AIAA AVIATION Forum, Modeling and Simulation Technologies Conference, 15–19 June 2020

A Modeling Approach for Handling Qualities and Controls Safety Analysis of Electric Air Taxi Vehicles

Stefan Schuet* Carlos Malpica† Thomas Lombaerts‡
John Kaneshige§ Shannah Withrow¶
Gordon Hardy‖ Jeremy Aires**
NASA Ames Research Center, Moffett Field, CA 94035

The combination of modern advances in electric propulsion, fly-by-wire controls, autonomy, and increasing demand for short range air taxi operations, is currently producing an outburst of vehicle designs more diverse than ever before. Advanced software tools are needed to support the rapid and safe introduction of any design into the airspace, including the safety of the deployed flight control system and vehicle handling qualities. This paper presents a methodology for building air taxi vehicle models with distributed electric propulsion for use in analyzing flight control system safety at the conceptual design level. The approach builds on existing software tools capable of outputting aeromechanics-based linear perturbation models for Vertical Take-off and Landing vehicles with multiple rotors. Rotor torque inputs are then converted into equivalent voltage control inputs, and the linear state and input dynamics matrices are modified to include electric motor dynamics with common parameters for direct-current electric motors. The linear perturbation dynamics are then stitched across multiple operating points into a quasi-Linear Parameter Varying model that covers the full flight envelope. A Model Predictive Controller is developed for use with the full envelope model, and a tradeoff analysis between handling quality and motor requirements is demonstrated using a six passenger NASA air taxi reference design.

Nomenclature

FlightCODE	Flight dynamics and control modeling tool for COnceptual DEsign
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
NDI	Nonlinear Dynamic Inversion
qLPV	quasi-Linear Parameter Varying
RPM	Rotations Per Minute
UAM	Urban Air Mobility
VTOL	Vertical Takeoff and Landing

I. Introduction

Recent demand for Urban Air Mobility (UAM), with advancements in control, automation, and electric propulsion, have fostered the rapid emergence of novel Vertical Takeoff and Landing (VTOL) air taxi vehicle concepts.[1,2] While interest in these vehicles is driven by an expected increase in demand for air taxi

*Computer Engineer, Intelligent Systems Division, Mail Stop 269-3, AIAA Member, stefan.r.schuet@nasa.gov.
†Aerospace Engineer, Aeromechanics Office, Mail Stop 243-11, AIAA Member.
‡Aerospace Research Engineer, KBR Wyle Services, Intelligent Systems Division, Mail Stop 269-1, AIAA Associate Fellow.
§Computer Engineer, Intelligent Systems Division, Mail Stop 269-1, AIAA member.
¶Aerospace Engineer, Aeromechanics Office, Mail Stop 243-11, AIAA Member.
‖Research Pilot, Flight Research Associates Inc., Moffett Field, California 94035, AIAA member.
**Computer Engineer, Intelligent Systems Division, Mail Stop 269-1.

transportation, to be satisfied ultimately with autonomous operation, the near term introduction of such vehicles into the airspace requires a careful analysis of overall safety. In this regard, there are many questions that should be addressed at the conceptual vehicle design level, chief among them, the degree to which a vehicle can meet mission-based performance metrics for handling qualities, whether operated by a pilot or autonomous system. Addressing these questions requires vehicle dynamics models sufficient for understanding the flight control system design trade-space between widely varying UAM vehicle concepts. This includes the ability to assess handling qualities ratings in accordance with accepted standards like ADS-33E-PRF[3] through piloted simulation study.

Over the past decade, NASA, working in collaboration with the U.S. Army Aviation Development Directorate, has established multi-disciplinary tools for evaluating rotorcraft conceptual designs including handling qualities.[4] Through this work, one can now use NASA Design and Analysis of Rotorcraft (NDARC)[5] to perform mission-based air vehicle design and optimization. With the Flight dynamics and control modeling tool for COnceptual DEsign (FlightCODE), flight dynamics models are generated from NDARC outputs, and then linked with the established CONtrol Designer's Unified InTerface (CONDUIT)[6] to produce a vehicle model with control system design suitable for handling qualities assessment.[a]

The established tool chain currently used at NASA is dependent on CONDUIT, and until recently, was missing the incorporation of models for electric propulsion that are needed for handling qualities and controls safety *design* studies, including the analysis of tradeoffs between traditional collective (or blade pitch) control with rotor-speed (or RPM) control.[7] In this paper we seek to extend the current capability by showing how to build basic electric propulsion into the linear perturbation flight dynamics models output from tools like FlightCODE or FLIGHTLAB,[8] in a manner that is suitable for use with other modern control methodologies such as Model Predictive Control and Nonlinear Dynamic Inversion.[9,10] This will eventually enable comparison with CONDUIT, and extend the tool set for studying the effects of reduced or failed motor operating capacity on high-level performance capability during a vehicle's conceptual design stage. Such studies are needed to meet emerging safety requirements for UAM operations, that will require continued safe operations in the event of any single failure, without requiring exceptional piloting skill or strength.[11]

We proceed as follows. In Section II, a direct-current electric motor model is reviewed as a useful means for capturing the conversion of electrical-power to mechanical-power, with design parameters for electrical system losses, mechanical transmission losses, operating voltage, as well as, the rotor rate and torque needed for hover. Section III then shows how the electric motor model is integrated with a given full-envelope quasi-Linear-Parameter-Varying (qLPV) bare-airframe model. In Section IV, a discrete Model Predictive Controller is developed for use with the integrated qLPV model. Finally, in Section V the proposed system is used to analyze the tradeoff between ADS-33E handling qualities and motor requirements for a conceptual NASA reference six-passenger quad-rotor air taxi design.

II. Electric Motor Model

The brushless Direct Current (DC) motor model is viewed as an approximate surrogate for the significantly more sophisticated electric propulsion system that might be adopted in an actual vehicle. In the basic model, applying Kirchhoff's Voltage Law to a loop through the windings of a DC motor circuit yields

$$V = IR + k_{\mathrm{b}}\omega \tag{1}$$

where V is the effective control voltage input to the motor terminals, I is the current, R is the armature resistance, k_{b} is the back Electro-Motive Force (EMF) constant, and ω is the angular frequency of the motor shaft. There is typically an inductance that acts in opposition to change in current through the motor, but so far our analysis has shown it to have negligible effect at the time scale of interest.[7]

Multiplying (1) by the current, produces an expression for the net electrical power delivered to the motor circuit, i.e.,

$$P = IV = I^2R + k_{\mathrm{b}}\omega I.$$

The I^2R term is the electrical power loss, and the $k_{\mathrm{b}}\omega I$ term is the power available for mechanical work. If

[a] FlightCODE is an integrated collection of software tools previously referred to as SIMPLI-FLYD (Simplified Flight Dynamics for Conceptual Design).[4]

the motor supplies torque τ_{motor}, then the mechanical work done by the motor per unit time is

$$P_{\text{motor}} = \omega \tau_{\text{motor}},$$

and it follows that $\omega \tau_{\text{motor}} = k_{\text{b}} \omega I$. Likewise, the mechanical work on the motor shaft per unit time is

$$P_{\text{rotor}} = \Omega \tau_{\text{rotor}},$$

where Ω is the angular rate of the rotors, and τ_{rotor} is the torque supplied to the rotor shaft. Allowing for loss in the motor-to-rotor transmission we set $P_{\text{rotor}} = \eta P_{\text{motor}}$, where $\eta \in [0, 1]$ is the transmission efficiency. It follows that

$$\tau_{\text{rotor}} = \eta \frac{\omega}{\Omega} \tau_{\text{motor}} = \eta r k_{\text{b}} I, \tag{2}$$

where $r = \omega / \Omega$ is the gear ratio. Next we substitute (1) into this expression and obtain

$$\tau_{\text{rotor}} = \eta r k_{\text{b}} \frac{V - k_{\text{b}} \omega}{R} = \frac{\eta r k_{\text{b}}}{R} V - \frac{\eta (r k_{\text{b}})^2}{R} \Omega. \tag{3}$$

Finally, to simplify notation for later define

$$k_V = \frac{\eta r k_{\text{b}}}{R}, \quad k_\Omega = \frac{\eta (r k_{\text{b}})^2}{R}, \quad \tau = \tau_{\text{rotor}}$$

so that the rotor torque is expressed as

$$\tau = k_V V - k_\Omega \Omega. \tag{4}$$

Furthermore, the equivalent motor voltage needed to hold the vehicle in a trim condition is

$$V_{\text{t}} = \tau_{\text{t}} \frac{R}{\eta r k_{\text{b}}} + k_{\text{b}} r \Omega_{\text{t}},$$

where τ_{t} and Ω_{t} are, respectively, the torque and rotor rotational rate required for trim.

Units

Because the above electrical quantities are in Volts, Amps, and Watts, the units for the electrical constants are all SI. The SI unit of torque is the $[\text{N} \cdot \text{m}]$, and the corresponding english unit is the $[\text{ft} \cdot \text{lbf}]$. Each $[\text{N} \cdot \text{m}]$ is 0.7375621493 $[\text{ft} \cdot \text{lbf}]$. For calculations in $[\text{ft} \cdot \text{lbf}]$ this conversion factor should be multiplied into k_V and k_Ω above. To keep the notation simple here though, the use of a consistent set of units commensurate with all physics model equations (i.e., either appropriate SI or English units) is assumed.

