Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference on
Guidance, Navigation & Control, Delft University of Technology, Delft,
The Netherlands, April 10-12, 2013

Application of Frequency-Limited Adaptive Quadrocopter Control

Kirk Y.W. Scheper, Daniel Magree, Tansel Yucelen, Gerardo De La Torre, and Eric N. Johnson

Abstract Adaptive control systems have long been used to effectively control dynamical systems without excessive reliance on system models. This is due mainly to the fact that adaptive control guarantees stability, the same however, cannot be said for performance; adaptive control systems may exhibit poor tracking during transient (learning) time. This paper discusses the experimental implementation of a new architecture to model reference adaptive control, specifically, the reference system is augmented with a novel mismatch term representing the high-frequency content of the system tracking error. This mismatch term is an effective tool to remove the high frequency content of the error signal used in the adaptive element update law. The augmented architecture therefore allows high-gain adaptation without the usual side-effect of high-frequency oscillations. The proposed control architecture is validated using the Georgia Tech unmanned aerial vehicle simulation tool (GUST) and also implemented on the Georgia Tech Quadrocpoter (GTQ). It is shown that the new framework allows the system to quickly suppress the effect of uncertainty without the usual side effects of high gain adaptation such as high-frequency oscillations.

1 Introduction

Unmanned Aerial Vehicles (UAV), like many other forms of robots, are often used to perform tasks too dangerous or repetitive for humans. Examples in industry show UAVs being used for surveillance and mapping of areas after natural disasters or in enemy territory in the case of military application. UAVs should operate with little user influence in hostile changing environments. That is, UAVs should have

K.Y.W Scheper, D. Magree, T. Yucelen, G. De La Torre, E. N. Johnson
Georgia Institute of Technology, Atlanta, GA 30332, USA,
e-mail: k.y.w.scheper@student.tudelft.nl, dmagree@gatech.edu, tansel.yucelen@ae.gatech.edu, glt3@ae.gatech.edu, eric.johnson@ae.gatech.edu

the ability to quickly adapt to their surroundings and act appropriately to changing conditions. This paper aims to extend this research by providing a solution to high-gain adaptive control without high-frequency oscillations, which ultimately allows for fast adaptation of advanced control systems in UAVs.

It is well known that standard model reference adaptive control (MRAC) approaches employ high-gain learning rates to achieve fast adaptation in order to rapidly reduce system tracking errors in the face of system uncertainties. High-gain learning rates, however, lead to increased controller efforts and can cause high-frequency oscillations, which may violate actuator rate saturation constraints [4] and/or excite unmodeled system dynamics [10, 11] resulting in system instability.

Motivated from this standpoint, this paper resorts to a recently developed frequency-limited adaptive control architecture [12]. The contribution of this framework comes from suppressing undesired high frequency system oscillations using a new reference system architecture. Specifically, the proposed reference system captures a desired closed-loop dynamical system behavior modified by a novel mismatch term representing the high-frequency content between the uncertain dynamical system and this reference system, i.e., the system error. In particular, this mismatch term allows the limiting of the frequency content of the system error dynamics both in transient and steady state, which is used to drive the adaptive controller. The purpose of this methodology is to prevent the update law from attempting to learn the high-frequency content of the system error. This key feature of the framework yields fast adaptation without incurring high-frequency oscillations in the transient performance.

The proposed augmentation to the reference system is implemented on a quadrocopter model simulated in the Georgia Institute of Technology unmanned aerial vehicle simulation tool (GUST) developed by the Unmanned Aerial Vehicle Research Facility (UAVRF). The GUST software package combines a high-fidelity vehicle and environment model, onboard flight control software, and ground station software. GUST may be operated in hardware in the loop (HITL) mode or software in the loop (SITL) mode. In HITL mode, the flight control software and ground station interface with physical sensors, actuators, and communication links. In SITL mode, the flight control software and ground station interface with the vehicle model and simulated communication links. This design ensures that the same flight control software is used in simulation and in flight reducing software development time.

The vehicle model used in the SITL mode is a six rigid body degree of freedom model complete with additional engine and rotor dynamics. The vehicle model simulates sensor noise, delay, location, orientation, actuator dynamics and saturation. For this paper, the SITL mode is used to obtain the simulation results and flight tests were conducted using the HITL mode onboard the flight platform.

The flight platform used in this paper is a quadrocopter UAV based on the commonly used Pelican airframe [2] from Ascending Technologies (AscTec). The UAVRF uses this platform to perform research on indoor flight navigation and guid-

ance algorithms and is referred to as the Georgia Tech Quadrocopter (GTQ). The GTQ is a fully autonomous vehicle which uses a laser SLAM based navigation system to traverse cluttered indoor environments. A detailed description of the platform and navigation system can be found in [3]. A picture of the GTQ can be seen below in Figure (1).

