Stall Recovery Guidance Using Fast Model Predictive Control

Stefan Schuet* Thomas Lombaerts† John Kaneshige‡
Kimberlee Shish§ Vahram Stepanyan¶
NASA Ames Research Center, Moffett Field, CA, 94035

Based on a detailed analysis of recent loss-of-control events, the Aircraft State Awareness Joint Safety Analysis Team has identified the need to develop algorithms and display strategies to provide control guidance for recovery from approach-to-stall or stall. In order to be effective, such guidance should enhance the pilot's ability to execute the Federal Aviation Administration's recommended stall recovery procedure. This paper explores the use of a fast model predictive control algorithm that determines near optimal recovery guidance, which quantifies the aircraft configuration and situation dependent recovery information required to maximize the effectiveness of the recovery. This information includes the magnitude of the initial pitch down maneuver, the specific amount of airspeed and thrust needed before pulling out of the recovery dive, as well as the maximum pitch-up rate that can be sustained without causing a secondary stall. The algorithm was integrated and tested with an in-house desktop simulator that implements the General Transport Aircraft model and the associated stall aircraft dynamics. Preliminary results are presented to demonstrate the use of the proposed approach as a recovery aid for pilots.

Nomenclature

α, β	angle-of-attack (rad.), side-slip angle (rad.)
θ, ϕ	pitch angle (rad.), roll angle (rad.)
V, γ	true airspeed (m/s), flight path angle (rad.)
g, ρ	acceleration due to gravity (m/s^2), air density (kg/m^3)
$S, \bar{q}$	wing surface area (m^2), dynamic pressure (Pa)
$\bar{c}, d_{\text{eng}}$	mean aerodynamic chord of the wing (m), engine diameter (m)
p, q, r	roll rate (rad./sec.), pitch rate (rad./sec.), yaw rate (rad./sec.)
f, x, u	continuous dynamics model, state vector, input vector
n	load factor, or dimension of state vector, depending on context
m	aircraft mass (kg), or dimension of input vector, depending on context
C_L, C_D, C_m	non-dimensional coefficients for lift, drag, and pitching moment
L, D, T	lift force (N), drag force (N), thrust produced by the engines (N)
TAS, CAS	true airspeed (m/s), calibrated airspeed (knots)
$\alpha_{\text{SR}}, \alpha_{\text{SW}}$	stall reference α (rad.), stall warning α (rad.)
$V_{\text{SR}}, V_{\text{SW}}$	stall reference airspeed (knots), stall warning airspeed (knots)
$\delta_{\text{e}}, \delta_{\text{fl}}$	elevator deflection angle (rad.), flap deflection angle (rad.)
$\delta_{\text{lg}}, \delta_{\text{sp}}$	landing gear deflection $\in [0, 1]$, speed brake deflection angle (rad.)
$\mathbb{R}^n$	set of real n-vectors
$\mathbb{S}^n_+$	set of $n \times n$ positive semi-definite symmetric matrices

*Researcher, Intelligent Systems Division, AIAA Member, email: stefan.r.schuet@nasa.gov
†Researcher, Stinger Ghaffarian Technologies, Intelligent Systems Division, AIAA Senior Member.
‡Researcher, Intelligent Systems Division, AIAA Member.
§Researcher, Millennium Engineering & Integration Company, Intelligent Systems Division, AIAA Member.
¶Senior Research Scientist, Universities Space Research Association, Intelligent Systems Division, AIAA Senior Member.

I. Introduction

The need for stall recovery guidance is driven by past accidents where pilots applied inappropriate control inputs that were opposite to what was necessary to recover the airplane.[1] This is just one aspect, though, of an overall joint industry and government team led effort to tackle the Loss of Control–Inflight (LOC-I) accident category for commercial aircraft.[1] The technology research addressing stall prevention and recovery spans multiple themes identified in accidents and incidents. These include a lack of aircraft state awareness due to factors such as poor visibility, sensor failure, automation system confusion, and pilot distraction.[1–6] In addition, improved simulator models with stall characteristics, and changes to pilot training for manual stall prevention and recovery were recommended.[7–10] In relation to this overall body of work, the focus of this paper is on the development of algorithms and guidance display strategies for helping pilots maximize the effectiveness of their stall training, on the rare occasion that it needs to be used.[2]

In an effort to address growing concerns over stall related training, the Federal Aviation Administration (FAA) recommends a stall recovery procedure to which all pilots should be trained.[11] This well considered procedure emphasizes reducing the angle-of-attack at the first indication of stall, while also being general enough to apply in a multitude of operational conditions including complications introduced by autopilot-induced excessive nose-up trim, and the need to avoid secondary stalls in the pull-out maneuver to the desired flight path angle. In this procedure there are, however, important aircraft and situational dependent details that are best provided by a computational recovery guidance routine. These include the magnitude of the initial pitch down maneuver, the specific amount of airspeed and thrust needed before pulling out of the recovery dive, as well as the maximum pitch-up rate that can be sustained without causing a secondary stall. Together, these elements affect how quickly the aircraft is returned to a safe flight path angle, and the loss of altitude required by the maneuver.

In this paper, we propose an online model predictive control computation for determining and providing scenario specific optimal stall recovery guidance for pilots of commercial transport aircraft. The approach uses the analytical kinematic and wind-axis force equations of motion to predict the expected aircraft dynamics and solves for the near optimal pitch maneuver that returns the aircraft state to a safe trimmed flight condition. The online computation of the maneuver ultimately requires solving a quadratic program, which is accomplished using the Fast Model Predictive Control (FMPC) interior point algorithm developed in Ref. 12. Furthermore, a preliminary model-based method for providing thrust guidance aimed at protecting against excessive nose-up pitch trim is presented. The end result, is a highly reliable algorithm for providing quantitative stall recovery guidance in agreement with the aforementioned FAA stall recovery procedure. The application of our approach is targeted at commercial transport class aircraft. As such, it does not explicitly consider some aspects of stall that are important for many smaller general aviation, unmanned aerial system, and military aircraft, such as stall induced spin.[13]

Extensive prior research has been conducted under the topic of upset prevention and recovery to address loss-of-control,[14] which is a multifaceted topic, where the type of stall considered here is often viewed as a simple case. For example, Gandhi et. al., develops a joint human automated recovery system for more generalized loss-of-control scenarios, in part by determining maneuver sequences that optimize loss of altitude subject to load factor and input constraints.[15] In that approach, however, a commercial nonlinear program solver[a] was used to precompute recovery procedures for storage in a lookup table that is accessed during flight operations. Reinforcement learning strategies for optimal upset recovery have also been considered for UAV applications.[16] In this approach large batches of simulation runs are used to train optimal recovery strategies for online use. As an alternative to the offline training based methods, constrained control approaches to the stall recovery guidance problem have been investigated.[17] One such approach uses Pseudo Control Hedging to adapt the system output commands to prevent control input saturation, for example by reducing the flight path angle to ensure angle-of-attack limits are not saturated. Another approach uses a physically motivated energy tradeoff analysis to derive the control guidance, while maintaining load factor limits and protecting against secondary stall.[18] The benefit of these methods is that the computational cost is significantly less than any optimal control method, and hence they are easily implemented for online use. The sacrifice of course, is that the sub-optimality of the resulting recovery maneuver is unclear.

In general, much of the prior work in upset recovery does not specifically address important operational complexities that can arise in real scenarios, such as an excessive nose-up pitching moment when too much thrust is applied, or the potential for a secondary stall to occur due to the coupled dynamics between pitch

[a] In this case, `fmincon` from MATLAB's optimization toolbox.

and the angle-of-attack during the pull-out phase of the recovery. The novel aspect of the approach presented in this paper is that it specifically addresses these important operational issues and provides quantitative pilot guidance by directly solving the associated optimal control problem at rates sufficiently fast and reliable enough for online use. We proceed as follows: in Sec. II we provide the motivation for our particular approach, in Sec. III we develop the flight dynamics model required to formulate our optimal control problem, in Sec. IV we show how to establish safe recovery targets, in Sec. V we review key aspects of the FMPC algorithm for computing solutions to the optimal control problem, in Sec. VI a pilot guidance display is proposed, in Sec. VII we explore a couple example recovery scenarios and report initial results before concluding in Sec. VIII.

II. Approach

The approach adopted in this paper is driven primarily by two factors. The first is of course to provide guidance that supports the manual stall prevention and recovery maneuver recommended in the FAA advisory on Stall Prevention and Recovery Training.[11] The second is to ensure the guidance is subjectively easy to follow, and that the guidance remains meaningful even if the pilot does not track it closely.

An abbreviated version of the FAA recommended stall recovery template from Ref. 11 is shown in Table 1. From the enumerated tasks (automation off, push, level, thrust, pull), one concludes that the recovery requires pitch, bank, and thrust guidance. Of these, the pitch guidance is the most important, since it is the primary means for arresting the stall, gaining airspeed, and recovering to a trimmed flight condition. Once the aircraft has sufficient airspeed, an existing method for providing bank guidance to level the wings will suffice. The application of thrust is also straightforward, in that it should be maximized to mitigate altitude loss, provided that care is taken to limit thrust in the presence of excessive nose-up pitch trim. However, the appropriate thrust depends on the pilot's desired objective after the completion of the recovery maneuver (such as, returning to level flight, or climbing out to a safe altitude), which should not be assumed known in advance. For this reason, we propose iteratively recomputing the pitch guidance at the current thrust setting, which may then change arbitrarily as the pilot follows (or does not follow) the thrust guidance cue. With this approach, the three guidance cues in pitch, bank, and thrust are independent of each other — and the pilot can follow them either separately, or together, as she or he deems appropriate.

Table 1: Abbreviated FAA Stall Recovery Template

1	**Disconnect autopilot and autothrottle/autothrust** Rationale: Leaving the autopilot or autothrottle/autothrust connected may result in inadvertent changes or adjustments that may not be easily recognized or appropriate, especially during high workload situations.
2	**(a) Nose down pitch control until impending stall indications are eliminated.** **(b) Nose down pitch trim as needed.** Rationale: Reducing the angle-of-attack is crucial for recovery. This will also address autopilot-induced excessive nose-up trim. If the control column does not provide sufficient response, pitch trim may be necessary.
3	**Bank wings level.** Rationale: This orients the lift vector for recovery.
4	**Apply thrust as needed.** Rationale: Amount of thrust depends on aircraft configuration and in some cases applying maximum thrust may create a strong nose-up pitching moment if airspeed is low.
5	**Retract speed brakes/spoilers.** Rationale: This will improve lift and stall margin.
6	**Return to the desired flightpath.** Rationale: Apply gentle action for recovery to avoid secondary stalls then return to desired flightpath.

III. Model Definition

The objective in this section is the development of a longitudinal flight dynamics model that is just complex enough to address the key operational aspects that are factored into the recovery template shown in Table 1. The desired end result is a model that can be used to quantify the amount of pitch down control required in step 2, the amount of thrust needed to effect a safe recovery in step 4, and the airspeed at which the pilot can begin step 6 along with maximum pitch up rate that can be sustained without exceeding the maximum angle-of-attack and causing a secondary stall.

