Proceedings of the ASME 2010 International Mechanical Engineering Congress & Exposition
IMECE2010
November 12-18, 2010, Vancouver, British Columbia, Canada

IMECE2010-38384

AN ITERATIVE FEEDBACK TUNING SCHEME FOR THE MULTILOOP PID CONTROL OF A UAV

Juan A. Ramirez
Dept. of Mechanical Engineering
Universidad Pontificia Bolivariana
P.O. Box 56006
Medellín, Colombia
juan.ramirezm@correo.upb.edu.co

Luis B. Gutierrez
Dept. of Electrical Engineering
Universidad Pontificia Bolivariana
P.O. Box 56006
Medellín, Colombia
lbgutie@ieee.org

Rafael E. Vasquez[*]
Dept. of Mechanical & Aerospace Engineering
University of Florida
P.O. Box 116300
Gainesville, Florida 32611
revasquez@ufl.edu[*]

ABSTRACT

An alternative for the steady-level flight control on an Unmanned Aerial Vehicle (UAV) is the use of decentralized multiloop PID controllers for each controlled variable. PIDs are linear structured low order controllers which are not so easy to tune in the presence of complex dynamics such as multivariable, non-minimum phase, oscillatory and high-order plants; and the use of conventional design techniques based on linearized models usually does not end in satisfactory results. In this work a design scheme based on iterative feedback tuning (IFT) for the multivariable nonlinear model of the Condor Andino UAV (Andean Condor UAV) is addressed. The method proposes the optimization of a quadratic performance target function using the closed loop response obtained from simulations. In this case the characteristics of the model are not used directly in the controller tuning process, but in simulations and in some other numeric manipulations. The optimization process is made using a modified version of the Levenberg-Marquardt algorithm. The simulation results show that a set of controller parameters can be found such that the target function has a local minimum.

Keywords: Controller Optimization; Iterative Feedback Tuning; Multiloop PID; UAV.

INTRODUCTION

The Automation and Design Research Group A+D from the Universidad Pontificia Bolivariana (Medellín, Colombia) has been developing the fixed-wing unmanned aerial vehicle Condor Andino, Fig. 1. The development process includes the design, construction, integration and test of the mechanical/electrical elements and the design, simulation and implementation of the control system which allows autonomous and radio-controlled flights. The vehicle has double tail-boom, 5-meter wingspan and is powered by a BT-64EI Fuji pusher engine with a 22-in diameter and 10-in pitch two-blade propeller.

Several advances have been made in the mechanical/aeronautical design [1], hardware and software architecture [2], modeling [3], simulation [4] and navigation [5] of the UAV. For the control system, a three-level architecture is used, based on the ideas shown in [6, 7]: at high-level the mission planner designs the trajectory in terms of a set of feasible waypoints, at mid-level the guidance system takes the waypoints and calculates filtered setpoints and, at low-level the controller follows the setpoints using the vehicle's state estimation and manipulating the control inputs. The design of the low-level control strategy is an open problem and several solutions can be used, from simple-linear to complex-nonlinear strategies. Perhaps the simplest effective solution when the system is multivariable is the use of decentralized PID (multiloop PID) controllers in some configuration, depending on the coupling between controlled variables and manipulated variables of the system, [8].

The behavior of the UAV is nonlinear, has several controlled variables and shows opposed-direction and coupled dynamics that lead to inverse response. Hence, the controller must be structured and multivariable, and its design process can be challeng-

[*] Address all correspondence to this author.

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

FIGURE 1. CONDOR ANDINO UAV

ing. Therefore, an alternative design technique based on the Iterative Feedback Tuning (IFT) idea addressed by Hjalmarsson [9] for linear SISO plants is proposed. This design scheme aims to the minimization of a quadratic target function using a gradient descent algorithm, based only on information from the closed loop response. The problem has been formulated for the linear MIMO case in [10], for the nonlinear SISO case in [11] and improved in [12]. This work addresses a design strategy for the low-level controller of a UAV, based on IFT for the MIMO nonlinear case.

DESIGN METHOD

Consider the nonlinear time-invariant plant described by the continuous-time model

$$\begin{aligned} \dot{x}(t) &= f(x(t),u(t)), \\ y(t) &= g(x(t),u(t)); \end{aligned} \tag{1}$$

where $x(t)$ is the state, $u(t)$ is the control input and $y(t)$ is the output. The effect of disturbances is neglected and the only exogenous variable considered is the controller reference signal. Suppose that we use a non-linear two-degree-of-freedom controller, which is defined by

$$\begin{aligned} \dot{x}_r(t) &= f_r(\rho,x_r(t),r(t)), \\ \dot{x}_y(t) &= f_y(\rho,x_y(t),y(t)), \\ u(t) &= g_r(\rho,x_r(t),r(t)) - g_y(\rho,x_y(t),y(t)); \end{aligned} \tag{2}$$

where $x_r(t)$ and $x_y(t)$ are the controller's state variables, $r(t)$ is the reference signal and ρ is a vector of tuning parameters. The closed-loop interconnection is depicted in Fig. 2. In Eqn. (2) the controller equations are parametrized because $x_r(t)$, $x_y(t)$, $u(t)$ and $y(t)$ are dependent on the values taken by ρ.

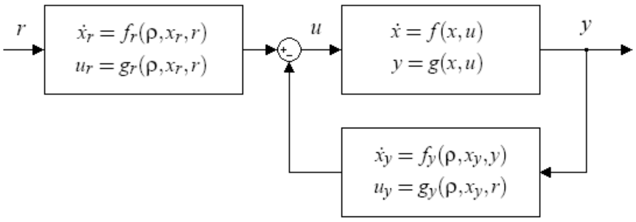

FIGURE 2. CLOSED-LOOP INTERCONNECTION

Now, the design objective is to find a ρ such that a quadratic performance target function has a local minimum. So, let us suppose that the target function is given by

$$J(\rho) = \frac{1}{2N}\sum_{k=1}^{N} \tilde{y}_k^T Q \tilde{y}_k + \frac{\lambda}{2N}\sum_{k=1}^{N} u_k^T R u_k, \tag{3}$$

where $\tilde{y} = y - y_d$ is the error between the desired response y_d and the actual response y; Q and R are constant weight matrices, λ is a scalar and k is the sample number. In Eqn. (3), J is calculated based on a discretization of $y(t)$. It is also possible to formulate J as a continuous time integral but it is easier to calculate it in discrete time if the information of y comes from simulations. However, this does not make a fundamental change in the problem formulation. The dependence of J on ρ in Eqn. (3) denotes that changes in J depends only on the particular value of ρ if the plant is assumed time-invariant and disturbance-free.

Now, using the definition in Eqn. (3) the design objective can be formulated as [9, 12]

$$\rho^* = \arg\min_{\rho} J(\rho), \tag{4}$$

subject to Eqns. (1) and (2).

Due to the nonlinear nature of the problem, the solution is obtained using the iterative gradient based local search algorithm

$$\rho_{i+1} = \rho_i - \eta_i H_i^{-1} J'(\rho_i), \tag{5}$$

where H is a positive definite matrix and J' is the gradient of J with respect to ρ. If $H = I$ the algorithm steps in the steepest descent direction, if $H = \partial^2 J(\rho)/\partial\rho^2$ is the exact Hessian, the Newton-Rhapson algorithm appears and if an approximation of the Hessian is used, the Gauss-Newton or Levenberg-Marquardt algorithms appear. As expected, the convergence properties are affected by the choice of H and η. The step size is defined by η

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

and can be determined using a sequence that meets the conditions

$$\sum_{i=1}^{\infty}\eta_i^2 < \infty, \quad \sum_{i=1}^{\infty}\eta_i = \infty. \tag{6}$$

A possible choice that satisfies Eqn. (6) is making $\eta_i = a/i$, with a being a positive constant.