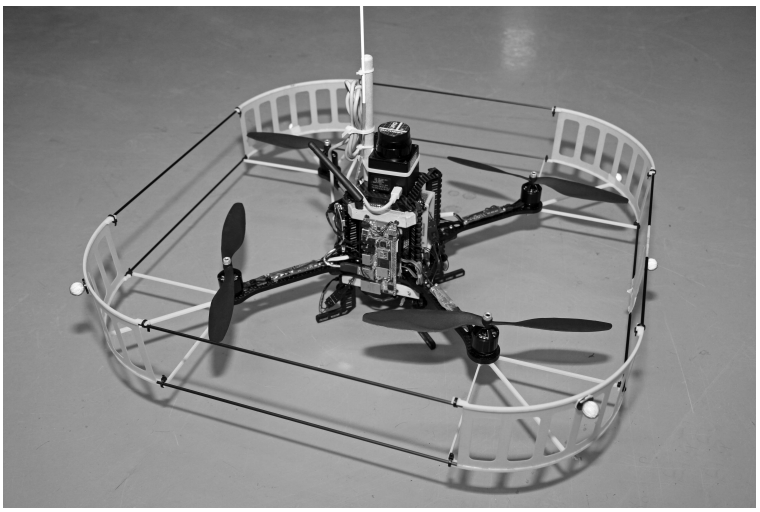

Fig. 1 Picture of the AscTec Pelican indoor flight platform known as GTQ. This is the flight platform used to evaluate the performance of the frequency-limited adaptive control architecture in this paper.

In the following sections, we first outline the adaptive update law used and give a description of the proposed augmented reference system architecture. Then the control system implemented is described in detail. High fidelity simulation and flight test results are then discussed. The performance of the frequency-limited architecture is then compared to a that of standard MRAC and the closed loop reference model framework as described in [9, 5].

The notation used in this paper is fairly standard. Specifically, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of $n \times 1$ real column vectors, $\mathbb{R}^{n\times m}$ denotes the set of $n \times m$ real matrices, $\mathbb{R}_+$ (resp., $\overline{\mathbb{R}}_+$) denotes the set of positive (resp., non-negative-definite) real numbers, $\mathbb{R}_+^{n\times n}$ (resp., $\overline{\mathbb{R}}_+^{n\times n}$) denotes the set of $n \times n$ positive-definite (resp., non-negative-definite) real matrices, $\mathbb{S}^{n\times n}$ denotes the set of $n \times n$ symmetric real matrices, $\mathbb{D}^{n\times n}$ denotes $n \times n$ real matrices with diagonal scalar entities, $(\cdot)^T$ denotes transpose, $(\cdot)^{-1}$denotes inverse, and $\triangleq$ denotes equality by definition.

2 Preliminaries

We will begin by considering the nonlinear dynamical system given by

$$\dot{x}_p = A_p x_p(t) + B_p \Lambda u(t) + B_p \delta_p(x_p(t)), \;\; x_p(0) = x_{p_0}, \;\; t \in \overline{\mathbb{R}}_+, \tag{1}$$

where $x_p(t) \in \mathbb{R}^{n_p}$ is the state vector available for feedback, $u(t) \in \mathbb{R}^m$ is the control input, $\delta_p : \mathbb{R}^{n_p} \to \mathbb{R}^m$ is an *uncertainty*, $A_p \in \mathbb{R}^{n_p \times n_p}$ is a known system matrix, $B_p \in \mathbb{R}^{n \times m}$ is a known control input matrix, and Λ is the unknown control effectiveness matrix. Furthermore, we assume the pair (A_p, B_p) is controllable and the uncertainty can be parametrized as

$$\delta_p(x) = W_p^T \sigma_p(x_p), \tag{2}$$

where $W_p \in \mathbb{R}^{s \times m}$ is an *unknown* weight matrix and $\sigma_p : \mathbb{R}_p^n \to \mathbb{R}^s$ is a known basis function of the form $\sigma_p(x_p) = [\sigma_{p_1}(x_p), \sigma_{p_2}(x_p), \ldots, \sigma_{p_s}(x_p)]^T$.

Remark 1. For the case where the basis function $\sigma_p(x_p)$ is unknown, parametrization in (2) can be relaxed, for example, by considering $\delta_p(x_p) = W_p^T \sigma_p^{nn}(V_p^T x_p) + \varepsilon_p^{nn}(x_p), x_p \in \mathfrak{D}_{x_p}$, where $W_p \in \mathbb{R}^{s \times m}$ and $V_p \in \mathbb{R}^{n_p \times s}$ are unknown weight matrices, $\sigma_p^{nn} : \mathfrak{D}_{x_p} \to \mathbb{R}^s$ is a known basis composed of neural network functional approximators, $\varepsilon_p^{nn} : \mathfrak{D}_{x_p} \to \mathbb{R}^m$ is an unknown residual error, and $\mathfrak{D}_{x_p}$ is a compact subset of $\mathbb{R}^{n_p}$ [6].

Remark 2. If the uncertainty is time varying the adaptive element update law can be extended using, for example, Theorem 7.1 from [12].

To address command following, let $c(t) \in R^{n_c}$ be a given bounded piecewise continuous command for tracking (or $c(t) = 0$ for stabilization) and $x_c(t) \in R^{n_c}$ be the integrator state satisfying

$$\dot{x}_c = E_p x_p(t) - c(t), \;\; x_c(0) = x_{c_0}, \tag{3}$$

where $E_p \in \mathbb{R}^{n_c \times n_p}$ allows to choose a subset of $x_p(t)$ to be followed by $c(t)$. Now (1) can be augmented with (2) as

$$\dot{x} = Ax(t) + B\Lambda u(t) + BW_p^T \sigma_p(x_p(t)) + B_r c(t), \;\; x(0) = x_0, \tag{4}$$

where $x(t) \triangleq \left[x_p^T(t), x_c^T(t)\right]^T \in \mathbb{R}^n, n = n_p + n_c$ is the augmented state vector, $x_0 \triangleq \left[x_{p_0}^T, x_{c_0}^T\right]^T \in \mathbb{R}^n$,

$$A \triangleq \begin{bmatrix} A_p & 0_{n_p \times n_c} \\ E_p & 0_{n_c \times n_c} \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{5}$$

$$B \triangleq \left[B_p^T \;\; 0_{n_c \times m}\right]^T \in \mathbb{R}^{n \times m} \tag{6}$$

$$B_r \triangleq \left[0_{n_p \times n_c} \;\; -I_{n_c \times n_c}\right]^T \in \mathbb{R}^{n \times n_c} \tag{7}$$

