A Control Allocation Technique to Recover From Pilot-Induced Oscillations (CAPIO) due to Actuator Rate Limiting

Yildiray Yildiz and Ilya V. Kolmanovsky

***Abstract*— This paper proposes a control allocation technique that can help pilots recover from pilot induced oscillations (PIO). When actuators are rate-saturated due to aggressive pilot commands, high gain flight control systems or some anomaly in the system, the effective delay in the control loop may increase depending on the nature of the cause. This effective delay increase manifests itself as a phase shift between the commanded and actual system signals and can instigate PIOs. The proposed control allocator reduces the effective time delay by minimizing the phase shift between the commanded and the actual attitude accelerations. Simulation results are reported, which demonstrate phase shift minimization and recovery from PIOs. Conversion of the objective function to be minimized and constraints to a form that is suitable for implementation is given.**

I. Introduction

A pilot induced oscillation (PIO) can be described as an inadvertent, sustained aircraft oscillation which is the consequence of an abnormal joint enterprise between the aircraft and the pilot [1]. There are several possible instigators of PIOs such as rate saturated actuators, high gain pilot/controller, system delays and phase lags. The focus of this paper is the rate saturation which is frequently observed during PIO events and has led to several crashes. An excellent overview of the effect of rate limiting in PIOs can be found in [2].

Figure 1 presents a basic model for a rate limited actuator [2], where u is the input to the actuator and δ is the actual actuator deflection. Without the rate limit, this dynamics is simply a first order lag, which is often used as an approximate model for actuators. Figure 2 shows time evolutions of input-output signals of such a rate saturated actuator [2], [3], [4], where $u_c = u$ represents the pilot command. Gain reduction and an increase in effective time delay are two detrimental results of rate saturation, as seen from this figure.

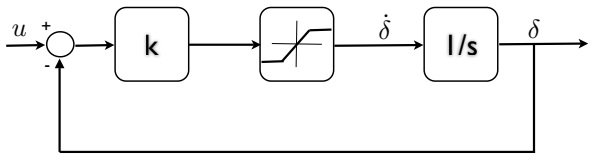

Fig. 1. Actuator model with rate saturation.

Yildiray Yildiz is with U. C. Santa Cruz, NASA Ames Research Center, Moffett Field, MS 269-1, CA 94035, USA yildiray.yildiz@nasa.gov

Ilya V. Kolmanovsky is with Research and Innovation Center, Ford Motor Company, 2101 Village Road, Dearborn, MI 48121, USA ikolmano@ford.gov

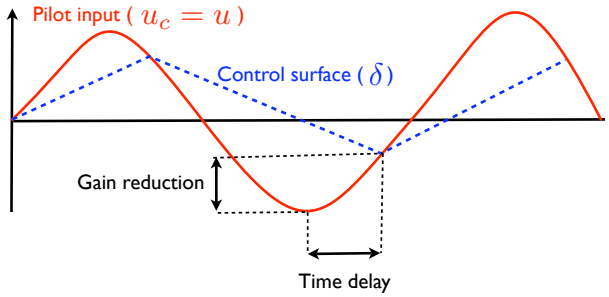

Fig. 2. Input u and output δ of a rate saturated actuator.

While gain reduction can not be avoided, there are several successful approaches in the literature that address eliminating the effective time delay: The main idea in [5], [6], [7], [8], [9] for eliminating the effective time delay is the differentiate-limit-integrate (DLI) approach. This simple yet effective method is implemented using a "software rate limiter" as shown in figure 3, where the software limiter is placed between the command signal and the input signal to the actuator. This method eliminates the effective time delay introduced by the rate saturation as seen in figure 4 and hence the onset of a PIO can be avoided.

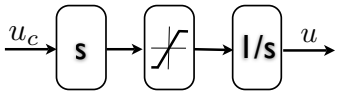

Fig. 3. Software rate limiter.

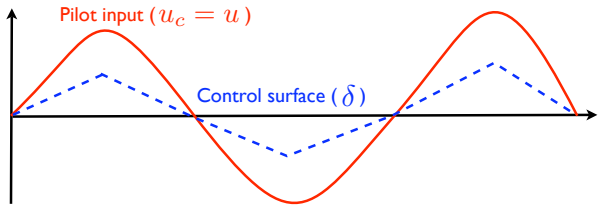

Fig. 4. Input u_c and output δ of a rate saturated actuator with a preceding software rate limiter.

The DLI method proved very successful both in simulation and actual flight tests. It has, however, some deficiencies such as introduction of a bias and susceptibility to noise. See figure 5. These problems may be handled using different techniques such as filtering and resetting/retrimming [6], [7], [8]. See also [9] for more improved results.

