Int. J. Appl. Math. Comput. Sci., 2015, Vol. 25, No. 1, 117–131
DOI: 10.1515/amcs-2015-0009

amcs

SUPERVISORY FAULT TOLERANT CONTROL OF THE GTM UAV USING LPV METHODS

TAMÁS PÉNI [a], BÁLINT VANEK [a,*], ZOLTÁN SZABÓ [a], JÓZSEF BOKOR [a]

[a] Institute for Systems and Control
Hungarian Academy of Sciences, Kende u. 13.–17., Budapest H-1111, Hungary
e-mail: vanek@sztaki.hu

A multi-level reconfiguration framework is proposed for fault tolerant control of over-actuated aerial vehicles, where the levels indicate how much authority is given to the reconfiguration task. On the lowest, first level the fault is accommodated by modifying only the actuator/sensor configuration, so the fault remains hidden from the baseline controller. A dynamic reallocation scheme is applied on this level. The allocation mechanism exploits the actuator/sensor redundancy available on the aircraft. When the fault cannot be managed at the actuator/sensor level, the reconfiguration process has access to the baseline controller. Based on the LPV control framework, this is done by introducing fault-specific scheduling parameters. The baseline controller is designed to provide an acceptable performance level along all fault scenarios coded in these scheduling variables. The decision on which reconfiguration level has to be initiated in response to a fault is determined by a supervisor unit. The method is demonstrated on a full six-degrees-of-freedom nonlinear simulation model of the GTM UAV.

Keywords: fault tolerant control, linear parameter-varying systems, supervisory architecture, flight control.

1. Introduction

A major goal in modern flight control system research is the need to pursue improved reliability and environmental sustainability of safety critical systems (Goupil and Marcos, 2012). Faults and failures may result in a loss of performance and even in a catastrophic harm or a loss of life. For that reason, increasing the safety and reliability of commercial aircraft via improving the pilots' abilities to counteract the faults and provide them with the flight envelope protection functions as long as possible are important priorities. Reconfigurable control methods promise a way to compensate for failures or damage of flight control effectors by using the remaining flight control surfaces to generate compensating forces and moments, via changes in the flight control algorithms. The majority of faults can be detected immediately with system checks, but a few of them require fault detection and isolation (FDI) methods. Combined with reconfigurable control methods, they are considered active fault tolerant control (FTC), which assumes a well defined mechanism to actively detect and counteract against faults in the system (Steinberg, 2005; Mahmoud *et al.*, 2003; Yang *et al.*, 2012; Montes de Oca *et al.*, 2012; Gáspár *et al.*, 2012b). In passive FTC, controllers are fixed and are designed to be robust against a class of presumed faults. This approach does not need a fault estimate (or detection) or controller reconfiguration, but provides only limited fault-tolerant capabilities (Zhang and Jiang, 2008).

In general, a fault tolerant flight control system is required to perform failure detection, identification and accommodation for sensor and actuator failures. Active FTC schemes can be broadly classified into projection based and online controller redesign based approaches (Sloth *et al.*, 2010). Numerous results have been proposed related to both the approaches in the past few years (Ganguli *et al.*, 2002; Alwi *et al.*, 2011; Edwards *et al.*, 2012). More recently, in numerous research projects within the EU FP7 framework, focus on the environmental impact of the aircraft and hence higher performance is required from the flight control system, with certifiable algorithms, leading to a paradigm shift from robust passive FTC towards active methods relying on switching, gain scheduled or linear parameter-varying (LPV) methods.

*Corresponding author

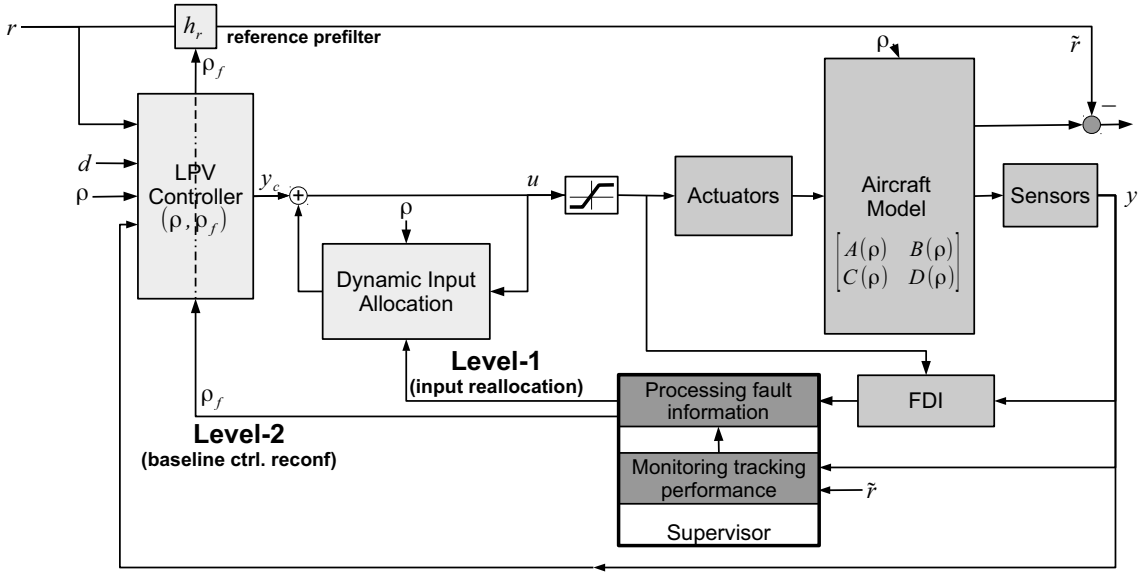


Fig. 1. General architecture of multi-level reconfigurable control.

The individual control components used in a vehicle system are often designed decoupled but are in interaction or even conflict with each other in terms of the full vehicle. An integrated control system is designed in such a way that the effects of a control system on other vehicle functions are taken into consideration in the design process by addressing the various performance specifications. Redundancy on sensor and actuator levels makes it possible to realize the same functionality using different sensor and actuator configurations, i.e., control reconfigurations. Thus integrated design is motivated by the needs of reconfigurable and reliable control (see, e.g., Staroswiecki, 2006; Stoustrup, 2009). Recently, several important papers have been presented in this topic (see, e.g., Yu *et al.*, 2008; Trachtler, 2004; Gáspár *et al.*, 2012a). A possible solution to integrated control is to set the design problem for the whole vehicle and include all the performance demands in a single specification. Besides the complexity of the resulting problem, the formulation of a suitable performance specification is the main obstacle for this direct global approach.

Another solution to integrated control is a quasi-decentralized control structure where the components are designed independently (see, e.g., Xiao *et al.*, 2011; Gáspár *et al.*, 2008). In the present paper the quasi-decentralized control system is augmented with a supervisor as illustrated in Fig. 1. The role of the supervisor is to meet a performance specification in different operating modes, i.e., fault conditions, and prevent the interference between flight control components. The supervisor has information about the current operational mode of the vehicle, i.e., the various vehicle maneuvers or the different fault operations gathered from monitoring components and fault-detection and identification (FDI) filters. The supervisor is able to make decisions about the necessary interventions into the vehicle components and guarantee the reconfigurable and fault-tolerant operation of the vehicle. These decisions are propagated to the lower layers through predefined interfaces encoded as suitable scheduling signals using the LPV framework.

The aim of the paper is to present a multi-level method combining dynamic control allocation and control reconfiguration. The baseline control system is designed for the nominal dynamics of the aircraft, while faults and actuator saturation limits are handled by the dynamic control allocation scheme if they pose a feasible challenge, given the original handling-qualities. If the level of the system degradation is beyond the limits of the original control system, the handling-qualities have to be revised and less demanding performance objectives along with increased robustness have to be provided for the faulty system. Coordination of these components is provided by a supervisor which re-allocates control authority based on health information, flight envelope limits and cross coupling between lateral and longitudinal motion. The monitoring components and FDI filters provide the supervisor with information about different fault operations, and based on that he or she is able to make decisions about necessary interventions into the vehicle motions and guarantee fault-tolerant operation of the aircraft. The design of the proposed reconfigurable control algorithm is based on linear parameter-varying (LPV) control methods that use a parameter dependent dynamic control allocation scheme. The prime advantage of this approach is that the synthesis results in a single multivariable controller with stability and robustness guarantees for the closed-loop system. The

design is demonstrated on a full six-degrees-of-freedom high-fidelity simulator of the NASA AirSTAR flight test vehicle.

The paper is organized as follows. Section 2 presents a brief introduction to the supervisory architecture. The vehicle model of the NASA AirSTAR UAV is presented in Section 3 together with the baseline controller synthesis. The dynamic input allocator is described in Section 4, which is followed by simulation results in Section 5. Conclusions are drawn in Section 6.