Next, let us consider the reference system capturing the desired closed-loop dynamical system performance given by

$$\dot{x_r} = A_r x_r(t) + B_r c(t), \;\; x_r(0) = x_{r0}, \tag{8}$$

where $x_r(t) \in \mathbb{R}^n$ is the reference state vector, $A_r \in \mathbb{R}^{n\times n}$ is the Hurwitz reference system matrix, and $B_r \in \mathbb{R}^{n\times m}$ is the command input matrix. Also, their exist a matrix $K \in \mathbb{R}^{m\times n}$ such that $A_r = A - BK$ holds.

Let us now consider the feedback control law given by

$$u(t) = u_n(t) + u_a(t), \tag{9}$$

where $u_n(t) \in \mathbb{R}^m$ and $u_a(t) \in \mathbb{R}^m$ are the nominal and adaptive control laws, respectively. Let u_n be given by

$$u_n(t) = -Kx(t), \tag{10}$$

Combining (9), (10) into (4) yields

$$\dot{x} = A_r x(t) + B_r c(t) + B\Lambda \left[u_a + W^T(t)\sigma(x(t))\right], \tag{11}$$

where $W \triangleq \left[\Lambda^{-1}W_p^T, (\Lambda^{-1} - I_{m\times m})K\right]^T \in \mathbb{R}^{(s+n)\times m}$ is an unknown (aggregated) weight matrix and $\sigma(x(t)) \triangleq \left[\sigma_p^T(x(t)), x^T(t)\right]^T \in \mathbb{R}^{s+n}$ is a known (aggregated) basis function. Now, considering (11), let the adaptive control law be defined as

$$u_a(t) = -\hat{W}(t)\sigma(x(t)), \tag{12}$$

where $\hat{W}(t) \in \mathbb{R}^{s\times m}$ is the estimate of W satisfying the weight update law given by

$$\dot{\hat{W}}(t) = \Gamma\sigma(x(t))e^T(t)PB, \;\; \hat{W}(0) = \hat{W}_0, \tag{13}$$

where $\Gamma \in \mathbb{R}_+^{s\times s} \cap \mathbb{S}^{s\times s}$ is the learning rate matrix, $e(t) \triangleq x(t) - x_r(t)$ is the system error state vector, and $P \in \mathbb{R}_+^{n\times n} \cap \mathbb{S}^{n\times n}$ is a solution of the Lyapunov equation

$$0 = A_r^T P + PA_r + R, \tag{14}$$

where $R \in \mathbb{R}_+^{n\times n} \cap \mathbb{S}^{n\times n}$ can be viewed as an additional learning rate. Note that since A_r is Hurwitz, it follows from converse Lyapunov theory [6] that there exists a unique P satisfying (14) for a given R.

The system error dynamics is given from (8), (11) and (12).

$$\dot{e}(t) = A_r e(t) - B\Lambda\tilde{W}^T(t)\sigma(x(t)), \;\; e(0) = e_0, \tag{15}$$

where $\tilde{W}(t) \triangleq \hat{W}(t) - W \in \mathbb{R}^{(s+n)\times m}$ is the weight error and $e_0 \triangleq x_0 - x_{r0}$. Proofs of Lyapunov stability of the weight matrix $W(t)$ and error vector $e(t)$ and the convergence of $e(t) \to 0$ as $t \to \infty$ can be found in reference [1].

3 Frequency-Limited Adaptive Control Framework

The augmented MRAC system architecture, as defined in [12], is achieved by introducing a mismatch term to the reference system dynamics. The mismatch term captures the high-frequency content between the uncertain dynamical system and the reference system. This augmenting of the reference system term allows the designer to limit the frequency content of the system error dynamics which is used to drive the adaptive element. The proposed architecture is visualized in the diagram shown below in Figure (2).

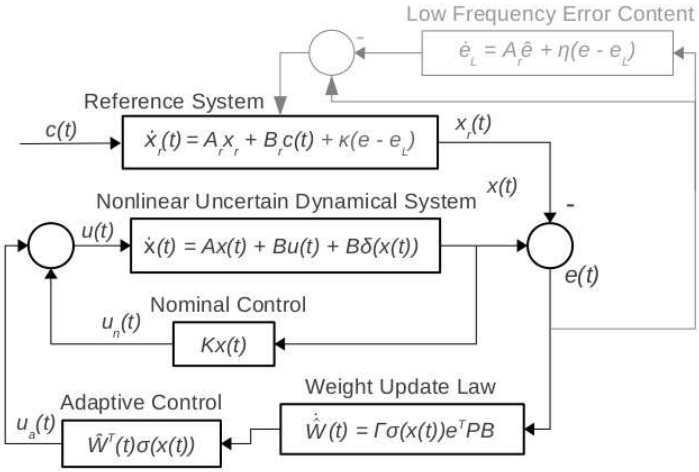

Fig. 2 Visualization of the proposed frequency-limited adaptive control architecture. Notice not only is the reference system driven by the external command but also by the mismatch term containing the high frequency error content. The augmented system is shown in grey.