Voltage Constraints

There are three primary sets of input limits that can be enforced on the system. The first is simply the minimum and maximum voltage input permitted by the system, denoted by $V_{\text{min}}^{\text{in}}$ and $V_{\text{max}}^{\text{in}}$. The second set of voltage limits prevents exceeding the maximum burst current I_{max} of the power supply. From (1), these are

$$V_{\text{min}}^{I} = k_{\text{b}} r \Omega - R I_{\text{max}}$$
$$V_{\text{max}}^{I} = k_{\text{b}} r \Omega + R I_{\text{max}}.$$

The third set of voltage limits ensure that the mechanical torque driven by the motor stays below the rotor shaft limit. From (3), these are

$$V_{\text{min}}^{\tau} = k_{\text{b}} r \Omega - \frac{R \tau_{\text{max}}}{\eta r k_{\text{b}}}$$
$$V_{\text{max}}^{\tau} = k_{\text{b}} r \Omega + \frac{R \tau_{\text{max}}}{\eta r k_{\text{b}}},$$

where τ_{max} is the maximum mechanical torque permitted on the rotor shaft. Notice that these limits change in time with the rotor-speed Ω, which is assumed positive in these equations. The most restrictive limits for the overall motor control voltage inputs are then

$$V_{\text{min}} = \max(V_{\text{min}}^{\text{in}}, V_{\text{min}}^{\text{I}}, V_{\text{min}}^{\tau})$$
$$V_{\text{max}} = \min(V_{\text{max}}^{\text{in}}, V_{\text{max}}^{\text{I}}, V_{\text{max}}^{\tau}).$$

Motor Parameters

When a particular motor specification is not available, the motor parameters can be chosen to meet other design criteria as follows. The trim analysis of the bare-airframe model determines the mechanical torque τ_h and rotor-speed Ω_h required for hover. With knowledge of an optimal operating point for the motor rate ω_h, set $r = \omega_\mathrm{h}/\Omega_\mathrm{h}$. From (1), the power used by the motor at hover is

$$V_\mathrm{h} I_\mathrm{h} = I_\mathrm{h}^2 R + k_\mathrm{b} r \Omega_\mathrm{h} I_\mathrm{h}.$$

With the electro-mechanical power system operating at a given efficiency η_e, it follows that

$$k_\mathrm{b} r \Omega_\mathrm{h} I_\mathrm{h} = \eta_\mathrm{e} V_\mathrm{h} I_\mathrm{h}, \tag{5}$$

$$I_\mathrm{h}^2 R = (1 - \eta_\mathrm{e}) V_\mathrm{h} I_\mathrm{h}, \tag{6}$$

for the motor's mechanical power and the electrical power loss, respectively. From (5) the back EMF constant is

$$k_\mathrm{b} = \frac{\eta_\mathrm{e} V_\mathrm{h}}{r \Omega_\mathrm{h}}.$$

With τ_h and (2) the hover current is found

$$I_\mathrm{h} = \frac{\tau_\mathrm{h}}{\eta r k_\mathrm{b}},$$

where, as before, η is the mechanical transmission efficiency of the gear box. Finally, with (6) we get the armature resistance

$$R = \frac{(1 - \eta_\mathrm{e}) V_\mathrm{h}}{I_\mathrm{h}}.$$

Thus, the motor parameters k_b, and R, are determined from the design driven values for τ_h, Ω_h, V_h, η, η_e, and knowledge of the desired motor operating frequency to get r.

III. Full Flight Envelope Model Buildup

The FlightCODE tool (or FLIGHTLAB) can be used to get state-space models for the bare-airframe dynamics in the form

$$\dot{x} = A(\nu)[x - x_\mathrm{t}(\nu)] + B(\nu)[u - u_\mathrm{t}(\nu)], \tag{7}$$

where A, B, x_t, and u_t are functions of a model parameter vector ν. For the examples considered later in this paper, ν is the aircraft x-body-axis airspeed, but in general ν may include other parameters such as altitude, rotor tilt, flap settings, rotor-speed, etc. While FlightCODE has many configuration options, it is used here for moderately sophisticated air taxi models with the following state and input vectors:[b]

$$x = \begin{bmatrix} \beta_{1k} & \text{lateral flapping angle for rotor } k \\ \beta_{2k} & \text{longitudinal flapping angle for rotor } k \\ \Omega_k & \text{angular rate of rotor } k \\ \phi & \text{roll angle} \\ \theta & \text{pitch angle} \\ u & x\text{-body-axis velocity} \\ v & y\text{-body-axis velocity} \\ w & z\text{-body-axis velocity} \\ p & \text{roll rate around } x\text{-body-axis} \\ q & \text{pitch rate around } y\text{-body-axis} \\ r & \text{yaw rate around } z\text{-body-axis} \end{bmatrix} \qquad u = \begin{bmatrix} \delta^\mathrm{col} & \text{collective input} \\ \delta^\mathrm{lat} & \text{lateral stick input} \\ \delta^\mathrm{lon} & \text{longitudinal stick input} \\ \delta^\mathrm{ped} & \text{pedal input} \\ \tau_k & \text{torque input to rotor } k \end{bmatrix}.$$

The use of subscript k, means one such term is included in the vector for each rotor in the model, $k = 1, \ldots N$ — so for a quad-rotor model $x \in \mathbb{R}^{20}$ and $u \in \mathbb{R}^8$. The use of consistent units is assumed, with the typical

[b] The general approach, however, is not limited to this case.

right-handed coordinate frame for the aircraft body axes.[c] Also, note that 'u' is used for both the input vector, and the x-body-axis velocity ('x' is also similarly overloaded), but the intended meaning is easily inferred from context. The functions $A(\nu)$, $B(\nu)$, $x_{\mathrm{t}}(\nu)$, and $u_{\mathrm{t}}(\nu)$ are often determined by linear interpolation across look-up tables. Such models are known as quasi-Linear Parameter Varying (qLPV) and are commonly used to stitch full flight envelope models together from a series of linear perturbation models.[12]

Rotor-speed and Torque Sign Convention

In the development to follow, it is assumed that the sign of the rotor rate Ω indicates direction of rotation. It follows that for a quad-rotor vehicle trimmed in a hover condition, two of the rotors are rotating in the positive direction, with the application of positive torque, and the other two rotors are rotating in the negative direction with the application of negative torque. In this case, the torque control derivatives are all-positive. This is not necessarily the default sign convention used by the modeling tool that outputs the bare-airframe model (7). For example, FlightCODE uses an all-positive rotor rate (and torque) sign convention, where the actual direction of the rotors is implicit, and the torque input derivatives have mixed positive and negative signs to establish the required balancing of torque in the dynamics.

A. Incorporating the Motor into the Bare-Airframe Model

The next step is to convert the mechanical torque input of the bare-airframe model into an equivalent voltage control input. Using the torque relation (4) we set

$$\tau_k = k_V V_k - k_\Omega \Omega_k,$$

for each rotor, assuming the motor constants are the same for all rotors. Substituting this expression into u produces

$$u = \begin{bmatrix} \delta^{\mathrm{col}} \\ \delta^{\mathrm{lat}} \\ \delta^{\mathrm{lon}} \\ \delta^{\mathrm{ped}} \\ k_V V_k \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_\Omega \Omega_k \end{bmatrix}.$$

Further substituting this, and the corresponding relation for u_{t}, into our state-space model

$$\dot{x} = A(x - x_{\mathrm{t}}) + B \left(\begin{bmatrix} \delta^{\mathrm{col}} \\ \delta^{\mathrm{lat}} \\ \delta^{\mathrm{lon}} \\ \delta^{\mathrm{ped}} \\ k_V V_k \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_\Omega \Omega_k \end{bmatrix} - \begin{bmatrix} \delta_{\mathrm{t}}^{\mathrm{col}} \\ \delta_{\mathrm{t}}^{\mathrm{lat}} \\ \delta_{\mathrm{t}}^{\mathrm{lon}} \\ \delta_{\mathrm{t}}^{\mathrm{ped}} \\ k_V V_{k\mathrm{t}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_\Omega \Omega_{k\mathrm{t}} \end{bmatrix} \right).$$

Expanding this result then produces

$$\dot{x} = A(x - x_{\mathrm{t}}) + B \left(\begin{bmatrix} \delta^{\mathrm{col}} \\ \delta^{\mathrm{lat}} \\ \delta^{\mathrm{lon}} \\ \delta^{\mathrm{ped}} \\ k_V V_k \end{bmatrix} - \begin{bmatrix} \delta_{\mathrm{t}}^{\mathrm{col}} \\ \delta_{\mathrm{t}}^{\mathrm{lat}} \\ \delta_{\mathrm{t}}^{\mathrm{lon}} \\ \delta_{\mathrm{t}}^{\mathrm{ped}} \\ k_V V_{k\mathrm{t}} \end{bmatrix} \right) - B \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_\Omega \Omega_k \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_\Omega \Omega_{k\mathrm{t}} \end{bmatrix} \right).$$

Next, the k_V motor constant is absorbed into B by multiplying it into the column of B that corresponds to τ_k. Referring to this modified B as $\bar{B}$, the model becomes

$$\dot{x} = A(x - x_{\mathrm{t}}) + \bar{B} \left(\begin{bmatrix} \delta^{\mathrm{col}} \\ \delta^{\mathrm{lat}} \\ \delta^{\mathrm{lon}} \\ \delta^{\mathrm{ped}} \\ V_k \end{bmatrix} - \begin{bmatrix} \delta_{\mathrm{t}}^{\mathrm{col}} \\ \delta_{\mathrm{t}}^{\mathrm{lat}} \\ \delta_{\mathrm{t}}^{\mathrm{lon}} \\ \delta_{\mathrm{t}}^{\mathrm{ped}} \\ V_{k\mathrm{t}} \end{bmatrix} \right) - B \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_\Omega(\Omega_k - \Omega_{k\mathrm{t}}) \end{bmatrix} \right).$$

[c] The positive x-axis points through the nose of the vehicle, the y-axis points out the right side of the vehicle (along its wing if it has one), and the z-axis points down through the floor.