2. Two-level reconfigurable control architecture

The reconfiguration framework proposed in the paper is based on a multi-level approach (Peni *et al*., 2013). The subsequent levels indicate how much authority is given to the reconfiguration task. On the lowest, first level, the effect of the fault is not significant and moderate maneuvers are commanded. Hence the fault is accommodated by modifying only the actuator/sensor configuration. In this case the fault remains hidden from the baseline controller. This is an important requirement in commercial aviation (Goupil and Marcos, 2011) from the perspective of the pilot and the flight control system (FCS), since only the flight envelope constraints might be impacted, but the autopilot functions can remain unchanged. A dynamic reallocation scheme, e.g., the one proposed by Zaccarian (2009), can be extended and applied on this level as proposed in Section 4. The allocation mechanism exploits the actuator/sensor redundancy available on the aircraft. It has already been shown, e.g., by Johansen and Fossen (2013), that most critical actuator/sensor faults can be managed in this way. Therefore we expect that most of the possible failures will be successfully accommodated without modifying the baseline controller. The present article discusses an actuator allocation scheme, but similar methods can be applied to the dual problem for sensor reconfiguration as discussed by Peni *et al*. (2014).

If a fault cannot be managed at the actuator/sensor level, then the reconfiguration process has to be allowed to access the baseline controller. This can be achieved via the fault specific scheduling parameters, which can be defined according to the impact of specific faults on the overall aircraft behaviour. To be more specific, faults impacting the longitudinal dynamics and those having most of their impact on the lateral dynamics can have different corresponding fault specific scheduling parameters. An invariant set based region of attraction estimation method (Chakraborty *et al*., 2011) can serve as a tool to assess the applicability of control laws and divide the different fault cases to be scheduled with a few fault dependent scheduling signals to maximise the stability region. The role of the supervisory logic is to handle these fault scheduling signals. In the present paper a single fault specific signal, ρ_f, is considered for simplicity and the corresponding control design is detailed in Section 3.3.

The decision on how the control system reconfigures in response to a fault is determined by a supervisor unit. Methods using controlled invariant set computation (Blanchini, 1999) can be employed to systematically study the region of stability in various fault cases. In the present method a simple logic is used. In the case of a single jamming fault of a surface, only Level-1 is used, while faults affecting multiple surfaces trigger Level-2 besides Level-1. The supervisor is also responsible for generating and providing all necessary fault information (surface affected, type of fault, failed position, etc.) to the reconfiguration subsystems: the supervisor manages the actuator/sensor reallocation procedure and constructs the fault-dependent scheduling parameters for the scheduled baseline controller. The entire control architecture is depicted in Fig. 1.

3. Aircraft model and baseline control design

An effective resource for experimentally testing flight control algorithms, including adaptive control algorithms, is the Airborne Subscale Transport Aircraft Research (AirSTAR) testbed at the NASA Langley Research Center (Murch, 2008). The primary AirSTAR flight test vehicle is a turbine powered 5.5% dynamically scaled model of a civilian transport aircraft, often referred to as the generic transport model (GTM). The GTM has a wing span of 7 ft, and weighs around 55 lbs. Under normal operations, it flies at an altitude of 700 to 1100 ft, with an airspeed between 70 and 85 knots. The currently used T-2 test aircraft is shown in Fig. 2. Significant wind tunnel and flight testing has been performed to identify the flight dynamics of the GTM (Cunningham *et al*., 2008). A nonlinear simulation model of the aircraft dynamics has been developed and is readily available to the research community.

Experimental control algorithms are easily embedded in this simulation model for verification prior to flight testing (Dorobantu *et al*., 2012). Hence, the AirSTAR testbed is highly effective for experimental flight control research through its flexible architecture as well as the rapid implementation and testing cycle. To fully understand the theoretical and practical aspects of the fault tolerant control scheme, the rigid body control of the aircraft motion has to be designed for both lateral and longitudinal planes. After obtaining a suitable LPV model, a baseline LPV control system is designed for pitch angle θ and airspeed V_{cas} control in the longitudinal plane as well as an advanced LPV controller for roll angle ϕ and sideslip angle β control in the lateral one.

Fig. 2. NASA AirSTAR vehicle.

3.1. Lateral model of the GTM aircraft. A four-state model, decoupled from the eight state reduced-order model, captures the primary lateral flight dynamics of the GTM. The states of the model are sideslip angle β [rad], roll rate p [rad/s], yaw rate r [rad/s], and roll angle ϕ [rad]. The control inputs to the model have significant redundancy, left and right aileron deflection $\delta_{a,L}, \delta_{a,R}$ [rad], upper and lower rudder $\delta_{r,U}, \delta_{r,L}$ [rad], inner and outer spoiler deflection $\delta_{sp,I}, \delta_{sp,O}$ [rad], and left and right throttle $\delta_{THR,L}, \delta_{THR,R}$ [%]. Due to the physical limits of the spoilers (they can be deflected only in positive direction) and their symmetric effect on the lateral dynamics, only the control inputs of the left spoilers are included in the model of the aircraft, i.e., the negative inputs on the left inboard (I) and left outboard (O) spoilers are implemented by deflecting in the positive direction the corresponding right inboard (RI) and right outboard (RO) spoilers. The lateral LPV state-space model of the GTM between 60 and 100 knots is approximated with an affine LPV model in the form of

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = (A_0 + A_V V_{cas}) \begin{bmatrix} \beta \\ p \\ r \\ \psi \end{bmatrix} + (B_0 + B_V V_{cas}) \begin{bmatrix} \delta_{a,L} \\ \delta_{a,R} \\ \delta_{r,U} \\ \delta_{r,L} \\ \delta_{sp,I} \\ \delta_{sp,O} \\ \delta_{THR,L} \\ \delta_{THR,R} \end{bmatrix}, \qquad (1)$$

where the outputs of the four states are expressed in degrees (the states are in radians), leading to a constant output readout map of $C = (180/\pi) I_{4\times 4}$. The affine LPV coefficients A_0, A_V, B_0, B_V are obtained with least-squares fit on the pointwise LTI plants. The plant is augmented with first order actuator dynamics of

$$G_{act} = \frac{10\pi}{s + 10\pi}$$

on ailerons, rudders and spoilers and

$$G_{eng} = \frac{-0.1474s + 0.7314}{s^2 + 1.336s + 0.7314}$$

on the throttle, which contains a right-half-plane zero. Sensor dynamics are omitted due to the high quality of the sensors onboard.

For fault injection purposes, the plant in Eqn. (1) is augmented with additional fault signals to include single and double rudder jamming failures. The upper rudder fault has the same input direction as the upper rudder input $B_{f,r,U} = B(\rho)(1 : 4, 3)$, but in the case of jamming at a constant angle (C_1) the fault signal, defined as a virtual input to the plant, is the opposite of the rudder command shifted by the offset of jamming position ($f_{r,U,j} = -\delta_{r,U} + C_1$), cancelling out the nominal rudder input to the plant. In the runaway case, the fault offset $f_{r,U,r}$ is also a function of time $f_{r,U,r} = -\delta_{r,U} + C_2 t$ with constant slope until the flight control surface reaches it physical limit. The same can be applied to the lower rudder $B_{f,r,L} = B(\rho)(1 : 4, 4))$, where jamming and runaway faults can be simulated similarly.

3.2. Longitudinal model of the GTM aircraft. The remaining four states, decoupled from the eight state reduced-order model, capture the primary longitudinal flight dynamics of the GTM. The states of the model are pitch angle θ [rad], pitch rate q [rad/s], angle of attack α [rad], and calibrated airspeed V_{cas} [m/s]. The control inputs to the model also have significant redundancy, left and right elevator deflection $\delta_{e,L}, \delta_{e,R}$ [rad], inner and outer spoiler deflection on both sides $\delta_{sp,I}, \delta_{sp,O}$ [rad], and left and right throttle $\delta_{THR,L}, \delta_{THR,R}$[%]. Due to the physical limits of the spoilers and their effect on the longitudinal dynamics, the control inputs are sign constrained, which has to be taken care by the control allocation method.

The longitudinal quasi-LPV state-space model of the GTM between 60 and 100 knots is approximated with an affine LPV model in the form similar to the lateral dynamics (Eqn. (1)), and hence the details are omitted here. Measured outputs are the four states, where the matrix C is constant. The affine LPV coefficients $A_{0,lon}, A_{V,lon}, B_{0,lon}, B_{V,lon}$ are obtained with least-squares fit on the pointwise LTI plants. The plant is augmented with the same actuator dynamics on elevators and spoilers as on the lateral control inputs, and the same engine dynamics described above are also used on the throttle. No fault is assumed on the longitudinal dynamics for the present investigation to keep

the overall system complexity tractable. Sensor dynamics on all eight measured outputs are omitted due to the high quality of the sensors onboard, but because of the onboard communication and digital implementation of the controllers a time delay of 0.03 s is assumed on both the lateral and longitudinal sensor channels, which is accounted by the fourth-order Padé approximation.