To define our augmented system let us first consider $e_L \in \mathbb{R}^n$, the low pass filtered error of $e(t)$ given by

$$\dot{e}_L = A_r e_L + \eta(e - e_L), \quad e_L(0) = 0, \tag{16}$$

where $\eta \in \mathbb{R}_+$ is a filter gain. From the low pass error update law it can be seen that the filter gain η is used to limit the bandwidth of this signal. This term must be large enough to capture the system dynamics to be input to the adaptive element but small enough to reject the high frequency content of this signal. Considering this in the classical loop shaping context, the user can now allow faster adaptation of the system to uncertainty by amplifying the adaptive learning rate while suppressing unwanted high frequency adaptation and subsequent high frequency oscillation.

Remark 3. As $\eta \to \infty$ the frequency-limited architecture approaches standard MRAC. As $\eta \to 0$ the frequency-limited architecture approaches the closed loop reference model framework described in [9, 5].

Now, as the purpose of the frequency limitation is to force our error to approach the behavior of the low pass filtered error, we can introduce a mismatch term which is a minimization of the relative distance of the trajectories of the two signals $e(t)$ and $e_L(t)$. This can be captured in the cost function given by $\mathfrak{J}(e,e_L) = \frac{1}{2}||e-e_L||_2^2$, and note that the negative gradient of this cost function with respect to $e(t)$ is given by

$$\frac{\delta[-\mathfrak{J}(e(t)-e_L(t))]}{\delta e(t)} = -(e(t)-e_L(t)) \tag{17}$$

Incorporating the mismatch term into the reference system in (8) results in the following augmented reference system

$$\dot{x}_r = A_r x_r + B_r c(t) + \kappa(e-e_L), \;\; x_r(0) = x_{r_0}, \tag{18}$$

where $\kappa \in \mathbb{R}_+$. Through singular perturbation analysis of the augmented reference system it can be shown that the high frequency content of the error signal is globally exponentially stable and vanishes in a fast manner for a sufficiently high κ[1]. Theorem 5.1 in [12] also discusses that even though the proposed architecture is predicated on a modified reference system given by (18) the augmented system actually converges to the ideal reference system given in (8).

Remark 4. It should be noted that increasing κ with a constant learning rate Γ causes the distance between the augmented system and the ideal reference system to increase so Γ and κ should be increased together for the best overall system performance.

Updating (15) with (18) results in a the system error dynamics now given by

$$\dot{e}(t) = A_r e(t) - B\Lambda\tilde{W}^T(t)\sigma(x(t)) - \kappa(e(t)-e_L(t)), \;\; e(0) = e_0, \tag{19}$$

This framework can now be extended to the flight platform as is discussed in the next section.

4 Application to a High-Fidelity Autonomous Quadrocopter Model

This section describes the implementation of the frequency limitation architecture in the Georgia Tech unmanned aerial vehicle simulation tool (GUST). Additional information of the control architecture can be found in [8].

[1] For a detailed description of the singular perturbation analysis, see Section VI in [12].

Let us first consider the nonlinear system which represents the dynamics a free flying body in the following form

$$\dot{p} = v, \tag{20}$$
$$\dot{v} = a(p, v, q, u_f, u_m), \tag{21}$$
$$\dot{q} = \dot{q}(q, \omega), \tag{22}$$
$$\dot{\omega} = \alpha(p, v, q, \omega, u_f, u_m), \tag{23}$$

where $p \in \mathbb{R}^3$ is vehicle position, $v \in \mathbb{R}^3$ is velocity, $q \in \mathbb{R}^4$ is the attitude quaternion and $\omega \in \mathbb{R}^3$ is angular velocity. $u_f \in \mathbb{R}^1$ and $u_m \in \mathbb{R}^3$ are primary force and moment actuators respectively. We may now define the state vector and control vector

$$x \triangleq [p^T \;\; v^T \;\; q^T \;\; \omega^T]^T, \tag{24}$$
$$u \triangleq [u_f^T \;\; u_m^T]^T, \tag{25}$$

It is assumed that psuedocontrol hedging as described in [7] copes with limitations to the control inputs such as actuator amplitude saturation, actuator rate saturation, and actuator dynamics so we will not include detailed actuator dynamics in this formulation. The approximated translational and rotational dynamics can be formulated from (20)-(23) as the following

$$a_{des} = \hat{a}(x, q_{des}, u_{f_{des}}) \tag{26}$$
$$\alpha_{des} = \hat{\alpha}(x, u_{m_{des}}) \tag{27}$$

where, a_{des} and α_{des} are commonly referred to as the pseudocontrol. $\hat{a}$ and $\hat{\alpha}$ represent the available approximation of the real rates obtained from the navigation system sensors. In addition, $u_{f_{des}}$, $u_{m_{des}}$ and q_{des} are the control inputs and attitude expected to achieve the desired pseudocontrol. When a_{des} and α_{des} are chosen such that they are invertible, the desired control and attitude can be written as

$$\begin{bmatrix} q_{des} \\ u_{f_{des}} \end{bmatrix} = \hat{a}^{-1}(x, a_{des}), \tag{28}$$
$$u_{m_{des}} = \hat{\alpha}^{-1}(x, \alpha_{des}), \tag{29}$$