Applying the same trick to the right-most term, let b_k be the column of B that corresponds to the τ_k input, so that the expression for $\dot{x}$ becomes

$$\dot{x} = A(x - x_\mathrm{t}) + \bar{B}\left(\begin{bmatrix}\delta^{\mathrm{col}}\\ \delta^{\mathrm{lat}}\\ \delta^{\mathrm{lon}}\\ \delta^{\mathrm{ped}}\\ V_k\end{bmatrix} - \begin{bmatrix}\delta_\mathrm{t}^{\mathrm{col}}\\ \delta_\mathrm{t}^{\mathrm{lat}}\\ \delta_\mathrm{t}^{\mathrm{lon}}\\ \delta_\mathrm{t}^{\mathrm{ped}}\\ V_{k\mathrm{t}}\end{bmatrix}\right) - b_k k_\Omega(\Omega_k - \Omega_{k\mathrm{t}}).$$

Now the right-most term can be absorbed into A by subtracting $b_k k_\Omega$ from the column of A that corresponds to Ω_k. Denoting this modified A matrix as $\bar{A}$ we have

$$\dot{x} = \bar{A}(x - x_\mathrm{t}) + \bar{B}\left(\begin{bmatrix}\delta^{\mathrm{col}}\\ \delta^{\mathrm{lat}}\\ \delta^{\mathrm{lon}}\\ \delta^{\mathrm{ped}}\\ V_k\end{bmatrix} - \begin{bmatrix}\delta_\mathrm{t}^{\mathrm{col}}\\ \delta_\mathrm{t}^{\mathrm{lat}}\\ \delta_\mathrm{t}^{\mathrm{lon}}\\ \delta_\mathrm{t}^{\mathrm{ped}}\\ V_{k\mathrm{t}}\end{bmatrix}\right).$$

With this procedure, the motor model is directly incorporated into the bare-airframe qLPV model by modifying the look-up tables for A and B. In addition, the tables for u_t are also modified to reflect that V_k now replaces τ_k for each rotor.

B. Accounting for the Motor Inertia

With the motor inertia included, the torque supplied by the motor is really

$$\tau_{\mathrm{motor}} = k_\mathrm{b} I - J_\mathrm{m}\dot{\omega},$$

where J_m is the motor inertia. With the power equations developed in Section II, i.e.,

$$P_{\mathrm{rotor}} = \Omega\tau_{\mathrm{rotor}} = \eta P_{\mathrm{motor}} = \eta\omega\tau_{\mathrm{motor}},$$

the torque transmitted to rotor from (3) becomes

$$\tau_{\mathrm{rotor}} = \frac{\eta r k_\mathrm{b}}{R}V - \frac{\eta(r k_\mathrm{b})^2}{R}\Omega - \eta r^2 J_\mathrm{m}\dot{\Omega}.$$

Expanding on our previous notation, let $k_{\dot{\Omega}} = \eta r^2 J_\mathrm{m}$ so that the torque imparted to the kth rotor becomes

$$\tau_k = k_V V_k - k_\Omega \Omega_k - k_{\dot{\Omega}}\dot{\Omega}_k.$$

Notice that in a trim state $\dot{\Omega}_k = 0$ for each rotor, so the extra term does not affect the trim condition.

Substituting this result into the bare-airframe input vector produces

$$u = \begin{bmatrix}\delta^{\mathrm{col}}\\ \delta^{\mathrm{lat}}\\ \delta^{\mathrm{lon}}\\ \delta^{\mathrm{ped}}\\ k_V V_k\end{bmatrix} - \begin{bmatrix}0\\ 0\\ 0\\ 0\\ k_\Omega\Omega_k\end{bmatrix} - \begin{bmatrix}0\\ 0\\ 0\\ 0\\ k_{\dot{\Omega}}\dot{\Omega}_k\end{bmatrix},$$

and after inserting this expression for u into the state equations we obtain

$$\dot{x} = A(x - x_\mathrm{t}) + B\left(\begin{bmatrix}\delta^{\mathrm{col}}\\ \delta^{\mathrm{lat}}\\ \delta^{\mathrm{lon}}\\ \delta^{\mathrm{ped}}\\ k_V V_k\end{bmatrix} - \begin{bmatrix}0\\ 0\\ 0\\ 0\\ k_\Omega\Omega_k\end{bmatrix} - \begin{bmatrix}\delta_\mathrm{t}^{\mathrm{col}}\\ \delta_\mathrm{t}^{\mathrm{lat}}\\ \delta_\mathrm{t}^{\mathrm{lon}}\\ \delta_\mathrm{t}^{\mathrm{ped}}\\ k_V V_{k\mathrm{t}}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ 0\\ k_\Omega\Omega_{k\mathrm{t}}\end{bmatrix}\right) - B\begin{bmatrix}0\\ 0\\ 0\\ 0\\ k_{\dot{\Omega}}\dot{\Omega}_k\end{bmatrix}.$$

From here, the steps from Section IIIA are repeated to absorb the terms involving k_V and k_Ω into the state and control matrices. The result is

$$\dot{x} = \bar{A}(x - x_\mathrm{t}) + \bar{B}\left(\begin{bmatrix} \delta^\mathrm{col} \\ \delta^\mathrm{lat} \\ \delta^\mathrm{lon} \\ \delta^\mathrm{ped} \\ V_k \end{bmatrix} - \begin{bmatrix} \delta_\mathrm{t}^\mathrm{col} \\ \delta_\mathrm{t}^\mathrm{lat} \\ \delta_\mathrm{t}^\mathrm{lon} \\ \delta_\mathrm{t}^\mathrm{ped} \\ V_{k\mathrm{t}} \end{bmatrix}\right) - B\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_{\dot{\Omega}}\dot{\Omega}_k \end{bmatrix}. \tag{8}$$

Now the term involving the state derivative $\dot{\Omega}$ is moved to the left-hand side, making the state equation

$$\dot{x} + B\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_{\dot{\Omega}}\dot{\Omega}_k \end{bmatrix} = \bar{A}(x - x_\mathrm{t}) + \bar{B}\left(\begin{bmatrix} \delta^\mathrm{col} \\ \delta^\mathrm{lat} \\ \delta^\mathrm{lon} \\ \delta^\mathrm{ped} \\ V_k \end{bmatrix} - \begin{bmatrix} \delta_\mathrm{t}^\mathrm{col} \\ \delta_\mathrm{t}^\mathrm{lat} \\ \delta_\mathrm{t}^\mathrm{lon} \\ \delta_\mathrm{t}^\mathrm{ped} \\ V_{k\mathrm{t}} \end{bmatrix}\right).$$

The left-hand side is then rewritten as

$$\dot{x} + B\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_{\dot{\Omega}}\dot{\Omega}_k \end{bmatrix} = \dot{x} + b_k k_{\dot{\Omega}}\dot{\Omega}_k = \dot{x} + b_k k_{\dot{\Omega}} e_k^T \dot{x} = \left(I + b_k k_{\dot{\Omega}} e_k^T\right)\dot{x},$$

where e_k is the natural basis vector corresponding to the Ω_k element of x. Doing this for each of the rotor torque inputs produces the matrix

$$M = I + \sum_{i=1}^{N_\mathrm{rotors}} k_{\dot{\Omega}} b_i e_i^T,$$

and (8) becomes

$$M\dot{x} = \bar{A}(x - x_\mathrm{t}) + \bar{B}\left(\begin{bmatrix} \delta^\mathrm{col} \\ \delta^\mathrm{lat} \\ \delta^\mathrm{lon} \\ \delta^\mathrm{ped} \\ V_k \end{bmatrix} - \begin{bmatrix} \delta_\mathrm{t}^\mathrm{col} \\ \delta_\mathrm{t}^\mathrm{lat} \\ \delta_\mathrm{t}^\mathrm{lon} \\ \delta_\mathrm{t}^\mathrm{ped} \\ V_{k\mathrm{t}} \end{bmatrix}\right).$$

Finally, if M is non-singular,

$$\dot{x} = M^{-1}\bar{A}(x - x_\mathrm{t}) + M^{-1}\bar{B}\left(\begin{bmatrix} \delta^\mathrm{col} \\ \delta^\mathrm{lat} \\ \delta^\mathrm{lon} \\ \delta^\mathrm{ped} \\ V_k \end{bmatrix} - \begin{bmatrix} \delta_\mathrm{t}^\mathrm{col} \\ \delta_\mathrm{t}^\mathrm{lat} \\ \delta_\mathrm{t}^\mathrm{lon} \\ \delta_\mathrm{t}^\mathrm{ped} \\ V_{k\mathrm{t}} \end{bmatrix}\right),$$

and the look-up tables for the bare-airframe model are updated to reflect this equation.

