

Trajectory Planning and Online Performance Model Estimation for Advanced Air Mobility

Tenavi Nakamura-Zimmerer*, Andrew Patterson†, Michael J. Acheson‡, and Irene M. Gregory§
NASA Langley Research Center, Hampton, VA 23681

We propose a framework for adaptive guidance based on optimal control and estimation using closed-loop performance models. In the proposed approach, parameters of a reduced-order performance model are estimated in real time so that trajectories and guidance commands are replanned with more accurate knowledge of the system’s response and performance capabilities. To apply this methodology to flight control systems, we introduce a simple yet expressive performance model modified from a reference linear design model. The proposed framework is applied to guidance of a simulated Advanced Air Mobility class concept aircraft experiencing control effector failures. We demonstrate that, by optimizing performance model parameters in real time, guidance commands can be intelligently adjusted to recover system stability and the performance of a full-order vehicle model, even in the event of effector failures. Furthermore, the reduced-order performance model requires a fraction of the computational cost of the full-order model, facilitating real-time use of adaptive, optimal guidance.

I. Introduction

ADVANCED Air Mobility (AAM) promises to revolutionize the air transportation system through the introduction of diverse electric Vertical Take-Off and Landing (VTOL) aircraft operating with high levels of automation and autonomy. Prototypical AAM class aircraft [1], like the NASA Lift-plus-Cruise (L+C) concept considered in this paper, are a blend between rotorcraft and conventional fixed-wing aircraft. These new vehicles are expected to exhibit three phases of flight: rotor-borne VTOL, wing-borne cruise, and a transition between the two. AAM class aircraft must contend with numerous, interacting control effectors, highly nonlinear dynamics, operations in crowded, dynamic, and uncertain environments, and the high potential for new propulsor technologies to experience failures, presenting a host of novel guidance and control challenges [2].

One line of research has explored the use of optimal control, specifically differential dynamic programming (DDP) [3, 4], as a means of trajectory planning with collision avoidance, control synthesis, and fault mitigation for AAM aircraft. In prior work [5, 6], DDP was applied (in simulation) to plan trajectories and control inputs at the individual effector level, using the full six-degree-of-freedom (6DoF) aircraft dynamics. This exploratory work is helpful in identifying how such novel AAM class aircraft can be flown effectively, as prescriptions based on engineering experience from conventional aircraft may not be helpful or even available.

While this previous work uses DDP to directly command effectors to control vehicle rates and velocities, we expect that many aerospace applications will use a preferred controller for this task. This is due in part to DDP’s considerable computational demands when applied to high-dimensional nonlinear systems, potential concerns about the real-time convergence of an iterative algorithm like DDP, and uncertainty around the potential performance gained by using the full 6DoF dynamics compared to a reduced-order surrogate. We expect that it is more likely for guidance and control systems on AAM class vehicles to operate at separate levels of abstraction, much like the current paradigm in aerospace systems [7, 8]. That is, an outer-loop trajectory planner is wrapped around an inner-loop controller which handles low-level stability and command tracking. Due to time and complexity scale separation between the inner and outer loops, a guidance algorithm can generally use a simplified *performance model* (i.e., *guidance model*) which approximates the aircraft’s closed-loop response to guidance commands. This is the approach we take in this work.

It is possible, however, that the inner-loop control system cannot achieve the expected response performance in the presence of significant disturbances, unmodeled dynamics, or effector failures which it cannot compensate for. In such situations, a simplified performance model could be too optimistic, and consequently, guidance commands could

*Research Aerospace Engineer, Flight Dynamics Branch, Member AIAA.

†Research Aerospace Engineer, Dynamic Systems and Control Branch.

‡Senior Research Engineer, Dynamic Systems and Control Branch, Member AIAA.

§Senior Technologist for Advanced Control Theory and Application, Fellow AIAA.

demand too much from a degraded control system, potentially jeopardizing performance and even stability. Real-time accurate knowledge of the aircraft and control system's dynamics can allow the guidance system to adjust commands to accomplish the mission (or abort if necessary) while remaining within the aircraft's achievable performance bounds.

To begin to address this challenge, we propose an *adaptive guidance* approach based on optimal control and estimation. In the proposed approach, a computational optimal control algorithm—such as DDP—is used to optimize parameters of the vehicle performance model to match previously observed behavior. Best-fit parameters are carried forward when replanning the future trajectory and guidance commands, again using an optimal control algorithm like DDP. In the context of flight control systems, we introduce a performance model of the inner loop's command-tracking response, with a structure based on the ideal linear reference dynamics which a typical robust and adaptive control system is designed to emulate.

The proposed framework is illustrated on two examples. First, we consider a synthetic problem from the $\mathcal{L}_1$ adaptive control ($\mathcal{L}_1$ -AC) literature [9]. This proof of concept serves to verify that our approach can match the performance of existing methods, while providing informative parameter estimates. This example is not intended to demonstrate superiority of the proposed method, as adaptive control fills a different role than optimal control-based guidance.

In the second example, we apply the proposed methodology to adaptive guidance of the NASA L+C AAM concept vehicle [1] with a robust servomechanism linear quadratic regulator (RSLQR) [8] inner-loop control law. We find that, when flying in nominal conditions, a reduced-order vehicle and controller performance model completely recovers the performance achieved when using the full 6DoF model at a fraction of the computational cost. When failures are introduced into the control effectors, the reduced-order model can still recover stability and performance if we perform real-time model parameter estimation. Conversely, ignoring the inner-loop controller dynamics can cause a complete loss of control when effector failures are introduced.