Combining the inverse law in (21) with (28) and (23) with (29), the following closed loop translational and rotational dynamics result

$$\dot{v} = a_{des} + \bar{\Delta}_a(x, u) - a_h, \tag{30}$$
$$\dot{\omega} = \alpha_{des} + \bar{\Delta}_\alpha(x, u) - \alpha_h, \tag{31}$$

where,

$$\bar{\Delta} = \begin{bmatrix} \bar{\Delta}_a \\ \bar{\Delta}_\alpha \end{bmatrix} \triangleq \begin{bmatrix} a(x,u) - \hat{a}(x, q_{des}, u_{f_{des}}) \\ \alpha(x,u) - \hat{\alpha}(x, u_{m_{des}}) \end{bmatrix} \tag{32}$$

are the static model error due to imperfect model inversion and the pseudocontrol hedging signals a_h and α_h are given by

$$a_h = \hat{a}(x, q_{des}, u_{f_{des}}) - \hat{a}(x, \hat{u}_f) \tag{33}$$
$$\alpha = \hat{\alpha}(x, u_{m_{des}}) - \hat{\alpha}(x, \hat{u}_m) \tag{34}$$

where $\hat{u}_f$ and $\hat{u}_m$ either estimations or measurements of the real control outputs. The pseudocontrols are given by

$$a_{des} = a_c + a_{pd} - \bar{a}_{ad} \tag{35}$$
$$\alpha_{des} = \alpha_c + \alpha_{pd} - \bar{\alpha}_{ad} \tag{36}$$

where a_c and α_c are outputs of reference systems for the translational and attitude dynamics, respectively; a_{pd} and α_{pd} are outputs of proportional-derivative (PD) compensators; and finally, a_{ad} and α_{ad} are the outputs of an adaptive element designed to cancel model error $\bar{\Delta}$. This formulation uses a single hidden layer neural network to update the adaptive law [8].

The full reference system dynamics are given by

$$\dot{p}_r = v_r, \tag{37}$$
$$\dot{v}_r = a_c(p_r, v_r, p_c, v_c), \tag{38}$$
$$\dot{q}_r = \dot{q}_r(q_r, \omega_r), \tag{39}$$
$$\dot{\omega} = \alpha_c(q_r, v_r, q_c, q_c \oplus q_{des}, \omega_c), \tag{40}$$

where p_r and v_r are the outer-loop reference system states and q_r and ω_r are the inner-loop reference system states. The external command signal is $x_c = [p_c^T v_c^T q_c^T \omega_c^T]^T$. Note that the attitude desired by the outer loop is now added to the commands for the inner-loop controller. Here, $q_c \oplus q_{des}$ denotes quaternion addition.

Now, the error dynamics can be stated as

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} p - p_r \\ v - v_r \\ Q(q, q_r) \\ \omega - \omega_r \end{bmatrix} \tag{41}$$

where $Q(\cdot,\cdot) : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}^3$ gives the three component error vector between quaternions. The output of the PD compensators can be written as

$$\begin{bmatrix} a_{pd} \\ \alpha_{pd} \end{bmatrix} = -\begin{bmatrix} R_p & R_d & 0 & 0 \\ 0 & 0 & K_p & K_d \end{bmatrix} e \tag{42}$$

where $R_p, R_d \in \mathbb{R}^{3\times3}$ and $K_p, K_d \in \mathbb{R}^{3\times3}$ are the linear gain, positive definite matrices determined from [8] as

$$R_p = \frac{\omega_o^2 \omega_i^2}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2}, \;\; R_d = \frac{\omega_o \omega_i (\zeta_o \omega_i + \omega_o \zeta_i)}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \tag{43}$$

$$K_p = \omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2, \;\; K_d = 2\zeta_i \omega_i + 2\zeta_o \omega_o \tag{44}$$

where the index i and o represent inner and outer loop respectively and ω and ζ are the frequency and damping ratio of the second order reference system dynamics.

The tracking error dynamics can be found directly by differentiating the error as

$$\dot{e} = \begin{bmatrix} v - v_r \\ \dot{v} - \dot{v}_r \\ \omega - \omega_r \\ \dot{\omega} - \dot{\omega}_r \end{bmatrix} \tag{45}$$

The overall tracking error dynamics can be expressed as

$$\dot{e} = A_r e + B_r [\nu_{ad} - \bar{\Delta}(x,u)] \tag{46}$$

where $\bar{\Delta}$ is given by (32) and the reference system is described by

$$A_r = \begin{bmatrix} 0 & I_3 & 0 & 0 \\ -R_p & -R_d & 0 & 0 \\ 0 & 0 & 0 & I_3 \\ 0 & 0 & -K_p & -K_d \end{bmatrix}, B_r = \begin{bmatrix} 0 & 0 \\ I_3 & 0 \\ 0 & 0 \\ 0 & I_3 \end{bmatrix}, \nu_{ad} = \begin{bmatrix} a_{ad} \\ \alpha_{ad} \end{bmatrix} \tag{47}$$

An approximate model for the attitude dynamics of the helicopter was generated by linearizing the nonlinear model around hover and neglecting coupling between the attitude dynamics

$$\alpha_{des} = \hat{\alpha} = \hat{A}_1 \omega_b + \hat{A}_2 v_b + B u_{m_{des}} \tag{48}$$

where $\hat{A}_1$ and $\hat{A}_2$ are linearized dynamics, and v_b and ω_b are body frame velocity and angular velocity. Choosing the control matrix B such that it is invertible, inverting (48), we are left with the following

$$u_{m_{des}} = B^{-1}(\alpha_{des} - \hat{A}_1 \omega_b - \hat{A}_2 v_b) \tag{49}$$