C. Gravity and Kinematics

The qLPV model (7) must include the gravitational acceleration in trim. This is because the net required acceleration in a trim condition is zero when $x = x_\mathrm{t}$ and $u = u_\mathrm{t}$. It follows that with the incorporation of gravity

$$\dot{x} = A(\nu)[x - x_\mathrm{t}(\nu)] + B(\nu)[u - u_\mathrm{t}(\nu)] + g(\phi, \theta) - g(\phi_\mathrm{t}(\nu), \theta_\mathrm{t}(\nu)),$$

where $g(\phi, \theta)$ is the gravitational acceleration resolved in the body-axis, using the associated roll ϕ and pitch θ Euler angles. With this equation, the net acceleration on the vehicle is zero when the state, input, and Euler angles are trimmed, i.e., equal to $x = x_\mathrm{t}$, $u = u_\mathrm{t}$, $\phi = \phi_\mathrm{t}$, and $\theta = \theta_\mathrm{t}$.

The force and moment equations for a rigid-body with constant mass are

$$\begin{aligned}\mathbf{F} &= m\dot{\mathbf{V}} + \boldsymbol{\omega} \times m\mathbf{V} \\ \mathbf{M} &= I_r\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times I_r\boldsymbol{\omega},\end{aligned}$$

where $\mathbf{F}$ is the net force, $\mathbf{M}$ is the net moment, $\mathbf{V} = [u, v, w]^T$, $\boldsymbol{\omega} = [p, q, r]^T$, I_r is the inertia matrix, and m is the net vehicle mass. With $x_\mathrm{v} = [\mathbf{V}^T, \boldsymbol{\omega}^T]^T$ we have

$$\begin{bmatrix}\mathbf{F} \\ \mathbf{M}\end{bmatrix} = \begin{bmatrix}mI & 0 \\ 0 & I_r\end{bmatrix} \dot{x}_\mathrm{v} + \begin{bmatrix}\boldsymbol{\omega} \times m\mathbf{V} \\ \boldsymbol{\omega} \times I_r\boldsymbol{\omega}\end{bmatrix},$$

here I is an appropriately sized identity matrix. It follows that

$$\dot{x}_\mathrm{v} = \begin{bmatrix}mI & 0 \\ 0 & I_r\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{F} \\ \mathbf{M}\end{bmatrix} - \begin{bmatrix}mI & 0 \\ 0 & I_r\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{\omega} \times m\mathbf{V} \\ \boldsymbol{\omega} \times I_r\boldsymbol{\omega}\end{bmatrix}.$$

This equation provides the body-axis dynamics contribution to our overall state vector x, which includes x_v. Using these equations the overall system is decomposed as

$$\dot{x} = f(x, u) + g(x) + h(x), \tag{9}$$

where,

$$f(x, u) = A(\nu)[x - x_\mathrm{t}(\nu)] + B(\nu)[u - u_\mathrm{t}(\nu)] - g(\phi_\mathrm{t}(\nu), \theta_\mathrm{t}(\nu))$$

is the gravity-adjusted airframe and propulsion model. The function $g(x)$ is the gravitational acceleration in the body-axis, which is only a function of the ϕ and θ components of the state vector x. The remaining kinematic terms are swept into $h(x)$, including the differential equations for the Euler angles ϕ, and θ.

D. Wind and Gust Models

As stated, the nonlinear model equation (9) does not yet include the effects of steady-wind or gusts. Let x_w represent the constant steady-wind state vector, which contains the appropriate body-axis velocity elements of the steady-wind vector. Since the aerodynamic accelerations are computed relative to the steady-wind, equation (9) becomes

$$\dot{x} = f(x - x_\mathrm{w}, u) + g(x) + h(x), \tag{10}$$

where x is now viewed as the inertial state of the vehicle, and $x - x_\mathrm{w}$ is viewed as the wind-relative state. In this case, the lookup parameter ν will also likely depend on the wind-relative airspeed (and hence the wind-relative state). Gust or turbulence effects are often modeled for handling qualities simulation as equivalent state or input disturbance signals that are injected into airframe model.[13,14] This paper focuses on an optimistic assessment of handling qualities where we assume the steady-wind is zero, i.e., $x_\mathrm{w} = 0$, with negligible turbulence. However, it is well recognized that turbulence can degrade the handling qualities ratings for a particular vehicle and control system.[15]

E. Linear Approximation

Model-based control system analysis and development typically requires the ability to establish an approximate linear model around an *arbitrary* operating point (x_0, u_0). The standard approach is to use a Taylor series approximation

$$\delta\dot{x} = \mathbf{J}_x\delta x + \mathbf{J}_u\delta u + \mathbf{f}(x_0, u_0), \tag{11}$$

where $\delta x = x - x_0$, $\delta u = u - u_0$, $\mathbf{f}(x_0, u_0)$ is the total model (9). The matrix $\mathbf{J}_x \in \mathbb{R}^{n\times n}$ is the total Jacobian matrix of $\mathbf{f}$ with respect to state x, i.e., $\mathbf{J}_x = J_x^f + J_x^g + J_x^h$, where J_x^f, J_x^g, and J_x^h are the respective Jacobian

matrices of f, g, and h with respect to x. The matrix $\mathbf{J}_u = J_u^f$, because the Jacobian matrix for g and h with respect to u is zero.

Though tedious, it is straight-forward to compute the required component Jacobian matrices. Care, however, is needed for J_x^f in particular, because $f(x)$ depends on $A(\nu)$, $B(\nu)$, $x_\mathrm{t}(\nu)$, $u_\mathrm{t}(\nu)$, and $g(x_\mathrm{t}(\nu))$, each of which depends on $\nu(x)$. Thus, an accurate calculation for J_x^f requires a rather involved application of the chain-rule. For the case with $\nu(x) : \mathbb{R}^n \to \mathbb{R}$, the result is

$$J_x^f = A(\nu) + \left[\frac{\partial A}{\partial \nu}(x - x_\mathrm{t}(\nu)) - (A(\nu) + J_x^g(x_\mathrm{t}))\frac{\partial x_\mathrm{t}}{\partial \nu} + \frac{\partial B}{\partial \nu}(u - u_\mathrm{t}(\nu)) - B(\nu)\frac{\partial u_\mathrm{t}}{\partial \nu}\right] \nabla \nu^T.$$

The partial derivative terms with respect to ν, are simply the slopes of the lines connecting the elements in the look-up tables for $A(\nu)$, $B(\nu)$, $x_\mathrm{t}(\nu)$, and $u_\mathrm{t}(\nu)$.

IV. Controller

In order to study handling qualities and flight safety, the integration of a specific flight control system is required. FlightCODE provides an automated means of accomplishing this through the use of CONDUIT, as noted in the introduction. However, with the approach presented here one can look at alternative controls technologies that utilize the flight dynamics model (9), such as Model Predictive Control (MPC) or Nonlinear Dynamic Inversion (NDI).

Here, an MPC design under investigation in ongoing work is summarized. A reduced order model is assumed for the controller design. In particular, the unobserved flapping states of the full nonlinear model state vector are eliminated. For notational simplicity, this model is also referred to as $\mathbf{f}(x, u)$, where x now represents an observable state vector.

A. Discrete MPC

Our approach is focused on MPC with quadratic cost, linear dynamics, and box constraints for the state and control inputs of a *discrete* control system. This problem class is convex, and highly-efficient solver algorithms are available that find the global optimal solution to the associated quadratic optimal control problem:

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=0}^{T-1} \bar{z}_\mathrm{d}(k)^T Q_\mathrm{d} \bar{z}_\mathrm{d}(k) + r_\mathrm{d}(k)^T R_\mathrm{d} r_\mathrm{d}(k) + \bar{z}_\mathrm{d}(T)^T Q_\mathrm{f} \bar{z}_\mathrm{d}(T) \\
\text{subject to} \quad & \bar{z}_\mathrm{d}(k) = z_\mathrm{d}(k) - z_\mathrm{cmd} \ \text{ for } k = 0, ..., T, \\
& z_\mathrm{d}(k+1) = A_\mathrm{d} z_\mathrm{d}(k) + B_\mathrm{d} r_\mathrm{d}(k) + w_\mathrm{d} \ \text{ for } k = 0, ..., T-1, \\
& z_\mathrm{min} \le z_\mathrm{d}(k) \le z_\mathrm{max} \ \text{ for } k = 1, ..., T, \\
& r_\mathrm{min} \le r_\mathrm{d}(k) \le r_\mathrm{max} \ \text{ for } k = 0, ..., T-1.
\end{aligned} \tag{12}$$

The variables in this problem are states $z_\mathrm{d}(1), z_\mathrm{d}(2), \ldots, z_\mathrm{d}(T)$ and inputs $r_\mathrm{d}(0), \ldots, r_\mathrm{d}(T-1)$. The given problem data is $z_\mathrm{cmd}, z_\mathrm{d}(0), A_\mathrm{d}, B_\mathrm{d}, w_\mathrm{d}, z_\mathrm{min}, z_\mathrm{max}, r_\mathrm{min}, r_\mathrm{max}$ and T. The optimal solution for $r_\mathrm{d}(0)$ is taken as the current input, and the process repeats in the next cycle.

In our implementation, the Fast MPC algorithm developed in [16] is used with a few modifications. First, diagonal input and state cost matrices R_d, Q_d, and Q_f are assumed. This was done to get even better computational performance. The external disturbance w_d is also included in the state equality constraints. This generalizes the approach in [16], for use with command tracking that permits non-linear dynamics model linearization around non-equilibrium points where the Taylor series approximation will generally carry a fixed bias term at the current state, i.e., $\mathbf{f}(x_0, u_0)$ in (11). However, care is needed here, as a fixed non-zero w_d can destabilize the system when its effect is beyond the limitations of the inputs. A more general discussion of stability for MPC is available in [17].