The remainder of this paper is organized as follows. In Section I.A we present the optimal control and estimation problems which form the basis of the adaptive guidance approach, and in Section I.B we provide an overview of DDP, the algorithm which we use to solve these problems. In Section II we present the performance model guidance and estimation formulation, and apply it to the $\mathcal{L}_1$ -AC benchmark example. Next in Section III we describe a formulation for performance models of robust and adaptive control systems, and in Section IV we apply this to adaptive guidance for the L+C vehicle. Concluding thoughts and future research directions are discussed in Section V.

A. Optimal Control and Estimation

In this paper, we consider aircraft guidance problems formulated as deterministic finite-horizon optimal control problems (OCPs) of the form

$$\begin{cases} \min_{\mathbf{u}(\cdot)} & \mathcal{J}_{\text{mpc}}[\mathbf{u}(\cdot)] := \int_{t_0}^{t_0+T_{\text{mpc}}} \mathcal{L}_{\text{mpc}}(t, \mathbf{x}(t), \mathbf{u}(t); \boldsymbol{\theta}) dt + \phi_{\text{mpc}}(\mathbf{x}(t_0 + T_{\text{mpc}})) \\ \text{s.t.} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}; \boldsymbol{\theta}) \\ & \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \quad (1)$$

Here $\mathbf{x} \in \mathbb{R}^{n_x}$ is the state, $\mathbf{u} \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ is the control input, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ are system parameters, $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{U} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}^{n_x}$ are the vehicle dynamics, $\mathcal{L}_{\text{mpc}} : [0, \infty) \times \mathbb{R}^{n_x} \times \mathbb{U} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{R}$ is the running cost, $\phi_{\text{mpc}} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the terminal cost, $t_0 \in \mathbb{R}$ is the initial time, $T_{\text{mpc}} > 0$ is the control horizon, and $\mathbf{x}_0 \in \mathbb{R}^{n_x}$ is a given initial condition.

During flight, disturbances and model mismatch between $\mathbf{f}(\cdot)$ and the true vehicle dynamics will inevitably lead to deviations from a previously planned trajectory. This means that we will need to repeatedly solve Eq. (1) online in a moving horizon fashion, incorporating new estimates of parameters and initial states as they become available. Note that in practice, we only partially solve Eq. (1) by performing just one or a few iterations of DDP per discrete timestep. This process is often called *model predictive control* (MPC).

Notice that the MPC problem (Eq. (1)) depends on finding suitable parameter and state estimates. This naturally leads to a *moving horizon estimation* (MHE) problem which is dual to Eq. (1). Let $t_0 \geq 0$ be the current time, $T_{\text{mhe}} > 0$ be the estimation time horizon, $\hat{\boldsymbol{\theta}} : [t_0 - T_{\text{mhe}}, t_0] \rightarrow \mathbb{R}^{n_\theta}$ the estimated parameter history, $\hat{\mathbf{x}}_{-T} := \hat{\mathbf{x}}(t_0 - T_{\text{mhe}})$ be the estimate of the past initial state, and $\mathbf{u} : [t_0 - T_{\text{mhe}}, t_0] \rightarrow \mathbb{U}$ be the known control history. We can then write a generic

estimation problem in continuous time as an OCP with free initial condition:

$$\begin{cases} \min_{\hat{\theta}(\cdot), \hat{\mathbf{x}}_{-T}} & \mathcal{J}_{\text{mhe}}[\hat{\theta}(\cdot), \hat{\mathbf{x}}_{-T}] := \phi_{\text{mhe}}(\hat{\mathbf{x}}(t_0 - T_{\text{mhe}})) + \int_{t_0 - T_{\text{mhe}}}^{t_0} \mathcal{L}_{\text{mhe}}(t, \hat{\mathbf{x}}(t), \hat{\theta}(t)) dt \\ \text{s.t.} & \dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}(t); \hat{\theta}) \\ & \hat{\mathbf{x}}(t_0 - T_{\text{mhe}}) = \hat{\mathbf{x}}_{-T} \end{cases} \quad (2)$$

It should be emphasized that Eq. (2) is a standard OCP with the parameters, $\hat{\theta}(\cdot)$, acting as controls and with a free initial condition, $\hat{\mathbf{x}}_{-T}$. Therefore, many algorithms for solving Eq. (1) can also be applied to Eq. (2).

To determine a practical cost function for estimation, suppose that we have access to (noisy) system measurements at discrete times t_k , $k = -N, \dots, 0$, with $t_{-N} = t_0 - T_{\text{mhe}}$:

$$\mathbf{y}(t_k) = \mathbf{h}(\mathbf{x}(t_k)) + \mathbf{w}_k \in \mathbb{R}^{n_y}, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_w). \quad (3)$$

Here $\mathbf{x}(\cdot)$ is the true (unknown) state, $\mathbf{h} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ is the sensor model, and $\mathbf{w}_k \in \mathbb{R}^{n_y}$ represents zero-mean Gaussian noise with known covariance $\Sigma_w \in \mathbb{R}^{n_y \times n_y}$. The joint probability of observing $\mathbf{y}(t_k)$, $k = -N, \dots, 0$, is given by

$$\rho(\mathbf{y}(t_k), k = -N, \dots, 0 | \hat{\theta}(\cdot), \hat{\mathbf{x}}_{-T}) = \prod_{k=-N}^0 \mathcal{N}(\mathbf{h}(\hat{\mathbf{x}}(t_k)), \Sigma_w), \quad (4)$$

where (with some abuse of notation) we have used $\mathcal{N}(\cdot)$ to denote the probability density function of a multivariate normal, and

$$\hat{\mathbf{x}}(t_k) = \hat{\mathbf{x}}_{-T} + \int_{t_0 - T_{\text{mhe}}}^{t_k} \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t); \hat{\theta}(t)) dt. \quad (5)$$