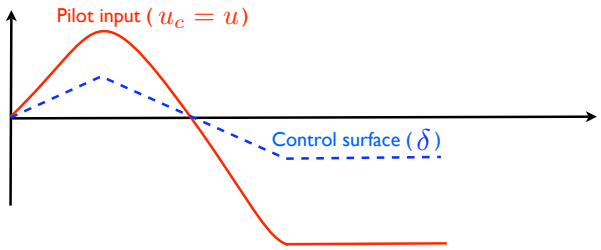

Fig. 5. Bias caused by the differentiate-limit-integrate approach

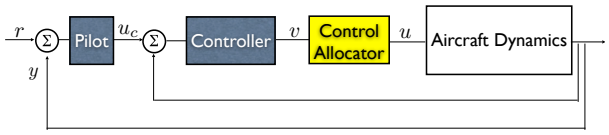

Fig. 6. Overall SISO system structure

There are also other methods for dealing with the effective time delay based on manipulating the input signal: In [10] a nonlinear adaptive filter is used to attenuate the pitch stick shaping function gain depending on the magnitude and frequency of the pilot input and the rate limits of the elevators. This approached was successfully implemented in Space Shuttle Enterprise ALT-5 and no further pitch PIO events were reported in the open literature since the 1977 event [2]. In [11], a phase compensating filter is used which reverses the direction of a rate saturated actuator when the derivatives of the input and the output signals have opposite sign and when the absolute value of the error is increasing. In [12], a similar actuator output reversing logic is developed wherein a feedback signal is passed through a lowpass filter. This approach also proved to be successful in actual flight tests. In [13], another phase compensator is developed for a rate limiting element without using any feedback or logic but employing describing function relationships developed in [14]. This filter also performed successfully in flight tests.

To the best of authors' knowledge, all the previously reported successful implementation results were for SISO applications without any redundant actuators. Consider the closed loop flight control structure in figure 6, where the pilot is also in the loop. In this structure, the pilot task may be to track an altitude reference r, by getting altitude measurement feedback y, and making necessary corrections via a pilot stick which gives pitch rate commands u_c as a reference to the inner flight controller. The inner controller may also be responding to roll and yaw rate commands at the same time. So, the "pilot command" u_c can be a vector of three elements. The controller then calculates the necessary attitude accelerations $v \in \Re^3$ and then control allocator allocates the available actuators $u \in \Re^m$, $m > 3$, to achieve these desired accelerations while possibly satisfying secondary objectives like drag minimization. In this scenario, it is not obvious where and how to use the DLI software limiter. An extension of using the DLI method to multi-input multi-output (MIMO) applications is given in [6], however, the authors had to use "ganged" actuators for successful implementation. Ganging of the actuators, on the other hand, prevents the use of redundant actuators for secondary objectives like drag minimization or reconfiguration after a failure. In addition, ganging becomes more cumbersome as the number of actuators increases [15].

The novel control allocation method proposed in this paper builds upon the previous works and is suitable for MIMO applications in the presence of redundant actuators. For ease of referencing, this technique will be called *Control Allocation (to recover from) Pilot-Induced Oscillations (CAPIO)*. The main idea behind CAPIO, for which a patent is pending, is to minimize the phase lag introduced into the system due to rate saturation by minimizing the error between the derivatives of desired and actual total control effort vectors as well as minimizing the error between them, using constrained optimization techniques. To achieve this goal, for example in a SISO case, one needs to minimize the phase lag between the pilot input and the control surface deflection. On the other hand, in a MIMO case, where there are multiple inputs and outputs, one needs to pinpoint where exactly the phase lag is being introduced to the system. For example, in a scenario where the flight control system produces the desired rate accelerations and a control allocator distributes these commands to redundant actuators using some predefined optimization routine, it makes more sense to minimize the phase lag between the desired and achieved accelerations than concentrating on individual actuator signals. It is noted that merely having a control allocation scheme that takes into account the rate limits of the actuators as constraints can not prevent phase shift between the desired and achieved accelerations when saturation is unavoidable, and thus may not be able to handle a PIO situation. In simulation studies, where PIOs are present with conventional control allocation techniques and for high gain pilot models it will be shown that the onset of these PIOs can be prevented using CAPIO.

It is noted that in [16] Durham and Bordignon extended the direct control allocation scheme to make it easier to implement for the case of rate-limited actuators and consequently ended up with a "moment-rate allocation" scheme. Although in [16] there is no implementation result showing a PIO preventing example, this control allocation scheme has a potential to handle PIOs despite being more complicated than CAPIO. Furthermore, the technique in [16] needs the calculation of a moment rate set which can introduce additional computational intensity.

The organization of this paper is as follows. In Section II, CAPIO is explained using a simple scalar example. In Section III, CAPIO is implemented for an aircraft model with redundant actuators, which is the case for which the use of CAPIO is particularly advantageous. In all cases, first, an example of a PIO formation is presented when a conventional control allocation is used, then it is shown that with the CAPIO the aircraft can recover from the PIO event. Finally, a summary of the paper is given.

II. Scalar example

In this section, it is assumed that the "controller" in figure 6 does not exist and hence, only the pilot is responsible for the control of the aircraft. The resulting system structure is presented in figure 7. In this scenario, the task is tracking a desired pitch angle θ_d. To simplify the analysis, the pilot is modeled as a static gain acting on the error between the desired pitch angle θ_d and the measured pitch angle θ. It is noted that often pilot behavior is assumed to have at least a lead compensation while out of a PIO event, yet a pure gain during a fully developed PIO gives reasonable accuracy [3], [4], [17]. For our purposes, it does not make a difference whether a pure gain or a lead is used, except that the "gain only" option is simpler to work with. For an overview of different pilot models and behaviors see the lecture by McRuer [1]. The actuators used in this scenario are the elevators that are ganged. For the purpose of illustration, it is assumed that the elevators have an 0.05 second time constant with a rate limit of 28.7 deg/s and without any position limits. Short period approximation model [18] is used for the aircraft dynamics, which is given as

$$\frac{\theta(s)}{\delta_e(s)} = \frac{1.39(s+0.306)}{s^3+0.805s^2+1.325s} \tag{1}$$

where δ_e is the elevator deflection. It is noted that the elevator dynamics together with the rate limit is used in series with (1) in the simulations.