B. Linearization and Rate Augmentation

It remains to map our continuous time nonlinear model $\mathbf{f}(x, u)$ to an acceptable discrete time approximation for use in the above quadratic MPC problem formulation. This is accomplished with a three step process: linearization, rate control augmentation, and discretization.

At the current state and input (x_0, u_0), the Taylor approximation (11) is applied, and rate control is augmented to the system producing,

$$\begin{bmatrix} \delta\dot{x} \\ \delta\dot{u} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_x & \mathbf{J}_u \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} r + \begin{bmatrix} I \\ 0 \end{bmatrix} \mathbf{f}(x_0, u_0). \tag{13}$$

The system input now becomes r, which just drives $\delta\dot{u}$. This processing step allows for the assumption of constant rate inputs between time samples in the continuous system, and has several other benefits. First, the discrete rate control system drives the actuator rates in the continuous system, which is more realistic than trying to drive the actuator positions directly. This is because the discrete controller assumes that the continuous input being driven can change instantaneously, an assumption which is less of an approximation for rate control than it is for position control. Second, we can simulate the continuous system response with the discrete controller using an ODE solver that requires continuous derivatives. Third, the final target input positions are often a-priori unknown, while the final input rates are known as they are usually controlled to zero. Finally, one can use the MPC problem box constraints to bound both the inputs (which are part of the state vector z) and the input rates.

To simplify notation (13) is expressed as

$$\dot{z} = Az + Br + w \tag{14}$$

where,

$$z = \begin{bmatrix} \delta x \\ \delta u \end{bmatrix},\ A = \begin{bmatrix} \mathbf{J}_x & \mathbf{J}_u \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ I \end{bmatrix},\ w = \mathbf{f}(x_0, u_0).$$

This system is the rate-augmented local linear approximation to the nonlinear model $\mathbf{f}(x, u)$. We also introduce the continuous-time Linear Quadratic Regulator problem for the augmented system evaluated at an equilibrium point (so $w = 0$),

$$\begin{aligned} \text{minimize} \quad & \int_0^\infty z(t)^T Q z(t) + r(t)^T R r(t) dt \\ \text{subject to} \quad & \dot{z} = Az + Br, \end{aligned} \tag{15}$$

with positive semi-definite state cost matrix $Q \in \mathbb{R}^{(n+m)\times(n+m)}$, and positive definite input cost matrix $R \in \mathbb{R}^{m\times m}$. This well known optimal control problem has a unique constant gain feedback solution if (A, B) is controllable (so that the system is always stabilizable), and (Q, A) observable (so that no nonzero state can go unpenalized). Furthermore, when R is diagonal it known that the system has infinite upper gain margin, gain reduction margin equal to 0.5, and at least 60 degree phase margin in each input control [18, Ch. 9.2.2].

C. Discretization

We are now in a position to discretize the rate augmented system (14) in a manner such that the solution to (12) without inequality constraints (at an equilibrium point) converges to the solution to (15) when the discretization step size h goes to zero, and the MPC controller time horizon T goes to infinity. This is accomplished by assuming a constant rate input over discretization step interval h, and integrating (14) over this interval with initial condition $z(kh)$. The integration for the linear system is accomplished by finding matrices A_d and F_d, such that

$$\exp\left(h \begin{bmatrix} A & I \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} A_\mathrm{d} & F_\mathrm{d} \\ 0 & I \end{bmatrix}.$$

Then, after setting $B_\mathrm{d} = F_\mathrm{d} B$, and $w_\mathrm{d} = F_\mathrm{d} w$, the discrete system

$$z_\mathrm{d}(k+1) = A_\mathrm{d} z_\mathrm{d}(k) + B_\mathrm{d} r(k) + w_\mathrm{d},$$

exactly evaluates the trajectory that satisfies (14) with initial condition $z(0)$, under the assumption of a constant rate input over each sample interval.

Furthermore, use of the trapezoidal rule to estimate the integral of the cost function in (15) over the sample time h with constant rate inputs produces

$$R_{\mathrm{d}} = \frac{h}{2}\left(2R + B_{\mathrm{d}}^T Q B_{\mathrm{d}}\right), \quad Q_{\mathrm{d}} = \frac{h}{2}\left(Q + A_{\mathrm{d}}^T Q A_{\mathrm{d}}\right).$$

This approximation ignores a cross term between $z_{\mathrm{d}}(k)$ and $r_{\mathrm{d}}(k)$ that appears in the trapezoidal rule. It also captures an effect of the current state and input across the integration interval h, which contributes to both Q_{d} and R_{d}. Finally, since we restricted our MPC formulation to diagonal Q_{d} and R_{d}, we also drop the off-diagonal components that may appear in the calculation, i.e., by taking $Q_{\mathrm{d}} = \mathbf{diag}(Q_{\mathrm{d}})$, and $R_{\mathrm{d}} = \mathbf{diag}(R_{\mathrm{d}})$.

D. Inner-Loop Control Algorithm

The algorithm for our discrete MPC controller runs at the simulation frame rate. In each frame, the linear model is evaluated and discretized at the current state. Given a control command and constraints relative to the current state, the MPC problem (12) is solved for the entire input sequence $r_{\mathrm{d}}(k)$, $k = 0, \ldots, T-1$. The value of $r_{\mathrm{d}}(0)$ drives the rate input $r(t)$ to the continuous qLPV model at the current simulation time, and the process repeats again on the next simulation frame. In effect, the MPC algorithm "thinks ahead of the aircraft" by planning a trajectory to hT [s] in the future with the linear model approximation at the current time, and applies the optimal input that it finds for the current time. The process repeats each simulator frame, and the linear approximation moves with the nonlinear dynamics model.

E. Simulation

With the flight dynamics model $\mathbf{f}(x, u)$ and our control law for the input rates $\dot{u} = \kappa(x_{\mathrm{cmd}})$ computed with MPC, the nonlinear system response is simulated by integrating

$$\dot{y} = \begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(x, u) \\ \kappa(x_{\mathrm{cmd}}(t)) \end{bmatrix},$$

from initial condition $y(0)$, where $x_{\mathrm{cmd}}(t)$ is a given state command signal.

F. Outer-Loop Considerations

Piloted handling qualities simulation requires the ability to implement different control response types, such as Attitude Command Attitude Hold (ACAH), Rate Command Attitude Hold (RCAH), Translational Rate Command (TRC), Rate Command Height Hold (RCHH), Rate Command Direction Hold (RCDH), and Turn Coordination (TC) in forward flight. For each of these, the ability to fine tune the controller response is also desired. In principle, this might all be accomplished within the MPC controller by adjusting the cost matrices Q and R. However, in practice we found it more convenient to use the MPC system to create a fast inner-loop ACAH controller, and combine it with a simple adjustable command filtering system to implement and tune the other control response types. Figure 1 shows a representative setup for an RCAH system, where ω_{cmd} [rad/sec] is a response bandwidth tuning parameter.

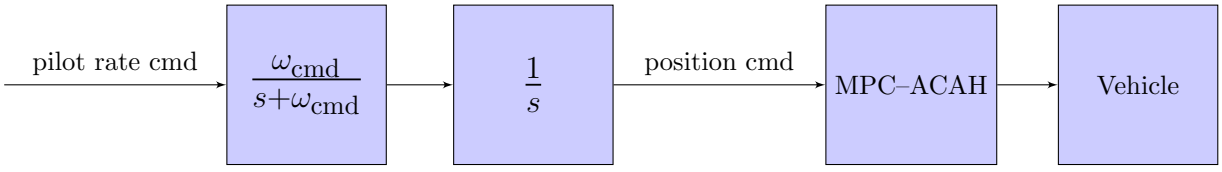

Figure 1: Basic RCAH setup with command filter.

V. Demonstration

To demonstrate some of the analysis one can perform with the proposed system, we explore the case of the NASA developed *conceptual* design shown in Figure 2. This is a 6-passenger, 50-nm range, 1200-lb

payload, quad-rotor vehicle with electric propulsion.[1] FlightCODE was used to generate the bare-airframe model in the qLPV form (see Eq. (7)), for collective controlled (D1) and rotor-speed controlled (D2) design variants. Details of the aero-mechanical modeling methodology are discussed in [7]. Using our motor modeling approach, the rotor torque inputs were powered by direct-drive DC motors with the parameters listed in Table 1.

Figure 2: Quad-rotor electric propulsion reference design.

	Symbol	D1	D2	units
Back EMF Constant	k_{b}	0.4048	0.3572	V-s
Armeture Resistance	R	0.0483	0.0484	Ohm
Motor Inertia	J_{m}	0.1266	0.1063	slug-ft
Gear Ratio	r	19.97	18.76	
Transmission Efficiency	η	1.00	1.00	
Electrical Efficiency	η_{e}	0.966	0.955	

Table 1: Motor Parameters for collective controlled (D1) and rotor-speed controlled (D2) design variants.

The Model Predictive Controller discussed earlier was then incorporated and setup for an Attitude-Command-Attitude-Hold response type, for each of the two vehicle configurations. For the first configuration (D1), collective blade-pitch is used to control maneuvering while the voltage inputs are used to maintain constant rotor-rate. For the second configuration (D2), the collective controls are held fixed, and the voltage inputs are used to manipulate rotor-speed (or RPM) for maneuvering capability. In both cases, the controllers have perfect knowledge of a reduced order qLPV model (9), with the eight blade flapping angle states removed.[d] Also, for the initial analysis we present here, the state and control box constraints in the MPC problem formulation (12) were effectively omitted, by setting $x_{\min} = u_{\min} = -10^8$, and $x_{\max} = u_{\max} = 10^8$.