Like [10], we assign Gaussian priors to the system parameter and past initial state estimates. For simplicity, we consider a finite-dimensional prior on $\hat{\theta}(t_k)$, $k = -N, \dots, 0$. Specifically, suppose that $\hat{\theta}(t_k) \sim \mathcal{N}(\underline{\theta}(t_k), \Sigma_\theta)$ and $\hat{\mathbf{x}}_{-T} \sim \mathcal{N}(\underline{\mathbf{x}}_{-T}, \Sigma_x)$, where the prior means $\underline{\theta}(t_k) \in \mathbb{R}^{n_\theta}$, $\underline{\mathbf{x}}_{-T} \in \mathbb{R}^{n_x}$ could be taken to be the estimates computed at the previous timestep, and the covariance matrices $\Sigma_\theta \in \mathbb{R}^{n_\theta \times n_\theta}$, $\Sigma_x \in \mathbb{R}^{n_x \times n_x}$ are user-specified parameters. Then the maximum a posteriori probability estimate is given by

$$\arg\max_{\hat{\theta}(\cdot), \hat{\mathbf{x}}_{-T}} \rho(\hat{\mathbf{x}}_{-T}) \prod_{k=-N}^0 \rho(\hat{\theta}(t_k)) \rho(\mathbf{y}(t_k) | \hat{\theta}(\cdot), \hat{\mathbf{x}}_{-T}). \quad (6)$$

This is equivalent to minimizing the negative log probability. Thus, with the specified Gaussian priors on $\hat{\mathbf{x}}_{-T}$ and $\hat{\theta}(t_k)$, $k = -N, \dots, 0$, some algebra leads to the following cost function:

$$\mathcal{J}_{\text{mhe}}[\hat{\theta}(\cdot), \hat{\mathbf{x}}_{-T}] = \|\hat{\mathbf{x}}_{-T} - \underline{\mathbf{x}}_{-T}\|_{\Sigma_x^{-1}}^2 + \sum_{k=-N}^0 \left(\|\mathbf{h}(\hat{\mathbf{x}}(t_k)) - \mathbf{y}(t_k)\|_{\Sigma_w^{-1}}^2 + \|\hat{\theta}(t_k) - \underline{\theta}(t_k)\|_{\Sigma_\theta^{-1}}^2 \right), \quad (7)$$

where $\|\mathbf{x}\|_{\mathbf{A}} := \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$ for a vector $\mathbf{x}$ and matrix $\mathbf{A}$ of compatible size.

B. Differential Dynamic Programming

In this paper, we use DDP [3, 4] as the computational tool for solving the MPC and MHE OCPs (Eqs. (1) and (2), respectively). Of course, DDP is just one of many viable algorithms for solving open-loop OCPs; see [11–13] for reviews of some of these methods. The guidance and estimation framework described in this paper can, in principle, be implemented with any reliable computational method. By choosing DDP we do not assert that it is superior to any other method; rather, we use it because it is convenient (we have access to an existing implementation) and because it has performed well in previous research on control of AAM class aircraft [5, 6].

At a high level, DDP is an iterative algorithm, with each iteration consisting of a backward pass followed by a forward pass. Starting from an initial guess for the optimal control signal $\mathbf{u}(\cdot)$ (or $\hat{\theta}(\cdot)$ in MHE) and the corresponding state trajectory $\mathbf{x}(\cdot)$, DDP computes a linear approximation of the dynamics about $\mathbf{x}(\cdot)$ and $\mathbf{u}(\cdot)$. This is used to propagate a quadratic approximation of the value function backwards in time. Then, in the forward pass, the full nonlinear dynamics are used to propagate the state forward and update the control sequence using gains computed based on the approximate

value function. Under reasonable technical conditions, DDP is guaranteed to converge to a locally optimal control sequence. For details we refer the reader to [3, 4, 6].

Variations of DDP have also been proposed to solve parameter-dependent OCPs [6, 10]. In addition to optimizing the control sequence, these methods can simultaneously optimize a set of *static-in-time parameters*. These are distinct from the time-varying parameters optimized in the MHE problem (Eq. (2)), though they have been used to solve problems like Eq. (2) by modeling some parameters as static-in-time. Although static parameter estimates can change as Eq. (2) is solved repeatedly over time, that approach can encounter difficulties when certain system parameters change rapidly over the estimation time horizon. In this paper, we treat the MPC and MHE problems separately, without optimizing any static parameters. However, our approach is compatible with parameterized DDP, and such interactions could be explored in future work.

II. Guidance and Estimation with Performance Models

Since DDP is expensive to evaluate and potentially difficult to certify for use as an inner-loop flight controller, it is more likely to take the role of an (adaptive) guidance algorithm. In this scenario, we leave the stability and tracking control to a fast, robust, and certifiable inner-loop controller. Meanwhile, DDP computes reference trajectories based on a simplified performance model, with model parameters which can be updated online through MHE. From the guidance perspective, the true, controlled vehicle dynamics are treated as a black box, into which DDP passes reference commands and receives output measurements. Note that although we do not use a full-order model, we have some choice about how to design the reduced-order dynamics because we have knowledge of the inner-loop controller (see Section III).