3.3. Baseline controller design with fault-dependent scheduling. The system interconnection (see Fig. 3) addressing the parameter dependent controller synthesis proposed to solve the roll angle and sideslip angle tracking problem, assuming noise and exogenous disturbances, is detailed in the following. The goal of the controller synthesis is to have robust performance across all operating points. To account for system health information, the LPV representation, scheduled with $\rho_V = V_{cas}$, is augmented with a fault specific scheduling variable $0 \leq \rho_f \leq 1$. This allows trading off high performance in healthy operation ($\rho_f = 0$) with robust, less demanding behavior in the case of failures ($\rho_f \leq 1$).

Model matching is achieved by filtering the reference signal through a "handling-qualities" model, scheduled with ρ_f, to achieve smooth behavior with adequate speed of response for bank angle and side speed commands.

The main control objectives, to keep the error between the plant outputs ϕ, β and the desired handling-quality responses

$$hq_\phi = \left(\frac{2.5 - \rho_f}{s + 2.5 - \rho_f}\right)^2,$$

$$hq_\beta = \left(\frac{1.5 - \rho_f}{s + 1.5 - \rho_f}\right)^2$$

are weighted across frequency and health status with

$$W_\phi = (7 - 2\rho_f)\left(\frac{3.5 - 2\rho_f}{s + 3.5 - 2\rho_f}\right)^2,$$

$$W_\beta = (3 - 2\rho_f)\left(\frac{1.5 - 0.75\rho_f}{s + 1.5 - 0.75\rho_f}\right)^2$$

across all parameter range, trading off good steady state tracking with degraded performance at frequencies higher than 5 rad/s.

The speed of response and tracking error requirements are reduced at higher frequencies as the aircraft is subjected to faults, which has direct impact on the control authority requirement in transient modes, especially when the reference command is changing. The fault dependent weights are tuned by a trial and error approach at different points of the flight envelope on LTI plants first, with the physical insight in mind that a system subjected to a loss of actuators will have lower achievable performance. Actuator usage is penalised in the design with weights of $W_{act} = (3+2\rho_f)\mathrm{diag}(1/20, 1/20, 1/30, 1/30, 1/15, 1/45, 1/25, 1/25)$, corresponding to the maximum actuator deflections respecting the physical limits of the individual control effectors, while also addressing the increased actuator usage during faults. Characteristics of the noise are captured by a weight of $W_n = (3 + 2\rho_f)\mathrm{diag}(0.04, 0.1, 0.1, 0.04)$ with a fault dependent magnitude, accounting for higher uncertainty. But the weight is constant across frequency, and assumes higher noise on pitch and yaw rate sensors than on the bank angle due to the sensor characteristics.

The uncertainty associated with the aircraft model is captured by an input multiplicative structure, where the uncertainty weights associated with the aileron, rudder, spoiler and throttle channels are $W_d = \mathrm{diag}(1.5, 1.5, 1.5, 1.5)$. The weights are optimized with linear point design first, at 18 points of the parameter space $[\rho_V; \rho_f] = [60 : 5 : 100; 0 : 1]$ defined in the LPV model. During the pointwise $\mathcal{H}_\infty$ synthesis, the γ performance level ranges between 0.835 and 1.001, with lower values at lower speeds and at $\rho_f = 1$, where lower performance is required. The LPV synthesis with an unbounded parameter rate, with a constant Lyapunov function, leads to a higher γ performance level, and the $\mathcal{L}_2$ gain is 1.1. This is a consequence of using a single parameter dependent LPV controller, where the parameter rates can be unbounded. The longitudinal LPV controller is designed in a similar way, but without fault scheduling for the tracking of θ. V_{cas} commands are followed using a setpoint tracking scheme, since V_{cas} is also used as a scheduling variable. Hence the trim velocity is always the measured one and the tracking error is always zero.

In the baseline control design all actuators are assumed to be used, with their maximum deflection limits, and the control allocation is assigning the correct amount respecting the supervisory commands, according to the health status and saturation of the actuators.

4. Dynamic input reallocation

This section extends the input reallocation method proposed by Zaccarian (2009) to parameter varying plants driven by dynamical actuators. In the work of Zaccarian (2009) only LTI models are assumed and the controller acts directly on the plant. This section considers the configuration when LTI actuators are connected to an LPV plant and the reconfiguration has to be performed through the actuator dynamics.

Let the actuator models be collected in a strictly proper, linear, time-varying state-space model as follows:

$$\begin{aligned} \dot{x}_a &= A_a x_a + B_a u_a, \\ y_a &= C_a x_a. \end{aligned} \tag{2}$$

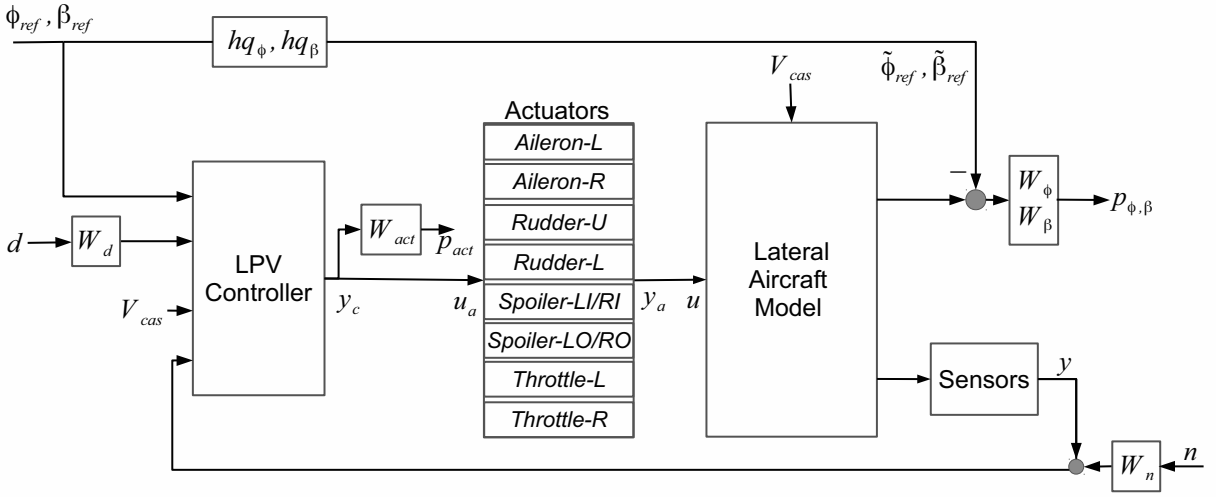


Fig. 3. Block diagram of the weighted interconnection for baseline lateral control design of the GTM aircraft. The controller is designed to minimise the induced $\mathcal{L}_2$-norm from inputs $[\phi_{ref}, \beta_{ref}, d, n]$ to performance outputs $[p_{act}, p_{\phi,\beta}]$.

Here $x_a \in \mathbb{R}^{n_{x_a}}$ collects the states of all actuators and $u_a \in \mathbb{R}^{n_{u_a}}, y_a \in \mathbb{R}^{n_{y_a}}$ denote the inputs and the outputs, respectively. The system to be controlled is assumed to be linear and parameter varying, given in state-space form as follows:

$$\dot{x} = A(\rho)x + B(\rho)u,$$

where $x \in \mathbb{R}^{n_x}, u \in \mathbb{R}^{n_u}$ are respectively the state and input of the plant, and $\rho : \mathbb{R} \to \mathbb{R}^{n_\rho}$ denotes the measurable time varying scheduling parameter. Since the plant is driven by the actuators, the input to the plant u equals y_a. In the nominal case, u_a equals the output of the controller, i.e., $u_a = y_c$. Input reallocation is performed if some failure occurs, and thus the control effort has to be redistributed among the healthy actuators. To modify the control input combination, an additional signal $\tilde{u}_a$ will be designed so that $y_c + \tilde{u}_a$ converges to the required configuration while $\tilde{u}_a$ remains totally unseen for the plant. (The latter requirement ensures that the reconfiguration does not affect the nominal closed loop performance.) These goals are intended to be achieved by exploiting the input redundancy of the system, which is available if $\operatorname{rank}(B(\rho)) < n_u, \forall \rho$).

It is assumed that the baseline controller has been designed for the input-redundant augmented plant containing the actuators as well. (This approach differs from the methods (Johansen and Fossen, 2013) where the controller is designed to generate only torques and forces, and thus the actuator dynamics are not explicitly taken into consideration during the synthesis procedure.)

The first step towards allocator design is to characterize the inputs u_a which do not have any effect on the plant. To this end, we consider the actuator dynamics with a modified output:

$$\begin{aligned} \dot{x}_a &= A_a x_a + B_a u_a, \\ v &= B(\rho) C_a x_a. \end{aligned} \tag{3}$$

Now, we make the following assumptions.

Assumption 1.

(a) The vector relative degree of (3) is $(1, 1, 1, \ldots, 1)$, i.e., the relative degree from each output $v_i(\rho)$ is 1.

(b) $B(\rho)$ has full row rank.