A. MPC Convergence to LQR

With the full envelope qLPV model setup, the model linearization is first checked for controllability across a hi-resolution sweep of the lookup parameter (ν). With the controllability property, one can drive the linearized model to any level of performance *if* there are no bounds on the inputs and states. While this is not a practical assumption, the initial interest is in quantifying the size of the inputs (voltages, currents, and torques) required to achieve particular ADS-33E handling qualities metrics.

Before moving onto the ADS-33E analysis however, the MPC computation is validated by checking convergence to the LQR response under the appropriate conditions. Figures 3 and 4 show the roll step and the command-to-attitude frequency response for the collective controlled model (D1) governed by LQR and MPC, with a particular choice of cost matrices Q and R. The figures show that the discrete MPC response converges to the continuous system LQR response as expected. Figures 5 and 6 show the same analysis for the rotor-speed controlled model (D2). Notice that the primary effect of the MPC discretization step is

[d] In MATLAB, by applying modred() to the constituent A and B matrices used to define the qLPV model.

to introduce delay in the system. For the remaining presentation, MPC is used with a three second time horizon, configured with $h = 0.05$ [s] and $T = 60$.

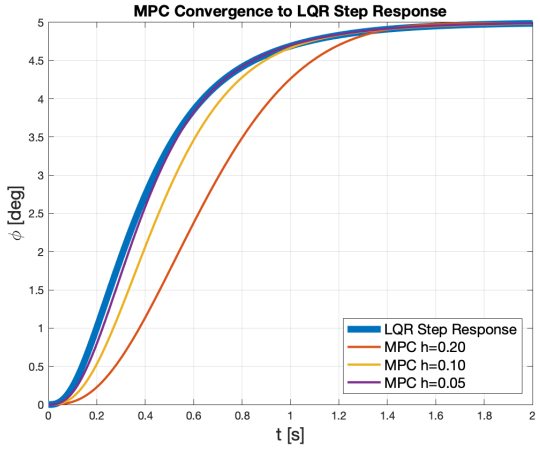

Figure 3: Collective controlled, roll step-response.

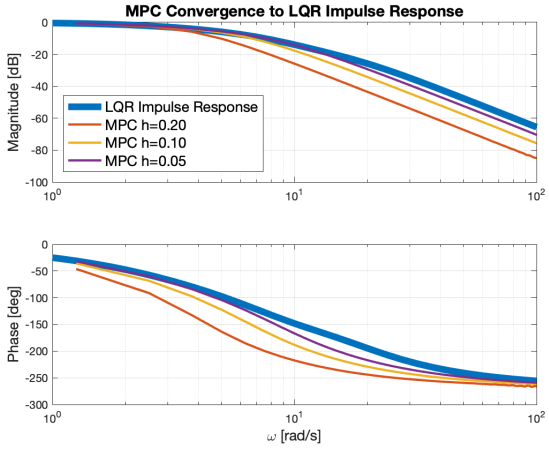

Figure 4: Collective controlled, roll frequency response.

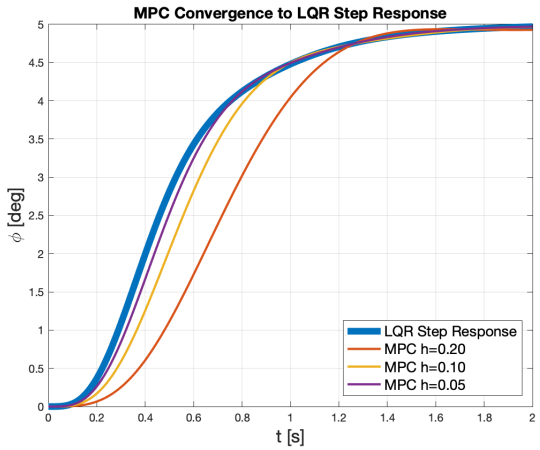

Figure 5: RPM controlled, roll step-response.

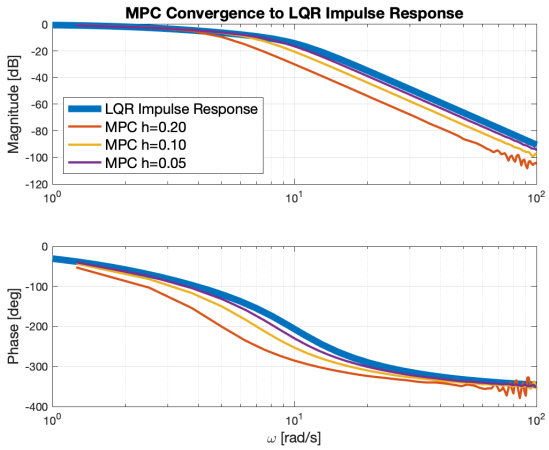

Figure 6: RPM controlled, roll frequency response.

B. A Performance Trade-Space

For the blade-pitch (collective) controlled design (D1), Figure 7 shows the ADS-33E moderate amplitude metric for 10 and 20 degree pitch forward commands, with three different controller aggressiveness settings. The aggressiveness setting is color-coded in the same way throughout this section. The most aggressive controller is red, the moderate controller is blue, and the least aggressive controller is magenta. The aggressiveness settings were obtained by adjusting the cost matrix (Q) terms associated with θ and q. In general, by increasing the θ-cost we improve tracking performance, and by increasing the q-cost we reduce aggressiveness (and the same for ϕ and p). Figures 8 and 9 show the motor input voltage and current that is needed to regulate the motor rates in color correspondence with the respective performance metrics in Figure 7. The voltage and current inputs are shown for the first and fourth motors as indicated in the plots. The inputs for the remaining two motors are symmetric with these. The torque required, which is directly proportional to the current, is also shown on the right y-axes of Figure 9. The collective controller was easily configured to exceed Level 1 handling qualities, and required substantially less motor current and torque than the rotor-speed controlled design, which is shown next.

Figures 10–12 show the analogous case for the rotor-speed controlled design (D2). For rotor-speed control,

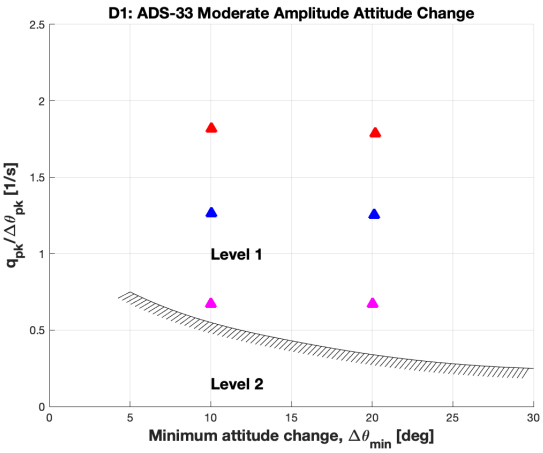

Figure 7: Collective controlled, ADS-33E moderate amplitude pitch response metrics.

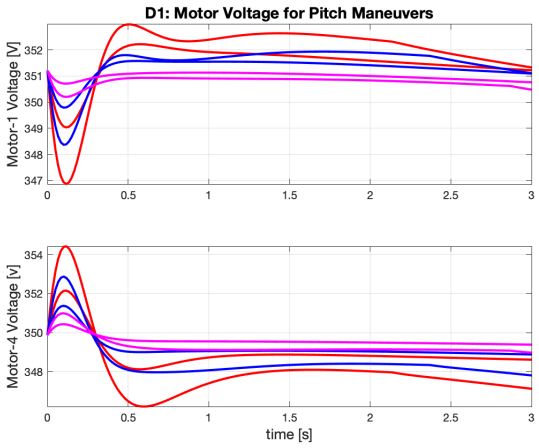

Figure 8: Collective controlled, Motor voltage required to regulate rotor speeds.

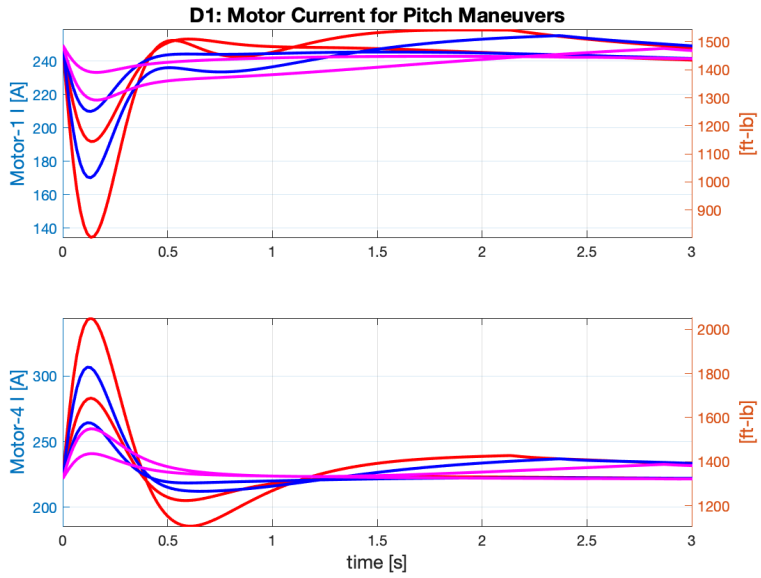

Figure 9: Collective Controlled, Motor current required to regulate rotor speeds.