A. General Formulation

The performance model MPC and MHE problem can be formulated as follows. Let $\mathbf{x} \in \mathbb{R}^{n_x}$ be the closed-loop vehicle state, which includes both vehicle states and inner-loop controller dynamics (e.g., integrator states). Let $\mathbf{r} \in \mathbb{U}_r \subseteq \mathbb{R}^{n_r}$ be a bounded reference input to the inner loop, and suppose that the closed-loop system satisfies

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{r}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{y} = \mathbf{h}(\mathbf{x}) \in \mathbb{R}^{n_y}. \quad (8)$$

Rather than apply DDP directly to this large, complicated system, we introduce a simpler dynamic model,

$$\dot{\mathbf{x}}_m = \mathbf{f}_m(\mathbf{x}_m, \mathbf{r}; \boldsymbol{\theta}_m), \quad \mathbf{x}_m(0) = \mathbf{x}_{m,0}, \quad (9)$$

where $\mathbf{x}_m \in \mathbb{R}^{n_{x,m}}$ is model state (often a reduced subset of $\mathbf{x}$), and $\boldsymbol{\theta}_m \in \mathbb{R}^{n_\theta}$ are a set of model parameters (which could be physical or notional). Here we have used the subscript m notation (instead of a “hat”) to distinguish these model states and parameters from the state and parameter *estimates* in Eq. (2).

We formulate an OCP using the model dynamics (Eq. (9)) in Eq. (1). Similarly, we formulate an appropriate optimal estimation problem (Eq. (2)) to update model parameters $\boldsymbol{\theta}_m$ and states $\mathbf{x}_{m,-T}$, based on measurements, $\mathbf{y}(t_k)$, collected from the physical aircraft. Online, we collect output measurements and run DDP to solve this MHE problem, thereby updating $\boldsymbol{\theta}_m(t_0)$ and $\mathbf{x}_m(t_0)$. Then we use DDP to solve Eq. (1), optimizing the input signal $\mathbf{r}(\cdot)$ for the model dynamics in Eq. (9). We pass the result to the inner-loop control which in turn affects the vehicle dynamics in Eq. (8).

This formulation is predicated on the assumption that it is possible to find parameters with which the model dynamics approximate the true dynamics. To make this precise, let $\mathbf{y}_m = \mathbf{h}_m(\mathbf{x}_m) \in \mathbb{R}^{n_y}$ describe the model states which we would like to approximate the measurements, $\mathbf{y}$. We assume that the model dynamics are sufficiently expressive in the following sense.

Assumption 1 (Expressive performance model) Let $\varepsilon_y > 0$ be a given tolerance and T_{mpc} the control time horizon. We assume that there exist constants $\rho_x, \rho_r, \rho_\theta > 0$ and a set $\mathbb{X} \subseteq \mathbb{R}^{n_x}$, such that for any $\mathbf{x}_0 \in \mathbb{X}$ and integrable reference signal $\mathbf{r} : [0, T_{\text{mpc}}] \rightarrow \mathbb{U}_r$, there exist $\mathbf{x}_{m,0}^* \in \mathbb{R}^{n_{x,m}}, \boldsymbol{\theta}_m^* \in \mathbb{R}^{n_\theta}$, such that for all $\mathbf{x}_{m,0} \in \mathbb{R}^{n_{x,m}}, \boldsymbol{\theta}_m \in \mathbb{R}^{n_\theta}$ satisfying $\|\mathbf{x}_{m,0} - \mathbf{x}_{m,0}^*\| < \rho_x$ and $\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*\| < \rho_\theta$, we have

$$\|\mathbf{h}(\mathbf{x}(t)) - \mathbf{h}_m(\mathbf{x}_m(t))\| \leq \varepsilon_y, \quad 0 \leq t \leq T_{\text{mpc}}, \quad (10)$$

where $\dot{\mathbf{x}}_m = \mathbf{f}_m(\mathbf{x}_m, \mathbf{r}; \boldsymbol{\theta}_m)$ and $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{r})$.

Remark 1 We would like to reiterate that separation of model complexity for guidance and inner-loop control is not a new concept—this is in fact the state of the practice. The novelty of this formulation is the introduction of real-time performance model estimation using DDP, and the specific control system performance models introduced later in Section III.

B. Adaptive Control Benchmark Example

In this section we illustrate the performance model framework on a benchmark example from the $\mathcal{L}_1$ -AC literature. We show that the proposed method yields very similar control performance as $\mathcal{L}_1$ -AC. In addition, this approach provides predictive system parameter estimates, in contrast to $\mathcal{L}_1$ -AC which collects all uncertainty into an additive dynamics modifier (σ below). We emphasize that the objective is *not* to claim that DDP is superior to $\mathcal{L}_1$ -AC, simply that using optimal control for MPC and MHE with performance models is feasible.