Remark 1. In practice, Assumption 1(a) is not too restrictive as most actuators can be well-approximated by a one or two dimensional models having explicit input dependence on the first derivatives of the output. For an example, see the case study in Section 5. As for Assumption 1(b), if $B(\rho)$ does not have full row rank, then it can be replaced in the forthcoming derivations by any full row rank matrix $\tilde{B}(\rho)$, the rows of which span the same (parameter-dependent) subspace as $B(\rho)$.

Due to Assumption 1, we can introduce the following coordinate transformation:

$$T(\rho) = \begin{bmatrix} B(\rho)C_a \\ T'(\rho) \end{bmatrix}, \quad z_a = T(\rho)x_a = \begin{bmatrix} z_{a,1} \\ z_{a,2} \end{bmatrix},$$

$$z_{a,1} = v,$$

where $T'(\rho)$ is chosen so that $T(\rho)$ is invertible for all ρ. Then we have

$$\dot{z}_a = \left[\frac{\partial T(\rho)}{\partial \rho} \dot{\rho} + T(\rho) A_a \right] T(\rho)^{-1} z_a + T(\rho) B_a u_a,$$

i.e., by partitioning the equations according to $z_{a,1}$ and $z_{a,2}$,

$$\dot{z}_{a,1} = A_1(\rho, \dot{\rho}) z_a + B_1(\rho) u_a, \tag{4a}$$

$$\dot{z}_{a,2} = A_2(\rho, \dot{\rho}) z_a + B_2(\rho) u_a. \tag{4b}$$

If

$$\tilde{u}_a = w_p(\rho, \dot{\rho}, z_a) + \ker(B_1(\rho))w,$$

where w is arbitrary and $w_p(\rho, \dot{\rho}, z_a)$ satisfies

$$A_1(\rho, \dot{\rho})z_a + B_1(\rho)w_p(\rho, \dot{\rho}, z_a) = 0, \tag{5}$$

then the dynamics (4a) at $u_a = \tilde{u}_a$ will reduce to $\dot{z}_{a,1} = 0$, i.e., if $z_{a,1}(0) = 0$, then $z_{a,1}$ remains zero for all time. As a consequence, (4b) will depend only on $z_{a,2}$, that is,

$$\begin{aligned}\dot{z}_{a,2} =& A_2(\rho, \dot{\rho}) \left[\begin{smallmatrix} 0 \\ z_{a,2} \end{smallmatrix}\right] \\ &+ B_2(\rho)(w_p(\rho, \dot{\rho}, z_{a,2}) + \ker(B_1(\rho))w).\end{aligned} \tag{6}$$

Since (2) is always stable, (6) (which is the zero-dynamics of (4)) is stable as well. The new input to the actuator is defined as follows:

$$\begin{aligned}u_a &:= y_c + w_p(\rho, \dot{\rho}, z_{a,2}) + \ker(B_1(\rho))w \\ &:= y_c + \tilde{w}.\end{aligned} \tag{7}$$

Since the system (4) is linear, its trajectory is the sum of the trajectories corresponding to input y_c and input $\tilde{w}$. Consequently, if $\tilde{w}$ at $y_c \equiv 0$ does not have any effect on the dynamics, then it will not influence the trajectory, either, if $y_c \neq 0$. Therefore, w_p can be constructed by assuming $z_{a,1} = 0$, i.e., we can use Eqn. (6) to generate $z_{a,2}$. Then z_a can be computed as

$$z_a = \begin{bmatrix} 0 \\ z_{a,2} \end{bmatrix}$$

and finally w_p can be obtained from (5).

It is important to emphasize that $z_{a,2}$ is generated independently of the actuator's state. Therefore, we do not need to measure x_a in order to compute the reconfiguration signal. This is because w_p is designed to compensate the effect of w and not the effect of y_c. If the model of the actuator is known up to an acceptable precision, then $z_{a,2}$ precisely approximates the effect of w on the real actuators.

The remaining component of the compensator is the reconfiguration signal w, which is constructed similarly to the LTI case:

$$\dot{w} = -K\ker(B_1(\rho))^T W u. \tag{8}$$

Here W is the (diagonal) weighting matrix controlling the contribution of each actuator and K is a feedback gain rendering (8) stable. It can be proved (Zaccarian, 2009) that, at constant parameter–control-output pair (ρ^*, y_c^*), w converges to the optimal value minimizing the quadratic cost function

$$J(w) = (y_c^* + \ker B_1(\rho^*)w)^T W (y_c^* + \ker B_1(\rho^*)w).$$

As for the entries of W, a large value at $W_{i,i}$ means switching off the i-th actuator. If the i-th actuator has upper and lower limits (e.g., $-M_i$ and M_i) which cannot be exceeded then, W_{ii} can be chosen as follows (Zaccarian, 2009):

$$W_{ii} = [(1 + \epsilon)M_i - |\mathrm{sat}_{M_i}(u_a(i))|]^{-1}. \tag{9}$$

5. Simulation results

In order to illustrate the relevance of the two-level reconfiguration architecture, a rudder fault scenario is investigated on the GTM, where both the levels of controller reconfiguration have to be activated to accommodate the fault.

The performance of the control allocation method is analyzed on a single (upper) rudder jamming case first. To perform the reconfiguration tasks, we assume that the fault has been detected at 0 seconds and precise fault information is available for the supervisor unit. In response to the jamming fault, the supervisor immediately switches off the rudder (the position remains constant, at the deflection where the fault occurred) and reconfigures the controller to attenuate the effect of actuator loss (see Fig. 4). The true deflections and the controller demands (after the allocation algorithm) are very close to each other in all flight control surfaces. It should be noted that ailerons are reaching their saturation limit at 4 seconds and that causes the overall allocation method to have a very small residual error on the upper rudder demand, which is not exactly zero when simultaneous compensation of ailerons and the faulted rudder are all necessary. It is also worth noticing that aileron deflections are not symmetric, which is due to the fact that the control allocation method uses all available control surfaces to help compensate the loss of the rudder. Ailerons are also used to create a yawing moment, but the left and right aileron has different control effectiveness on the yaw rate channel as described in Appendix by the corresponding B_0 and B_V LPV state space matrices.

The control task is a simultaneous doublet tracking problem on both the sideslip-angle (β) and the bank-angle (ϕ) as shown in Fig. 5, where a minor tracking error can be observed due to the combined maneuver, where cross-coupling is fundamentally difficult to handle. Notice that the aircraft has calibrated airspeed of 80 knots at the beginning of the maneuver in each simulation case, which slightly decreases towards the end since velocity control has a slower response. It can be stated that the input allocator efficiently handles the loss of the surface, and tracking of both reference signals is good. On the other hand, the baseline controller without control allocation would provide adequate ϕ tracking with an unacceptably poor β tracking response.

The performance of the allocation scheme is further analyzed in the case when both rudders fail and have to

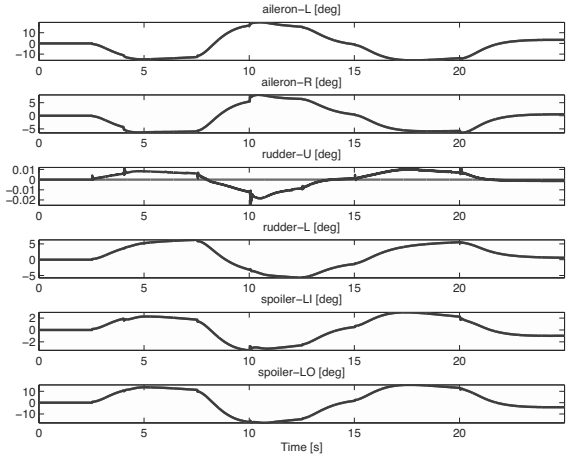


Fig. 4. First six control inputs in the upper rudder jamming fault case, input allocation with a baseline controller. Black: the output of the controller, grey: the saturated input signals. (The negative inputs on the left inboard (LI) and left outboard (LO) spoilers are implemented by deflecting in the positive direction the corresponding right inboard (RI) and right outboard (RO) spoilers.)

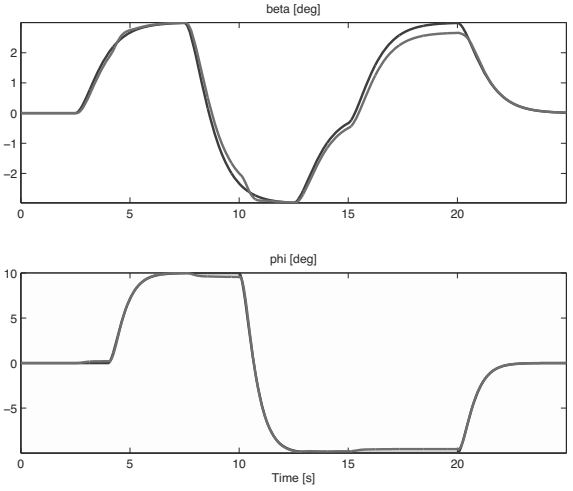


Fig. 5. Output tracking in the upper rudder jamming fault case, input allocation with a baseline controller. Black: reference signals filtered by hq_β, hq_ϕ, grey: outputs of the GTM model.

be switched off. To compare the results of the different control configurations, the first simulation is performed without any reconfiguration. In this case the baseline controller tries to control the faulty aircraft. It can be seen in Figs. 6 and 7 that the capability of tracking β-reference is lost but the aircraft remains stable.