A key element in the algorithm is determining the gradient J'. Hjalmarsson et al. [9] show how to deal with linear single-input single-output plants, Hjalmarsson and Birkeland [10] show how to deal with linear multivariable plants and Sjöberg et al. [11] show how to deal with nonlinear single-input single-output plants. In this case an alternative is proposed for the case in which the plant and/or the controller are multivariable and nonlinear, when a simulation model is available.

From Eqn. (3) the derivative of J with respect to ρ is given by

$$J'(\rho) = \frac{1}{N}\sum_{k=1}^{N}\left[\left(\frac{\partial y}{\partial \rho}\right)_k^T Q\tilde{y}_k + \lambda\left(\frac{\partial u}{\partial \rho}\right)_k^T R u_k\right], \tag{7}$$

considering that $\partial\tilde{y}/\partial\rho = \partial y/\partial\rho$ since y_d is independent of ρ.

If the Gauss-Newton algorithm is used, an approximation of the Hessian is given by

$$\hat{H}(\rho) = \frac{1}{N}\sum_{k=1}^{N}\left[\left(\frac{\partial y}{\partial \rho}\right)_k^T Q\left(\frac{\partial y}{\partial \rho}\right)_k + \lambda\left(\frac{\partial u}{\partial \rho}\right)_k^T R\left(\frac{\partial u}{\partial \rho}\right)_k\right], \tag{8}$$

and in Eqn. (5) $H_i = \hat{H}(\rho_i)$. However, as shown in [12], if the Levenberg-Marquardt algorithm is used then, in Eqn. (5),

$$H_i = \hat{H}(\rho_i) + \mu_i I; \tag{9}$$

where μ is a damping coefficient that regularizes the Hessian estimate, as proposed by Levenberg [13,14]. As suggested in [12], a starting value for the damping coefficient is

$$\mu_0 = \tau\max(\mathrm{diag}(\hat{H}_0)), \tag{10}$$

where τ is 10^{-6} for a good initial guess and 10^{-3} to 1 for a poor initial guess. The damping parameter can be updated based on a gain ratio that evaluates the quality of the previous step. The gain ratio ϑ is defined as the ratio between the actual and the expected J and is given by

$$\vartheta = \frac{J_{i-1} - J_i}{\frac{1}{2}h_i^T(\mu_i h_i - J_i)}, \tag{11}$$

where h_i is the step in the tuning parameters, given by

$$h_i = -\eta_i H_i^{-1} J(\rho_i).$$

The Levenberg-Marquardt algorithm updates the damping coefficient as follows

$$\mu_{i+1} = \begin{cases} 2\mu_i, & \vartheta \in (-\infty, 0.25] \\ \mu_i, & \vartheta \in (0.25, 0.75) \\ \mu_i/3, & \vartheta \in [0.74, \infty) \end{cases}. \tag{12}$$

Another option, as suggested in [12, 14], is given by a continuous version of (12)

$$\begin{aligned} &\vartheta < 0 \begin{cases} \mu_{i+1} = \mu_i \nu_i \\ \nu_{i+1} = 2\nu_i \end{cases}, \\ &\vartheta \geq 0 \begin{cases} \mu_{i+1} = \mu_i \max(\frac{1}{3}, 1 - (2\vartheta - 1)^3)\nu_i \\ \nu_{i+1} = 2 \end{cases}; \end{aligned} \tag{13}$$

where $\nu_0 = 2$.

As seen in Eqns. (7) and (8), the gradient and Hessian estimate depend on the Jacobians $\partial y/\partial\rho$ and $\partial u/\partial\rho$. These gradients are obtained by solving a system of matrix differential equations. The equation system is obtained partially differentiating Eqns. (1) and (2) with respect to ρ. The obtained system of equations is illustrated in Fig. 3 and given by

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial x}{\partial \rho}\right) &= \frac{\partial f}{\partial x}\left(\frac{\partial x}{\partial \rho}\right) + \frac{\partial f}{\partial u}\left(\frac{\partial u}{\partial \rho}\right), \\ \frac{\partial y}{\partial \rho} &= \frac{\partial g}{\partial x}\left(\frac{\partial x}{\partial \rho}\right) + \frac{\partial g}{\partial u}\left(\frac{\partial u}{\partial \rho}\right); \end{aligned} \tag{14}$$

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial x_r}{\partial \rho}\right) &= \nabla_\rho(f_r) + \frac{\partial f_r}{\partial x_r}\left(\frac{\partial x_r}{\partial \rho}\right), \\ \frac{d}{dt}\left(\frac{\partial x_y}{\partial \rho}\right) &= \nabla_\rho(f_y) + \frac{\partial f_y}{\partial x_y}\left(\frac{\partial x_y}{\partial \rho}\right) + \frac{\partial f_y}{\partial y}\left(\frac{\partial y}{\partial \rho}\right), \\ \frac{\partial u}{\partial \rho} &= \nabla_\rho(g_r) - \nabla_\rho(g_y) + \frac{\partial g_r}{\partial x_r}\left(\frac{\partial x_r}{\partial \rho}\right) \\ &\quad - \left[\frac{\partial g_y}{\partial x_y}\left(\frac{\partial x_y}{\partial \rho}\right) + \frac{\partial g_y}{\partial y}\left(\frac{\partial y}{\partial \rho}\right)\right]; \end{aligned} \tag{15}$$

where $\nabla_\rho(f) = \left[\frac{\partial f}{\partial \rho_1}\ \frac{\partial f}{\partial \rho_2}\ \cdots\ \frac{\partial f}{\partial \rho_n}\right]$.

As expected, the ODE system given by Eqns. (1), (2), (14) and (15) can be solved simultaneously in order to obtain $\partial y/\partial\rho$ and $\partial u/\partial\rho$ when a particular ρ is chosen. Then, with $\partial y/\partial\rho$ and $\partial u/\partial\rho$ known, J' and $\hat{H}$ can be calculated in order to solve Eqn. (4) through the algorithm in Eqn. (5) by using any of the possibilities to choose H (steepest descent, Newton-Rhapson, Gauss-Newton or Levenberg-Marquardt) [12].