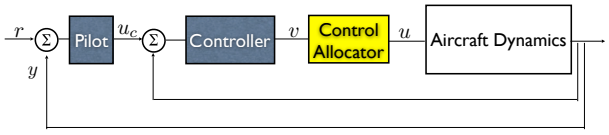

Fig. 7. Overall SISO system structure

A. Pitch angle control with conventional control allocation

A control allocator distributes the total control effort, $v \in \Re^p$, to redundant actuators, $u \in \Re^m$, $m > p$, so that total control effort is produced by the actuators through an input matrix, $B \in \Re^{p\times m}$, possibly together with a secondary objective like drag minimization. One way of achieving this goal is using mixed optimization [15], which requires minimizing the following objective function

$$J = \|Bu - v\| + \epsilon\|u - u_p\|, \tag{2}$$

subject to $u_{min} \le u \le u_{max}$ and $\dot{u}_{min} \le \dot{u} \le \dot{u}_{max}$, where u_p is a "preferred" actuator position and ϵ is a weight. This weight may be chosen sufficiently small to emphasize the main goal of minimizing the error between the desired total control effort v and the produced control effort Bu by the actuators.

In our example, the total control effort, $u_c = v$, is a scalar and there is only one effective actuator (ganged elevators), which makes $B = 1$. To simplify further, it is assumed that there are no secondary objectives, thus $\epsilon = 0$ and the elevators have only rate limits. As a result, the control allocator needs to minimize

$$J = |u - u_c|, \tag{3}$$

subject to $\dot{u}_{min} \le \dot{u} \le \dot{u}_{max}$. In this form, the main task of the control allocator becomes trying to achieve u_c as close as possible while preventing the rate saturation. However, it is noted that the control allocator will still make the elevators work as close as possible to rate saturation limit to be able to follow u_c, whenever necessary. Therefore, one will observe almost the same input output relationship as in figure 2 between u_c and δ_e. As a result, it turns out that a conventional control allocator for a scalar system is equivalent to a pure gain of 1, in terms of the system response.

Figure 8 presents a simulation result where a 1 radian pitch step is given as a reference to the pilot and the pilot acts on the tracking error with a gain of 1.2. Although during the first 4.5 seconds the actuator is rate saturated, after 4.5 seconds the pilot command (u_c) rate gets smaller and thus the actual elevator deflection δ_e can catch it up. The result is a stable tracking of the step reference.

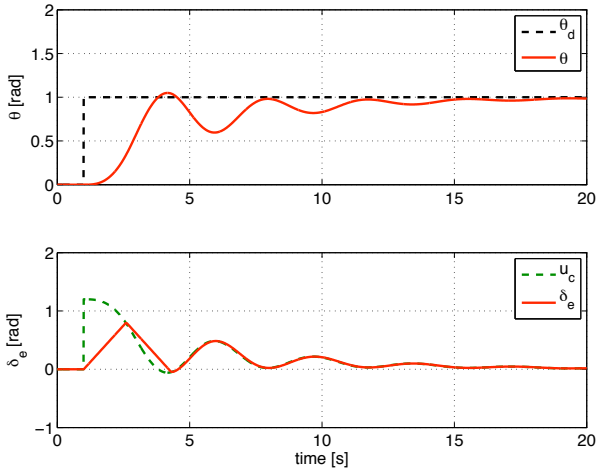

Fig. 8. Pitch reference tracking with conventional control allocation

Figure 9 presents a simulation result when a more aggressive pilot with a gain of 1.65 is controlling the aircraft. Due to a more aggressive pilot control, the elevators are always saturated and an introduced effective time delay is visible (note the similarity with figure 2). The result is formation of a PIO observed as sustained oscillations in the system response.

Figure 10 presents the tracking error and the pitch angle in the same plot. It is noted that the signals have the same magnitude with a -180 degrees phase shift, which shows that the closed loop system is pushed to the marginally stable point with the introduction of an effective time delay. This, in fact, confirms that this is a PIO event, not just temporary maneuvers conducted by the pilot to stabilize the system. As in a typical PIO event, the pilot is not driving the oscillation but she is driven by the oscillation, meaning that she must

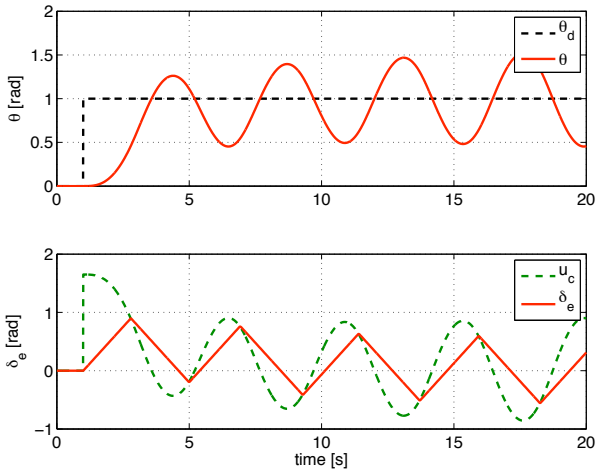

Fig. 9. PIO formation during pitch reference tracking with conventional control allocation.