Figures 8 and 9 present simulation results obtained by using only local control allocation. To compensate the faulty rudders, the allocator tries to generate larger inputs on the healthy actuators, but this is prevented by the limit constraints (9). These two conflicting goals generate a trade-off in the allocator dynamics, which leads to significant performance degradation and loss of stability. (During the simulation runs, the actuator limits introduced in Section 3.3 are applied, i.e., $M = [20, 20, -, -, 15, 45, 25, 25]$ are used in (9). The third and fourth diagonal entries of W, which correspond to the inactive rudders, are set to 1000. This large weight is used to decrease the contribution of the faulty components of the control input vector. The remaining parameters of the allocator are chosen as follows: $\epsilon = 0.01$, $K = \text{diag}(0.1, 0.1, \ldots, 0.1)$.)

Simulations are also performed to see how setting the scheduling variable ρ_f of the LPV controller to 1 affects the response alone. By shifting ρ_f from 0 (nominal case) to 1 (faulty case), the required speed of response is adjusted. At $\rho_f = 0$, fast handling quality filters are set, which means a fast system response in the nominal case. As ρ_f is shifted towards 1, the poles of the prefilters decrease, which means slower reference tracking for the faulty aircraft. The results show that the double rudder fault cannot be accommodated by controller scheduling alone, and the ability to track β-reference is still lost.

Finally, Figs. 10 and 11 present simulation results when both input reallocation and fault-dependent scheduling are applied. Compared with the previous simulation scenarios, the improvement is significant. The aircraft managed to track both reference signals with acceptably small tracking errors. Errors can be observed due to the demanding nature of the problem and left aileron commands are reaching the saturation limits for longer periods, but the plane remains controllable in both the ϕ and β axes.

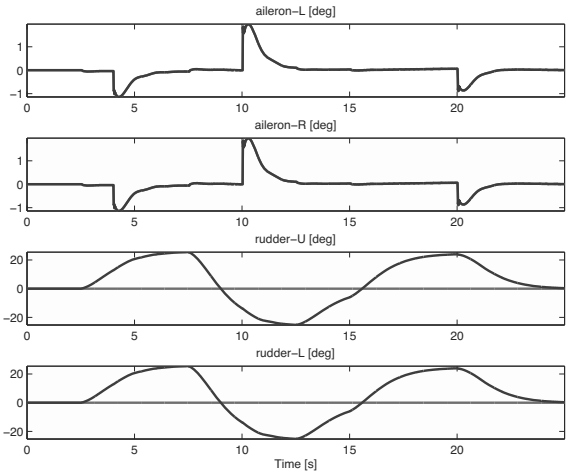


Fig. 6. First four control inputs in the complete rudder jamming case with a baseline controller. Black: the output of the controller, grey: the saturated input signals. (The rudders are switched off.)

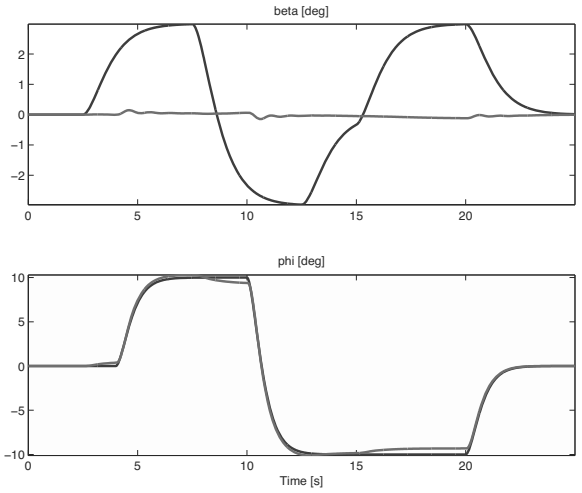


Fig. 7. Output tracking in the complete rudder jamming case with a baseline controller. Black: reference signals filtered by hq_β, hq_ϕ, grey: outputs of the GTM model.

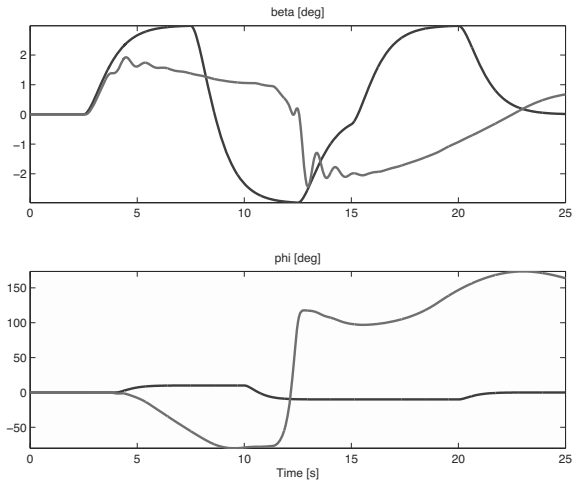


Fig. 9. Output tracking in the complete rudder jamming case using only input allocation. Black: reference signals filtered by hq_β, hq_ϕ, grey: outputs of the GTM model.

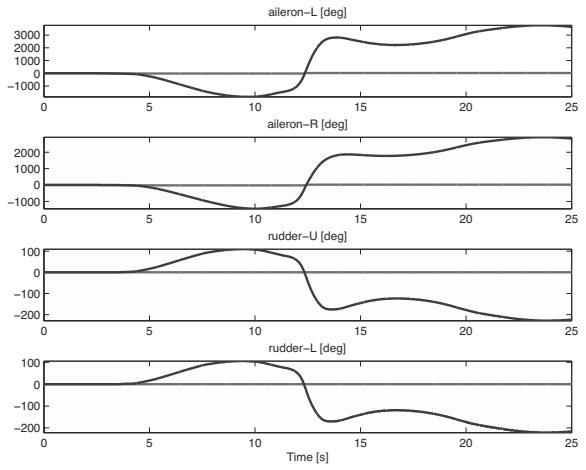


Fig. 8. First four control inputs in the complete rudder jamming case using only input allocation. Black: modified control inputs, grey: saturated input signals. (The rudders are switched off.)

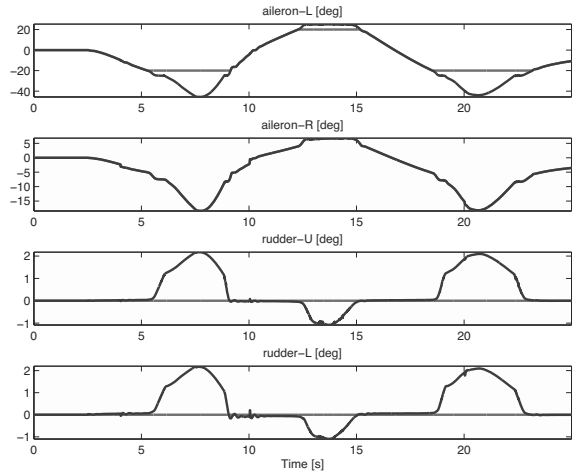


Fig. 10. First four control inputs in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: modified control inputs, grey: saturated input signals. (The rudders are switched off.)

Results of the dynamic allocation framework have to be compared with static allocation methods, where the allocator only depends on the null space of the system but no dynamics are included to shape its effect. Due to the numerical properties of the allocator implementation, the gain K shaping the dynamics of the allocator, as described in Eqn. (8), is changed from the baseline case of $K_b = 0.1$ to an almost static case of $K_b = 50$. Since the solver became stiff at higher values of K, this represents a quasi-static allocator case when the allocator dynamics do not play a role, only the kernel computation is taking care of the control effort distribution and the internal dynamics are changing much more rapidly than the dynamics of the closed-loop plant.

It can be seen in Fig. 12 that the tracking performance does not differ significantly between the two cases; on the other hand, the control input usage in the quasi-static case, shown in Figs. 13–14, is much more significant. The aileron is reaching higher positive deflections, and the rudder demand is closer to the ideal 0 deg deflection in the static case, but spoilers are in saturation for significantly longer time and the throttle usage is also drastically increased. This indicates that an additional degree of freedom in the allocator dynamics can be used to trade off time spent in saturation vs. tracking performance, while dynamic allocation is capable of handling sudden changes

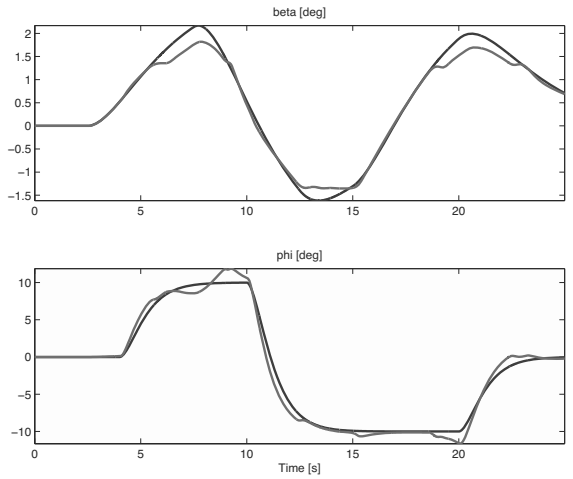


Fig. 11. Output tracking in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: reference signals filtered by hq_β, hq_ϕ, grey: outputs of the GTM model.