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

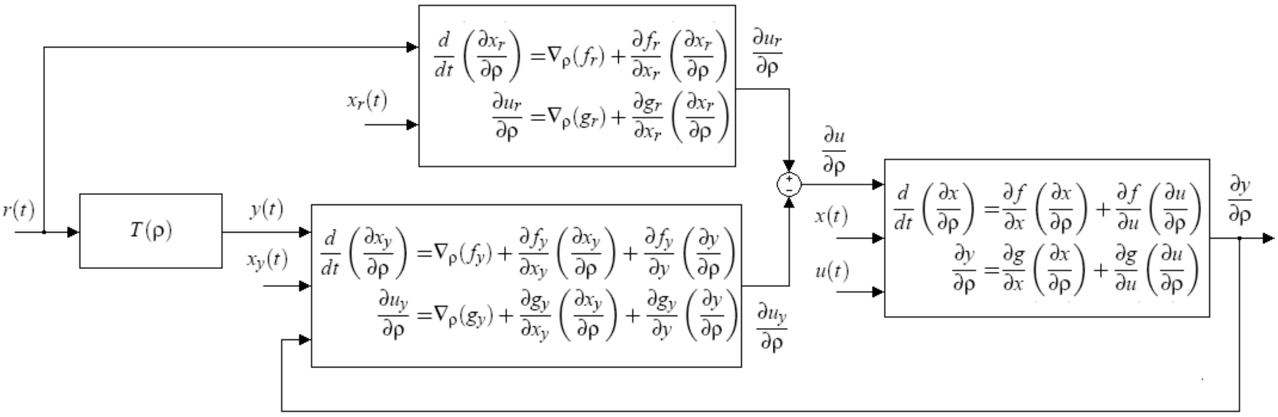
$\frac{d}{dt}\left(\frac{\partial x_r}{\partial\rho}\right) = \nabla_\rho(f_r) + \frac{\partial f_r}{\partial x_r}\left(\frac{\partial x_r}{\partial\rho}\right)$
$\frac{\partial u_r}{\partial\rho} = \nabla_\rho(g_r) + \frac{\partial g_r}{\partial x_r}\left(\frac{\partial x_r}{\partial\rho}\right)$
$\frac{\partial u_r}{\partial\rho}$
$x_r(t)$
$\frac{\partial u}{\partial\rho}$
$r(t)$
$T(\rho)$
$y(t)$
$x_y(t)$
$\frac{d}{dt}\left(\frac{\partial x_y}{\partial\rho}\right) = \nabla_\rho(f_y) + \frac{\partial f_y}{\partial x_y}\left(\frac{\partial x_y}{\partial\rho}\right) + \frac{\partial f_y}{\partial y}\left(\frac{\partial y}{\partial\rho}\right)$
$\frac{\partial u_y}{\partial\rho} = \nabla_\rho(g_y) + \frac{\partial g_y}{\partial x_y}\left(\frac{\partial x_y}{\partial\rho}\right) + \frac{\partial g_y}{\partial y}\left(\frac{\partial y}{\partial\rho}\right)$
$\frac{\partial u_y}{\partial\rho}$
+
-
$x(t)$
$u(t)$
$\frac{d}{dt}\left(\frac{\partial x}{\partial\rho}\right) = \frac{\partial f}{\partial x}\left(\frac{\partial x}{\partial\rho}\right) + \frac{\partial f}{\partial u}\left(\frac{\partial u}{\partial\rho}\right)$
$\frac{\partial y}{\partial\rho} = \frac{\partial g}{\partial x}\left(\frac{\partial x}{\partial\rho}\right) + \frac{\partial g}{\partial u}\left(\frac{\partial u}{\partial\rho}\right)$
$\frac{\partial y}{\partial\rho}$

FIGURE 3. MATRIX ODE SYSTEM

AIRCRAFT MODEL

Two different models of the aircraft's dynamics are used, one for simulation and one for control design. The simulation model used to test the controllers is a complete model that considers the rigid-body dynamics, the nonlinear behavior of aerodynamic forces and moments, and a blade element theory model of the thrust forces and moments in the propeller. The model used in the control design process is a reduced model that only considers the uncoupled longitudinal/lateral dynamics of the aircraft.

Both models use the following nomenclature:

- m is the mass and J_x, J_y, J_z, J_{xz} are the moments and products of inertia, and $\Gamma = J_x J_z - J_{xz}^2$;
- ϕ, θ, ψ are the roll, pitch and yaw Euler's angles respectively;
- α, β are the angle of attack and sideslip angle respectively;
- U, V, W are the body frame $x-y-z$ velocity components;
- p, q, r are the roll, pitch and yaw angular rates respectively;
- D, Y, L are the drag, crossforce and lift forces respectively;
- l, m, n are the roll, pitch and yaw moments;
- T is the thrust force;
- τ_m is the propeller's torque;
- ω_m is the propeller's angular speed;
- J_p is the propeller/motor's moment of inertia; and
- g is the acceleration due to gravity.

The simulation model uses flat-earth dynamic equations defined in the body frame [15] given by

$$\begin{aligned}\dot U =& \frac{1}{\mathrm{m}}\left(-D\cos\alpha\cos\beta - Y\cos\alpha\sin\beta + L\sin\alpha + T\right)\\ &- g\sin\theta - (qW - rV),\end{aligned} \tag{16}$$

$$\dot V = \frac{1}{\mathrm{m}}\left(-D\sin\beta + Y\cos\beta\right) + g\sin\phi\cos\theta - (rU - pW), \tag{17}$$

$$\begin{aligned}\dot W =& \frac{1}{\mathrm{m}}\left(-D\sin\alpha\cos\beta - Y\sin\alpha\sin\beta - L\cos\alpha\right)\\ &+ g\cos\phi\cos\theta - (pV - qU),\end{aligned} \tag{18}$$

$$\begin{aligned}\dot p =& \frac{1}{\Gamma}\left[(J_z\mathrm{c}\alpha + J_{xz}\mathrm{s}\alpha)(l\mathrm{c}\beta - m\mathrm{s}\beta) - (J_z\mathrm{s}\alpha + J_{xz}\mathrm{c}\alpha)n\right.\\ &- \left(J_z(J_z - J_y) + J_{xz}^2\right)qr + J_{xz}(J_x - J_y + J_z)pq\\ &\left.+ J_z\tau_m + J_{xz}J_p q\omega_m\right],\end{aligned} \tag{19}$$

$$\begin{aligned}\dot q =& \frac{1}{J_y}\left[l\mathrm{s}\beta + m\mathrm{c}\beta + (J_z - J_x)pr - J_{xz}\left(p^2 - r^2\right)\right.\\ &\left.- J_p r\omega_m\right],\end{aligned} \tag{20}$$

$$\begin{aligned}\dot r =& \frac{1}{\Gamma}\left[(J_z\mathrm{s}\alpha + J_{xz}\mathrm{c}\alpha)(l\mathrm{c}\beta - m\mathrm{s}\beta) - (J_{xz}\mathrm{s}\alpha - J_x\mathrm{c}\alpha)n\right.\\ &- \left(J_x(J_x - J_y) + J_{xz}^2\right)pq - J_{xz}(J_x - J_y + J_z)qr\\ &\left.+ J_{xz}\tau_m + J_x J_p q\omega_m\right];\end{aligned} \tag{21}$$

The kinematic equations are given by

$$\dot\phi = p + q\tan\theta\sin\phi + r\tan\theta\cos\phi, \tag{22}$$

$$\dot\theta = q\cos\phi - r\sin\phi, \tag{23}$$

$$\dot\psi = q\frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta}. \tag{24}$$

The translational dynamic equations used in controller design are defined in the wind frame, *i.e.* using V_T, α and β instead of U, V and W, assuming zero wind velocity. Let us define

$$\begin{bmatrix} V_T \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{U^2 + V^2 + W^2} \\ \mathrm{atan2}(W, U) \\ \mathrm{atan2}\left(V, \sqrt{U^2 + W^2}\right) \end{bmatrix}. \tag{25}$$

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

Then, taking the time-derivative of Eqn. (25) and substituting into Eqns. (16) - (18) yields

$$\begin{aligned} \mathrm{m}\dot{V}_T =& -D + T\mathrm{c}\alpha\mathrm{c}\beta \\ & + \mathrm{m}g\left(-\mathrm{c}\alpha\mathrm{c}\beta\mathrm{s}\theta + \mathrm{s}\beta\mathrm{s}\phi\mathrm{c}\theta + \mathrm{s}\alpha\mathrm{c}\beta\mathrm{c}\phi\mathrm{c}\theta\right), \end{aligned} \tag{26}$$

$$\begin{aligned} \mathrm{m}\mathrm{c}\beta V_T\dot{\alpha} =& -L - T\mathrm{s}\alpha + \mathrm{m}g\left(\mathrm{s}\alpha\mathrm{s}\theta + \mathrm{c}\alpha\mathrm{c}\phi\mathrm{c}\theta\right) \\ & + \mathrm{m}V_T\left(q\mathrm{c}\beta - \left(p\mathrm{c}\alpha + r\mathrm{s}\alpha\right)\mathrm{s}\beta\right), \end{aligned} \tag{27}$$

$$\begin{aligned} \mathrm{m}V_T\dot{\beta} =& Y - T\mathrm{c}\alpha\mathrm{s}\beta \\ & + \mathrm{m}g\left(\mathrm{c}\alpha\mathrm{s}\beta\mathrm{s}\theta + \mathrm{c}\beta\mathrm{s}\phi\mathrm{c}\theta - \mathrm{s}\alpha\mathrm{s}\beta\mathrm{c}\phi\mathrm{c}\theta\right) \\ & - \mathrm{m}V_T\left(r\mathrm{c}\alpha - p\mathrm{s}\alpha\right). \end{aligned} \tag{28}$$