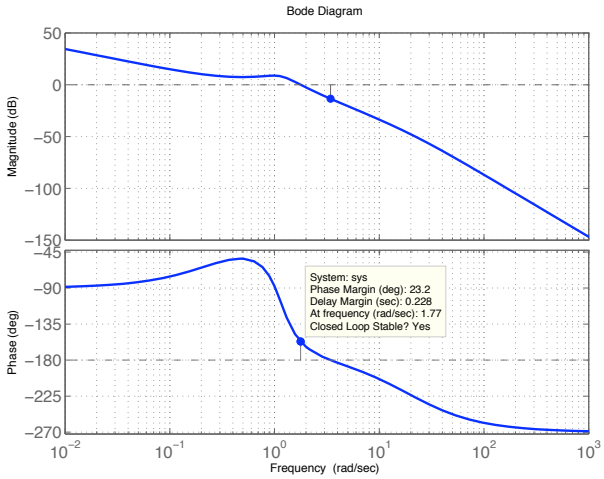

Fig. 11. Bode plot of the overall system without the rate limiting element.

redirect efforts away from the primary task by a noticable amount [17].

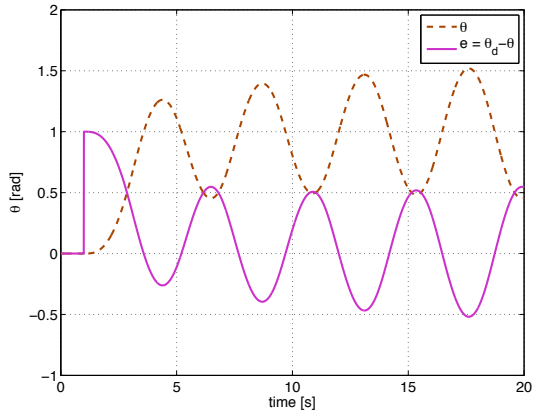

Fig. 10. Evolution of pitch angle and pitch angle tracking error.

Figure 11 shows the bode plot of the loop transfer function neglecting the rate limiting element. The phase margin is 23.2 degrees with a crossover frequency of 1.77 rad/s which gives a 0.228 seconds time delay margin. Introduced effective time delay is larger than this value (see figure 9) which actually makes the system unstable (instead of marginally stable). It is also noted that there is a gain reduction due to rate limiting which increases the time-delay margin.

B. Pitch angle control with CAPIO

As discussed in the previous section, the effective time delay introduced due to rate saturation can be given as one of the reasons for the formation of the PIO observed in figure 9. This effective time-delay is the result of the phase lag between u_c and δ_e, where "phase lag" refers to the shift between the extremum points of the signals. Hence, the main goal of CAPIO is to minimize the phase lag between these two signals. This is achieved by modifying the original objective function (3) as

$$J = |u - u_c| + k\,|\dot{u} - \dot{u}_c|\,, \tag{4}$$

where $k > 0$. It is noted that with this modified objective function, the control allocator is trying to realize $\dot{u}_c$ as well as u_c. A very high value of k makes the two signals, u_c and δ_e, have approximately the same derivative at all times, which eliminates the phase lag completely but causes a bias as shown in figure 5. On the other hand, a very small value of k may not be sufficient for the control allocator to be any different than the conventional one and thus does not prevent PIOs. Therefore, the designer needs to decide on a suitable value of k which minimizes the phase lag and at the same time prevents a bias. As an alternative, the designer can choose to "activate" k, i.e. set it to a positive value, only when it is needed, and keep it 0 at all other times. The latter approach is taken in this paper, assuming that there exists a PIO detection algorithm on board.

For implementation, the objective function (4) needs to be put in a form that can be minimized using available optimization packages. To achieve this goal, the derivatives in the objective function are approximated as follows

$$\dot{u}(t) \approx \frac{u(t) - u(t-T)}{T}, \tag{5}$$

where T is the sampling interval. In all the simulations in this paper, we use $T = 10$ms. Using the approximation in (5), the following objective function is obtained

$$J^{'}(t) = T|u(t) - u_c(t)| + k|u(t) - u(t-T) - T\dot{u}_c(t)| \tag{6}$$

where, $J^{'} = TJ$.

Figure 12 shows the simulation result where the system is provided with the same pitch angle reference and the same pilot gain as in figure 9, but this time CAPIO is the governing control allocator. $k > 0$ is set with the detection of the PIO and deactivated (set to 0) when the system is

out of the PIO. With the activation of k, actual elevator deflection δ_e comes in phase with the commanded elevator deflection u_c and this eliminates the previously introduced effective time-delay and drives the system out of the PIO. Then, k is deactivated and CAPIO continues to perform as a conventional control allocator ensuring that δ_e tracks u_c. As a result, the system eliminates the PIO without any bias formation.

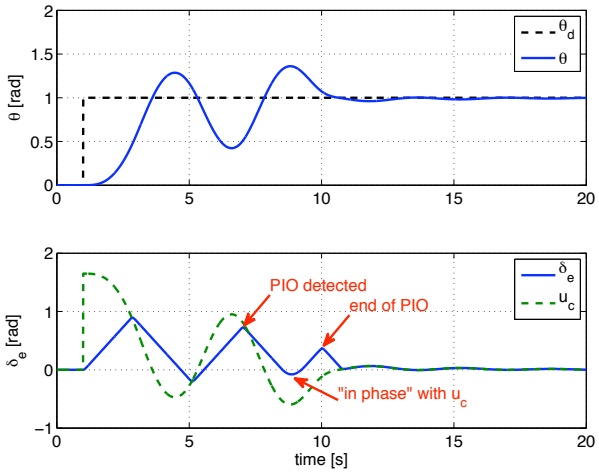

Fig. 12. Pitch reference tracking with CAPIO

Remark 1 The objective function (6) contains the derivative of the commanded elevator deflection u_c. This derivative can be computed using differentiation filters, such as $s/(\tau s + 1)$. The measurement noise can be alleviated by using low pass filters, which proved to be successful in previous PIO preventing algorithms [9].