The translational dynamics are modeled as a point mass with a thrust vector that is a function of the plant orientation. A simple model can be used here as any unmodeled dynamics is compensated in the adaptive element. The simple relationships between thrust, attitude, and accelerations can be described as

$$a_{des} = \begin{bmatrix} 0 \\ 0 \\ Z_{u_{coll}} \end{bmatrix} (u_{coll_{des}} - u_{coll_{trim}}) + L_{bv} g \tag{50}$$

where the subscript *coll* represents collective control input and $Z_{u_{coll}}$ is the control derivative for acceleration in the vertical axis. L_{bv} is the direction cosine matrix that

transforms a vector from the vehicle (or local) frame to the body frame and g is an assumed gravity vector. The desired specific force along the body z axis may be evaluated as

$$f_{sf} = a_{des} - L_{bv}g \tag{51}$$

The required collective input can be evaluated as

$$u_{coll_{des}} = f_{sf}/Z_{U_{coll}} + u_{coll_{trim}} \tag{52}$$

The attitude augmentation required to orient the thrust vector to attain the desired translational accelerations are given by the following small-angle corrections from the current reference body attitude and attitude command:

$$\Delta\Phi_1 = a_{des_2}/f_{sf}, \quad ,\Delta\Phi_2 = a_{des_1}/f_{sf}, \quad \Delta\Phi_1 = 0, \tag{53}$$

For this simplified model, heading change has no effect on accelerations in the x,y plane, and hence, $\Phi_3 = 0$. These three correction angles may now be used to generate the attitude quaternion correction desired by the outer loop, thus,

$$q_{des} = q(\Delta\Phi_1, \Delta\Phi_2, \Delta\Phi_3) \tag{54}$$

where $q(\cdot)$ is a function that expresses an Euler-angle-based rotation as a quaternion.

Now, with the framework for the implementation of the control scheme and the adaptive element described let us describe the formulation of the frequency limitation within that scheme. As the inner and outer loops are fully decoupled, applying the frequency-limited framework to the outerloop results in the following.

$$\dot{e}_L = \begin{bmatrix} 0 & I_3 \\ -R_p & -R_d \end{bmatrix} e_L + \eta \left(\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} - e_L \right) \tag{55}$$

This can be expanded to the following

$$\dot{e}_{L_1} = e_{L_2} + \eta(e_1 - e_{L_1}) \tag{56}$$

$$\dot{e}_{L_2} = -R_p e_{L_1} - R_d e_{L_2} + \eta(e_2 - e_{L_2}) \tag{57}$$

Inserting this result into (46) results in the following updated error dynamics

$$\dot{e} = A_r e + B_r[\nu_{ad} - \bar{\Delta}(x,u)] - \kappa \left(\begin{bmatrix} e_1 \\ e_2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} e_{L_1} \\ e_{L_2} \\ 0 \\ 0 \end{bmatrix} \right) \tag{58}$$

5 Simulation and Flight Test Results

The following section shows the results of both Simulation and flight tests for translational position step commands to the GTQ quadrocopter using the GUST high fidelity flight simulator. For the tests with the frequency limited MRAC active, η was chosen as $5 rad/s$ so as to pass the frequency range of the expected system dynamics of UAV while rejecting the high frequency content of the control signal. The κ term was determined by simple trail and observation.

First, let us consider the simulation results starting with Figure (3) which shows the results of standard MRAC with high gain adaptation, the outerloop learning rate was set as $\Gamma_{ol} = 20$. The figure shows oscillation in the position tracking, large amplitude oscillations in the adaptive control signal and large outputs of the actuators, with the actuators often hitting their magnitude limits. This flying performance would not be suitable for a real flight.

Figure (4) shows the effect of the frequency-limited architecture. The test was completed with the same adaptive gain as above but with the additional settings of $\kappa = 20, \eta = 5$. It can be seen that the large oscillation in the position tracking is reduced, the tracking error itself is reduced and the actuator output is now within an acceptable magnitude and frequency range.

Now, compare the performance of the frequency-limited architecture to the closed loop reference architecture suggested in [9, 5] as shown in Figure (5). Notice that the adaptive signal appears to have more high frequency content, this results in a higher frequency content in the actuator output. This observation can be explained as possible noise amplification in the adaptive signal due to the raw error feedback in the reference system.

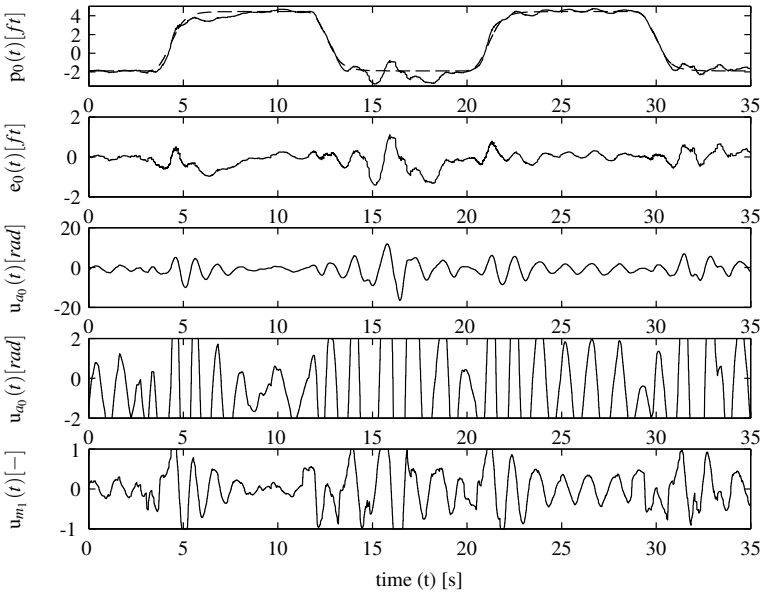

Fig. 3 Simulation test results for translational commands using standard MRAC with the following settings ($\Gamma_{ol} = 20 \cdot I_3, \kappa = 0, \eta = 0$). (Dashed line is commanded position).