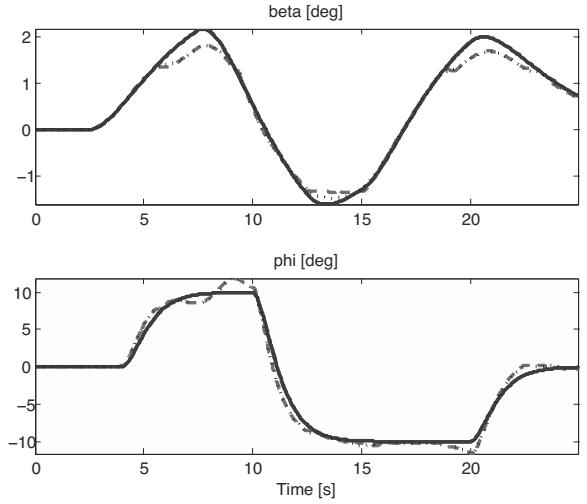


Fig. 12. Comparison of quasi-static allocation vs. dynamic allocation: output tracking in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: reference signals filtered by hq_β, hq_ϕ, grey dashed: outputs of the GTM model with baseline dynamic allocation, black dotted: outputs of the GTM model with quasi-static allocation.

in configurations (healthy and faulty) with bumpless transfer in control demand.

Fault detection and isolation results are not always correct, and the estimated position in the case of actuator jamming might differ from the true position of the surface. The allocation framework is tested against FDI related uncertainty first, with introducing a 0.5 deg offset between the detected jamming position and the actual location of the rudder surfaces; the true jamming is at 0.5 deg, while

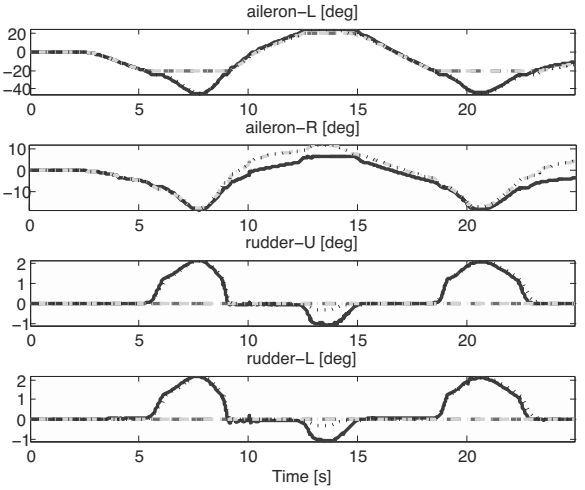


Fig. 13. Comparison of quasi-static allocation vs. dynamic allocation: aileron and rudder control inputs in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: baseline reference deflections, grey dashed: real actuator deflections, black dotted: quasi-static allocation deflection demands, light grey dash-dotted: quasi-static real actuator deflections.

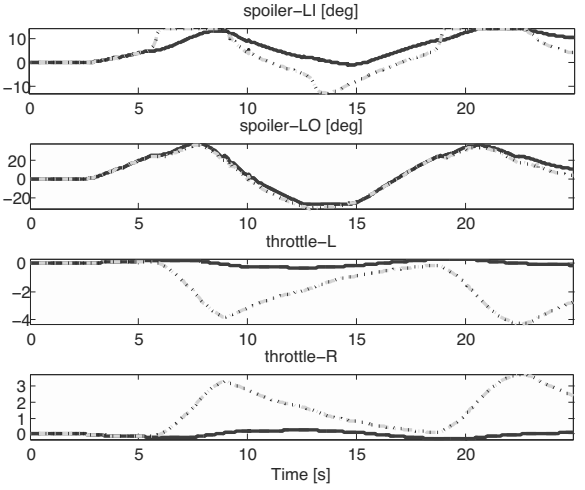


Fig. 14. Comparison of quasi-static allocation vs. dynamic allocation: spoiler and throttle control inputs in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: baseline reference deflections, grey dashed: real actuator deflections, black dotted: quasi-static allocation deflection demands, light grey dash-dotted: quasi-static real actuator deflections.

FDI assumes it to be 0 deg.

It can be seen clearly that the proposed control design architecture is able to handle the uncertainty caused by the offset. The tracking performance in positive sideslip commands are slightly better in the uncertain

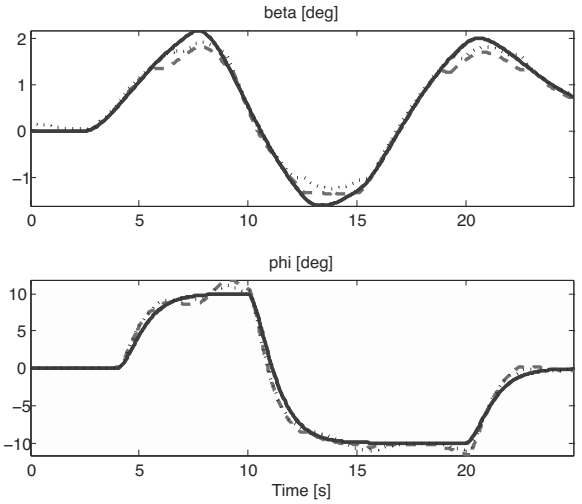


Fig. 15. Comparison of baseline with the FDI offset error case: output tracking in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: reference signals filtered by hq_β, hq_ϕ, grey dashed: outputs of the GTM model with baseline dynamic allocation, black dotted: outputs of the GTM model with FDI offset.

case, while negative sideslip commands is tracked better in the baseline case, as shown in Fig. 15. This is due to the fact that the true jamming is at 0.5 deg in the uncertain case, which helps slightly to achieve positive sideslip and reduces the control authority in negative sideslip commands.

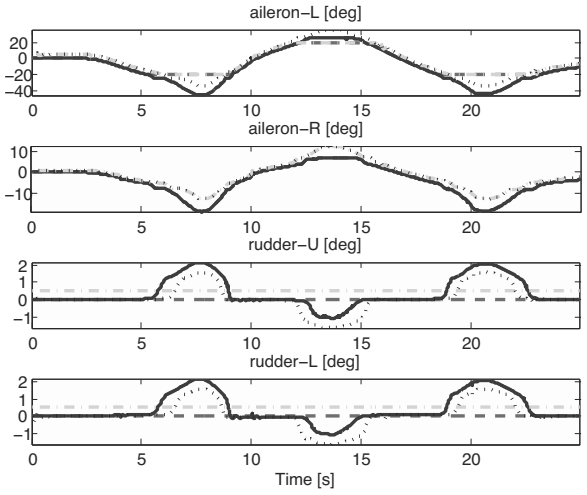


Fig. 16. Comparison of baseline with the FDI offset error case: aileron and rudder control inputs in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: baseline reference deflections, grey dashed: real actuator deflections, black dotted: FDI offset allocation deflection demands, light grey dash-dotted: FDI offset real actuator deflections.

It is also visible in Fig. 16 that rudder demands and actual deflections in the nominal and the uncertain case are different, following a similar trends as shown in the tracking error comparison plot.

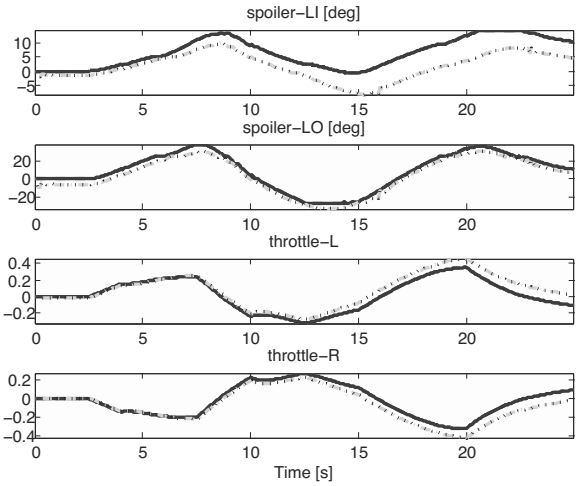


Fig. 17. Comparison of baseline with the FDI offset error case: spoiler and throttle control inputs in complete rudder jamming case using input allocation and fault-dependent scheduling. Black: baseline reference deflections, grey dashed: real actuator deflections, black dotted: FDI offset allocation deflection demands, light grey dash-dotted: FDI offset real actuator deflections.