As mentioned, the uncoupled equations are defined in the longitudinal and lateral modes: the longitudinal mode assumes that $\beta \equiv p \equiv r \equiv \phi \equiv 0$ and the lateral mode assumes that $\dot{V}_T \equiv \dot{\alpha} \equiv q \equiv 0$. Thus, the longitudinal dynamics equations obtained from (26), (27), (20) and (23) are given by

$$\mathrm{m}\dot{V}_T = -D + T\cos\alpha + \mathrm{m}g\left(-\cos\alpha\sin\theta + \sin\alpha\cos\theta\right), \tag{29}$$

$$\begin{aligned} \mathrm{m}V_T\dot{\alpha} =& -L - T\sin\alpha + \mathrm{m}g\left(\sin\alpha\sin\theta + \cos\alpha\cos\theta\right) \\ & + \mathrm{m}V_T q, \end{aligned} \tag{30}$$

$$\dot{q} = \frac{m}{J_y}, \tag{31}$$

$$\dot{\theta} = q, \tag{32}$$

$$\dot{h} = V_T\left(\cos\alpha\sin\theta - \sin\alpha\cos\theta\right), \tag{33}$$

and the lateral dynamics equations obtained from (28), (19), (21), (22) and (24) are given by

$$\begin{aligned} \mathrm{m}V_{T0}\dot{\beta} =& Y - T_0\mathrm{c}\alpha_0\mathrm{s}\beta \\ & + \mathrm{m}g\left(\mathrm{c}\alpha_0\mathrm{s}\beta\mathrm{s}\theta_0 + \mathrm{c}\beta\mathrm{s}\phi\mathrm{c}\theta_0 - \mathrm{s}\alpha_0\mathrm{s}\beta\mathrm{c}\phi\mathrm{c}\theta_0\right) \\ & - \mathrm{m}V_{T0}\left(r\mathrm{c}\alpha_0 - p\mathrm{s}\alpha_0\right), \end{aligned} \tag{34}$$

$$\begin{aligned} \dot{p} =& \frac{1}{\Gamma}\left[\left(J_z\mathrm{c}\alpha_0 + J_{xz}\mathrm{s}\alpha_0\right)\left(l\mathrm{c}\beta - m_0\mathrm{s}\beta\right)\right. \\ & \left. - \left(J_z\mathrm{s}\alpha_0 + J_{xz}\mathrm{c}\alpha_0\right)n + J_z\tau_m\right], \end{aligned} \tag{35}$$

$$\begin{aligned} \dot{r} =& \frac{1}{\Gamma}\left[\left(J_z\mathrm{s}\alpha_0 + J_{xz}\mathrm{c}\alpha_0\right)\left(l\mathrm{c}\beta - m_0\mathrm{s}\beta\right)\right. \\ & \left. - \left(J_{xz}\mathrm{s}\alpha_0 - J_x\mathrm{c}\alpha_0\right)n + J_{xz}\tau_m\right], \end{aligned} \tag{36}$$

$$\dot{\phi} = p + r\tan\theta_0\cos\phi, \tag{37}$$

$$\dot{\psi} = r\frac{\cos\phi}{\cos\theta_0}. \tag{38}$$

Equations (29) to (38) can be used to obtain two state space models, one for each mode. For the longitudinal mode

$$\begin{aligned} \dot{x}_{\mathrm{long}} &= f_{\mathrm{long}}(x_{\mathrm{long}}, u_{\mathrm{long}}), \\ y_{\mathrm{long}} &= \left[\, V_T \;\; h \;\; \theta - \alpha \;\; q \,\right]^{\mathrm{T}}, \end{aligned} \tag{39}$$

with $x_{\mathrm{long}} = \left[\, V_T \;\; \alpha \;\; q \;\; \theta \;\; h \,\right]^T$ and $u_{\mathrm{long}} = \left[\, \delta_\omega \;\; \delta_E \,\right]^T$; and for the lateral mode

$$\begin{aligned} \dot{x}_{\mathrm{lat}} &= f_{\mathrm{lat}}(x_{\mathrm{lat}}, u_{\mathrm{lat}}), \\ y_{\mathrm{lat}} &= \left[\, \psi \;\; \beta \;\; r \;\; \phi \;\; p \,\right]^{\mathrm{T}}, \end{aligned} \tag{40}$$

with $x_{\mathrm{lat}} = \left[\, \beta \;\; p \;\; r \;\; \phi \;\; \psi \,\right]^T$ and $u_{\mathrm{lat}} = \left[\, \delta_A \;\; \delta_R \,\right]^T$.

PROPOSED CONTROLLERS

The proposed controllers are multiloop PIDs, this means they are structured linear low-order controllers whose structure is defined based on the knowledge of the system's dynamics.

The longitudinal mode controller has the structure shown in Fig. 4 (left), and is defined by the state space equations

$${}^{\mathrm{long}}\dot{x}_c = \begin{bmatrix} {}^{\mathrm{vel}}k_2\left(V_{Td} - V_T\right) \\ {}^{\mathrm{alt}}k_2\left(h_d - h\right) \end{bmatrix}, \tag{41}$$

$$u_{\mathrm{long}} = \begin{bmatrix} {}^{\mathrm{vel}}x_c + {}^{\mathrm{vel}}k_1\left(V_{Td} - V_T\right) \\ {}^{\mathrm{alt}}x_c + {}^{\mathrm{alt}}k_1\left(h_d - h\right) - {}^{\mathrm{alt}}k_3\gamma - {}^{\mathrm{alt}}k_4 q \end{bmatrix}; \tag{42}$$

where ${}^{\mathrm{alt}}k = \left[\, {}^{\mathrm{alt}}k_1 \;\; {}^{\mathrm{alt}}k_2 \;\; {}^{\mathrm{alt}}k_3 \;\; {}^{\mathrm{alt}}k_4 \,\right]^T$ and ${}^{\mathrm{vel}}k = \left[\, {}^{\mathrm{vel}}k_1 \;\; {}^{\mathrm{vel}}k_2 \,\right]^{\mathrm{T}}$ are the tuning parameters; ${}^{\mathrm{long}}x_c = \left[\, {}^{\mathrm{vel}}x_c \;\; {}^{\mathrm{alt}}x_c \,\right]^T$ is the controller state; V_{Td}, h_d are the reference (set-point) values; α is the angle of attack, θ is the pitch angle, q is the pitch rate and $\gamma = \theta - \alpha$ is the flight-path angle. This controller has two loops: a PI velocity feedback loop and an altitude 3-loop P-P-PI which uses the fight-path angle, the pitch rate and the altitude feedback.