A. Airspeed and Attitude Dynamics

The dynamics between true airspeed V, the angle-of-attack α, and the aircraft pitch θ, are captured by the wind-axes force equations[19]

$$\dot{V} = -\frac{SV^2\rho}{2m}C_D(\alpha) + \frac{T}{m}\cos(\alpha)\cos(\beta) + g\sin(\alpha)\cos(\beta)\cos(\phi)\cos(\theta) + g\sin(\beta)\sin(\phi)\cos(\theta) - g\sin(\theta)\cos(\alpha)\cos(\beta) \quad (1)$$

$$\dot{\alpha} = -\frac{SV\rho C_L(\alpha)}{2m\cos(\beta)} - \frac{T\sin(\alpha)}{Vm\cos(\beta)} + q - (p\cos(\alpha) + r\sin(\alpha))\tan(\beta) + \frac{g}{V\cos(\beta)}(\sin(\alpha)\sin(\theta) + \cos(\alpha)\cos(\phi)\cos(\theta)) \quad (2)$$

along with the kinematic equation

$$\dot{\theta} = q\cos(\phi) - r\sin(\phi), \quad (3)$$

where ϕ is the bank angle, β is the side-slip angle, p is the body-axis roll rate, q is the body-axis pitch rate, r is the body-axis yaw rate, S is the aerodynamic reference wing area, ρ is the air density at the current altitude, m is the aircraft mass, T is the engine thrust, and g is the acceleration due to gravity. These values are assumed known or estimated from current sensor measurements on board the aircraft.

The pilot uses control deflection to adjust the pitch rate according to a guidance signal u, the effect of which is modeled by setting

$$\dot{\theta} = k_\theta u, \quad (4)$$

where k_θ is a proportionality constant.[b] We then solve (3) for the body-axis pitch rate

$$q = \frac{1}{\cos(\phi)}\left[k_\theta u + r\sin(\phi)\right]. \quad (5)$$

After substituting this result for q into (2), the dynamics between the variables V, α, and θ are captured in equations (1), (2), and (4), assuming all other quantities are known. Finally, for values of α less than the critical stall angle, we assume the lift and drag coefficients are given by

$$C_D(\alpha) = C_{D_0} + C_{D_\alpha}\alpha + C_{D_{\alpha^2}}\alpha^2 \quad (6)$$

$$C_L(\alpha) = C_{L_0} + C_{L_\alpha}\alpha, \quad (7)$$

where the aerodynamic derivatives $C_{D_0}, C_{D_\alpha}, C_{D_{\alpha^2}}, C_{L_0}$, and C_{L_α} are functions of the aircraft configuration, i.e., flap and speed brake settings, known ahead of time, either through look-up tables or online system identification methods such as the one developed in Refs. 20 and 21. With these assumptions in place, our nonlinear system is represented by the expression $\dot{x} = f(x, u)$, where $x = (V, \alpha, \theta)$ and u is the pilot pitch-rate input defined above.

Our stall recovery guidance objectives will be met without requiring any specific stall dynamics modeling. While this is at first counter-intuitive given the nature of our problem, the decision is well motivated by the fact that the initial pitch down maneuver (step 2 in the FAA's recovery template), will almost immediately bring α into a valid region for the flight mechanics model just presented. That said, pilot stall recovery training, and the evaluation of any stall recovery guidance algorithm, are only properly performed with simulators that adequately model the aircraft's stall dynamics.[9,10]

[b] We also considered the use of a simple pilot model, where the pilot was expected to control the pitch in proportion to the pitch guidance error, i.e., by driving $\dot{\theta} = k_\theta(u - \theta)$. When used in a model predictive control framework however, this approach produces a guidance signal that tries to game the pilot into following the optimal trajectory. This guidance was subjectively viewed as frustrating to follow, and prone to causing pilot induced oscillation.

B. Model Discretization

Our model predictive control algorithm requires an approximate linear discrete dynamics model. This model is derived from the nonlinear model, by using a first order Taylor approximation to linearize $f(x,u)$

$$\dot{x} = f(x,u) \approx f(x_0,u_0) + J_x(x-x_0) + J_u(u-u_0),$$

where J_x and J_u are the Jacobian matrices of f with respect to x and u. Near x_0 and u_0 the model behaves according to

$$\delta\dot{x} = J_x\delta x(t) + J_u\delta u(t) + f(x_0,u_0),$$

where $\delta x = x - x_0$, and $\delta u = u - u_0$.

This linear system is then discretized by approximating $\delta u(t)$ as constant between samples, i.e., $\delta u(t) = u(k)$ for $kh \le t < (k+1)h$, where h is the sample time, and by using the following trick. First, let $\delta v(t) = J_u\delta u(t) + f(x_0,u_0)$, and notice that $\delta\dot{v} = 0$, for $kh < t < (k+1)h$, since $\delta u(t)$ is being treated as constant across each time sample. Without loss of generality, we seek the solution for $z(h)$ given $z(0)$, where

$$\dot{z}(t) = \begin{bmatrix} \delta\dot{x} \\ \delta\dot{v} \end{bmatrix} = \begin{bmatrix} J_x & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta v \end{bmatrix} = Gz. \tag{8}$$

The solution is provided by the matrix exponential, which can be expressed as

$$z(h) = \begin{bmatrix} \delta x(h) \\ \delta v(h) \end{bmatrix} = \exp(Gh)z(0) = \begin{bmatrix} A & F \\ 0 & I \end{bmatrix} \begin{bmatrix} \delta x(0) \\ \delta v(0) \end{bmatrix}. \tag{9}$$

We conclude $\delta x(h) = A\delta x(0) + F\delta v(0) = A\delta x(0) + FJ_u\delta u(0) + Ff(x_0,u_0)$. In the development below, we drop the δ notation when referring to the approximate discrete model, so that $x(k)$ and $u(k)$ always represent the deviation from the current set point (x_0,u_0). The approximate discretized system of interest is then expressed as

$$x(k+1) = Ax(k) + Bu(k) + w, \tag{10}$$

where A and F are taken from $\exp(Gh)$ as proscribed in (9), $B = FJ_u$, and $w = Ff(x_0,u_0)$. While the matrix exponential $\exp(Gh)$ is always defined, even if J_x is singular as in the present case, care must be taken in the computation.[22]

The use of the matrix exponential approach introduces significantly less discretization error over larger time steps compared to more basic integration schemes, like the first order Euler method. This enables the use of larger time steps and corresponding reduction in the computational cost of the fast model predictive controller we develop below.

C. Example

A comparison between the linearized discrete model and the nonlinear continuous model prediction for a pitch-up maneuver is shown in Fig. 1. The figure plots the state variable response for V, α, and θ from a zero degree pitch condition and input pitch rate of 3 deg/sec applied between 5 and 10 seconds. The solid blue line shows the continuous nonlinear system response, obtained by a Runga-Kutta integration of the nonlinear dynamics, using MATLAB's `ode23` function, along with the linearized discrete approximation obtained by iterating the linearized discrete system (10). The plot shows good agreement between the nonlinear and discrete approximation while the aircraft holds zero degree pitch and speeds up. Towards the end of the pitch-up maneuver the airspeed deviates from the discrete approximation due to the quadratic drag model employed by the nonlinear system, which kicks in at larger values of α. There is, however, excellent agreement in α across the entire 30 second time horizon. This is explained by the fact that (2) is nearly linear in α, for small values of α.

The solid red-line on the α plot in Fig. 1 shows the stall warning angle-of-attack α_{SW} (see Sec. IV.A.1 below). For this particular maneuver, a secondary stall warning would occur as α crosses α_{SW}, just after 5 seconds. For values of α much larger than α_{SW}, the nonlinear model equations (1) and (2) are not representative of the true aircraft dynamics, in particular, because equations (6) and (7) are not valid for high α. This is an important reason for developing guidance that recovers the aircraft to a safe state without exceeding α_{SW}.

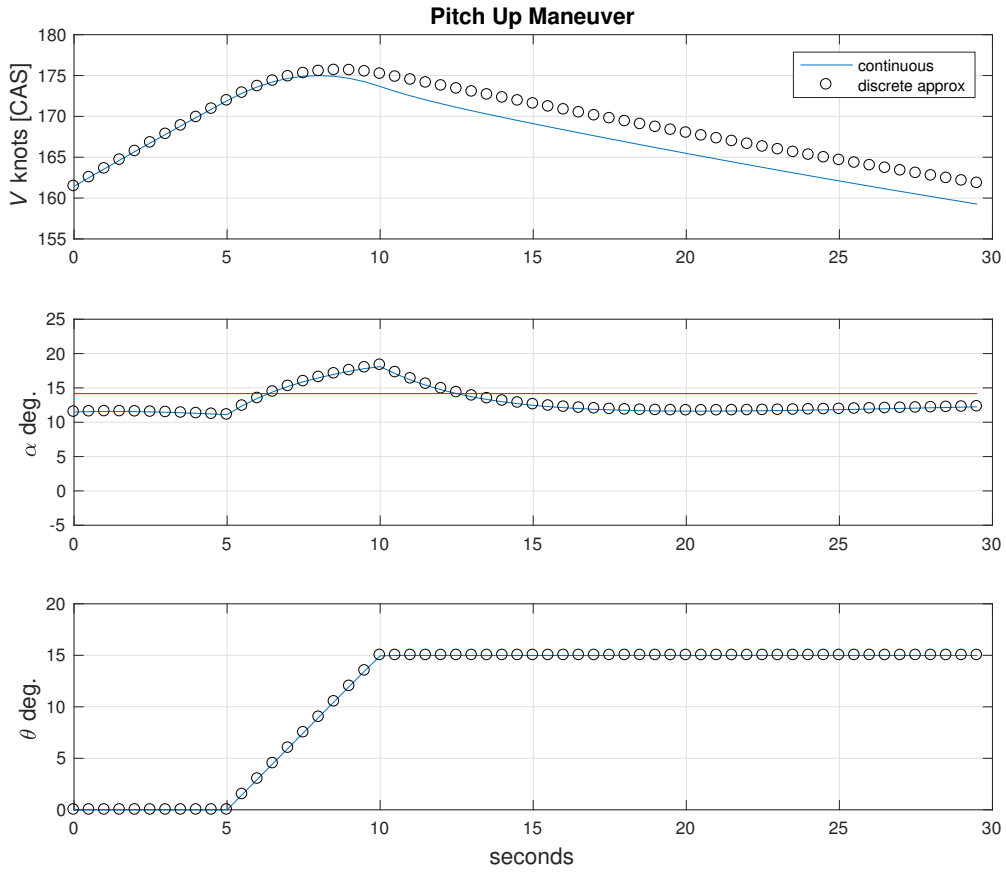

Figure 1: Comparison of nonlinear and discrete linear approximate dynamics model during an aggressive pitch up maneuver.