Remark 2: PIO detection algorithm that is assumed to be present on board can detect the PIO earlier or later than what is assumed in figure 12. In addition, this algorithm can decide that the system is out of the PIO earlier or later than what is assumed in figure 9. These timing decisions can affect system performance. However, the main point made here is that although the timing of the activation and deactivation of the k term may effect the overall system performance, once k is activated, CAPIO will force the actuator deflection to be in phase with the commanded signal and eliminate the introduced effective time delay. The CAPIO will continue to perform as a regular control allocator once k is deactivated. This dual behavior both helps the aircraft go out of the PIO and prevents bias formation.

Remark 3 This scalar example is given mainly for a simple introduction to the CAPIO. It is clear that the DLI method would also prevent the PIO in this scenario. However, the real power of the CAPIO becomes apparent in more realistic cases where the system is MIMO and there are redundant actuators in the system.

III. Multi Input Multi Output (MIMO) Example with Redundant Actuators

To illustrate the advantages of CAPIO, a flight control example using a simplified ADMIRE model from [19] is used. This model includes redundant actuators which makes the DLI method hard to apply if one does not want to gang the actuators.

The linearized aircraft model at Mach 0.22, altitude 3000m is given by

$$\begin{aligned} x &= [\alpha \quad \beta \quad p \quad q \quad r]^T - x_{\text{lin}}, \\ y &= Cx = [p \quad q \quad r]^T - y_{\text{lin}}, \\ \delta &= [\delta_c \quad \delta_{re} \quad \delta_{le} \quad \delta_r]^T - \delta_{\text{lin}}, \\ u &= [u_c \quad u_{\text{re}} \quad u_{le} \quad u_r]^T - u_{\text{lin}} \\ \begin{bmatrix} \dot{x} \\ \dot{\delta} \end{bmatrix} &= \begin{bmatrix} A & B_x \\ 0 & -B_\delta \end{bmatrix} \begin{bmatrix} x \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ B_\delta \end{bmatrix} u, \end{aligned} \quad (7)$$

where α, β, p, q and r are the angle of attack, sideslip angle, roll rate, pitch rate and yaw rate, respectively. δ and u represent the actual and the commanded control surface deflections, respectively. Control surfaces are canard wings, right and left elevons and the rudder. $(.)_{\text{lin}}$ refers to values at the operating points where the linearization was performed. The actuators have the following position and rate limits

$$\begin{aligned} &\delta_c \in [-55, 25] \times \frac{\pi}{180}; \quad \delta_{re}, \delta_{le}, \delta_r \in [-30, 30] \times \frac{\pi}{180} \\ &\dot{\delta}_c, \dot{\delta}_{re}, \dot{\delta}_{le}, \dot{\delta}_r \in [-70, 70] \times \frac{\pi}{180} \end{aligned} \quad (8)$$

and have first order dynamics with a time constant of 0.05 seconds. It is noted that the position limits given are the same as in [19] but the rate limits are assumed to illustrate CAPIO properties. At trimmed flight, we have

$$\begin{aligned} x &= x_0 = (12.7 \quad 0 \quad 0 \quad 0 \quad 0) \\ \delta &= \delta_0 = (0 \quad 5.4 \quad 5.4 \quad 0) \end{aligned} \quad (9)$$

To make this model suitable for control allocation implementation, the actuator dynamics are neglected and the control surfaces are viewed as pure moment generators and their influence on $\dot{\alpha}$ and $\dot{\beta}$ is neglected. It is noted that the actuators dynamics are present during the simulations, i.e. they are neglected only during the control allocation algorithm derivation. These assumptions lead to the following approximate model

$$\begin{aligned} \dot{x} &= Ax + B_u u = Ax + B_v v, \\ v &= Bu, \end{aligned} \quad (10)$$

where

$$B_u = B_v B, \quad B_v = \begin{bmatrix} 0_{2\times 3} \\ I_{3\times 3} \end{bmatrix},$$

$$A = \begin{bmatrix} -0.5432 & 0.0137 & 0 & 0.9778 & 0 \\ 0 & -0.1179 & 0.2215 & 0 & -0.9661 \\ 0 & -10.5128 & -0.9967 & 0 & 0.6176 \\ 2.6221 & -0.0030 & 0 & -0.5057 & 0 \\ 0 & 0.7075 & -0.0939 & 0 & -0.2127 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & -4.2423 & 4.2423 & 1.4871 \\ 1.6532 & -1.2735 & -1.2735 & 0.0024 \\ 0 & -0.2805 & 0.2805 & -0.8823 \end{bmatrix}.$$

The virtual (total) control effort, v, consists of the angular accelerations in roll, pitch and yaw.

In this flight control example the pilot task is to track a given pitch angle reference, θ_d, using a pitch rate, q_d, stick. In addition, roll rate, p, and the yaw rate, r, are to be controlled independently to track their references p_d and r_d. The overall system structure is given in figure 13.