The lateral mode controller has the structure shown in Fig. 4 (right), and is defined by the state space equations

$${}^{\mathrm{lat}}\dot{x}_c = \begin{bmatrix} {}^{1}k_2 \\ {}^{2}k_2 \end{bmatrix}\left(\psi_d - \psi\right), \tag{43}$$

$$u_{\mathrm{lat}} = \begin{bmatrix} {}^{\mathrm{lat}}x_{c1} + {}^{1}k_1\left(\psi_d - \psi\right) - {}^{1}k_3\phi - {}^{1}k_4 p \\ {}^{\mathrm{lat}}x_{c2} + {}^{2}k_1\left(\psi_d - \psi\right) - {}^{2}k_3\beta - {}^{2}k_4 r \end{bmatrix}; \tag{44}$$

where ${}^{1}k_{\mathrm{lat}} = \left[\, {}^{1}k_1 \;\; {}^{1}k_2 \;\; {}^{1}k_3 \;\; {}^{1}k_4 \,\right]^{\mathrm{T}}$ and ${}^{2}k_{\mathrm{lat}} = \left[\, {}^{2}k_1 \;\; {}^{2}k_2 \;\; {}^{2}k_3 \;\; {}^{2}k_4 \,\right]^{\mathrm{T}}$ are the tuning parameters, ${}^{\mathrm{lat}}x_c = \left[\, {}^{\mathrm{lat}}x_{c1} \;\; {}^{\mathrm{lat}}x_{c2} \,\right]^{\mathrm{T}}$ is the controller state, ψ_d is the reference (set-point) value, ψ is the heading, ϕ is the bank angle, β is the angle of sideslip, p is the roll rate and r is the yaw rate. This controller has four P inner loops which use the bank angle, the sideslip angle, the roll rate and the yaw rate feedback and two PI heading feedback outer loops.

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

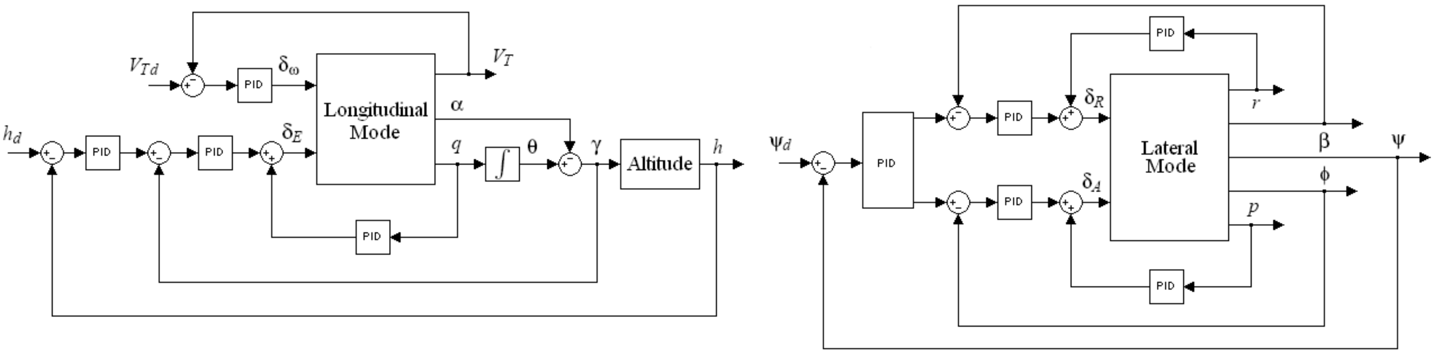

FIGURE 4. PROPOSED CONTROLLERS: LONGITUDINAL MODE (LEFT), LATERAL MODE (RIGHT)

CONTROLLER DESIGN

The controller parameter optimization with the iterative algorithm shown in Eqn. (5), using Eqn. (7) and Eqns. (8)/(9), requires solving systems in Eqns. (14) and (15). The equations with the form of Eqn. (14) are obtained from the model of Eqns. (29)-(33)/(39) and from Eqns. (34)-(38)/(40) analytically or numerically using *v.g.* centered differences. In this work, fourth-order centered differences are used. Equations (15) are obtained from the controller Eqns. (41)-(42) and (43)-(44), analytically, taking the partial derivative with respect to ρ. These equations depend on the selected particular controller structure.

Longitudinal mode optimization

The optimization of the longitudinal controller is done in two steps. First, the altitude loop parameters ${}^{\mathrm{alt}}k$ are optimized while the velocity loop parameters ${}^{\mathrm{vel}}k$ are held constant, through the response to an exponential set-point change in the desired altitude h_d. Second, the velocity loop parameters ${}^{\mathrm{vel}}k$ are optimized while the altitude loop parameters ${}^{\mathrm{alt}}k$ are held constant, through the response to an exponential set-point in the desired altitude. In both cases the velocity set-point V_{Td} is held constant.

In the altitude case, taking the partial derivative of Eqns. (41)-(42) with respect to ${}^{\mathrm{alt}}k$, Eqns. (15) and (14) become

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial^{\mathrm{alt}}x_c}{\partial^{\mathrm{alt}}k}\right) &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & h_d-h & 0 & 0 \end{bmatrix} \\ &\quad - \begin{bmatrix} {}^{\mathrm{vel}}k_2 & 0 & 0 & 0 \\ 0 & {}^{\mathrm{alt}}k_2 & 0 & 0 \end{bmatrix} \frac{\partial y_{\mathrm{long}}}{\partial^{\mathrm{alt}}k},\end{aligned} \tag{45}$$

$$\begin{aligned}\frac{\partial u_{\mathrm{long}}}{\partial^{\mathrm{alt}}k} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ h_d-h & 0 & -\gamma & -q \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{\partial^{\mathrm{alt}}x_c}{\partial^{\mathrm{alt}}k} \\ &\quad - \begin{bmatrix} {}^{\mathrm{vel}}k_1 & 0 & 0 & 0 \\ 0 & {}^{\mathrm{alt}}k_1 & {}^{\mathrm{alt}}k_3 & {}^{\mathrm{alt}}k_4 \end{bmatrix} \frac{\partial y_{\mathrm{long}}}{\partial^{\mathrm{alt}}k};\end{aligned} \tag{46}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial x_{\mathrm{long}}}{\partial^{\mathrm{alt}}k}\right) =& \frac{\partial f_{\mathrm{long}}\left(x_{\mathrm{long}},u_{\mathrm{long}}\right)}{\partial x_{\mathrm{long}}}\left(\frac{\partial x_{\mathrm{long}}}{\partial^{\mathrm{alt}}k}\right) \\ &+ \frac{\partial f_{\mathrm{long}}\left(x_{\mathrm{long}},u_{\mathrm{long}}\right)}{\partial u_{\mathrm{long}}}\left(\frac{\partial u_{\mathrm{long}}}{\partial^{\mathrm{alt}}k}\right),\end{aligned} \tag{47}$$

$$\frac{\partial y_{\mathrm{long}}}{\partial^{\mathrm{alt}}k} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}\left(\frac{\partial x_{\mathrm{long}}}{\partial^{\mathrm{alt}}k}\right). \tag{48}$$

In the design process this loop uses the target function

$$J_{\mathrm{alt}} = \frac{1}{2N}\sum_{n=1}^{N}\left[\tilde{y}_{\mathrm{long}n}^{\mathrm{T}}\,{}^{\mathrm{alt}}Q\tilde{y}_{\mathrm{long}n} + \lambda u_{\mathrm{long}n}^{\mathrm{T}}\,{}^{\mathrm{alt}}Ru_{\mathrm{long}n}\right];$$

with weights

$${}^{\mathrm{alt}}Q = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{\mathrm{alt}}R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \lambda = 1;$$

initial controller parameters ${}^{\mathrm{alt}}k = \begin{bmatrix} -0.5 & -0.01 & -2 & -2 \end{bmatrix}^{\mathrm{T}}$; fixed velocity controller parameters ${}^{\mathrm{vel}}k_0 = \begin{bmatrix} 0.5 & 0.1 \end{bmatrix}^{\mathrm{T}}$ and an exponential altitude 5-meter set-point change with a time constant of 5 seconds. Hence, when the Levenberg-Marquardt optimization algorithm is implemented, the target function and its gradient behaves as shown in Fig. 5 and the evolution of the time response of altitude and velocity is shown in Fig. 6.