IV. Stall Recovery Targets

The purpose of a guidance algorithm is ultimately to bring the aircraft to a target state. For stall recovery guidance, though, there is no existing well defined target state that applies to all scenarios. In this section we investigate a few candidate possibilities, based on some basic requirements for what constitutes a successful recovery.

The first requirement is that the aircraft should be returned to a trim state, which by definition is a state for which the airspeed, flight path angle, thrust, and angle-of-attack are held constant in time. Furthermore, to align with the FAA recovery procedure, and human factors considerations, it also makes sense to decouple the pitch, thrust, and bank guidance systems, as previously discussed in Sec. II. It follows that the optimal pitch maneuver should be iteratively recomputed assuming the current bank and thrust setting. This enables the bank and thrust setting to change independently in time, though the bank target should be zero at the completion of the recovery. With these requirements in mind, a good strategy for defining a target recovery state, is to first determine a target airspeed, and then to use the current thrust to approximately determine both the flight path angle and angle-of-attack needed to hold the aircraft in trim.

A. Recovery Airspeed

In this section we define several reasonable values for the recovery airspeed target V_{T}. In the next section we show how the remaining components of the relevant target state vector $(V_{\mathrm{T}}, \gamma_{\mathrm{T}}, \alpha_{\mathrm{T}})$ are determined by enforcing a trim condition on the aircraft dynamics at the current thrust T, altitude, and aircraft configuration (i.e., mass, flap, and speed brake setting).

1. Stall Warning Airspeed — V_{SW}

The stall warning speed V_{SW} is defined in the Code of Federal Regulations (CFR) §1.1 as the speed at which onset of natural or artificial stall warning occurs. Here, we propose a working definition, that is meant to satisfy the more specific requirements of CFR §25.207, while also relating it to several other important quantities relevant to stall. This is accomplished in a manner that is consistent with the recovery dynamics model presented in Sec. III.

First we define the stall reference angle-of-attack α_{SR}, as the angle-of-attack that first maximizes the underlying *nonlinear* lift curve $C_L(\alpha)$ for the aircraft in its current configuration. The value of α_{SR} can be determined from representative aerodynamic coefficient lookup tables, or by flying the particular maneuver defined in CFR §25.103, while measuring the load factor corrected lift coefficient. With the value α_{SR} that maximizes C_L, and the load-factor defined as,

$$n = \frac{L}{W} = \frac{\rho S V^2 C_L(\alpha)}{2mg}, \tag{11}$$

we can solve for the stall reference speed V_{SR} as a function of the load-factor (for $n \geq 0$)

$$V_{\mathrm{SR}}(n) = \sqrt{\frac{2nmg}{\rho S C_L(\alpha_{\mathrm{SR}})}} \times \frac{V_{\mathrm{CAS}}}{V_{\mathrm{TAS}}} \text{ knots [CAS]}, \tag{12}$$

where the ratio $V_{\mathrm{CAS}}/V_{\mathrm{TAS}}$ converts true airspeed to calibrated airspeed. With this definition, $V_{\mathrm{SR}}(1)$ is the level flight airspeed, in knots, that corresponds to α_{SR}. Finally, in accordance with CFR §25.207, we choose to define

$$V_{\mathrm{SW}}(n) = \max(1.05 V_{\mathrm{SR}}(n), V_{\mathrm{SR}}(n) + 5) \text{ knots [CAS]}, \tag{13}$$

which has an explicit dependency on the load-factor, such that V_{SW} increases when the aircraft is "pulling g's." We can also find the corresponding stall warning angle-of-attack, at $n = 1$, by using (7) and (11) to obtain

$$\alpha_{\mathrm{SW}} = \frac{2mg}{\rho S C_{L_\alpha} \left[V_{\mathrm{SW}}(1) \times V_{\mathrm{TAS}}/V_{\mathrm{CAS}}\right]^2} - \frac{C_{L_0}}{C_{L_\alpha}}. \tag{14}$$

The use of (7) is warranted, because the lift curve should be linear in the neighborhood of α_{SW} to a good degree of approximation. Furthermore, by using $\hat{n} = \max(n, 1)$, the pair $V_{\mathrm{SW}}(\hat{n})$ and α_{SW}, can

be used to drive the stall warning (stick-shaker) indicator, and Pitch Limit Indicator (PLI), such that $\theta_{PLI} - \theta = \alpha_{\rm SW} - \alpha$, on the Primary Flight Display (PFD).

Recovery to $V_{\rm T} = V_{\rm SW}$ should be considered the minimum permissible target airspeed. In most scenarios, this airspeed would be an inadequate recovery target, at least not without adding additional margin. However, the stall warning airspeed typically corresponds to the stick-shaker warning on commercial aircraft, and it is central to the operational use of stall recovery guidance.

2. *Maneuvering Airspeed* — $V_{\rm mn}$

We define this airspeed as the minimum airspeed needed to sustain a full bank at $\phi = \phi_{\max}$ (typically, $\phi_{\max} = 35°$) at the target recovery flight path angle corresponding to the current thrust. In light of Fig. 2, trimmed flight requires

$$mg = L\cos\gamma\cos\phi = \frac{\rho S V_{\rm mn}^2 C_L(\alpha)}{2}\cos\gamma\cos\phi, \tag{15}$$

where γ is the flight path angle. For most commercial aircraft, the flight path angle sustained in trim is small, and one can use $\cos\gamma \approx 1$ in (15), immediately concluding that

$$V_{\rm mn} = \sqrt{\frac{2mg}{\rho S C_L(\alpha_{\rm SW})\cos\phi_{\max}}}, \qquad -90° < \phi_{\max} < 90°, \tag{16}$$

where $\alpha_{\rm SW}$ is the stall warning angle-of-attack defined above, which contains some margin to stall.

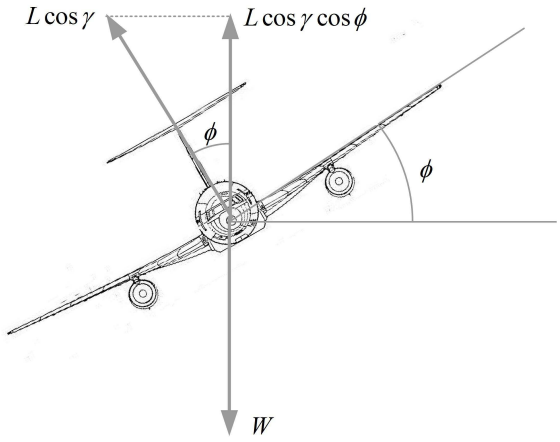

Figure 2: Force balance between lift and weight in a turn

The primary issue with using $V_{\rm mn}$ as the recovery target airspeed, however, is that it may still be on the back-side of the power curve for the aircraft, a state that can be counter-intuitive to a distracted pilot. This is because increasing airspeed from the back-side of the power curve may require an additional dive, particularly at high altitudes.

3. *Reference Airspeed* — $V_{\rm REF}$

Another airspeed target that may be suitable for stall recovery is $V_{\rm REF}$ as defined in the CFR. Technically, $V_{\rm REF}$ is the minimum final approach speed in landing configuration, which by design includes appreciable margin to stall. For commercial aircraft, CFR §25.125 specifies

$$V_{\rm REF} = 1.23 V_{\rm SR}(1) \ \text{ knots [CAS]},$$

where $V_{\rm SR}(1)$ is the stall reference speed, determined by (12) with the aircraft in landing configuration. Even though the FAA defines $V_{\rm REF}$ for landing configuration, it makes some sense to extend the definition to other configurations, by using the appropriate value for $V_{\rm SR}$. With this extension, setting $V_{\rm T} = V_{\rm REF}$ provides a target airspeed with the same margin to stall that is mandated by regulation for landing.

4. Minimum Drag Airspeed — V_{md}

Finally, the minimum drag airspeed V_{md}, defined as the airspeed that maximizes the lift-to-drag ratio, seems to be a very reasonable selection for the target recovery airspeed. Its value is computable using the same approximations we have employed so far, and it has the benefit of being the altitude dependent minimum airspeed on the front side of the power curve. It is also the airspeed that minimizes the glide slope at idle thrust, which may be useful in engine out scenarios.

B. Completing the Recovery Target

Once a target airspeed is specified, and converted to true airspeed if necessary, the target angle-of-attack and flight path angle can be determined from approximate conditions required to trim the aircraft at the current thrust. First, by combining (7) and (11) with $n \approx 1$ (for trimmed flight) one obtains

$$\alpha_{\mathrm{T}} = \frac{2mg}{\rho V_{\mathrm{T}}^2 S C_{L_\alpha}} - \frac{C_{L_0}}{C_{L_\alpha}}. \tag{17}$$

With this value for α_{T}, the required drag from (6) is

$$D = \rho S V_{\mathrm{T}}^2 (C_{D_0} + C_{D_\alpha}\alpha_{\mathrm{T}} + C_{D_{\alpha^2}}\alpha_{\mathrm{T}}^2)/2. \tag{18}$$

Finally, trim in the direction of the velocity vector at the current thrust setting T with $\cos\beta \approx 1$ requires

$$T\cos\alpha_{\mathrm{T}} = D - mg\sin\gamma_{\mathrm{T}}, \tag{19}$$

from which we can conclude

$$\gamma_{\mathrm{T}} = \arcsin\left(\frac{D - T\cos\alpha_{\mathrm{T}}}{mg}\right). \tag{20}$$

The final, approximate, trimmed stall recovery target is then $(V_{\mathrm{T}}, \gamma_{\mathrm{T}}, \alpha_{\mathrm{T}})$ defined at the current thrust T. Also note, in practice the current thrust must be estimated. In our approach, we use a look-up table that maps throttle position (or N1), airspeed, and altitude to the approximate net force T generated by the engines.

C. Elevator Constrained Max Thrust

The disconnection of the autopilot in accordance with step one of the FAA recovery template may cutout the auto-stab trim system. This can leave the aircraft in an excessive nose-up trim condition, which becomes dangerous when increased thrust causes a nose-up pitching moment that can not be countered by the elevators. For the general transport aircraft model of interest, the pitching moment coefficient can be modeled as

$$C_m = C_0 + C_{m_{\delta_{\mathrm{e}}}}\delta_{\mathrm{e}} + C_{m_{\delta_{\mathrm{e}}^2}}\delta_{\mathrm{e}}^2 + C_{m_T}\frac{T}{\bar{q}d_{\mathrm{eng}}^2}, \tag{21}$$

where,

$$C_0(\alpha, q, i_{\mathrm{h}}, \delta_{\mathrm{sp}}, \delta_{\mathrm{fl}}, \delta_{\mathrm{lg}}) = C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\alpha^2}}\alpha^2 + C_{m_q}\frac{q\bar{c}}{V} + C_{m_{i_{\mathrm{h}}}}i_{\mathrm{h}} + C_{m_{\delta_{\mathrm{sp}}}}\delta_{\mathrm{sp}} + C_{m_{\delta_{\mathrm{fl}}}}\delta_{\mathrm{fl}} + C_{m_{\delta_{\mathrm{lg}}}}\delta_{\mathrm{lg}},$$

and it is assumed that $\alpha = \alpha_T$, is fixed along with the positions of the stabilizer i_{h}, the spoilers δ_{sp}, the flaps δ_{fl}, the landing gear δ_{lg}, the mean aerodynamic chord $\bar{c}$, the engine diameter d_{eng}, and the dynamic pressure $\bar{q} = (1/2)\rho V^2$. In addition, to avoid the effects of the sometimes rapidly changing body-axis pitch rate, we assume as a worst case that $q = 0$ in this equation.[c] After setting (21) to zero, one can find the thrust T required to balance any elevator δ_{e} position, and in particular,

$$T_{\max}^{\mathrm{elev}} = -\frac{\bar{q}d_{\mathrm{eng}}^2}{C_{m_T}}(C_0 + C_{m_{\delta_{\mathrm{e}}}}\delta_{\mathrm{e}_{\max}} + C_{m_{\delta_{\mathrm{e}}^2}}\delta_{\mathrm{e}_{\max}}^2), \tag{22}$$

where $\delta_{\mathrm{e}_{\max}}$ is the maximum achievable elevator deflection. The sign convention for the coefficients in the above model is such that pushing the control column (or stick) forward causes the positive δ_{e} that would be required to arrest the pitch-up moment caused by increasing thrust ($C_{m_T} > 0$).