The main way to counteract the model mismatch caused by the FDI error is to use the spoilers, as shown in Fig. 17. A significant difference can be seen between the two cases in terms of spoiler usage. This is caused by the particular tuning of the baseline LPV control systems and cannot be considered a general conclusion, but it shows how efficiently the uncertainty is handled.

To further account for uncertainties, a case with model mismatch is also included, as shown in Fig. 18. The LPV control law is not scheduled with the correct V_{cas} value, but is subjected to a 5 knots constant offset. This introduces a significant uncertainty between the design model and the true aircraft dynamics, which is handled exceptionally well by the control law. No visible difference or performance degradation can be seen on the tracking plots and the control inputs are also almost identical, only a slight glitch can be seen at 23 seconds in the aileron deflections, which indicates that at some scheduling variable values the performance might degrade. Since these parameter values are not kept for a long time, the plant is able to tolerate the scheduling variable offset.

The simulations above all assume perfect sensor information, since the original GTM simulation does not include any sensor model. To test the performance of the method, all the simulations above were also assessed with sensor noise included. 0.25 and 0.5 degree standard

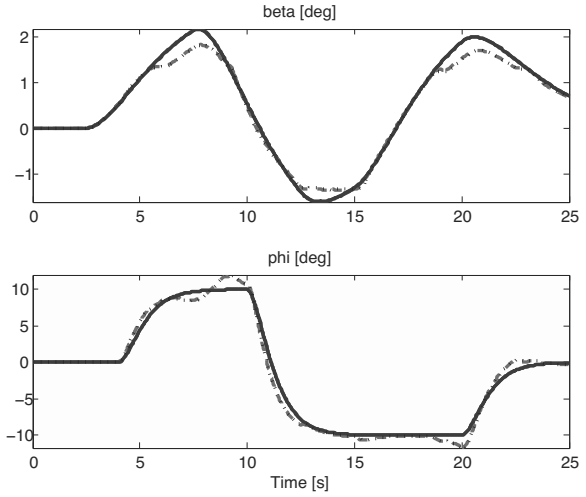


Fig. 18. Comparison of the nominal vs. uncertain system model: output tracking in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: reference signals filtered by hq_β, hq_ϕ, grey dashed: outputs of the GTM model with baseline dynamic allocation, black dotted: outputs of the GTM model with five knots uncertainty in LPV controller scheduling.

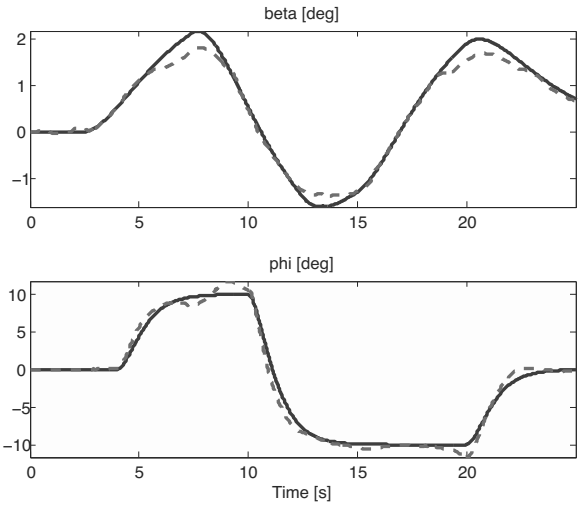


Fig. 19. System response with sensor noise: output tracking in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: reference signals filtered by hq_β, hq_ϕ, grey dashed: outputs of the GTM model with baseline dynamic allocation.

deviation Gaussian noise values are assumed on the angle and angular rate measurements, respectively. It can be seen in Fig. 19 that the tracking performance is virtually the same, while the control demands are subjected to the high frequency disturbance caused by the noisy signal sampled with 100 Hz, as shown in Fig. 20. In all the cases above, the responses with additional sensor noise are very similar and the trends are the same, but these results are not included in the paper to have a clearer view on the general behavior of the pure sensor outputs and the control commands.

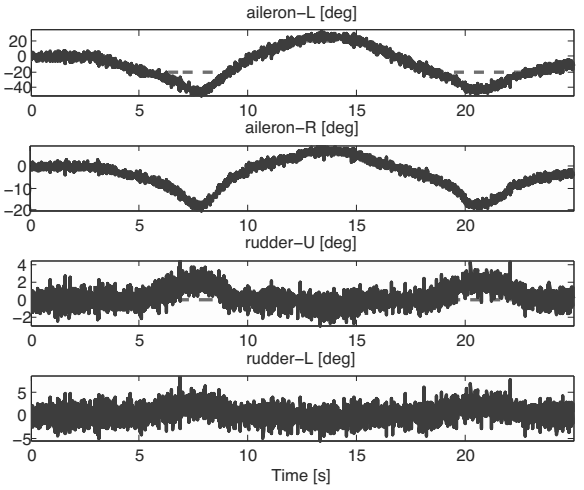


Fig. 20. Simulation with sensor noise: aileron and rudder control inputs in the complete rudder jamming case using input allocation and fault-dependent scheduling. Black: baseline reference deflections, grey dashed: real actuator deflections.

6. Conclusion

The present paper shows the advantages of the proposed two-level reconfigurable control approach based on control allocation and LPV performance scheduling. Due to the hardware redundancy of actuation surfaces onboard the current aircraft, it is often advantageous to re-allocate the control authority among the healthy actuators, but this is mostly done in a discrete way with pre-programmed routines which have to be tailored to each individual fault case, while the dynamic input allocation method presented here is fairly general to handle various kinds of failures within a common framework. On the other hand, in certain fault cases, the original handling quality requirements are no longer feasible due to reduced control authority, and hence graceful degradation of the performance is inevitable, which is scheduled with an LPV controller tuned to trade off performance vs. robustness. The complimentary properties of the two reconfiguration methods are handled by a supervisory logic which activates the required level when necessary. The advantages of the proposed method are highlighted in the paper with simulation results, where lateral commands are followed with various rudder fault conditions.

The method is presented on a simulation model of the GTM aircraft, but it is the aim of the authors to implement them on the high fidelity simulation model

of the RECONFIGURE benchmark to be assesed by Airbus. Furthermore, the allocation framework has to be working for all fault scenarios for various actuator and sensor fault cases. Hence the present results have to be embedded into a higher level overall framework, where fault tolerant control, fault detection and supervisory logic are all designed with taking their tradeoffs into account.

Acknowledgment

The authors greatly acknowledge the help of Dr. Irene Gregory and David Cox of the NASA Langley Research Center for making the GTM simulation model available for the purpose of this research.

The research leading to these results had received funding within the European Union's 7th Framework Programme (FP7/2007-2013) under the grant agreement no. FP7-AAT-2012-314544.

References

Alwi, H., Edwards, C. and Tan, C. (2011). *Fault Detection and Fault-Tolerant Control Using Sliding Modes*, Springer-Verlag, London.

Blanchini, F. (1999). Set invariance in control, *Automatica* **35**(11): 1747–1767.

Chakraborty, A., Seiler, P. and Balas, G.J. (2011). Nonlinear region of attraction analysis for flight control verification and validation, *Control Engineering Practice* **19**(4): 335–345.

Cunningham, K., Foster, J.V., Murch, A.M. and Morelli, E. (2008). Practical application of a subscale transport aircraft for flight research in control upset and failure conditions, *AIAA Guidance, Navigation, and Control Conference, Honolulu, HI, USA*, pp. 1–14.

Dorobantu, A., Murch, A.M. and Balas, G.J. (2012). H_∞ robust control design for the NASA AirSTAR flight test vehicle, *50th AIAA Aerospace Sciences Meeting, Nashville, TN, USA*, pp. 1–14.

Edwards, C., Alwi, H. and Tan, C.P. (2012). Sliding mode methods for fault detection and fault tolerant control with application to aerospace systems, *International Journal of Applied Mathematics and Computer Science* **22**(1): 109–124, DOI: 10.2478/v10006-012-0008-7.

Ganguli, S., Marcos, A. and Balas, G. (2002). Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis, *American Control Conference, Anchorage, AK, USA*, Vol. 5, pp. 3612–3617.

Gaspar, P., Nemeth, B. and Bokor, J. (2012). Design of an LPV-based integrated control for driver assistance systems, *Robust Control Design (7th ROCOND), Aalborg, Denmark*, pp. 511–516.

Gáspár, P., Szabó, Z. and Bokor, J. (2012). LPV design of fault-tolerant control for road vehicles, *International Journal of Applied Mathematics and Computer Science* **22**(1): 173–182, DOI: 10.2478/v10006-012-0013-x.

Gaspar, P., Szabo, Z. and Bokor, J. (2008). An integrated vehicle control with actuator reconfiguration, *IFAC World Congress, Seoul, Korea*, pp. 2087–2092.