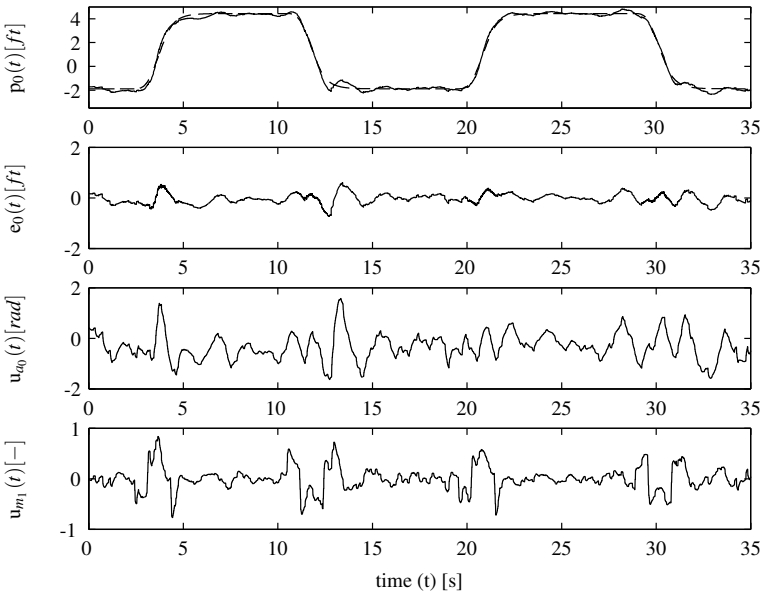

Fig. 4 Simulation test results for translational commands using the frequency-limited adaptive control architecture with the following settings ($\Gamma_{ol} = 20 \cdot I_3, \kappa = 20, \eta = 5$). (Dashed line is commanded position)

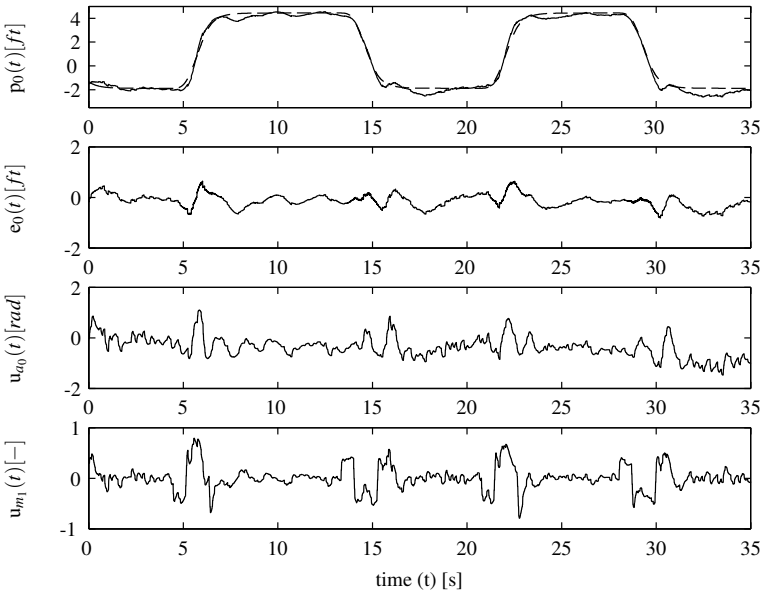

Fig. 5 Simulation test results for translational commands using approach from [9] with the following settings ($\Gamma_{ol} = 20 \cdot I_3, \kappa = 20, \eta = 0$). (Dashed line is commanded position)

These simulations show that the frequency-limited architecture improves the performance of high adaptive gain maneuverer by reducing the high frequency oscillations while ensuring the actuator output signal is within practicable limits.

The flight results are shown in Figures (6) and (7). Figure (6) shows the performance of the frequency-limited architecture with high gain learning rate of $\Gamma_{ol} = 3$. At this learning rate, without the augmented architecture activated, the platform would become unstable due to the influence of computational delays and noisy sensor readings. The performance of the frequency-limited architecture can be compared to the nominal case of standard MRAC with a learning rate of $\Gamma_{ol} = 1.5$ in Figure (7). Notice that the tracking error observed with the nominal case are approximately a factor of two higher than that of the frequency-limited adaptive control, this highlights the improvement in tracking performance using the new architecture.