[c] Since $C_{m_q} < 0$, C_0 becomes more negative as the pitch rate increases, $T_{\max}$ gets bigger, and hence less restrictive under this model. If on the other hand, $q < 0$, and δ_{e} is at its maximum pitch down deflection, then of course there is no uncontrollable pitch up moment.

D. Thrust Guidance

Effecting a stall recovery essentially involves increasing the kinetic energy of the aircraft (and reducing α). This happens by converting fuel into thrust, and by sacrificing altitude. Minimizing altitude loss, therefore requires as much fuel to thrust conversion as possible, and this in turn means applying all available thrust as early as possible. In our implementation, we are evaluating the use of thrust guidance determined as $T_{\mathrm{g}} = \min(T_{\max}, T_{\max}^{\mathrm{elev}})$, i.e., the lesser of the maximum available thrust, or the maximum thrust that the elevator authority will allow.

It is important to consider the application of thrust from the training perspective as well, since the reduction of the angle-of-attack should be the first priority, as indicated in the FAA stall recovery template. From the training perspective, thrust should also only be added when it is safe to do so without contributing to an inappropriate pitch input, and it is this aspect that we want to help ensure with our approach.

V. Fast Model Predictive Control

Finally, we turn to the central computation considered in this paper: the real-time approximation of the optimal pitch recovery maneuver. Recall the definition of the state vector $x = (V, \alpha, \theta)$ and input $u \propto \dot{\theta}$. Together with the model equations (1), (2), and (4), the relevant nonlinear system $\dot{x} = f(x, u)$ is determined, as discussed in Sec. III.A. In each computational cycle, the nonlinear system is then linearized, and discretized, around the current aircraft state according to the process presented in Sec. III.B. The end results are the discretized state and input transition matrices A and B, respectively, as well as the exogenous error term w. Using the results of Sec. IV, a specific recovery airspeed is selected, and used to complete the specification of the recovery state target $x_{\mathrm{T}} = (V_{\mathrm{T}}, \alpha_{\mathrm{T}}, \theta_{\mathrm{T}})$, where $\theta_{\mathrm{T}} = \gamma_{\mathrm{T}} + \alpha_{\mathrm{T}}$, and the input target $u_{\mathrm{T}} \propto \dot{\theta}_{\mathrm{T}} = 0$. The optimal control problem is then formulated as a discrete finite horizon optimal control problem:

$$\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{k=1}^{N-1} [x(k) - x_{\mathrm{T}}]^T Q [x(k) - x_{\mathrm{T}}] \\
& \qquad + [x(N) - x_{\mathrm{T}}]^T Q_f [x(N) - x_{\mathrm{T}}] \\
& + \textstyle\sum_{k=0}^{N-1} [u(k) - u_{\mathrm{T}}]^T R [u(k) - u_{\mathrm{T}}] \\
\text{subject to} \quad & x(k+1) = Ax(k) + Bu(k) + w, \; x(0) = 0 \\
& x_{\min} \le x(k) \le x_{\max}, \; u_{\min} \le u(k) \le u_{\max}, \\
& \text{for } k = 0, \ldots, N-1,
\end{aligned} \tag{23}$$

where N is the time horizon, Q is the positive semi-definite state tracking cost matrix, R is the positive definite input tracking cost matrix, Q_f is the positive semi-definite final state cost matrix, $x_{\min/\max}$ specifies the given state box constraints, and $u_{\min/\max}$ specifies the given input box constraints. A unique feature of the optimal control formulation, is that the computed guidance is determined by a weighted positive-definite error with respect to the state target (V_{T}, α_{T}, θ_{T}). In this sense, the optimization objective is equivalent to minimizing a weighted mean-square error between the recovery trajectory and the recovery target, with a regularization term on the input control. By tuning the values of the weights (diagonal elements of Q) one can change the guidance to prioritize particular target values, such as the angle-of-attack, or the airspeed. By setting $Q = \mathbf{diag}(0, 1, 0)$, one obtains guidance that functions like an α controller; or by setting $Q = \mathbf{diag}(1, 0, 0)$, one obtains guidance that functions like an airspeed controller, etc.

The optimal control problem formulation (23) can be equivalently cast as the quadratic program

$$\begin{aligned}
\text{minimize} \quad & z^T H z + g^T z \\
\text{subject to} \quad & A_{\mathrm{eq}} z = b_{\mathrm{eq}} \\
& z_{\min} \le z \le z_{\max},
\end{aligned} \tag{24}$$

where

$$
\begin{aligned}
z &= \begin{bmatrix} u(0) & x(1) & u(1) & x(2) & \dots & u(N-1) & x(N) \end{bmatrix}^T \\
A_{\text{eq}} &= \begin{bmatrix}
-B & I & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\
0 & -A & -B & I & 0 & \dots & 0 & 0 & 0 \\
0 & 0 & 0 & -A & -B & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \dots & I & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \dots & -A & -B & I
\end{bmatrix} \\
b_{\text{eq}} &= \begin{bmatrix} Ax(0)+w \\ w \\ w \\ \vdots \\ w \end{bmatrix} \\
H &\in \mathbb{S}_+^{N(n+m)}, \text{ and block diagonal} \\
g &\in \mathbb{R}^{N(n+m)},
\end{aligned}
$$

and where $z_{\min/\max}$ specifies the state and input box constraints in correspondence with the definition of z. For our stall recovery problem it is sufficient to choose $N = 60$, for our system with $n = 3, m = 1$, so $z \in \mathbb{R}^{240}$. In addition, we will only require the use of diagonal matrices for Q, Q_f, and R (R is in fact a scalar in our specific problem formulation), and this implies that H is also a diagonal matrix.

By taking the fast model predictive control (FMPC) approach developed in Ref. 12, one gains the capability to solve (24) rapidly enough for online implementations. The FMPC algorithm acquires its computational speedup principally through three mechanisms. First, it takes advantage of the sparse banded block structure inherent to both A_{eq} and H. Second, it trades off accuracy with respect to the true optimal solution. This is done however, in such a way that the near optimal solutions returned by the algorithm are still guaranteed to fall inside the feasible set, even if the algorithm is terminated early. In practice this means all output solutions have some degree of built-in safety margin, which increases as the algorithm becomes more approximate. In our preliminary analysis, however, the FMPC solution was not significantly different from the commercial quadratic program solver with which it was compared (see Fig. 4). Finally, the computational cost can be reduced by using *warm start*, where the solution from a previous iteration is used to initialize the starting point for the next iteration. Each of these efficiencies improves the computational cost by an order of magnitude (roughly by a factor of 10). The net result is an algorithm that scales only linearly with N, and is 100-1000 times faster than some generic commercial tools such as MATLAB's quadratic program solver. To obtain a better understanding of the FMPC advantages summarized here, a closer examination of the algorithm is needed. This is the topic of the next subsection, which can be skipped on a first reading.

A. FMPC, the Main Idea

The main idea is centered around solving the convex quadratic program (24), through a series of approximations of the form

$$
\begin{aligned}
\text{minimize} \quad & f_0(z) = z^T H z + g^T z + \kappa \mathbf{1}^T \Phi(z, z_{\min}, z_{\max}) \\
\text{subject to} \quad & A_{\text{eq}} z = b_{\text{eq}},
\end{aligned}
\tag{25}
$$

where $\mathbf{1}$ is an appropriately sized vector of ones, and Φ is a log-barrier function defined as

$$
\Phi(z, z_{\min}, z_{\max}) = -\log(z - z_{\min}) - \log(z_{\max} - z).
$$

Here, we interpret the log of a vector as the component-wise log of each element in the vector, i.e., for $x \in \mathbb{R}^n, \log(x) = [\log(x_1), \log(x_2), \dots, \log(x_n)]^T$. Notice that Φ is only defined for values of z that are inside the inequality constraints (i.e., $z : z_{\min} \leq z \leq z_{\max}$), and that Φ goes to infinity as z approaches either

$z_{\min}$ from above or $z_{\max}$ from below. It is in this sense that Φ serves as an approximate surrogate for the inequality constraints in (24), and one that improves as the value of κ in (25) goes to zero from above.

The approximate problem (25), is a convex optimization problem, without inequality constraints, which is solved using Newton's method. In a basic version of Newton's method, one starts with a *feasible* initial guess z_0, and the second order Taylor approximation for the objective function in (25), i.e.,

$$\hat{f}_0(\Delta z) = \Delta z^T (H + H_\Phi)\Delta z + (g + g_\Phi)^T \Delta z + f_0(z_0), \tag{26}$$

where $\Delta z = z - z_0$, $H_\Phi = \nabla^2[(\kappa/2)\mathbf{1}^T\Phi(z_0, z_{\min}, z_{\max})]$, and $g_\Phi = \nabla[\kappa\mathbf{1}^T\Phi(z_0, z_{\min}, z_{\max})]$. After substituting (26) for f_0 in (25), and noting that $A_{\rm eq}(z + \Delta z) = b_{\rm eq} \Rightarrow A\Delta z = 0$ when $A_{\rm eq}z_0 = b_{\rm eq}$, one can solve the resulting equality constrained quadratic problem using linear algebra. To see this, start by forming the Lagrangian,

$$\mathcal{L}(\Delta z, \nu) = \hat{f}(\Delta z) + \nu^T A_{\rm eq}\Delta z,$$

where $\nu \in \mathbb{R}^{Nn}$ is the vector of Lagrange multipliers for the equality constraints. Minimizing (26) subject to $A_{\rm eq}\Delta z = 0$ is achieved by minimizing the Lagrangian jointly with respect to Δz and ν. This is accomplished by setting the gradient of $\mathcal{L}$ with respect to Δz and ν equal to zero, which produces the Karush-Kuhn-Tucker (KKT) system

$$\begin{bmatrix} \hat{H} & A_{\rm eq}^T \\ A_{\rm eq} & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \nu \end{bmatrix} = \begin{bmatrix} -\hat{g} \\ 0 \end{bmatrix}, \tag{27}$$

where $\hat{H} = 2(H + H_\Phi)$ and $\hat{g} = g + g_\Phi$. The KKT system is then solved for the update direction Δz, also known as the Newton step. The Newton step Δz is a descent direction for f_0 at z_0. The current guess is then updated to $z_0 = z_0 + t\Delta z$, where t is computed using a line search method that ensures f_0 decreases, and that the updated z_0 still satisfies the inequality constraints $z_{\min} \le z \le z_{\max}$, required to evaluate Φ. This process is then iterated until a stopping condition is satisfied, producing the numerical solution to (25). In general, Newton methods converge quadratically when the initial guess is close to the solution. This means that, for a good initial guess, no more than a few, e.g., 2–5, iterations are required for convergence.