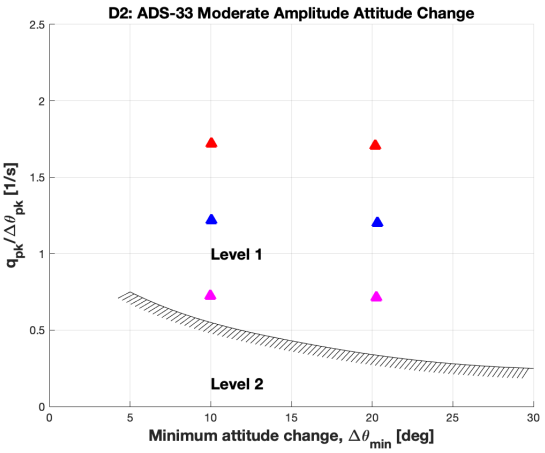

Figure 10: RPM controlled, ADS-33E moderate amplitude pitch response metrics.

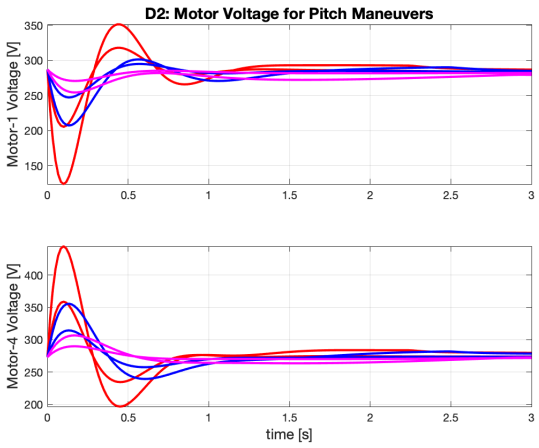

Figure 11: RPM controlled, Motor voltage required for pitch response metrics.

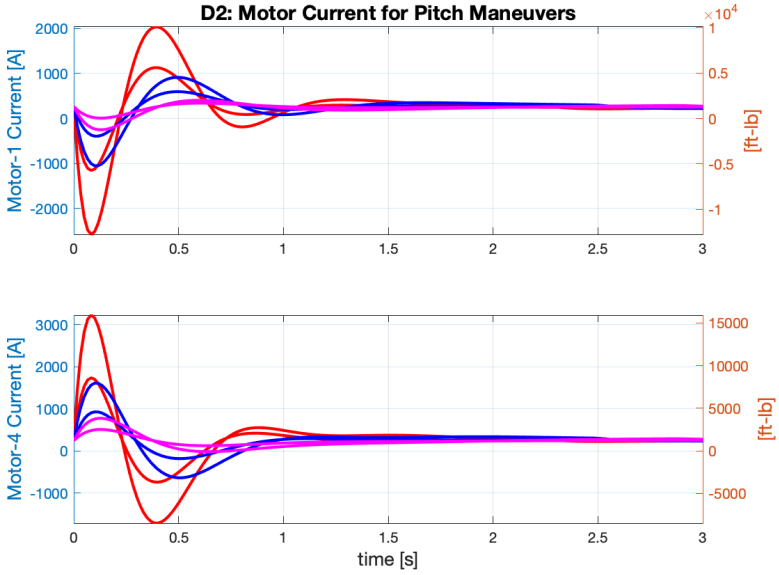

Figure 12: RPM controlled, Motor current required for pitch response metrics.

the collective inputs are held fixed and the motor input voltages are manipulated to fully control the vehicle. Notice the dramatic increase in the required motor current (and associated rotor torques).

Figure 13 shows the ADS-33E metrics, for the RPM controlled design, with roll commands of 10, 20, and 40 degrees. For the same commanded input, the roll and pitch ADS-33E metrics are roughly the same due to the symmetry of the vehicle. However, the ADS-33E requirement for Level 1 handling quality is more stringent for roll, and the least aggressive controller does not satisfy the Level 1 rating. Figure 13 also shows that the MPC controller, with fixed state and input cost matrices (Q and R), consistently maintains $p_{\text{pk}}/\Delta\phi_{\text{pk}}$ for each of the roll commands. However, Figures 14 and 15 show that a significant cost is paid in the control voltage and motor current, as roughly double the peak voltage and current is required to hit the same metric with 40 degree roll command.

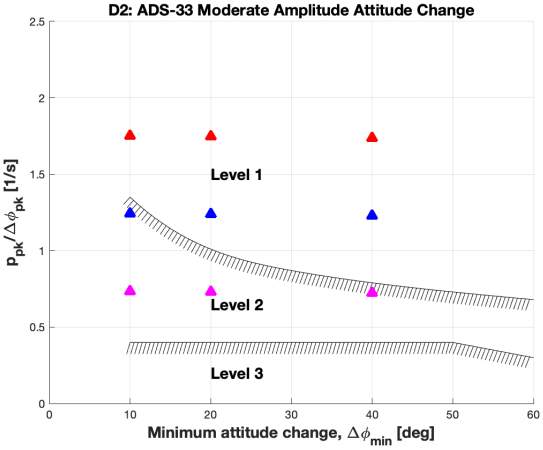

Figure 13: RPM controlled, ADS-33E moderate amplitude roll response metrics.

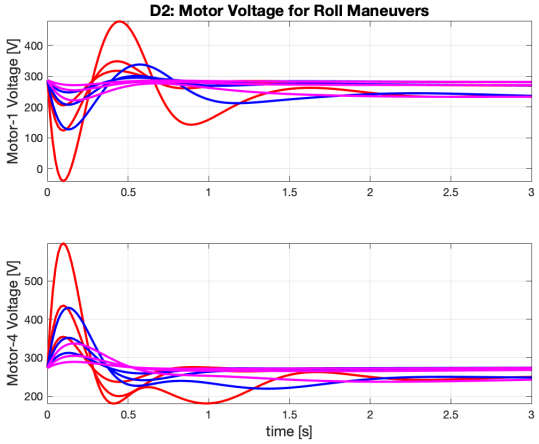

Figure 14: RPM controlled, Motor voltage required for roll response metrics.

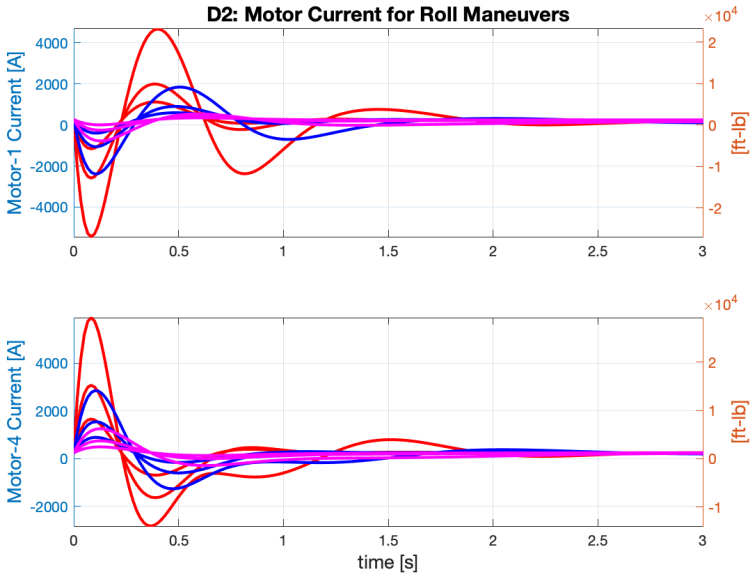

Figure 15: RPM controlled, Motor current required for roll response metrics.

Figures 16 and 17 show the ADS-33E small amplitude metrics (phase-delay vs. bandwidth) associated with each of the controller aggressiveness levels for the RPM controller, using the same color correspondence. Cross referencing with Figures 12 and 15, we observe that increasing bandwidth and decreasing the phase delay come at a significant cost in terms of the peak current and torque required.

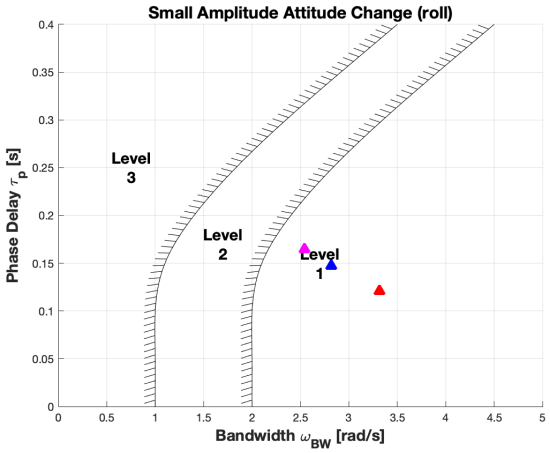

Figure 16: RPM controlled, ADS-33E small amplitude roll response metrics.

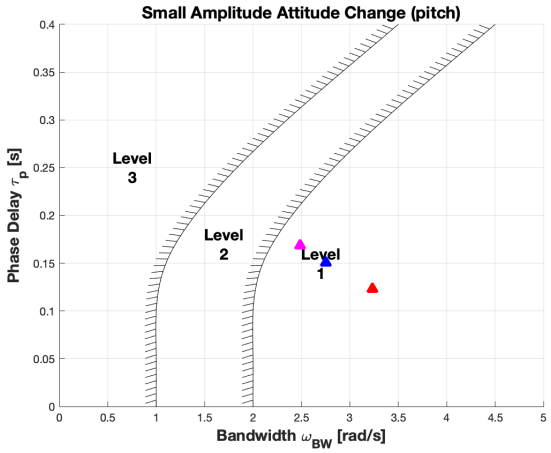

Figure 17: RPM controlled, ADS-33E small amplitude pitch response metrics.

