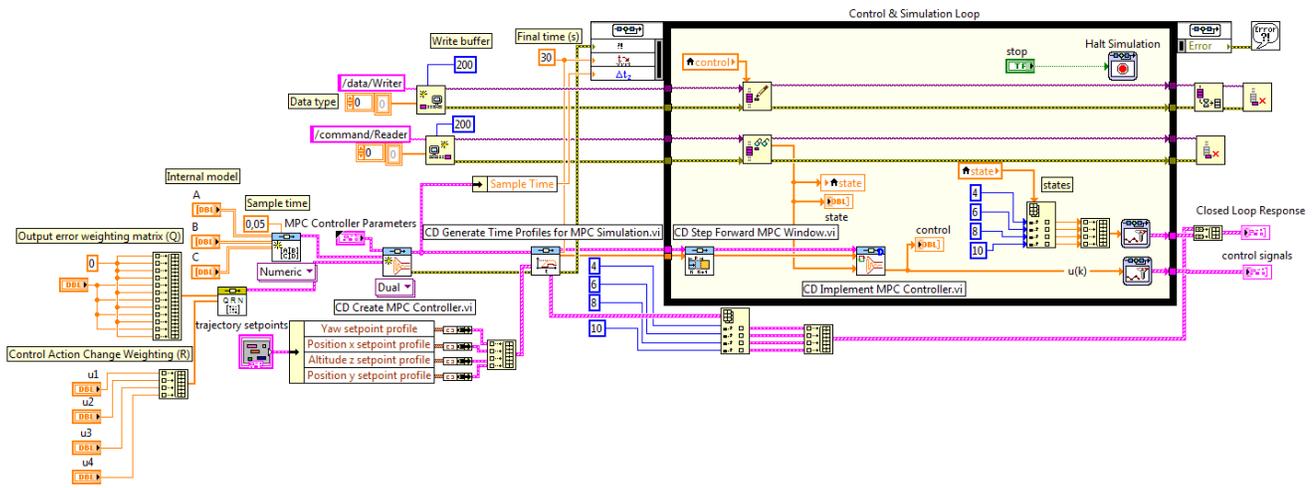


Fig. 14. LabVIEW diagram for the Network Streams “Client” based implementation of the MPC algorithm model