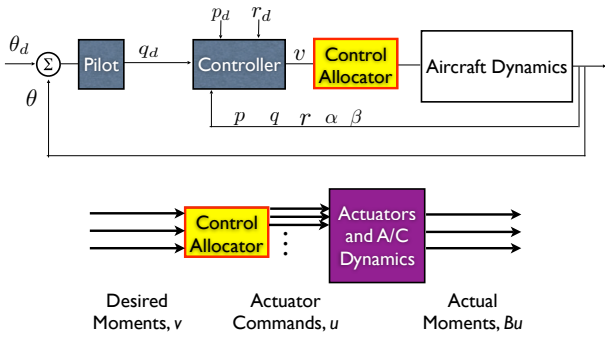

Fig. 13. Overall MIMO system structure

The inner controller is a dynamic inversion controller which uses q_d, p_d and r_d as references and produces the necessary attitude accelerations, $v \in \Re^3$, to track these references. Dynamic inversion control laws, v, make the closed loop dynamics follow a desired reference model

$$\dot{y}_m = A_m y_m + B_m r_m \tag{11}$$

where $y_m = [p_m \quad q_m \quad r_m]^T$ represents the desired output and $r_m = [p_d \quad q_d \quad r_d]^T$ is the reference input. In this example, $A_m = -2 \times I_{3\times 3}$ and $B_m = 2 \times I_{3\times 3}$. Reference model tracking can be achieved by inverting the dynamics [15] as

$$v = (CB_v)^{-1}[A_m y + B_m r_m - CAx]. \tag{12}$$

The control allocator distributes this total control effort, v, to individual control surfaces via the actuator commands, $u \in \Re^4$. The control surfaces then produces actual attitude accelerations, Bu, where B is the control input matrix. The pilot is modeled as a pure gain for simplicity.

A. Flight control with conventional control allocation

The conventional control allocation used in this example minimizes the following objective function

$$J = ||Bu - v||_2^2 + \epsilon||u||_2^2 \tag{13}$$

subject to $max(\dot{u}_{min}T + u_{k-1}, u_{min}) \leq u \leq \min(\dot{u}_{max}T + u_{k-1}, u_{max})$, where T is the sampling interval. It is noted that norms, instead of square-norms, can be used in the objective function. Note that (13) is in the form of a typical objective function used in conventional control allocators [15], where the main objective is to minimize the error between the desired and the actual total control efforts. As $\epsilon \to 0$, minimizing (13) becomes equivalent to achieving the main objective explained above and picking the solution that gives the minimum control surface deflection, among different solutions. In this example $\epsilon = 10^{-5}$.

Figure 14 presents the simulation result with the conventional control allocation where the pilot receives a step pitch angle reference at $t = 3$ seconds and the inner controller receives a pulse roll rate reference at $t = 0.5$ seconds and a zero yaw rate reference at all times. The pilot is aggressive and has a gain of 4.07. Because of this high gain, the aircraft goes into a divergent PIO in the pitch axis. In addition, yaw axis also goes into a sustained oscillation mode which diverges towards the end. The inner controller can still manages to track the roll rate pulse reference. Canard wings and the ailerons saturate both in position and the rate. The results of this saturation can best be observed as a phase shift between the desired pitch acceleration v_2 and the actual pitch acceleration created by the control surfaces Bu_2, although the same phenomenon is observed between v_3 and Bu_3. These phase shifts, or the effective time delays, is something that is almost always observed in PIO events due to actuator saturation.

B. Flight control with CAPIO

To recover from a PIO event, CAPIO forces the virtual (total) control effort v, to be in phase with the actual control effort Bu produced by the actuators. To achieve this, a derivative error term is added to objective function (13) to obtain the following CAPIO objective function

$$J' = ||Bu - v||_2^2 + ||W_d(B\dot{u} - \dot{v})||_2^2 + \epsilon||u||_2^2 \tag{14}$$

where $W_d \in R^{3\times 3}$ represents a weighting matrix on the derivative term. The cost function J' is minimized with respect to u, with $\dot{u} = (u - u^-)/T$, where u^- denotes the value of u at the previous sampling instant. As discussed in the previous section, one can affect the phase lag by changing the weighing matrix. In addition, one can achieve axis prioritization by properly choosing W_d. As in the scalar case, it is assumed that there exits a PIO detector on board and we use an implementation in which W_d is set to identity or zero depending on whether the aircraft is in a PIO event or not.

The objective function (14) needs to be transformed into a form that can be minimized numerically. To achieve this goal,

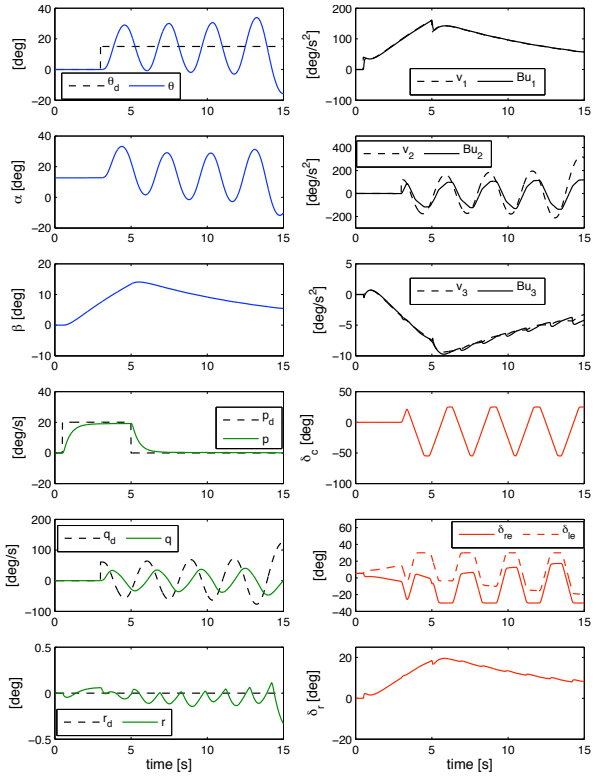

Fig. 14. Pitch angle θ and aircraft states x , on the left. Desired (commanded) and actual attitude accelerations v and Bu, and the control surface deflections δ, on the right, when a conventional control allocator is used.