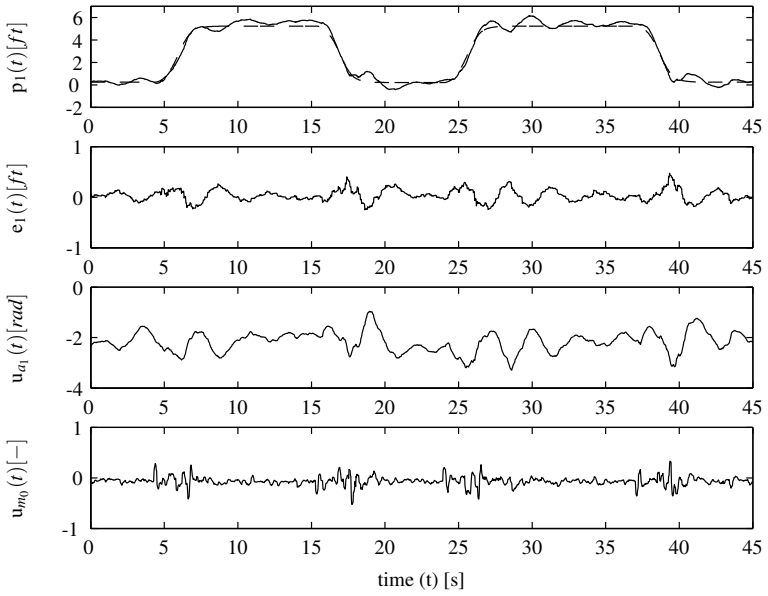

Fig. 6 Flight test results for translational commands using the frequency-limited adaptive control architecture with the following settings ($\Gamma_{ol} = 3 \cdot I_3, \kappa = 6, \eta = 5$). (Dashed line is commanded position)

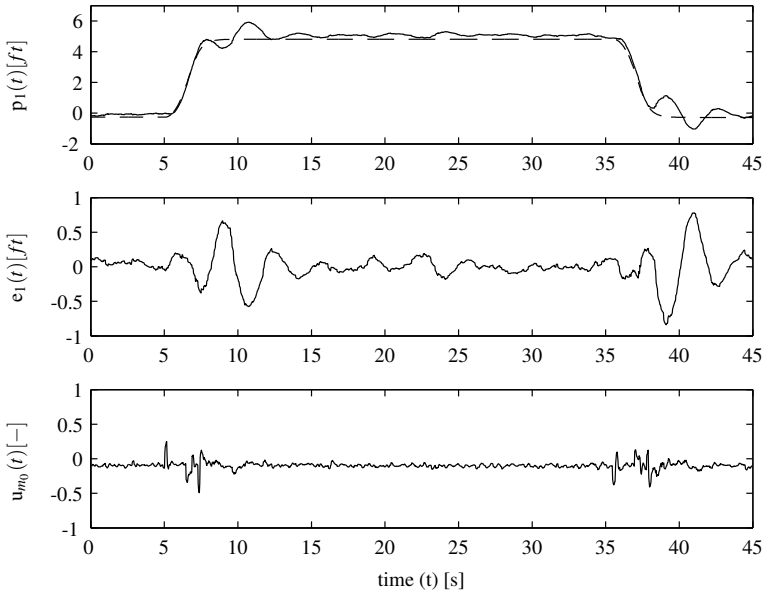

Fig. 7 Flight test results for translational commands using standard MRAC with the following settings ($\Gamma_{ol} = 1.5 \cdot I_3, \kappa = 0, \eta = 0$). (Dashed line is commanded position)

6 Conclusion

In this paper, we presented an application of a frequency-limited adaptive control framework to a high-fidelity autonomous quadrocopter model. We first presented the framework of frequency-limited adaptive control. Next we presented the implementation of the controller on the outerloop of an existing model inversion adaptive controller for an autonomous quadrocopter. Finally we compared standard and frequency-limited controller simulation results during translation maneuver in simulations and show the performance of the controller in real world flight tests. All results showed that the frequency-limited controller improved tracking as compared to the standard model reference adaptive control and kept control outputs within reasonable limits.

References

1. A. M. Annaswamy and K. S. Narendra. Stable adaptive systems. 1989.
2. Ascending Technologies: Astec Pelican.
3. G. Chowdhary, D.M. Sobers Jr., C. Pravitra, C. Christmann, A. Wu, H. Hashimoto, C. Ong, R. Kalghatgi, and E.N. Johnson. Self-contained ranging sensor aided autonomous guidance, navigation, and control for indoor flight. *AIAA Journal of Guidance Control and Dynamics*, 2011.
4. Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky. Adaptive control and the nasa x-15-3 flight revisited. *IEEE Control Systems Magazine*, vol. 30, pp. 32-48, 2010.
5. T. E. Gibson, A. M. Annaswamy, and E. Lavretsky. Adaptive systems with closed-loop reference models: Stability, robustness, and transient performance. *IEEE Transactions on Automatic Control. Submitted, available online*, 2012.
6. W. M. Haddad and V. Chellaboina. Nonlinear dynamical systems and control. a lyapunov-based approach. 2008.
7. E. N. Johnson and A. J. Calise. Limited authority adaptive flight control for reusable launch vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, vol. 26, pp. 906–913, 2003.
8. E. N. Johnson and S. K. Kannan. Adaptive trajectory control for autonomous helicopters. *AIAA Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 524538, 2005.
9. E. Lavretsky. Reference dynamics modification in adaptive controllers for improved transient performance. *AIAA Guidance, Navigation and Control Conference*, 2011.
10. C. E. Rohrs, L. S. Valavani, M. Athans, and G. Stein. Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Transactions on Automatic Control*, vol. 30, pp. 881–889, 1985.
11. T. Yucelen and W. M. Haddad. A robust adaptive control architecture for disturbance rejection and uncertainty suppression with l_∞ transient and steady-state performance guarantees. *International Journal of Adaptive Control and Signal Processing*, 2012. doi: 10.1002/acs.2281.
12. T. Yucelen, G. De La Torre, and E. N. Johnson. Frequency-limited adaptive control architecture for transient response improvement. *American control conference, to appear*, 2012.