The final controller parameters are given by

$${}^{\mathrm{alt}}k = \begin{bmatrix} -0.1251 & -0.0084 & -2.0032 & -1.9940 \end{bmatrix}.$$

As it can be seen from the results, the altitude overshoot and settling time are reduced but the velocity peak is big.

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

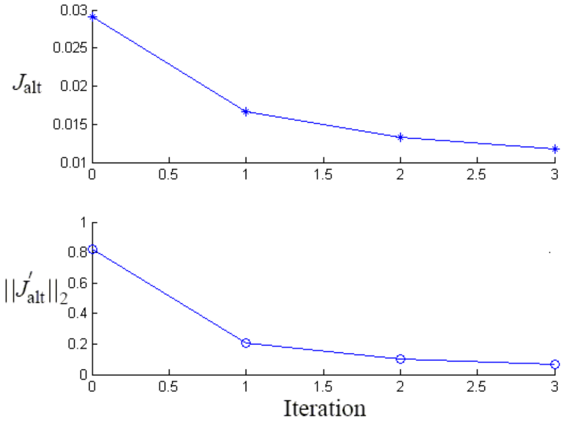

FIGURE 5. ALTITUDE CONTROL TARGET FUNCTION AND GRADIENT NORM

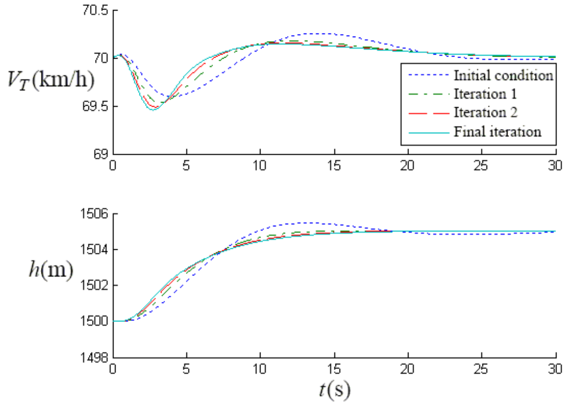

FIGURE 6. TIME RESPONSE OF ALTITUDE AND VELOCITY IN THE ALTITUDE LOOP OPTIMIZATION

In the velocity case, taking the partial derivative of Eqns. (41)-(42) with respect to ${}^{\mathrm{vel}}k$, Eqns. (15) and (14) become

$$\frac{d}{dt}\left(\frac{\partial\,{}^{\mathrm{vel}}x_c}{\partial\,{}^{\mathrm{vel}}k}\right) = \begin{bmatrix} 0 & V_{Td}-V_T \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} {}^{\mathrm{vel}}k_2 & 0 & 0 & 0 \\ 0 & {}^{\mathrm{alt}}k_2 & 0 & 0 \end{bmatrix} \frac{\partial y_{\mathrm{long}}}{\partial\,{}^{\mathrm{vel}}k}, \tag{49}$$

$$\begin{aligned} \frac{\partial u_{\mathrm{long}}}{\partial\,{}^{\mathrm{vel}}k} = {} & \begin{bmatrix} V_{Td}-V_T & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{\partial\,{}^{\mathrm{vel}}x_c}{\partial\,{}^{\mathrm{vel}}k} \\ & - \begin{bmatrix} {}^{\mathrm{vel}}k_1 & 0 & 0 & 0 \\ 0 & {}^{\mathrm{alt}}k_1 & {}^{\mathrm{alt}}k_3 & {}^{\mathrm{alt}}k_4 \end{bmatrix} \frac{\partial y_{\mathrm{long}}}{\partial\,{}^{\mathrm{vel}}k}; \end{aligned} \tag{50}$$

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial x_{\mathrm{long}}}{\partial\,{}^{\mathrm{alt}}k}\right) = {} & \frac{\partial f_{\mathrm{long}}\left(x_{\mathrm{long}}, u_{\mathrm{long}}\right)}{\partial x_{\mathrm{long}}} \left(\frac{\partial x_{\mathrm{long}}}{\partial\,{}^{\mathrm{alt}}k}\right) \\ & + \frac{\partial f_{\mathrm{long}}\left(x_{\mathrm{long}}, u_{\mathrm{long}}\right)}{\partial u_{\mathrm{long}}} \left(\frac{\partial u_{\mathrm{long}}}{\partial\,{}^{\mathrm{alt}}k}\right), \end{aligned} \tag{51}$$

$$\frac{\partial y_{\mathrm{long}}}{\partial\,{}^{\mathrm{alt}}k} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \left(\frac{\partial x_{\mathrm{long}}}{\partial\,{}^{\mathrm{alt}}k}\right). \tag{52}$$

In the design process this loop uses the target function

$$J_{\mathrm{vel}} = \frac{1}{2N} \sum_{n=1}^{N} \left[\tilde{y}_{\mathrm{long}n}^{\mathrm{T}}\,{}^{\mathrm{vel}}Q\tilde{y}_{\mathrm{long}n} + \lambda u_{\mathrm{long}n}^{\mathrm{T}}\,{}^{\mathrm{vel}}Ru_{\mathrm{long}n}\right];$$

with weights

$${}^{\mathrm{vel}}Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, {}^{\mathrm{vel}}R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \lambda = 1;$$

initial controller parameters ${}^{\mathrm{vel}}k_0 = \left[0.5\ 0.1\right]^{\mathrm{T}}$; fixed altitude controller parameters ${}^{\mathrm{alt}}k_0 = \left[-0.5\ -0.01\ -2\ -2\right]^{\mathrm{T}}$ and an altitude 5-meter set-point change with a time constant of 5 seconds. Hence, when the Levenberg-Marquardt optimization algorithm is implemented, the target function and its gradient behaves as shown in Fig. 7 and the evolution of the time response of altitude and velocity is shown in Fig. 8.

The final controller parameters are given by

$${}^{\mathrm{vel}}k = \left[5.22\ 2.82\right].$$

As it can be seen from the results, the parameters change has little effect on the altitude response and the velocity peak response and settling time are reduced.