Following [9], consider a dynamical system of the form

$$\begin{cases} \dot{\mathbf{x}} = (\mathbf{A} + \mathbf{A}_\Delta) \mathbf{x} + \mathbf{B}\omega\mathbf{u} + \mathbf{f}_\Delta(t, \mathbf{x}, \mathbf{z}), & \mathbf{x}(0) = \mathbf{x}_0, \\ \dot{\mathbf{z}} = \mathbf{g}(t, \mathbf{x}, \mathbf{z}), & \mathbf{z}(0) = \mathbf{z}_0, \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases} \quad (11)$$

where $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$, and $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ are fixed and known, with $n_y \leq n_u$, $\mathbf{A}$ Hurwitz, $\mathbf{B}$ full column rank, $(\mathbf{A}, \mathbf{B})$ controllable, and $(\mathbf{A}, \mathbf{C})$ observable; $\mathbf{A}_\Delta \in \mathbb{R}^{n_x \times n_x}$ is unknown and bounded; $\omega \in \mathbb{R}^{n_u \times n_u}$ is unknown, bounded, and strictly row-diagonally dominant with positive diagonal entries; and $\mathbf{f}_\Delta : [0, \infty) \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_x}$ are unmodeled nonlinear dynamics with unknown internal state $\mathbf{z} \in \mathbb{R}^{n_z}$. We assume $\mathbf{f}_\Delta(\cdot)$ is Lipschitz and bounded, and that the internal dynamics $\mathbf{g}(\cdot)$ are bounded-input-bounded-output stable with respect to initial condition and $\mathbf{x}$. The goal is to optimize $\mathbf{u}$ such that the system output, $\mathbf{y}$, tracks the output of the ideal reference system

$$\dot{\mathbf{x}}_{\text{id}} = \mathbf{A}\mathbf{x}_{\text{id}} + \mathbf{B}\mathbf{K}_{\text{DC}}\mathbf{r}, \quad \mathbf{y}_{\text{id}} = \mathbf{C}\mathbf{x}_{\text{id}}, \quad \mathbf{x}_{\text{id}}(0) = \mathbf{x}_0, \quad \mathbf{K}_{\text{DC}} := -(\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}, \quad (12)$$

for bounded reference inputs $\mathbf{r} : [0, \infty) \rightarrow \mathbb{U}_r \subset \mathbb{R}^{n_r}$. We consider the numerical example from [9] with true system parameters specified in ‘‘Scenario 1’’, and compare the results of DDP with the $\mathcal{L}_1$ -AC as designed in [9]. While only the results for ‘‘Scenario 1’’ are presented here, similar results were obtained with all other scenarios.

For DDP we consider the model system,

$$\dot{\mathbf{x}}_m = (\mathbf{A} + \hat{\mathbf{A}}_\Delta) \mathbf{x}_m + \mathbf{B}\hat{\omega}\mathbf{u} + \boldsymbol{\sigma}, \quad \mathbf{x}_m(0) = \mathbf{x}_0, \quad (13)$$

where the parameters $\hat{\mathbf{A}}_\Delta$, $\hat{\omega}$ are allowed to vary within the bounds specified in [9], and $\boldsymbol{\sigma} = \boldsymbol{\sigma}(t) \in \mathbb{R}^{n_x}$ is a bounded, time-varying disturbance estimate. The parameter estimates are initialized as $\hat{\mathbf{A}}_\Delta(t=0) = \mathbf{0}_{n_x \times n_x}$, $\hat{\omega}(t=0) = \mathbf{I}_{n_u \times n_u}$, and $\boldsymbol{\sigma}(t=0) = \mathbf{0}_{n_x}$. The most recent estimate of $\boldsymbol{\sigma}$ is held constant when optimizing controls.

We compute control inputs in an MPC fashion by applying DDP to solve Eq. (1) with the model dynamics in Eq. (13) and the following continuous time cost function:

$$\mathcal{J}_{\text{mpc}}[\mathbf{u}(\cdot)] = \int_{t_0}^{t_0+T_{\text{mpc}}} \left(100 \|\mathbf{C}\mathbf{x}_m(t) - \mathbf{y}_{\text{id}}(t)\|_2^2 + \|\mathbf{u}(t) - \mathbf{u}_{\text{id}}(t)\|_2^2 \right) dt + \|\mathbf{C}\mathbf{x}_m(t_0+T_{\text{mpc}}) - \mathbf{y}_{\text{id}}(t_0+T_{\text{mpc}})\|_2^2 \quad (14)$$

where $\mathbf{y}_{\text{id}}(\cdot)$ is the reference trajectory obtained from Eq. (12), $\mathbf{u}_{\text{id}}(t) := \mathbf{K}_{\text{DC}}\mathbf{r}(t)$, and $\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ as in [9]. Here we employ a quadratic cost function to strongly penalize deviations from the output of the ideal system (Eq. (12)), while including a small control penalty to reduce excessive control effort. Note that control penalties also serve to improve numerical conditioning of the OCP.

For MHE we use the Gaussian prior cost function shown in Eq. (7), where the measurements are simply taken to be the known, modeled states from the the true system. In particular,

$$\begin{aligned} & \mathcal{J}_{\text{mhe}}(\hat{\mathbf{A}}_\Delta(\cdot), \hat{\omega}(\cdot), \boldsymbol{\sigma}(\cdot), \mathbf{x}_{m,-T}) \\ &= \|\mathbf{x}_{m,-T} - \underline{\mathbf{x}}_{m,-T}\|_2^2 + \int_{t_0-T_{\text{mhe}}}^{t_0} \left(100 \|\mathbf{x}_m(t) - \mathbf{x}(t)\|_2^2 + \|\hat{\mathbf{A}}_\Delta(t) - \underline{\mathbf{A}}_\Delta(t)\|_F^2 + \|\hat{\omega}(t) - \underline{\omega}(t)\|_F^2 + \|\boldsymbol{\sigma}(t) - \underline{\boldsymbol{\sigma}}(t)\|_2^2 \right) dt, \end{aligned} \quad (15)$$

where $\underline{\mathbf{x}}_{m,-T}$, $\underline{\mathbf{A}}_\Delta$, $\underline{\omega}$, and $\underline{\boldsymbol{\sigma}}$ denote prior means on the initial condition and parameter estimates (i.e., the estimates from the previous discrete timestep), and $\|\cdot\|_F$ denotes the Frobenius norm. Here we apply a strong penalty on deviations from past state observations and small penalties on model parameters to prevent overfitting and improve numerical