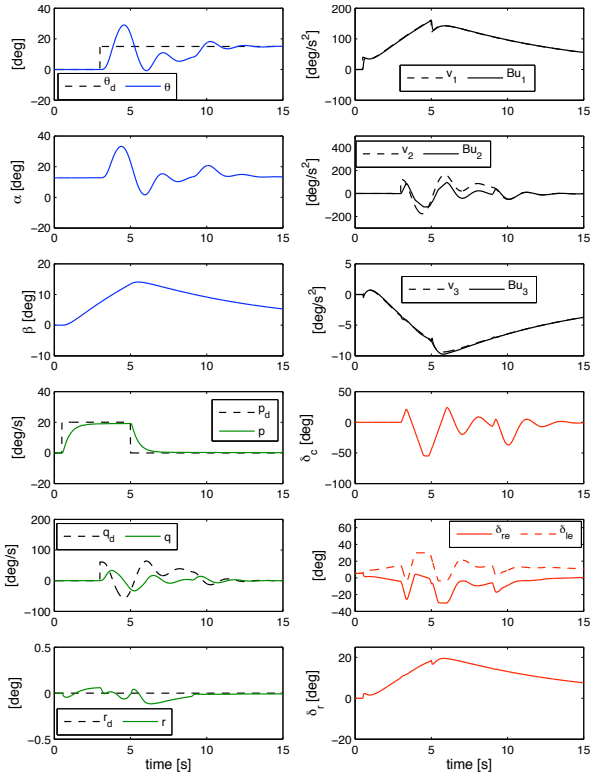

Fig. 15. Pitch angle θ and aircraft states x , on the left. Desired (commanded) and actual attitude accelerations v and Bu, and the control surface deflections δ, on the right, when CAPIO is used.

the derivatives in the objective function are approximated as in (5). After some algebra (14) can be rewritten as

$$\begin{aligned} J^{'} \quad = \quad & u^T \left(B^T T^2 B + B^T R B + \epsilon I_{4\times 4}\right) u \\ & +2\left(-v^T T^2 B - u^{-T} B^T R B - \dot{v}^T T R B\right) u \\ & +v^T T^2 v + u^{-T} B^T R B u^- + 2u^{-T} B^T R T \dot{v} \\ & +\dot{v}^T T^2 R \dot{v} \end{aligned} \tag{15}$$

subject to $max(\dot{u}_{min}T + u^-, u_{min}) \leq u \leq \min(\dot{u}_{max}T + u^-, u_{max})$, where $R = {W_d}^T W_d$.

Figure 15 presents the simulation result when CAPIO is used as the control allocator. All the settings including the pilot gain are the same as in the previous example with the conventional control allocation. Since CAPIO prevents the effective time delay introduction, the aircraft now recovers from the PIO.

To show the difference that CAPIO makes in control effort realization, the pitch axis accelerations are presented again in figure 16 for both cases. It is noted as as soon as the PIO detection signal is obtained, CAPIO forces the control surfaces to produce accelerations "in phase" with the commanded accelerations, eliminating the time delay due to phase shift. When the aircraft recovers from PIO, control surfaces arrange themselves to track the commanded acceleration. The result is recovery from the PIO without any bias formation.

The constrained optimization of the cost (15) is a low dimensional quadratic programming problem with linear inequality constraints. This problem depends on a vector of parameters, specifically, on $p = [u^- \quad v \quad \dot{v}]$. Note that the parameters enter linear in the cost and in the constraints and, hence, such a quadratic programming problem can be solved explicitly using off-line multi-parametric QP solvers. The solution is known to be a piecewise affine continuous function of the parameter vector and have the following form, $u = f_i p + g_i$, if $F_i + G_i \leq 0$, $i = 1, ..., N_r$ where N_r is the finite number of polyhedral regions and each region is associated with its set of linear inequality constraints and its affine map. Such an explicit solution is computed off-line and can be deployed on-line in the aircraft software using a set of simple if-then-else rules, additions, multiplications and comparisons. The need to embed a quadratic programming solver to perform constrained optimization of the cost (15) within aircraft software can thus be avoided altogether. A cross-section of the explicit solution polyhedral regions is illustrated in figure 17; the explicit solution has $N_r = 223$ regions.

IV. Summary

In this paper, a new control allocation scheme for recovery from pilot induced oscillations (CAPIO) was proposed. CAPIO functions by minimizing the error between the derivatives of desired total control effort and the achieved

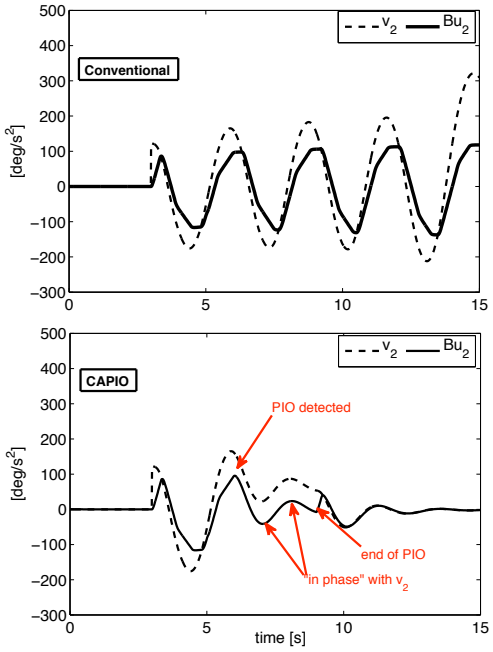

Fig. 16. Pitch angle θ and aircraft states x , on the left, desired (commanded) and actual attitude accelerations v and Bu, and the control surfaces δ, on the right, when a CAPIO is used.