Figure 18 shows the rate-command heave step response for each of our controller aggressiveness settings. Figure 19 shows motor voltage and current (per motor) required in color correspondence with Figure 18. Notice the time delay before the step response starts to ramp up. This is caused by the additional time required for the rotor speeds to increase before the rotors can produce the lift needed to climb. For reference, the collective controlled heave response (which in our configuration is the same for all three aggressiveness levels) is also shown as the dotted red line in Figure 18. The higher order heave response of the rotor-speed controlled vehicle is not well suited to the ADS-33E performance metric, which is based on a least-squares fit to a time-delayed first order response. The performance analysis defined in NASA TM110399 for a shipboard landing task is preferred, and is shown for our rotor-speed controllers in Figure 20.[19]

Overall, handling-qualities are synonymous with control safety because vehicles with good handling qualities can avoid sudden obstacles and safely respond to wind gusts or other unexpected events. This is needed for both human and autonomous pilots. Even though the ADS-33E standards may not be the most appropriate metrics for air taxi operations in urban environments, the above analysis shows that a significant tradeoff in the peak electrical current (or torque) is required for rotor-speed based control, independent of the adopted standard. While designs with rotor-speed control will need to handle large peak-demands for worst-case rapid maneuvering, the analysis also shows potential for significant power savings by maneuvering slowly during nominal operations.

VI. Conclusion

An approach for developing electric air taxi models suitable for analysis of handing qualities and controls safety was proposed and demonstrated. The method leverages the use of established tools for the conceptual design of Vertical Takeoff and Landing aircraft, which includes the ability to output linear perturbation models for the bare-airframe flight dynamics at various operating conditions. We then showed how to augment a distributed electric propulsion system model in state-space form, and extended it for use across the full flight envelope using a standard quasi-Linear Parameter Varying model stitching technique. Using a Model Predictive Controller, an analysis of the trade-off between handling qualities and the transient motor current and torque requirements was demonstrated for a six passenger quad-rotor NASA reference design.

Models developed with the proposed approach are now integrated into the Vertical Motion Simulator facility at NASA Ames for pilot testing, and are producing qualitatively reasonable results. However, there is currently no actual flight data available with which to perform a quantitative validation of the flight dynamics. There are also many important effects not explicitly treated in our approach, especially pertaining to thermal effects on the electrical system performance or operating condition dependence of motor parameters. However, if these effects can be mapped to the expected power transmission efficiency (η) or other already included parameters then they too can be accounted for in the modeling framework.

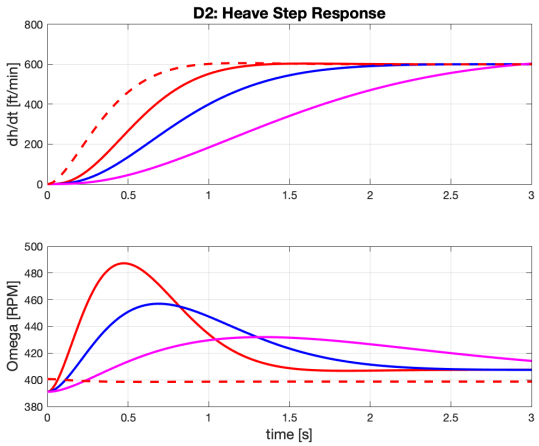

Figure 18: RPM controlled, heave step response from hover.

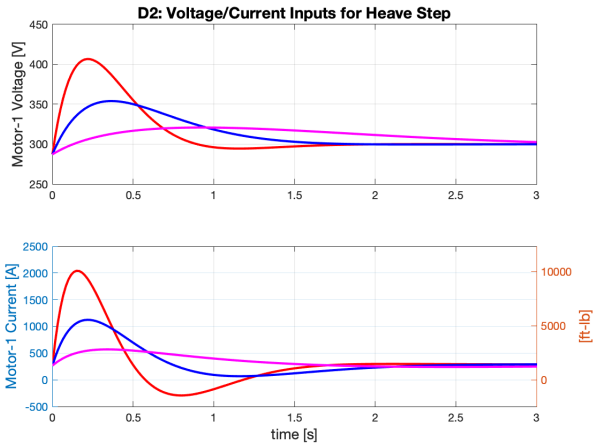

Figure 19: RPM controlled, voltage and current inputs for heave step response.

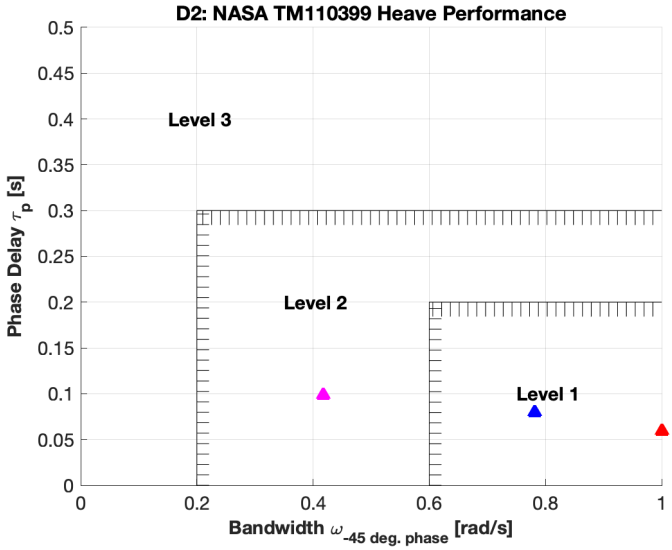

Figure 20: RPM controlled, NASA heave performance metrics.

Acknowledgements

This work was supported by NASA's Revolutionary Vertical Lift Technology project.

References

[1]Wayne Johnson, Christopher Silva, and Eduardo Solis. Concept vehicles for vtol air taxi operations. In *AHS Technical Conference on Aeromechanics Design for Transformative Vertical Flight*. American Helicopter Society, January 2018.

[2]Christopher Silva, Wayne R. Johnson, Eduardo Solis, Michael D. Patterson, and Kevin R. Antcliff. Vtol urban air mobility concept vehicles for technology development. In *2018 Aviation Technology, Integration, and Operations Conference*, Dallas, TX, June 2018. DOI: 10.2514/6.2018-3847.

[3]*Aeronautical Design Standard Performance Specification Handling Qualities Requirements for Military Rotorcraft*. United States Army Aviation and Missile Command, March 2000. ADS-33E-PRF.

[4]B. Lawrence, C. R. Theodore, W. Johnson, and T. Berger. A handling qualities analysis tool for rotorcraft conceptual designs. *The Aeronautical Journal*, 122(1252):960–987, June 2018.

[5]W. Johnson. NDARC, NASA design and analysis of rotorcraft. Technical Report TP 2009-215402, National Aeronautics and Space Administration, 2009.

[6]M.B. Tischler, J. Colbourne, M. Morel, D. Biezad, K. Cheung, W. Levine, and V. Moldoveanu. A multidisciplinary flight control development environment and its application to a helicopter. *IEEE Control Systems Magazine*, 19(4):22–33, August 1999.

[7]Carlos Malpica and Shannah Withrow-Maser. Handling qualities analysis of blade pitch and rotor speed controlled evtol quadrotor concepts for urban air mobility. In *VFS International Powered Lift Conference*. Vertical Flight Society, January 2020.

[8]Ben Lawrence, Gareth Padfield, and Philip Perfect. Flexible uses of simulation tools in an academic environment. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, August 2006. DOI: 10.2514/6.2006-6808.

[9]Utku Eren, Anna Prach, Başaran Bahadr Koçer, Saša V. Raković, Erdal Kayacan, and Behçet Açkmeşe. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017. DOI: 10.2514/1.G002507.

[10]Thomas Lombaerts, John Kaneshige, Stefan Schuet, Gordon Hardy, Bimal Aponso, and Kimberlee Shish. Dynamic inversion based full envelope flight control for an evtol vehicle using a unified framework. In *AIAA SciTech Forum*, January 2020. DOI: 10.2514/6.2020-1619.

[11]SC-VTOL-01. EASA special condition vertical take-off and landing (vtol) aircraft. European Union Aviation Safety Agency, July 2019.

[12]Lior Zivan and Mark B. Tischler. Development of a full flight envelope helicopter simulation using system identification. *Journal of the American Helicopter Society*, 55, 2010. DOI: 10.4050/JAHS.55.022003.

[13]Jeff A. Lusardi, Mark B. Tischler, Chris L. Blanken, and Steven J. Labows. Empirically derived helicopter response model and control system requirements for flight in turbulence. *Journal of the American Helicopter Society*, 49(3):340–349, 2004. DOI: 10.4050/JAHS.49.340.

[14]Gopal H. Gaonkar. Review of turbulence modeling and related applications to some problems of helicopter flight dynamics. *Journal of the American Helicopter Society*, 53(1):87–107, 2008.

[15]R. A. Hess. Rotorcraft handling qualities in turbulence. *Journal of Guidance, Control, and Dynamics*, 18(1):39–45, 1995.

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

[17]D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000. DOI: 10.1016/S0005-1098(99)00214-9.

[18]Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, 2nd edition, 2001.

[19]James A. Franklin and Michael W. Stortz. Moving base simulation evaluation of translational rate command systems for stovl aircraft in hover. Technical Report TM110399, National Aeronautics and Space Administration, June 1996.