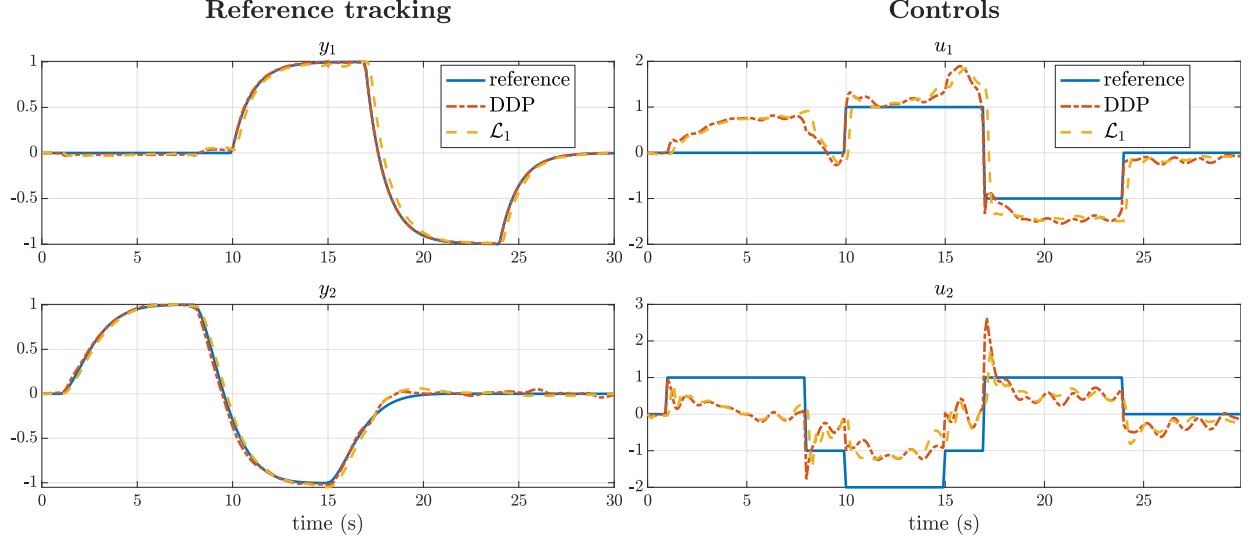


Fig. 1 Reference commands, $\mathbf{r}(t)$, tracked outputs $\mathbf{y}(t)$, and controls, $\mathbf{u}(t)$, for DDP-based MPC and MHE compared with $\mathcal{L}_1$ -AC for the example from [9].

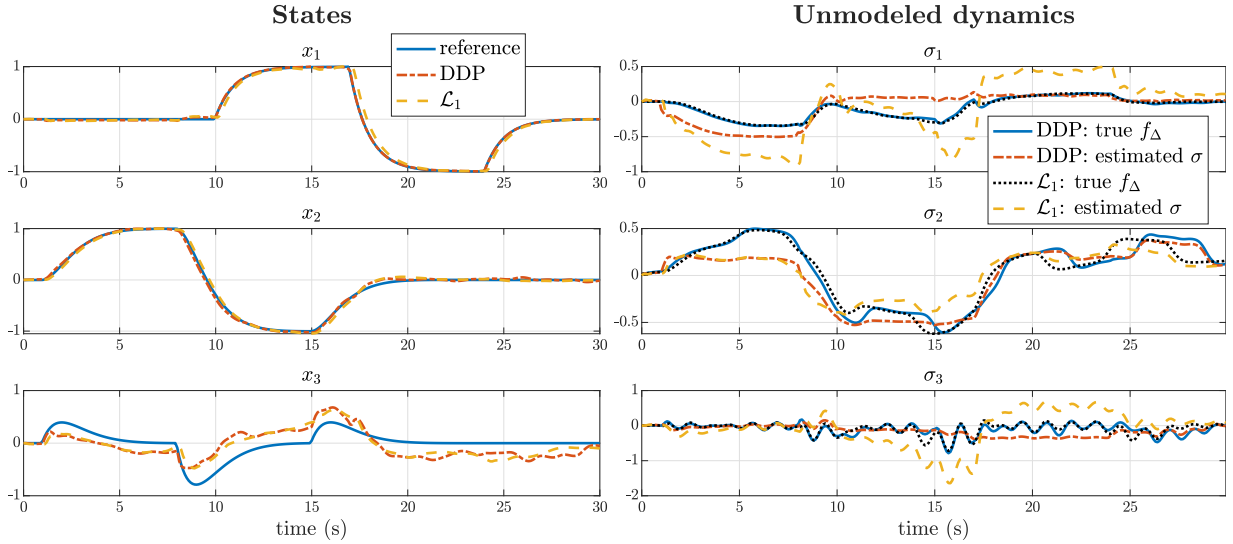


Fig. 2 State histories, $\mathbf{x}(t)$, unmodeled dynamics, $\mathbf{f}_\Delta(t, \mathbf{x}, \mathbf{z})$, and additive uncertainty estimates, $\boldsymbol{\sigma}(t)$, for the example from [9]. Note that $\mathbf{f}_\Delta(t, \mathbf{x}, \mathbf{z})$ varies depending on the different state histories resulting from applying DDP or $\mathcal{L}_1$ -AC.

conditioning. Notice that although we have complete and perfect state observations in this example problem, rather than fixing $\mathbf{x}_{m,-T} = \mathbf{x}(t_0 - T_{\text{mhe}})$, we allow $\mathbf{x}_{m,-T}$ to vary and impose a penalty on deviation from its prior. Freeing the initial condition in this way, even when not needed, can slightly improve numerical conditioning.