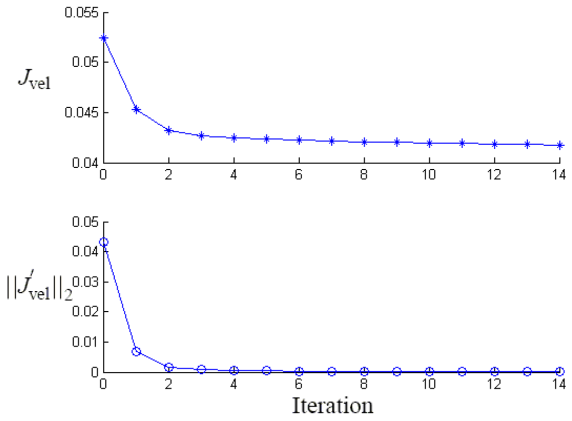

FIGURE 7. VELOCITY CONTROL TARGET FUNCTION AND GRADIENT NORM

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

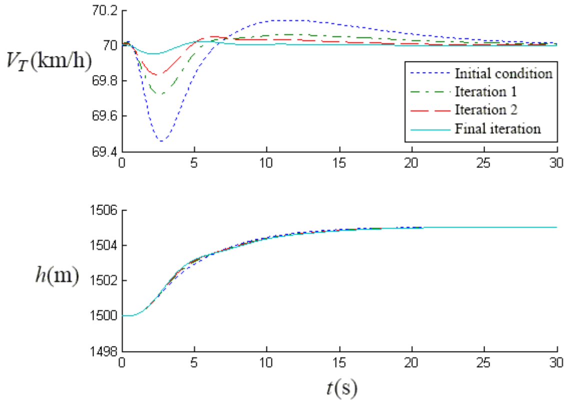

FIGURE 8. TIME RESPONSE OF ALTITUDE AND VELOCITY IN THE VELOCITY LOOP OPTIMIZATION

Lateral mode

Opposed to the longitudinal mode case, the optimization of the lateral mode parameters is done in a single step. Therefore, the controller parameters ${}^{1}k_{\text{lat}}$ and ${}^{2}k_{\text{lat}}$ are optimized at the same time. Then, taking the partial derivative of Eqns. (43)-(44) with respect to ${}^{\text{lat}}k = \left[\,{}^{1}k_{\text{lat}}^{\text{T}}\ {}^{2}k_{\text{lat}}^{\text{T}}\,\right]^{\text{T}}$, Eqns. (15) and (14) become

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial^{\text{lat}}x_c}{\partial^{\text{lat}}k}\right) &= \begin{bmatrix} 0 & 0 & 0\ 0\ 0 & \psi_d-\psi & 0\ 0 \\ 0 & \psi_d-\psi & 0\ 0\ 0 & 0 & 0\ 0 \end{bmatrix} \\ &+ \begin{bmatrix} {}^{1}k_2 & 0\ 0\ 0\ 0 \\ {}^{2}k_2 & 0\ 0\ 0\ 0 \end{bmatrix} \frac{\partial y_{\text{lat}}}{\partial^{\text{alt}}k},\end{aligned} \tag{53}$$

$$\begin{aligned}\frac{\partial u_{\text{lat}}}{\partial^{\text{lat}}k} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \psi_d-\psi & 0 & -\phi & -p \\ \psi_d-\psi & 0 & -\beta & -r & 0 & 0 & 0 & 0 \end{bmatrix} \\ &+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{\partial^{\text{lat}}x_c}{\partial^{\text{lat}}k} \\ &- \begin{bmatrix} {}^{1}k_1 & 0 & 0 & {}^{2}k_3 & {}^{2}k_4 \\ {}^{2}k_1 & {}^{2}k_3 & {}^{2}k_4 & 0 & 0 \end{bmatrix} \frac{\partial y_{\text{lat}}}{\partial^{\text{lat}}k}\end{aligned} \tag{54}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial x_{\text{lat}}}{\partial^{\text{lat}}k}\right) &= \frac{\partial f_{\text{lat}}\left(x_{\text{lat}}, u_{\text{lat}}\right)}{\partial x_{\text{lat}}}\left(\frac{\partial x_{\text{lat}}}{\partial^{\text{lat}}k}\right) \\ &+ \frac{\partial f_{\text{lat}}\left(x_{\text{lat}}, u_{\text{lat}}\right)}{\partial u_{\text{lat}}}\left(\frac{\partial u_{\text{lat}}}{\partial^{\text{lat}}k}\right)\end{aligned} \tag{55}$$

$$\frac{\partial y_{\text{lat}}}{\partial^{\text{lat}}k} = \begin{bmatrix} 0\ 0\ 0\ 0\ 1 \\ 1\ 0\ 0\ 0\ 0 \\ 0\ 0\ 1\ 0\ 0 \\ 0\ 0\ 0\ 1\ 0 \\ 0\ 1\ 0\ 0\ 0 \end{bmatrix}\left(\frac{\partial x_{\text{lat}}}{\partial^{\text{lat}}k}\right). \tag{56}$$

The design process is done using the target function

$$J_{\text{lat}} = \frac{1}{2N}\sum_{n=1}^{N}\left[\tilde{y}_{\text{lat}n}^{\text{T}}\ {}^{\text{lat}}Q\tilde{y}_{\text{lat}n} + \lambda u_{\text{lat}n}^{\text{T}}\ {}^{\text{lat}}Ru_{\text{lat}n}\right];$$

with weights

$${}^{\text{lat}}Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}, {}^{\text{lat}}R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \lambda = 1;$$

initial controller parameters

$$\begin{aligned}{}^{1}k_{\text{lat}0} &= \left[-1.30\ -0.50\ -1.50\ -2.00\right]^{\text{T}}, \\ {}^{2}k_{\text{lat}0} &= \left[0.50\ 0.25\ 1.45\ 2\right]^{\text{T}}\end{aligned}$$

and a heading 10-degree set-point change with a time constant of 5 seconds. Hence, when the Levenberg-Marquardt optimization algorithm is implemented, the target function and its gradient behaves as shown in Fig. 9 and the evolution of the time response of altitude and velocity is shown in Fig. 10.

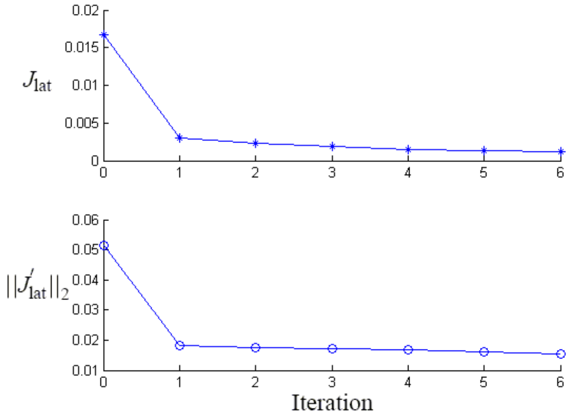

FIGURE 9. LATERAL CONTROL TARGET FUNCTION AND GRADIENT NORM

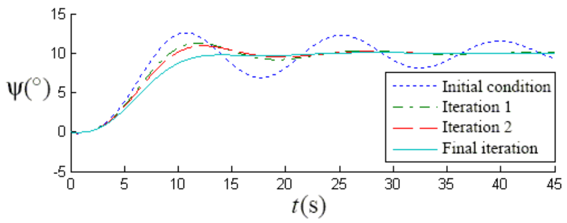

FIGURE 10. TIME RESPONSE OF ALTITUDE AND VELOCITY IN THE VELOCITY LOOP OPTIMIZATION

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

The final controller parameters are given by

$$^{1}k_{\text{lat}} = \left[-1.59\ -0.11\ -1.55\ -2.09\right]^{\text{T}},$$
$$^{2}k_{\text{lat}} = \left[0.37\ 0.42\ 1.27\ 2.01\right]^{\text{T}}.$$

As it can be seen from the results, the heading overshoot and settling time are reduced and the highly oscillatory initial response is considerably improved.

SIMULATION RESULTS

The designed controllers are tested through numeric simulation of the vehicle's complete model (Eqns. (16)-(24)) using a three-level hierarchical architecture. The high-level is a mission planner with a trajectory generator which gives a set of feasible waypoints, the mid-level is a guidance system that reads the waypoints and calculates the filtered setpoints for the low-level multiloop PID controller that stabilizes the vehicle and follows the set-point changes manipulating the vehicle's control inputs.

The trajectory waypoints used in the controller test are summarized in Tab. 1 and illustrated in Fig. 11. In the mode column 1 = passthrough, 2 = cut and 3 = loiter; the North and East columns are objective-point coordinates in the passthrough and cut modes and center-point coordinates in the loiter mode; and in the turn radius column a positive radius means clockwise loiter and a negative one means counter-clockwise loiter.