Interior-point algorithms for solving (24) proceed as follows. Starting from an initial value for $\kappa > 0$, the surrogate problem (25) is solved by iterating Newton's method from an initial z_0 to convergence. This produces a solution within the interior of the constraint set that numerically solves (25), but is sub-optimal with respect to (24), roughly, to the degree that κ is greater than zero. The solution is then used to seed the initial guess in another iteration of the process at a reduced value of κ, and so on, until a stopping criterion is met.

This is the minimum theory required to appreciate the four most important aspects of the FMPC algorithm. First, observe that since H_Φ is a positive-definite diagonal matrix, it follows that $\hat{H}$ is a positive-definite block diagonal matrix.[d] The KKT matrix appearing on the left hand side of (27) is therefore always non-singular when $A_{\rm eq}$ is full rank. This property makes the algorithm numerically robust, in that it will not hang on non-singularity issues. Second, the computational efficiency of the FMPC algorithm stems from the following simple rearrangement of the KKT system,

$$\begin{aligned} \Delta z &= -\hat{H}^{-1}(g + A_{\rm eq}^T\nu) \\ A_{\rm eq}\hat{H}^{-1}A_{\rm eq}^T\nu &= -A_{\rm eq}\hat{H}^{-1}g. \end{aligned}$$

Computation with $\hat{H}^{-1}$ is trivial for the diagonal or block diagonal matrices we require. Furthermore, it can be shown that the matrix

$$A_{\rm eq}\hat{H}^{-1}A_{\rm eq}^T$$

is a banded block tridiagonal matrix, with $n \times n$ blocks and bandwidth $2n-1$, see Ch.10, of Ref. 23. A system of equations involving such a matrix can be solved in order n^3N floating point operations, a computation that grows only linear with the time horizon N (and in our case $n = 3$ is small). Third, in many practical optimal control applications, including ours, very little benefit is achieved from the reduction of κ beyond a certain point. In other words, for a reasonable value of κ, the solution to (25) is a sufficient approximation for the solution to (24), and one can forgo the additional computation required to converge on the precise solution to (24). Also, due to the nature of the log-barrier function, this approximate solution will be more

[d] In our particular case, $\hat{H}$ is diagonal, not just block diagonal.

conservative with respect to the constraints — and this is typically a good thing, especially if the constraints are safety related. Finally, additional computation can be saved by using a previous solution as the starting point for the next problem. This process, known as warm start, leverages the fact that, for optimal control problems, the formulation data changes only slightly between iterations of the control calculation. Often, this means the updated solution will not lie far from the previous solution. Furthermore, since Newton's method converges quadratically when the initial guess is close to the solution, a significant amount of computation is saved when compared to a method that starts from a less informed initial guess in each iteration of the control calculation.

There are important extensions to the basic theory presented above, such as, the use of an infeasible start Newton method, more general linear constraints, and time varying discrete linear systems. For the in depth discussion of the FMPC algorithm, please refer to Ref. 12. Also, chapters 10–11 of Ref. 23, detail the basic theory of interior point methods, and other optimization algorithm details.

Reference 12 also linked to available C-code for solving a less general version of (24) than we required to simultaneously work with an exogenous input term (produced by linearizing the flight dynamics model around a non-stationary set point), while also driving the system to a nonzero control target. To meet these requirements we implemented our own version, also in C using the BLAS and LAPACK libraries. In addition, a phase I method was developed for finding an initial feasible starting point from which to begin the FMPC algorithm. More importantly, the phase I method enables the determination of infeasible problem specifications that would cause the FMPC algorithm to fail, which allows one to employ a backup strategy if necessary. Finally, within our implementation of Newton's method, a backtracking line search from pg. 464 of Ref. 23 was used (with backtracking parameters $\alpha = 0.3, \beta = 0.8$), and our FMPC solutions were found by solving (25) with fixed $\kappa = 10$.

B. Constraints

A primary benefit of our optimal control approach is the ability to factor both state and input box constraints into the computation. The input constraints are used to ensure the guidance is reasonable. The state constraints ensure that the recovery maneuver is safe, by enforcing limits on V, α, and θ. Caution must be taken in specifying the constraints, however, as an overly constrained system can result in an infeasible optimization problem for which no guidance can be computed.

An important infeasibility issue occurs, for example, when the current α exceeds the maximum permitted value, and the lower limit on the pitch rate input constraint is insufficient to reduce α below this maximum in one time step h. The guidance algorithm, however, must produce a pitch down command in this critical situation. In our approach, we resolved this issue simply by making the minimum pitch-rate constraint sufficiently negative, i.e., loose, so that a solution can always be found, even if the pilot does not, or can not, exactly follow the computed guidance. To prevent overly aggressive pitch down guidance with this approach, a rate limit should be placed on the computed result before presenting it on the pilot display.

For stall recovery guidance, the upper limit on α is the most critical. Setting this limit to the value that maximizes the lift with margin, e.g., $\alpha_{\max} = \alpha_{\mathrm{SR}} - \alpha_{\mathrm{mrg}}$, enables the computation of the optimal pull-out (step 6 of the FAA recovery template) without causing a secondary aerodynamic stall. With this definition of $\alpha_{\max}$ one can further set the maximum permissible load factor for the pull out maneuver by observing that during the recovery $V \leq V_{\mathrm{T}}$, and that $C_L(\alpha)$ typically increases monotonically with α, up to $\alpha_{\max}$. From (11) one then obtains the upper bound

$$n \leq \frac{\rho V_{\mathrm{T}}^2 S C_L(\alpha_{\max})}{2mg},$$

which can be computed from the available problem data. It follows that a reasonable requirement on n may be satisfied by adjusting α_{mrg}, accordingly, while requiring that V_{T} is defined such that the corresponding $\alpha_{\mathrm{T}} < \alpha_{\max}$. A similar approach can be used to obtain a lower bond on n. In our work, with the particular transport aircraft model discussed below, we found that using $\alpha_{\max} = \alpha_{\mathrm{SW}}$ was sufficient to ensure guidance with reasonable load factor values for the recovery maneuver. Furthermore, it was not necessary to enforce tight constraints on either V or θ.

Finally, note that the box constraints specified in (23) and (24) are relative to current values for the state variables V, α, and θ that were used to set the linear approximation in Sec. III.B. In particular, with respect

to the notation in (23),

$$x_{\min} = \begin{bmatrix} V_{\min} - V \\ \alpha_{\min} - \alpha \\ \theta_{\min} - \theta \end{bmatrix} \quad x_{\max} = \begin{bmatrix} V_{\max} - V \\ \alpha_{\max} - \alpha \\ \theta_{\max} - \theta \end{bmatrix}.$$

C. Determining the Pitch Guidance

In each computational cycle or simulator frame, the pitch guidance is derived from the (nearly) optimal solution $z^\star$ to the problem (24) found by the FMPC algorithm. Based on an investigation of several (obvious) approaches, we propose driving the pitch guidance with $\theta_g = \max(u^\star(1), -3) + \theta$ deg., where $u^\star(1) = z_5^\star$. This way, θ_g is an approximation for the pitch angle the pilot should achieve in one second, when the pitch rate is driven by the *rate limited* (to greater than -3 deg./s) optimal value, 0.5 seconds into the future. This half-second lead time is motivated by the intuition that the pilot requires a short amount of time to achieve the guided pitch rate. The other elements of the solution vector $z^\star$ represent the optimal future inputs and states that would occur if the system were to evolve exactly according to the linearized predictive model. These future values represent a planning exercise for deciding the best current input without jeopardizing the future system safety. For stall recovery, this means ensuring no stall occurs during the recovery phase, due to the coupled dynamics between V, α, and θ. Of course, the pilot will not fly the guidance exactly (and may choose to ignore it completely), so when new information is available in the next simulator frame, the computation is repeated, yielding a revised estimate $z^\star$ and update to the current guidance command.

VI. Displays

In an effort to isolate the stall recovery algorithm from the effect of the guidance display, we use the standard, and well established, horizontal pitch director bar, as shown on the Attitude Directional Indicator (ADI) in Fig. 3. For the bank guidance, we use the standard vertical director bar (and algorithm) to level the wings when the airspeed crosses V_{SW}. While the airspeed is less than V_{SW}, no bank guidance is provided, i.e., the vertical flight director bar remains centered on the aircraft nose. Also shown are the standard stall warning (stick-shaker) limit on the airspeed tape, and the PLI in yellow on the ADI — driven by the calculations presented in Sec. IV.A.1. For thrust guidance however, there was no existing well established display. Based in part on work developed in Ref. 15, we propose the symbology shown on the left edge of the ADI in Fig. 3, which indicates the current thrust (left pointing triangle in white), and the guided thrust (right pointing triangle in magenta) on a scale of zero to one. The red band appearing on the thrust guidance indicator shows limited thrust due to excessive nose-up pitch trim — computed using the calculation discussed in Sec. IV.D. The scaling of the thrust information needed for this display requires tabulating the, aircraft dependent, minimum and maximum available engine thrust as a function of airspeed and altitude. The thrust guidance display appears when the system transitions to Stall Recovery Guidance (SRG) mode, as shown on the flight mode annunciator.

VII. Results

A. GTM Simulation Model

The stall recovery guidance algorithm was integrated and tested with the General Transport Model (GTM) simulator.[24] This simulator is representative of a generic aircraft similar to a Boeing 757, and it includes the stall dynamics. It was developed from a sub-scale polynomial aerodynamic database, extended to cover the stall regime with wind-tunnel and spin-tunnel test data. The model was further adapted to full-scale by making Reynolds Number corrections.