We use a timestep of $\Delta_t = 0.1\text{s}$, one iteration of DDP per timestep, and time horizons of $T_{\text{mpc}} = 5\text{s}$ and $T_{\text{mhe}} = 1\text{s}$. The reference trajectory, $\mathbf{r}(t)$, is defined over a 30s interval as in [9].

As shown in Fig. 1, we find that DDP—knowing only about the linear dynamics with an additive disturbance—achieves comparable tracking performance to $\mathcal{L}_1$ -AC with similar control effort. Next, Figs. 2 and 3 show how DDP’s parameter estimates evolve over time, compared to the additive disturbance estimates to those obtained with $\mathcal{L}_1$ -AC. Observe that some (but not all) estimates of entries of $\hat{\mathbf{A}}_\Delta$ and $\hat{\boldsymbol{\omega}}$ converge to their true values. It should not be surprising that full convergence was not achieved; this is because the model dynamics do not include structural knowledge of the true nonlinear dynamics.

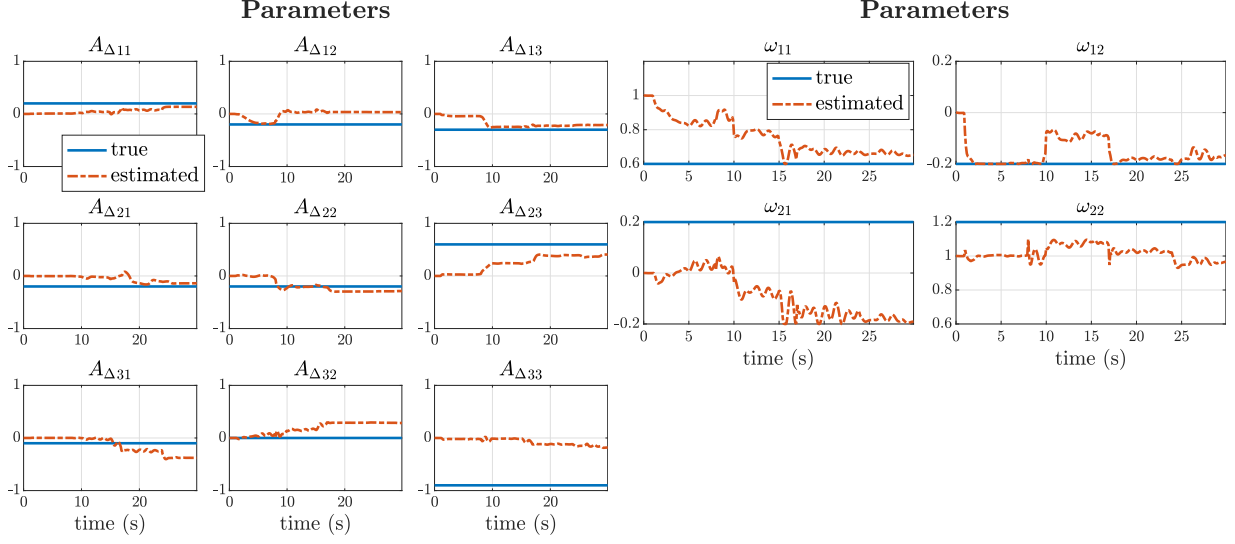


Fig. 3 DDP model parameter estimates, $\hat{\mathbf{A}}_{\Delta}(t)$ and $\hat{\boldsymbol{\omega}}(t)$, for the example from [9].

Nevertheless, by correctly estimating some components of these matrices, DDP relies less on the additive uncertainty estimate than $\mathcal{L}_1$ -AC, which treats all uncertainty as additive (i.e., $\mathcal{L}_1$ -AC only estimates $\boldsymbol{\sigma}$). Furthermore, the ability to compute structured parameter estimates could potentially be useful for other autonomous decision-making algorithms.

Lastly, we note that for this problem, the optimal control-based algorithm can achieve stability and tracking by estimating either $\boldsymbol{\sigma}$ or the matrices $\hat{\mathbf{A}}_{\Delta}$ and $\hat{\boldsymbol{\omega}}$. When only $\boldsymbol{\sigma}$ was estimated, performance was almost identical to $\mathcal{L}_1$ -AC. Best performance with the new method was obtained when all three parameters were estimated.

III. Reference Inner-loop Performance Models

In this section we consider a generic, continuous-time vehicle guidance model,

$$\dot{\mathbf{x}}_g = \mathbf{f}_g(\mathbf{x}_g, \mathbf{u}_g; \boldsymbol{\theta}_g), \quad \mathbf{x}_g(0) = \mathbf{x}_{g,0}, \quad (16)$$

where the guidance input, $\mathbf{u}_g \in \mathbb{R}^{n_u}$, is the output of an inner-loop control system. A common paradigm in $\mathcal{L}_1$ -AC [9, 14] and model reference adaptive control (MRAC) [8] is to design the control law such that $\mathbf{u}_g$ tracks the output of an ideal reference model. The reference model is often the closed-loop linear dynamics of a RSLQR [8, 15].