The trajectory achieved using the multiloop PID in the low-level controller is shown in Fig. 11, the time response of the controlled variables is shown in Fig. 12 and the control inputs behavior is shown in Fig. 13. These results show that an adequate system behavior can be achieved through the chosen control strategy, thus the controller can handle gradual smooth set-point changes.

TABLE 1. WAYPOINTS

ID	Mod	North	East	Alt.	Speed	Rad.	No.
		(m)	(m)	(m)	(km/h)	(m)	turns
1	1	100	0	1500	70	-	-
2	1	500	0	1500	75	-	-
3	1	1000	0	1500	70	-	-
4	3	1000	500	1500	70	500	1
5	1	1500	0	1500	70	-	-
6	1	2000	0	1550	70	-	-
7	3	2000	-500	1550	70	-500	1/2
8	1	1000	-1000	1500	70	-	-
9	2	500	-1000	1500	70	250	-
10	1	0	0	1500	70	-	-

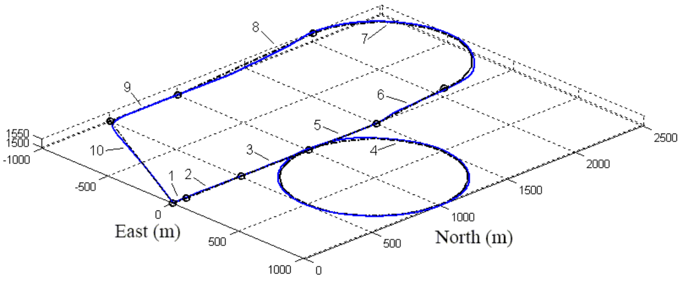

FIGURE 11. ACHIEVED TRAJECTORY

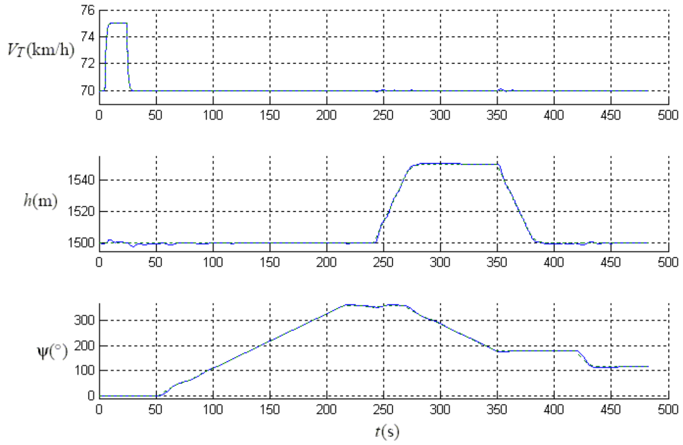

FIGURE 12. CONTROLLED VARIABLES: VELOCITY (FIRST), ALTITUDE(SECOND) AND HEADING (THIRD)

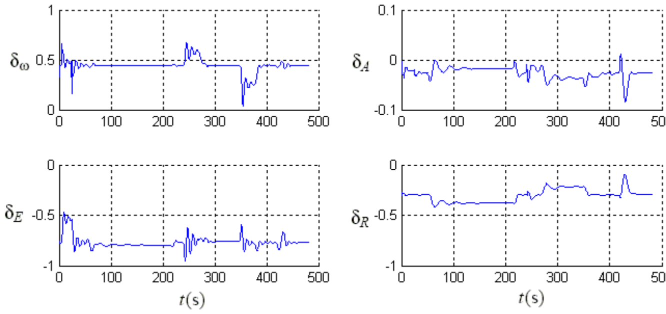

FIGURE 13. CONTROL INPUTS

CONCLUSIONS

In this work the design of a multiloop PID controller for the low-level control of an UAV using an iterative feedback tuning scheme was addressed.

Because a PID is a linear structured low order controller and the UAV is a multivariable non-minimum phase oscillatory high-order plant, the design scheme was proposed in a feedback-iterative-optimization fashion. The design objective was the minimization of a quadratic target function through an iterative gra-

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm

dient based local search algorithm, which uses a continuous version of the Levenberg-Marquardt rules. The algorithm required the calculation of the target function gradient and an estimation of the Hessian obtained from the plant's closed-loop response, after solving a matrix differential equation system, obtained from the controller equations and the plant's nonlinear simulation model.

The results showed that the method optimizes the tuning parameters in the direction of a local minimum of the target function and improves the system's closed-loop performance.

ACKNOWLEDGMENT

The authors gratefully acknowledge the partial financial support given by the Universidad Pontificia Bolivariana for the Master of Science studies of Juan A. Ramirez.

REFERENCES

[1] Vásquez, J. M., 2007. "Mechanical pre-design of an automatic system for power transmision lines inspection". Master's thesis, School of Engineering, Universidad Pontificia Bolivariana, Colombia.

[2] Múnera, D. A., 2005. "Design and integration of a modular system for navigation control of an unmanned vehicle". Master's thesis, School of Engineering, Universidad Pontificia Bolivariana, Colombia.

[3] Ortiz, P., 2008. "Mathematical model for a fixed-wing uav using parametric identification methods". Master's thesis, School of Engineering, Universidad Pontificia Bolivariana, Colombia.

[4] Zuluaga, C. A., 2007. "Simulation platform for autonomous vehicles". Master's thesis, School of Engineering, Universidad Pontificia Bolivariana, Colombia.

[5] Cardona, L., 2008. "Navigation system for unmanned vehicles". Master's thesis, School of Engineering, Universidad Pontificia Bolivariana, Colombia.

[6] Gutiérrez, L., 2004. "Adaptive mode transition control architecture with an application to unmannedaerial vehicles". PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology.

[7] Vachtsevanos, G., Tang, L., Drozeski, G., and Gutiérrez, L., 2005. "From mission planning to flight control of unmanned aerial vehicles: Strategiesandimplementation tools". *Annual Reviews in Control*, ***29***, pp. 101 – 115.

[8] Åström, K., and Hägglund, T., 2001. "Future of PID Control". *Control Engineering Practice*, ***9***, pp. 1163–1175.

[9] Hjalmarsson, H., Gunnarsson, S., and Gevers, M., 1994. "A convergent iterative restricted complexity control design scheme". In Proceedings of the 33rd IEEE Conference on Decision and Control, pp. 1735 – 1740.

[10] Hjalmarsson, H., and Birkeland, T., 1998. "Iterative feedback tuning of linear time-invariant MIMO systems". In Proceedings of the 37th IEEE Conference on Decision and Control, pp. 3893 – 3898.

[11] Sjöberg, J., et al., 2003. "Iterative controller optimization for nonlinear systems". *Control Engineering Practice*, ***11***(9), September, pp. 1079 – 1086.

[12] Huusom, J., et al., 2009. "Improving convergence of iterative feedback tuning". *Journal of Process Control*, ***19***, pp. 570 – 578.

[13] Levenberg, K., 1944. "A method for the solution of certain problems in least squares". *Quarterly of Applied Mathematics*, ***2***, pp. 164 – 168.

[14] Nielsen, H., 1999. Damping parameter in Marquardt's method. Tech. Rep. IMM-REP-1999-05, Technical University of Denmark.

[15] Stevens, B., and Lewis, F., 2003. *Aircraft Control and Simulation*. J. Wiley.

Copyright © 2010 by ASME

Downloaded From: http://proceedings.asmedigitalcollection.asme.org/pdfaccess.ashx?url=/data/conferences/imece2010/72225/ on 08/02/2017 Terms of Use: http://www.asm