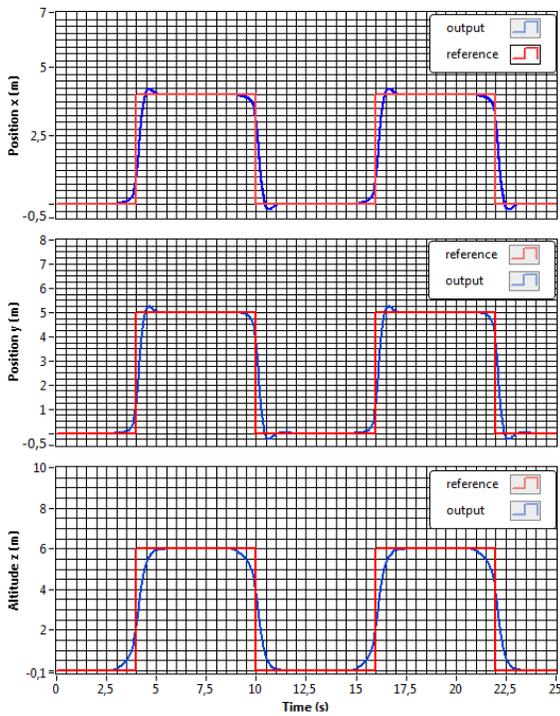


Fig. 15. MPC prototyping results for the Quadrotor position

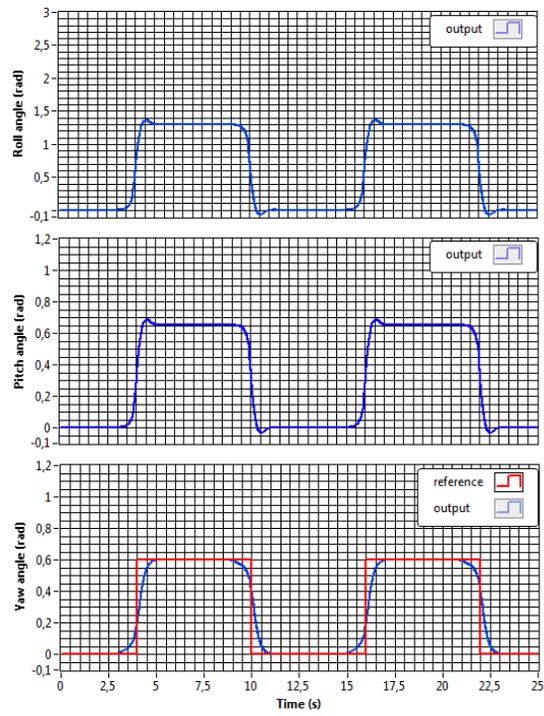


Fig. 16. MPC prototyping results for the Quadrotor attitude

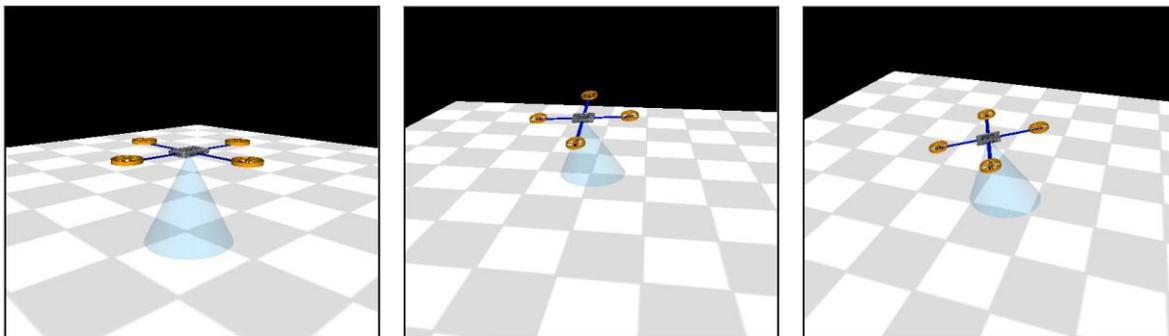


Fig. 17. Flight motion illustrations of the co-simulated Quadrotor in a 3D frame

V. CONCLUSION

In this paper, a new CAD methodology for PIL co-simulation is proposed and successfully implemented for the rapid flight control prototyping of a Quadrotor UAV. The proposed SW/HW solution is based on an embedded NI myRIO-1900 platform and a host PC equipped with the LabVIEW/CDSim graphical programming environment. An efficient set-up Network Streams data communication protocol is further established for such MBD platform. The nonlinear dynamic model of the Quadrotor as well as those of PID and MPC algorithms are implemented and deployed on two separate VIs within a LabVIEW RT myRIO project thanks to the developed and given LabVIEW patterns for the host PC and deployed embedded target parts. All hardware co-simulation results, obtained for a built 3D prototype of the studied rotorcraft, show the effectiveness of our proposed myRIO-based prototyping platform. These results improve further the low cost and simplicity of the future real-world implementation of the prototyped control laws while using the same embedded NI myRIO-1900 platform.