As discussed in Sec. III.A, our stall recovery guidance algorithm requires only an approximation for the flight dynamics in the nominal flight regime, i.e., not stalled. For the GTM model this corresponds to $\alpha \leq \alpha_{SR} = 16$ deg. Within this regime, suitable approximations for the non-dimensional drag and lift coefficients are

$$C_D(\alpha) = C_{D_0} + C_{D_\alpha}\alpha + C_{D_{\alpha^2}}\alpha^2 + C_{D_{\delta_{sp}}}\delta_{sp} + C_{D_{\delta_{fl}}}\delta_{fl} + C_{D_{\delta_{lg}}}\delta_{lg} + C_{D_{\alpha\delta_{fl}}}\alpha\delta_{fl} \tag{28}$$

$$C_L(\alpha) = C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta_{sp}}}\delta_{sp} + C_{L_{\delta_{fl}}}\delta_{fl} + C_{L_{\delta_{lg}}}\delta_{lg} \tag{29}$$

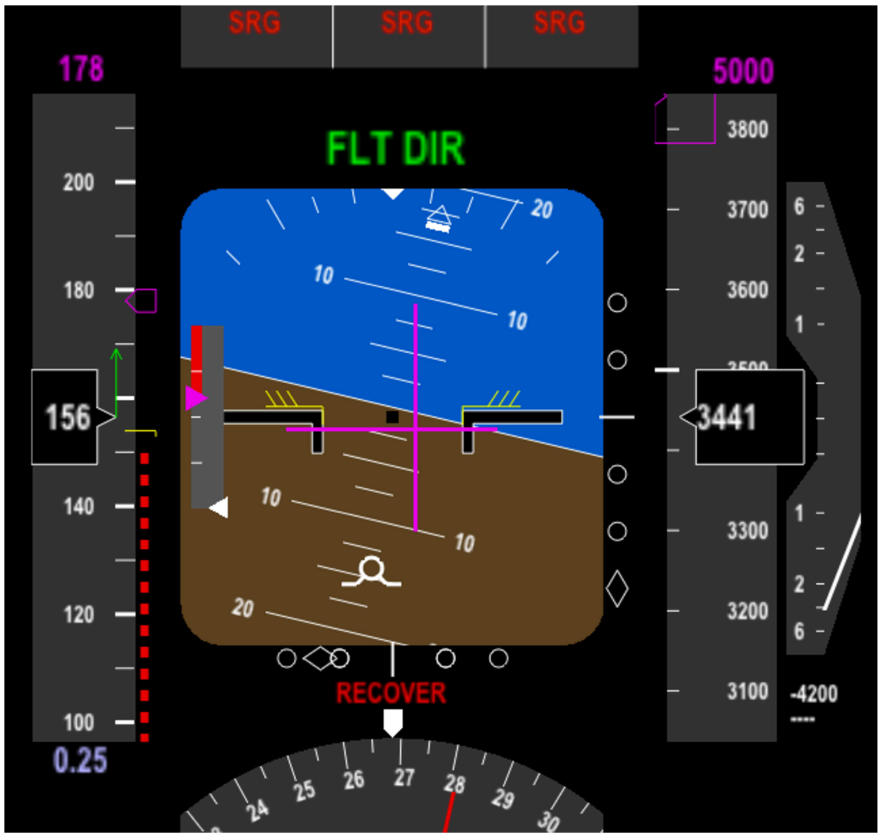

Figure 3: Pitch, bank, and thrust guidance displays during recovery from aerodynamic stall.

where δ_{sp} is the spoiler deflection, δ_{fl} is the flap deflection, and $\delta_{\text{lg}} \in [0, 1]$ is the gear setting. The equation for C_m was provided earlier, see (21). Equations (6) and (7), follow from (28) and (29) by fixing the associated terms for the aircraft configuration, i.e., values for δ_{sp}, δ_{fl}, and δ_{lg}. Representative values for the required derivatives are presented in Table 2. In the examples discussed below, the aircraft mass $m = 83{,}806$ kg (approximately landing weight), the wing surface area $S = 181.25$ m^2, mean aerodynamic chord $\bar{c} = 5.072$ m, engine diameter $d_{\text{eng}} = 2.146$ m, and the air density ρ varied with altitude according to the 1976 U.S. Standard Atmosphere.[25]

Table 2: Aerodynamic Coefficient Approximation for the GTM Simulator Model ($\alpha \leq 16$ deg.)

C_{D_0}	C_{D_α}	$C_{D_{\alpha^2}}$	$C_{D_{\delta_{\text{sp}}}}$	$C_{D_{\delta_{\text{fl}}}}$	$C_{D_{\delta_{\text{lg}}}}$	$C_{D_{\alpha\delta_{\text{fl}}}}$
0.02	-0.086	2.7	-0.011	0.13	0.037	0.81

C_{L_0}	C_{L_α}	$C_{L_{\delta_{\text{sp}}}}$	$C_{L_{\delta_{\text{fl}}}}$	$C_{L_{\delta_{\text{lg}}}}$
0.11	4.6	0.88	1.5	-0.027

C_{m_0}	C_{m_α}	$C_{m_{\alpha^2}}$	C_{m_q}	$C_{m_{\delta_{\text{e}}}}$	$C_{m_{\delta_{\text{e}}^2}}$	$C_{m_{i_{\text{h}}}}$	C_{m_T}	$C_{m_{\delta_{\text{sp}}}}$	$C_{m_{\delta_{\text{fl}}}}$	$C_{m_{\delta_{\text{lg}}}}$
0.33	-3.2	6.0	-15	-1.7	-0.54	-3.3	-0.093	-0.12	-0.35	0.013

B. Initial Results

Figure 4 shows an optimal recovery plan obtained by solving the model predictive control problem (23) for a scenario involving an aircraft at high altitude 35,000 ft (10,668 m, $\rho = 0.373$ kg/m^3, $g = 9.77$ m/s^2), in a clean configuration, slowing into a stall from a pitch-up attitude with 15 degree bank. The FMPC guidance calculation was setup with $k_\theta = 1$, $h = 0.5$ and $N = 60$. The nonlinear dynamics model was linearized around the initial state shown at $t = 0$ in the plot. The cost matrices were selected to penalize tracking error in units of m/s for airspeed and deg. for angles, i.e., by selecting $Q = \mathbf{diag}(1, a, a)$, and $R = a$, where $a = (180/\pi)^2$. Note, the diagonal entries of Q can be adapted to adjust the nature of the controller, for example by discounting the weight on the airspeed error term to create more of an α controller. For the constraints, we used $20 \leq V \leq 240$ m/s, $-2 \leq \alpha \leq \alpha_{\text{SW}}$ deg., $-30 \leq \theta \leq 30$ deg., and $-180 \leq u \leq 10$ deg./s. As discussed in Sec. V.B, the only active constraint is the upper limit on α, while the constraints on V, θ, and especially u were intentionally left loose. Two different computations of the optimal solution are also shown in the figure: the first computed using the `quadprog` function in MATLAB's optimization toolbox (taking 0.28 seconds), and the second computed using the FMPC algorithm (taking 0.0029 seconds), *without warm start*. As expected, the FMPC algorithm achieves a close approximation to the solution obtained with the commercial solver, but is roughly 100 times faster in this case. It can also be observed that the FMPC algorithm is slightly more conservative with respect the constraints, as one should expect.

The optimal solution shown in Fig. 4 contains the quantitative, non-obvious and situation dependent, information missing from the FAA's stall recovery template. First, it shows that in order to ultimately recover to $V_{\text{T}} = V_{\text{REF}} = 180$ knots [CAS], with $\alpha_{\text{T}} = 9.76$ deg., the pilot should pitch down as aggressively as possible (but within safety limits), to about -7 deg., which returns α to a safe angle in less than 0.5 seconds. The plan then indicates the pitch-up maneuver should begin after about 3 seconds at a smooth rate sufficient to ensure α does not exceed $\alpha_{\text{SW}} = 14$ deg., its maximum permissible in the problem formulation.

Of course, the computed solution is only a plan, and the pilot may not follow it exactly at each time step. One reason, pertinent to the the present example, is that the computed maneuver may be too aggressive for the pilot to follow.[e] It is for this reason that the optimal solution is recomputed at each time step (in this case at the simulator frame rate of 50 Hz). This approach ensures that the optimal guidance provided represents the best policy from the current aircraft state, independent of what the pilot actually decides to do.

Figure 5 shows a down-sampled subset of the continuously updated optimal recovery trajectories in gray, as the aircraft flies its course, shown in blue before the start of the recovery maneuver, and in magenta during

[e]This was intentional per the discussion in Sec. V.B.

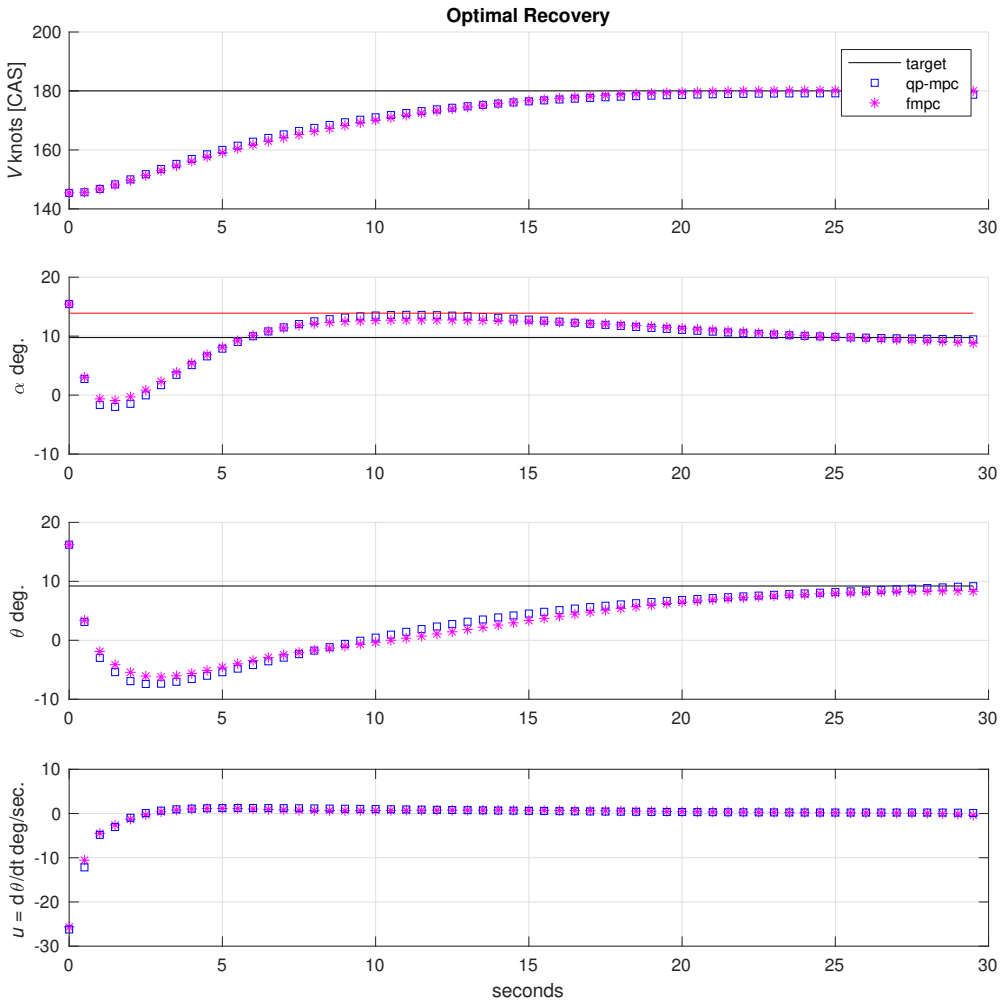

Figure 4: Optimal recovery plan.

the recovery maneuver. In this baseline example, a rate limited autopilot controls the aircraft. As shown, the autopilot was setup to ignore the recovery guidance, essentially flying into a stall when α exceeds $\alpha_{\mathrm{SR}} = 16$ deg. During this time the algorithm is continuously advising increasingly severe pitch down maneuvers. When the recovery maneuver begins, at 0 seconds, the autopilot attempts to follow the guidance, which includes commanding thrust and bank ϕ (to level the aircraft), as shown in the bottom two plots. In this case, the autopilot lags the optimal recovery, which is not rate limited, but is still able to recover the aircraft to level flight at $V_{\mathrm{T}} = V_{\mathrm{REF}} = 180$ knots (CAS) without exceeding $\alpha_{\mathrm{SW}} = 14$ deg.