Goupil, P. and Marcos, A. (2011). Advanced diagnosis for sustainable flight guidance and control: The European ADDSAFE project, *SAE Technical Paper 2011-01-2804*.

Goupil, P. and Marcos, A. (2012). Industrial benchmarking and evaluation of ADDSAFE FDD designs, *Fault Detection, Supervision and Safety of Technical Processes (8th SAFEPROCESS), Mexico City, Mexico*, pp. 1131–1136.

Johansen, T.A. and Fossen, T.I. (2013). Control allocation—a survey, *Automatica* **49**(5):1087–1103.

Mahmoud, M., Jiang, J. and Zhang, Y. (2003). *Active Fault Tolerant Control System*, Springer-Verlag, Berlin/Heidelberg.

Montes de Oca, S., Puig, V., Witczak, M. and Dziekan, Ł. (2012). Fault-tolerant control strategy for actuator faults using LPV techniques: Application to a two degree of freedom helicopter, *International Journal of Applied Mathematics and Computer Science* **22**(1): 161–171, DOI: 10.2478/v10006-012-0012-y.

Murch, A.M. (2008). A flight control system architecture for the NASA AirSTAR flight test infrastructure, *AIAA Guidance, Navigation, and Control Conference, Honolulu, HI, USA*, pp. 1–8.

Peni, T., Vanek, B., Szabo, Z. and Bokor, J. (2014). Dynamic sensor allocation framework for fault tolerant flight control, *19th IFAC World Congress, Cape Town, South Africa*, pp. 3482–3487.

Peni, T., Vanek, B., Szabo, Z., Gaspar, P. and Bokor, J. (2013). Supervisory fault tolerant control of the GTM UAV using LPV methods, *International Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France*, pp. 655–660.

Sloth, C., Esbensen, T. and Stoustrup, J. (2010). Active and passive fault-tolerant LPV control of wind turbines, *American Control Conference (ACC), 2010, Baltimore, MD, USA*, pp. 4640–4646.

Staroswiecki, M. (2006). Robust fault tolerant linear quadratic control based on admissible model matching, *45th IEEE Conference on Decision and Control, San Diego, CA, USA*, pp. 3506–3511.

Steinberg, M. (2005). A historical overview of research in reconfigurable flight control, *Aerospace Control and Guidance Systems Committee Meeting No. 95, Salt Lake City, UT, USA*, Subcomitee E, pp. 1–38.

Stoustrup, J. (2009). Plug and play control: Control technology towards new challenges, *European Journal of Control* **15**(3–4): 311–330.

Trachtler, A. (2004). Integrated vehicle dynamics control using active brake, steering and suspension systems, *International Journal of Vehicle Design* **36**(1): 1–12.

Xiao, H., Chen, W., Zhou, H. and Zu, J. (2011). Integrated control of active suspension system and electronic stability programme using hierarchical control strategy: Theory and experiment, *Vehicle System Dynamics* **49**(1–2): 381–397.

Yang, H., Jiang, B., Cocquempot, V. and Lu, L. (2012). Supervisory fault tolerant control with integrated fault detection and isolation: A switched system approach, *International Journal of Applied Mathematics and Computer Science* **22**(1): 87–97, DOI: 10.2478/v10006-012-0006-9.

Yu, F., Li, D. and Crolla, D. (2008). Integrated vehicle dynamics control: State-of-the art review, *IEEE Vehicle Power and Propulsion Conference, Harbin, China*, pp. 1–6.

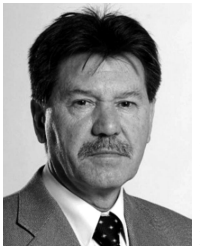
Zaccarian, L. (2009). Dynamic allocation for input redundant control systems, *Automatica* **45**(6): 1431–1438.

Zhang, Y. and Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems, *Annual Reviews in Control* **32**(2): 229–252.

Tamás Péni received an M.Sc. in informatics and a Ph.D. in electrical engineering from the Budapest University of Technology and Economics, Hungary, in 1999 and 2009, respectively. He is currently a senior research fellow with the Institute for Computer Science and Control, Hungarian Academy of Sciences (HAS). His main research interest includes constrained and fault tolerant control of linear parameter varying systems, as well as saturation compensation with applications in aerospace and transportation systems.

Bálint Vanek received an M.Sc. degree in mechanical engineering from the Budapest University of Technology and Economics in 2003, and a Ph.D. in aerospace engineering and mechanics from the University of Minnesota in 2008. He was a senior scientist in R&D at Honeywell Aerospace Advanced Technology, where he worked on flight control systems, unmanned aerial vehicles and navigation. He is currently a senior research fellow in the Institute for Computer Science and Control, Hungarian Academy of Sciences, where he leads the Aerospace Guidance Navigation and Control Group. His research interest includes fault detection and reconfigurable flight control of aerospace vehicles, robust control, fault-tolerant systems and insertion of unmanned aerial vehicles into the common airspace.

Zoltán Szabó obtained his B.Sc. and MSc. degrees in computer science in 1991 and 1993, respectively, and his a Ph.D. degree in applied mathematics from Eotvos Lorand University, Budapest, in 2003. He has been a doctor of the Hungarian Academy of Sciences (HAS) since 2011. Since 1996 he has been with the Systems and Control Laboratory (SCL) of the Institute for Computer Science and Control, HAS. As a research advisor of the SCL he is the head of the Theory of Dynamical Systems Research Group. His research interests include multivariable systems and robust control, system identification, fault detection with applications in power systems safety operations and also in the control of mechanical and vehicle structures. Currently he is involved in activities related to reconfigurable control and robust control of linear parameter varying systems.



József Bokor received Dr.Univ. and Ph.D. degrees in electrical engineering from the Budapest University of Technology and Economics (BUTE), in 1977 and 1983, respectively, and a D.Sc. degree in control from the Hungarian Academy of Sciences (HAS) in 1990. He was elected a corresponding member and later a full member of the HAS in 1999 and 2002, respectively. Currently, he is a research director and the head of the Systems and Control Laboratory of the Institute for Computer Science and Control, HAS. He is also a professor at the Department of Control and Transport Automation, Faculty of Transportation Engineering, BUTE. He has held visiting positions at the Imperial College of Science and Technology, London, the Systems and Control Group, Delft University, the MIT LIDS, Cambridge, MA, and at the Department of Aerospace and Mechanics, University of Minnesota, Minneapolis. His research interests include multivariable systems and robust control, system identification, fault detection with applications in power systems safety operations and also in the control of mechanical and vehicle structures. He is a fellow of the IFAC and the IEEE.

Appendix

The LPV state space matrices of the model are the following:

$$A_0 = \begin{bmatrix} -0.0214 & 0.2822 & -0.9661 & 0.4824 \\ 74.6640 & 2.5820 & 3.3225 & 0 \\ -35.2724 & -0.2907 & 0.0050 & 0 \\ 0 & 1 & 0.2828 & 0 \end{bmatrix},$$

$$A_V = \begin{bmatrix} -0.007 & -0.0025 \\ -2.1438 & -0.1102 \\ 0.8570 & 5.269\text{e}{-5} \\ 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} -2.2095\text{e}{-4} & -0.003 \\ -0.0144 & 0 \\ -0.0185 & 0 \\ -0.0025 & 0 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} 1.1585\text{e}{-4} & 1.3505\text{e}{-4} & 8.6333\text{e}{-5} \\ 0.4457 & 0.8599 & -0.2519 \\ -0.0038 & 0.0232 & 0.231 \\ 0 & 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 8.6333\text{e}{-5} & 1.4412\text{e}{-5} & 1.7615\text{e}{-5} \\ -0.1071 & 0.1693 & 0.4817 \\ 0.2341 & 0.0175 & 0.0498 \\ 0 & 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 3.7855\text{e}{-5} & -2.0125\text{e}{-5} \\ 0.0042 & -0.0046 \\ 0.0208 & -0.0207 \\ 0 & 0 \end{bmatrix},$$

$$B_V = \begin{bmatrix} -4.1098\text{e}{-6} & -1.3556\text{e}{-6} & 2.0292\text{e}{-5} \\ -0.0108 & -0.0178 & 0.0059 \\ 1.8183\text{e}{-5} & -6.0439\text{e}{-4} & -0.0058 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{matrix} 2.0292\text{e}{-5} & 3.2352\text{e}{-7} & 3.9542\text{e}{-7} \\ 0.0025 - 0.0038 & -0.0107 & \\ -0.0059 & -3.7078\text{e}{-4} & -0.0011 \\ 0 & 0 & 0 \end{matrix}$$

$$\begin{bmatrix} -2.2505\text{e}{-7} & 1.1964\text{e}{-7} \\ 8.1258\text{e}{-6} & -8.9243\text{e}{-6} \\ 3.9962\text{e}{-5} & -3.9736\text{e}{-5} \\ 0 & 0 \end{bmatrix}.$$

Received: 31 January 2014
Revised: 24 September 2014