REFERENCES

- [1]. R. Lozano (Ed.), *Unmanned aerial vehicles: Embedded control*, John Wiley & Sons, ISTE, UK, 2010.
- [2]. R. Austin, *Unmanned Aircraft Systems: UAVs design, development and deployment*, 1st edition, John Wiley & Sons Ltd, London, 2010.
- [3]. K. Nonami, F. Kendoul, S. Suzuki, W. Wang and D. Nakazawa, *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*, Springer, New York, 2010.
- [4]. F. Torres, A. Rabhi, D. Lara, G. Romero and C. Pégard, Fuzzy State Feedback for Attitude Stabilization of Quadrotor, *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, DOI: 10.5772/61934, 2016.
- [5]. T. Espinoza, A. Dzul and M. Llama, Linear and Nonlinear Controllers Applied to Fixed-Wing UAV, *International Journal of Advanced Robotic Systems*, vol. 10, DOI: 10.5772/53616, 2013.
- [6]. K.P. Valavanis (Ed.), *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, 1st edition, Springer Netherlands, New York, 2007.
- [7]. L.M. Argentim, W.C. Rezende, P.E. Santos and R.A. Aguiar, PID, LQR and LQR-PID on a Quadcopter platform, In *the 2013 International Conference on Informatics, Electronics & Vision*, pp. 1-6, Dhaka, Bangladesh, May 2013.
- [8]. S. Khatoun, D. Gupta, and L.K. Das, PID & LQR control for a Quadrotor: Modeling and simulation, In *the 2014 International Conference on Advances in Computing, Communications and Informatics*, pp. 796-802, New Delhi, September 2014.
- [9]. J. Han, L. Di, C. Coopmans and Y.Q. Chen, Pitch Loop Control of a VTOL UAV Using Fractional Order Controller, *International Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 187-195, 2014.
- [10]. H. Liu, Y. Bai, G. Lu, Z. Shi and Y. Zhong, Robust Tracking Control of a Quadrotor Helicopter, *International Journal of Intelligent & Robotic Systems*, vol. 75, no. 3, pp. 595-608, 2014.
- [11]. L. Besnard, Y.B. Shtessel and B. Landrum, Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer, *Journal of the Franklin Institute*, vol. 349, no. 2012, pp. 658-684, 2011.
- [12]. E-H. Zheng, J-J. Xiong and J-L. Luo, Second order sliding mode control for a Quadrotor UAV, *ISA Transactions*, vol. 53, no. 4, pp. 1350-1356, 2014.
- [13]. J-J. Xiong and E-H. Zheng, Position and attitude tracking control for a Quadrotor UAV, *ISA Transactions*, vol. 53, no. 3, pp. 725-731, 2014.
- [14]. G.V. Raffo, M.G. Ortega and F.R. Rubio. Backstepping/nonlinear H_{∞} control for path tracking of a Quadrotor unmanned aerial vehicle, In *the American control conference*, pp. 3356-3361, Seattle, 2008.
- [15]. X. Lei, S.S. Ge and J. Fang, Adaptive Neural Network Control of Small Unmanned Aerial Rotorcraft, *International Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 331-341, 2014.
- [16]. K. Alexis, G. Nikolakopoulos and A. Tzes, Model predictive Quadrotor control: attitude, altitude and position experimental studies, *IET Control Theory Applications*, vol. 6, no. 12, pp. 1812-1827, 2012.
- [17]. G.V. Raffo, M.G. Ortega and F.R. Rubio. An integral predictive/nonlinear H_{∞} control structure for a Quadrotor helicopter, *Automatica*, vol. 46, no. 2010, pp. 29-39, 2010.
- [18]. N. Slegers, J. Kyle and Mark Costello, Nonlinear Model Predictive Control Technique for Unmanned Air Vehicles, *Journal of Guidance, Control and Dynamics*, vol. 29, no. 5, pp. 1179-1188, 2006.
- [19]. R.V. Lopes, P.H. Rodrigues, G. Borges and J. Ishihara, Model Predictive Control applied to tracking and attitude stabilization of a VTOL Quadrotor aircraft, *ABCM Symposium Series in Mechatronics*, vol. 5, pp. 176-185, 2012.
- [20]. National Instruments, *NI myRIO-1900: User Guide and Specifications*, 376047A-01, www.ni.com, Accessed on 05/September/2015.
- [21]. National Instruments, *LabVIEW Control Design User Manual*, 371057G-01, www.ni.com, Accessed on 13/July/2015.
- [22]. National Instruments, *NI LabVIEW for Compact RIO Developer's Guide*, www.ni.com, Accessed on 16/September/2015.
- [23]. K.J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, Research Triangle Park, North Carolina, 1995.
- [24]. J.B. Rawlings, D.Q. Mayne, *Model Predictive Control: Theory and Design*, Nob Hill Publishing, Madison, USA, 2013.
- [25]. J.A. Rossiter, *Model Based Predictive Control: A practical Approach*, Control Series, CRC Press, New York, 2004.