The guidance, however, was not intended for driving an autopilot. Figure 6 shows a manual piloted recovery for the same scenario, using an open loop controller with yaw damper — as flown by a professional pilot. The result shows some minor pilot oscillations as compared to the autopilot controlled example. For the open loop aircraft, the sensitive pitch control response was a contributing factor to the observed oscillation. Overall, though, the pilot was able to return to the recovery target (level flight at 180 knots), in about the same amount of time as the autopilot example. Furthermore, the α plot shows no secondary stall warnings (indicated by crossings of the red line) occurred during the recovery. The average computation time for each FMPC trajectory update was 0.47 ± 0.14 ms when our realtime implementation compiled from C-code was used.[f]

C. Recovery Metrics

Important objective measures for evaluating the effectiveness of a stall recovery include: time to reduce α below α_{SW}, load factor, and the number of secondary stick-shaker warnings that occur during the recovery. While the FAA has deemphasized altitude loss for stall prevention and recovery *training*,[11] we propose that altitude loss is still a valid metric for evaluating recovery guidance effectiveness.

Figure 7 compares these metrics for three recoveries: manual piloted control with no stall recovery guidance, manual piloted control with FMPC recovery guidance, and auto-piloted control with FMPC recovery guidance. This time, an approach stall scenario was used to setup each of the recoveries. In this scenario, the aircraft was initialized at 3,000 ft (914.4 m, $\rho = 1.121$ kg/m^3, $g = 9.804$ m/s^2) in landing configuration (full flaps and gear), descending at 700 ft/min (-3 deg. flight path angle), and decelerating towards stall with the throttle at idle. At $\alpha = 16$ deg., corresponding to $t = 0$ in the figure, the pilot (or auto-pilot) is given the go ahead to begin the recovery maneuver. The recovery task is to catch the airspeed target $V_{\mathrm{T}} = V_{\mathrm{REF}} = 136$ knots [CAS], or to accelerate through it, at level or climbing flight. The FMPC algorithm was configured in exactly the same way as for the high altitude scenario used in the previous section.

All three recoveries shown in Fig. 7, are achieved in less than about 10 seconds, with the no guidance manual recovery taking a few seconds longer to level off than the others. In addition, when the guidance was provided, the pilot applied max thrust earlier and more aggressively than in the other cases, which resulted in about a 200 ft altitude savings. The guided recoveries also exhibited better load factors. No secondary stick shakers occurred in all three cases, though increased margin to α_{SW} is observed during the pull-out phase of the recovery when guidance was provided. Both the piloted and auto-piloted recoveries lagged the pitch guidance cue, and this resulted in an overshooting of the target airspeed (which is acceptable for stall recovery). Also, notice the significantly reduced time to recover with full flaps, and greater engine thrust authority at lower altitude, as compared to the high altitude recovery discussed in the previous section, which took about 30 seconds to complete. The manual recovery was flown by a test pilot, first without and then with the guidance switched on.

VIII. Conclusion

The use of model predictive control to provide recovery guidance for stall prevention and recovery was proposed. The motivation for this approach was intrinsic, and based on using the coupled flight dynamics between airspeed, pitch, and angle-of-attack to determine the non-obvious optimal pitch maneuver that recovers the aircraft to a given target airspeed without causing a secondary stall. In addition, a model-based approach for protecting against over-thrust in the presence of an excessive nose-up pitch trim condition was proposed as a method for driving thrust guidance during the recovery. Furthermore, our problem adaptation for use with the Fast Model Predictive Control algorithm was shown to produce a significant speed improvement (roughly by a factor of 1000), over a commercially available, but general purpose, quadratic

[f] On a Macbook Pro with 2.6 GHz Intel Core i7 processor.

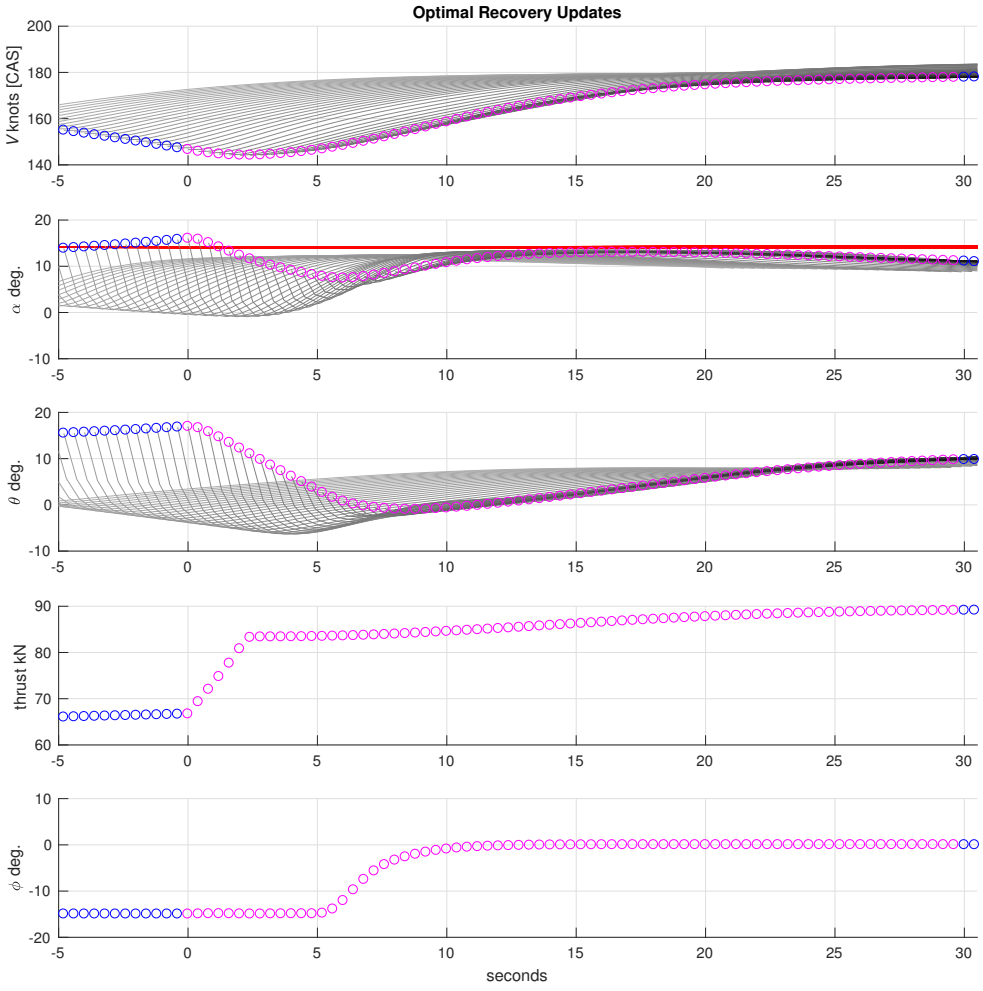

Figure 5: Autopilot controlled aircraft: optimal recovery updates through time, with changing thrust and bank angle.

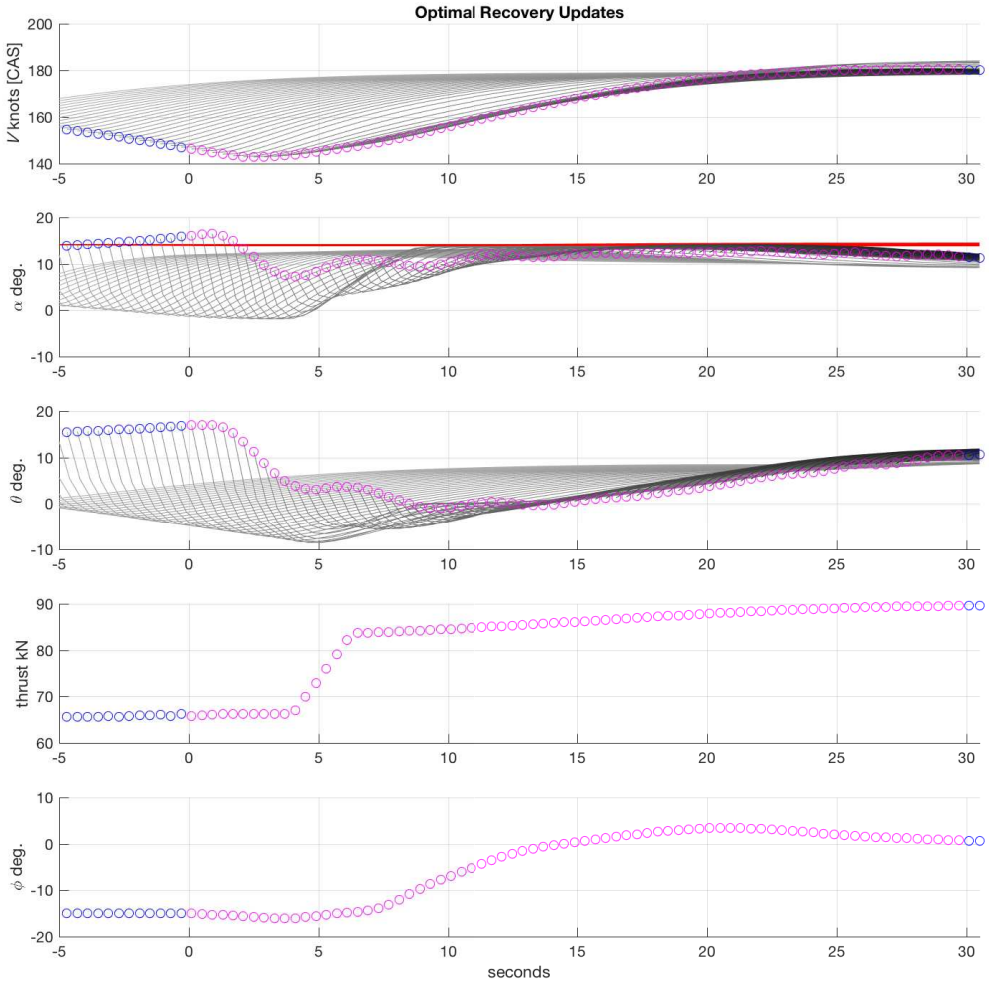

Figure 6: Manually controlled aircraft: optimal recovery updates through time, with changing thrust and bank angle.