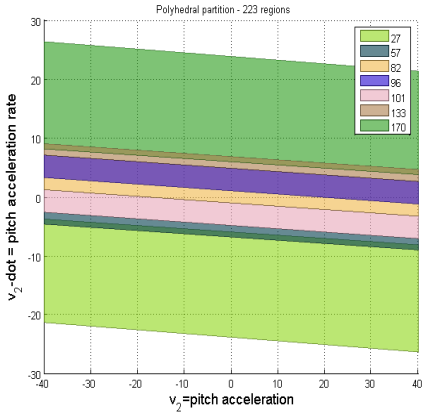

Fig. 17. Cross-section of regions of explicit solution by pitch acceleration - pitch acceleration rate plane.

total control effort vectors, as well as minimizing the error between these two vectors. The derivative error minimization forces the elements of these two vectors to have the same phase and the error minimization makes them converge to each other. This dual behavior results in recovery from dangerous PIO events without creating any bias. The technique was explained first on a scalar example, and then a MIMO example with redundant actuators was given. In each example, first a PIO was created using high gain pilot models with a conventional control allocator and then recovery from this PIO event was shown by the employment of CAPIO. Conversion of the allocation scheme to a form that is easy to implement was also given. It is noted that in real aircraft implementation, handling of measurement noise, digital realization of the derivative and the integration of CAPIO with a PIO detector are important points to be addressed for a successful technology transfer.

REFERENCES

[1] D. McRuer, "Human dynamics and pilot-induced oscillations," *Minta Martin Lecture*, 1992, Massachusetts Institute of Technology, Cambridge, MA.

[2] D. H. Klyde and D. G. Mitchell, "Investigating the role of rate limiting in pilot-induced oscillations," in *Proc. AIAA Atmospheric Flight Mechanics Conference and Exhibit*. Austin, Texas: AIAA, Aug. 2003, pp. 1–12.

[3] D. McRuer, D. H. Klyde, and T. T. Myers, "Development of a comprehensive pio theory," in *Proc. AIAA Atmospheric Flight Mechanics Conference*. San Diego, CA: AIAA, July 1996, pp. 581–597.

[4] D. H. Klyde and D. G. Mitchell, "A pio case study - lessons learned through analysis," in *Proc. AIAA Atmospheric Flight Mechanics Conference and Exhibitt*. San Francisco, CA: AIAA, Aug. 2005, pp. 1–17.

[5] P. Deppe, C. Chalk, and M. Shafer, "Flight evaluation of an aircraft with side and centerstick controllers and rate-limited ailerons," Advanced Technology Center, Calspan Corp., Buffalo, NY, Final Rept. 8091-2, Apr. 1994.

[6] R. A. Hess and S. A. Snell, "Flight control system design and rate saturating actuators," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 90–96, 1997.

[7] S. A. Snell and R. A. Hess, "Robust decoupled, flight control design with rate-saturating actuators," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 3, pp. 361–367, 1998.

[8] M. Chapa, "A nonlinear pre-filter to prevent departure and/or pilot-induced oscillations (pio) due to actuator rate limiting," Graduate School of Engineering, Air Force Inst. of Technology (AU), Wright-Patterson AFB, OH, M.S. Thesis AFIT/GAE/ENY/99M-01, Mar. 1999.

[9] B. S. Liebst, M. J. Chapa, and D. B. Leggett, "Nonlinear prefilter to prevent pilot-induced oscillations due to actuator rate limiting," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 740–747, 2002.

[10] J. W. Smith and J. W. Edwards, "Design of a nonlinear adaptive filter for suppression of shuttle pilot-induced oscillation tendencies, NASA TM-81349," Apr. 1980.

[11] J. Koper, "An approach for compensating actuator rate saturation," Air Vehicle and Crew systems Technology Dept., Naval Air Development Center, Warminster, PA, Interim Rept. NADC-87120-60, Aug. 1987.

[12] L. Rundqwist and R. Hillgren, "Phase compensation of rate limiters in jas 39 gripen," in *Proc. AIAA Atmospheric Flight Mechanics Conference*. San Diego, CA: AIAA, July 1996, pp. 69–79.

[13] D. Hanke, "Phase compensation: A means of preventing aircraft-pilot coupling caused by rate limitation," DLR - Forschungsbericht 98-15, Tech. Rep., 1998.

[14] ——, "Handling qualities analysis on rate limiting elements in flight control systems," in *Proc. Flight vehicle Integration Panel workshop in Pilot Induced Oscillations*. AGARD-AR-335, Feb. 1995.

[15] M. Bodson, "Evaluation of optimization methods for control allocation," in *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*. Montreal, CA: AIAA, Aug. 2001, pp. 1–15.

[16] W. C. Durham and K. A. Bordignon, "Multiple control effector rate limiting," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 1, p. 3037, 1996.

[17] D. G. Mitchell and D. H. Klyde, "Recommended practices for exposing pilot-induced oscillations or tendencies in the development process," in *Proc. USAF Development Test and Evaluation Summit*. Woodland Hills, CA: AIAA, Nov. 2004, pp. 1–20.

[18] J. H. Blakelock, *Automatic Control of Aircraft and Missiles*, 2nd ed. John Wiley & Sons, 1991.

[19] S. Ola Harkegard and T. Glad, "Resolving actuator redundancy - optimal control vs. control allocation," *Automatica*, vol. 41, pp. 137–144, 2005.