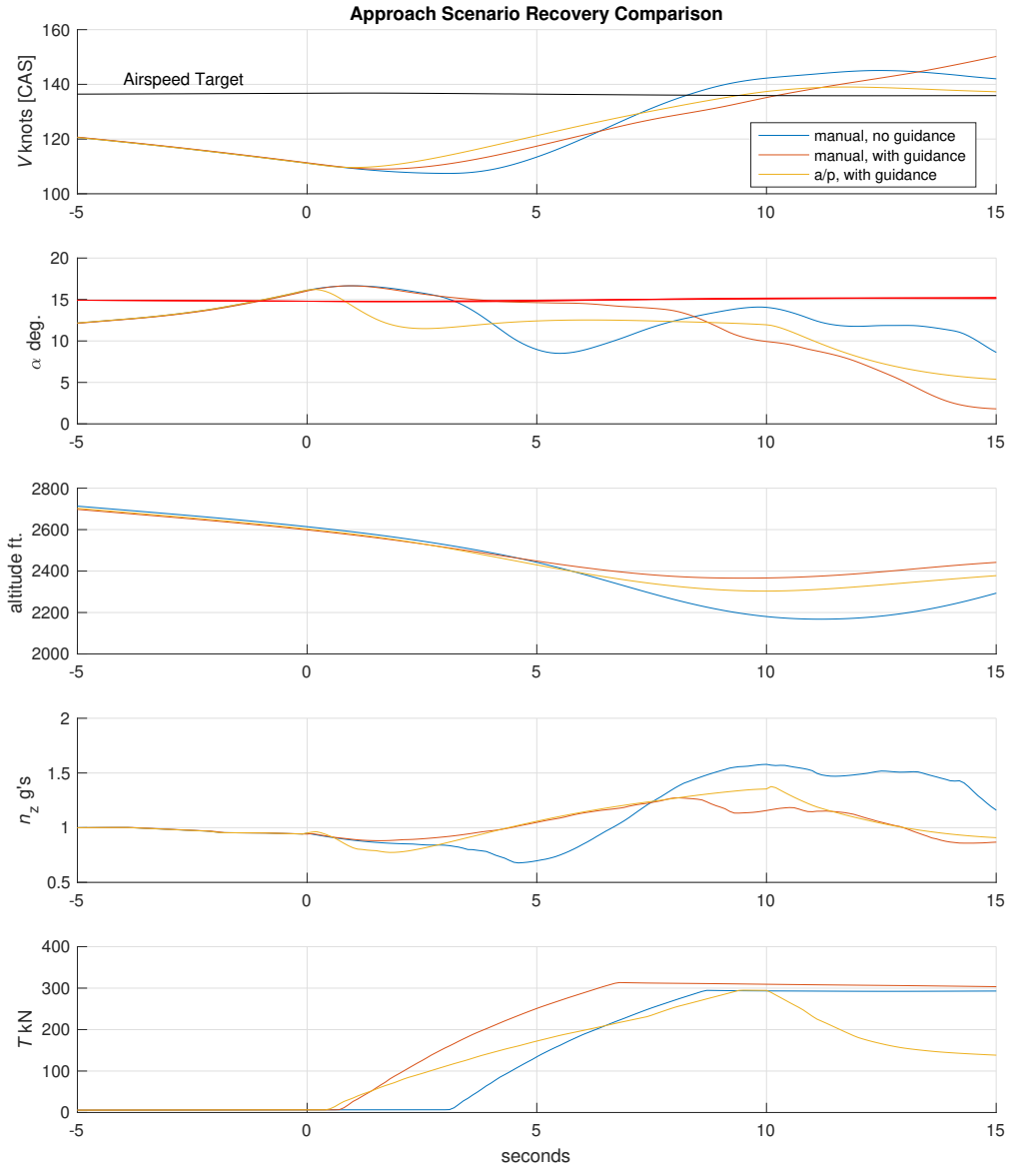

Figure 7: Recovery comparison.

program solver. This enables the real-time use of the proposed guidance algorithm, without having to rely on precomputed look-up tables that may not cover the full variability of potential conditions that one may require (e.g., changes in altitude, thrust, stall entry condition, as well as, flap, gear, and speed brake settings).

We also examined a few specific piloted recoveries in both a high altitude, and an approach scenario. This was meant only to gain a better understanding for how the resulting guidance may serve as a pilot aid during the recovery. The algorithms and results presented so far are preliminary. In the future, a planned simulation study in the Vertical Motion Simulator Facility, at NASA Ames Research Center, will be used to achieve more comprehensive evaluation, involving many different pilots. More work is also required to establish a concept for operational use on aircraft and integration with the existing flight director and automation system.

Acknowledgements

This work was supported by NASA's Airspace Technology Demonstrations (ATD) project, through the Technologies for Airplane State Awareness (TASA) activity. The authors also sincerely thank test pilot Gordon Hardy for his evaluation and feedback on the proposed guidance algorithm, which were based on initial fixed-base simulation runs in the Advanced Controls Technologies lab at NASA Ames Research Center.

References

[1]Airplane State Awareness Joint Safety Analysis Team. Final report analysis and results. Technical report, Commercial Aviation Safety Team, 2014.

[2]Commercial Aviation Safety Team. Safety enhancement SE 207 ASA research – attitude and energy state awareness technologies. http://www.skybrary.aero/bookshelf/books/2538.pdf. Accessed online December 6, 2016.

[3]Commercial Aviation Safety Team. Safety enhancement SE 208 ASA research – airplane systems awareness. http://www.skybrary.aero/bookshelf/books/2539.pdf. Accessed online December 6, 2016.

[4]Steven D. Young, Maarten Uijt De Haag, Taumi Daniels, Emory Evans, Kimberlee H. Shish, Stefan Schuet, Timothy Etherington, and Daniel Kiggins. Evaluating technologies for improved airplane state awareness and prediction. In *AIAA Infotech @ Aerospace*. American Institute of Aeronautics and Astronautics, January 2016. doi:10.2514/6.2016-2043.

[5]Kimberlee Shish, John Kaneshige, Diana Acosta, Stefan Schuet, Thomas Lombaerts, Lynne Martin, and Avinash N. Madavan. Aircraft mode and energy-state prediction, assessment, and alerting. *Journal of Guidance, Control, and Dynamics*, accessed December 6, 2016. doi: http://dx.doi.org/10.2514/1.G001765.

[6]Thomas Lombaerts, Stefan Schuet, Diana Acosta, John Kaneshige, Kimberlee Shish, and Lynne Martin. Piloted simulator evaluation of safe flight envelope display indicators for loss of control avoidance. *Journal of Guidance, Control, and Dynamics*, accessed September 15, 2016. doi: http://dx.doi.org/10.2514/1.G001740.

[7]Commercial Aviation Safety Team. Safety enhancement SE 196 R1.3 ASA – training - effective upset prevention and recovery training, including approach -to -stall. http://www.skybrary.aero/bookshelf/books/2520.pdf. Accessed online December 6, 2016.

[8]Commercial Aviation Safety Team. Safety enhancement concept SE 209.1 (R&D) ASA – research – simulator fidelity. http://www.skybrary.aero/bookshelf/books/2537.pdf. Accessed online December 6, 2016.

[9]Jeffery A. Schroeder, Judith S. Burki-Cohen, David Shikany, David R. Gingras, and Paul P. Desrochers. An evaluation of several stall models for commercial transport training. In *AIAA Modeling and Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics, January 2014.

[10]Peter M. T. Zaal, Jeffery A. Schroeder, and William W. Chung. Transfer of training on the vertical motion simulator. *Journal of Aircraft*, 52(6):1971–1984, 2015.

[11]Federal Aviation Administration. Advisory circular on stall prevention and recovery training. U.S. Department of Transportation, AC120-109A, 2015.

[12]Yang Wang and S. Boyd. Fast model predictive control using online optimization. *Control Systems Technology, IEEE Transactions on*, 18(2):267–278, March 2010.

[13]Roberto Bunge, Felipe Munera Savino, and Ilan Kroo. Approaches to automatic stall/spin detection based on small-scale uav flight testing. In *AIAA Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics, June 2015. doi:10.2514/6.2015-2235.

[14]Christine Belcastro. Loss of control prevention and recovery: Onboard guidance, control, and systems technologies. In *AIAA Guidance, Navigation, and Control Conference*, number AIAA 2012-4762. American Institute of Aeronautics and Astronautics, 2012.

[15]Neha Gandhi, Nathan Richards, and Alec Bateman. Desktop simulator demonstration of a joint human/automated upset recovery system. In *AIAA Guidance, Navigation, and Control Conference*, number AIAA 2012-4820. American Institute of Aeronautics and Astronautics, 2012.

[16]Brian Dutoi, Nathan Richards, Neha Gandhi, David Ward, and John Leonard. Hybrid robust control and reinforcement learning for optimal upset recovery. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2008.

[17]Vahram Stepanyan, Kalmanje S. Krishnakumar, John Kaneshige, and Diana M. Acosta. Stall recovery guidance al-

gorithms based on constrained control approaches. In *AIAA Guidance, Navigation, and Control Conference*, number AIAA 2016-0878. American Institute of Aeronautics and Astronautics, January 2016.

[18]Thomas Lombaerts, Stefan Schuet, John Kaneshige, Kimberlee Shish, and Vahram Stepanyan. Stall recovery guidance using an energy based algorithm. In *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, January 2017.

[19]Vladislav Klein and Eugene A. Morelli. *Aircraft System Identification, Theory and Practice*, chapter 3. American Institute of Aeronautics and Astronautics, Inc., 2006.

[20]Stefan Schuet, Thomas Lombaerts, Diana Acosta, Kevin Wheeler, and John Kaneshige. An adaptive nonlinear aircraft maneuvering envelope estimation approach for online applications. In *AIAA Guidance, Navigation, and Control Conference*, number AIAA 2014-0268, January 2014.

[21]Stefan Schuet, Thomas Lombaerts, Diana Acosta, John Kaneshige, Kevin Wheeler, and Kimberlee Shish. Autonomous flight envelope estimation for loss-of-control prevention. *Journal of Guidance, Control, and Dynamics*, 2016. doi: 10.2514/1.G001729.

[22]Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1), 2003.

[23]S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[24]Richard M. Hueschen. Development of the transport class model (tcm) aircraft simulation from a sub-scale generic transport model (gtm) simulation. Technical Report NASA/TM–2011-217169, NASA Langley Research Center, Hampton, Virginia, August 2011. Available at: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110014509.pdf.

[25]*U.S. Standard Atmosphere, 1976*. National Aeronautics and Space Administration, Washington, D.C., October 1976.