Adaptive control offers a number of desirable properties, including guaranteed stability and tracking performance. However, performance can vary within some bounds, and could degrade further if the vehicle experiences disturbances or effector failures which cannot be rejected by the adaptive controller. Therefore, we expect that it is beneficial to include a model of the controller's (time-varying) performance when using DDP for trajectory planning. While the true inner-loop control dynamics are available, a simplified performance model of the inner loop can be sufficient to capture the most important dynamics and reduce computational cost. Motivated by the results of Section II.B, we propose a performance model which uses the ideal reference dynamics that the inner loop is designed to track.

Suppose that the inner-loop control is designed such that, for $\mathbf{r} = \mathbf{r}(t) \in \mathbb{U}_r \subset \mathbb{R}^{n_r}$, $n_r \leq n_u$, the inner loop tracks the output of the ideal reference system*,

$$\dot{\mathbf{z}}_{\text{id}} = \mathbf{A}\mathbf{z}_{\text{id}} + \mathbf{B}\mathbf{r}, \quad \mathbf{u}_{\text{id}} = \mathbf{C}\mathbf{z}_{\text{id}}, \quad \mathbf{z}_{\text{id}}(0) = \mathbf{z}_{\text{id},0}, \quad (17)$$

where $\mathbf{A} \in \mathbb{R}^{n_z \times n_z}$, $\mathbf{B} \in \mathbb{R}^{n_z \times n_r}$, and $\mathbf{C} \in \mathbb{R}^{n_u \times n_z}$ are known and defined analogously to Eq. (11). To account for variations in tracking performance, as well as degraded performance resulting from major disturbances, we propose an inner-loop performance model which resembles the kinds of dynamics typically considered in adaptive control [8, 9]:

$$\dot{\mathbf{z}} = (\mathbf{A} + \hat{\mathbf{A}}_{\Delta})\mathbf{z} + \mathbf{B}\hat{\boldsymbol{\omega}}\mathbf{r} + \boldsymbol{\sigma}, \quad \hat{\mathbf{u}}_g = \mathbf{C}\mathbf{z}, \quad \mathbf{z}(0) = \mathbf{z}_0. \quad (18)$$

*For a detailed performance analysis of $\mathcal{L}_1$ -AC and MRAC, we refer the reader to [14] and [8], respectively.

Here $\mathbf{z} \in \mathbb{R}^{n_z}$ and $\hat{\mathbf{u}}_g \in \mathbb{R}^{n_u}$ represent *predictions* of the inner loop's response to reference commands, $\hat{\mathbf{A}}_\Delta \in \mathbb{R}^{n_z \times n_z}$ and $\hat{\omega} \in \mathbb{R}^{n_r \times n_r}$ are unknown and defined analogously to Eq. (11), and $\sigma \in \mathbb{R}^{n_z}$ is an estimate of additive disturbances. The combined vehicle and inner-loop performance model can then be expressed as

$$\mathbf{x}_m := \begin{pmatrix} \mathbf{x}_g \\ \mathbf{z} \end{pmatrix}, \quad \theta_m := \{\theta_g, \hat{\mathbf{A}}_\Delta, \hat{\omega}, \sigma\}, \quad \dot{\mathbf{x}}_m = \mathbf{f}_m(\mathbf{x}_m, \mathbf{r}; \theta_m) := \begin{pmatrix} \mathbf{f}_g(\mathbf{x}_g, \mathbf{C}\mathbf{z}; \theta_g) \\ (\mathbf{A} + \hat{\mathbf{A}}_\Delta)\mathbf{z} + \mathbf{B}\hat{\omega}\mathbf{r} + \sigma \end{pmatrix}. \quad (19)$$

Note that additive uncertainty, σ , can also be included in the guidance dynamics, $\mathbf{f}_g(\cdot)$.

IV. AAM Trajectory Planning with Control Effector Failures

Finally, we present an illustrative example of how the performance model estimation and guidance approach introduced in Section II can be applied to the NASA L+C concept aircraft. The L+C has 8 vertical rotors for VTOL operation, as well as a fixed wing, pusher propeller, and aerodynamic surfaces for forward flight [1]. For the inner loop, we design a RSLQR [8] to track heading frame velocity and yaw rate commands. DDP is applied to a simplified kinematic guidance model (see Section IV.B) combined with an inner-loop performance model of RSLQR as introduced in Section III. An overview of the guidance and control architecture is presented in Fig. 4.

Simulation results presented in Section IV.E suggest that DDP with the combined kinematic-RSLQR performance model can match the performance obtained when using the full 6DoF L+C-RSLQR dynamic model, at a tiny fraction of the computational cost. Furthermore, the framework is shown to be able to compensate for degraded inner-loop command-tracking performance as a result of effector failures, maintaining vehicle stability and completing the maneuver.

A. Vehicle dynamics

We begin with an overview of the 6DoF nonlinear dynamics of the L+C aircraft. For a more detailed introduction to flight dynamics, we refer the reader to e.g., [7, 8]. Following [15–17], we formulate the dynamics in the *heading* frame, which is convenient for providing guidance commands applicable in “cruise” (forward flight like a fixed-wing aircraft), “hover” (VTOL operations), and “transition” modes (a blend between cruise and hover, using vertical rotors and the pusher propeller).

Let $\boldsymbol{\eta} = (\phi \ \theta \ \psi)^T$ denote the vehicle's attitude relative to the Earth-fixed inertial reference frame, described by Euler angles, roll, pitch, and yaw, in the body-fixed 3-2-1 rotation sequence ($\psi - \theta - \phi$). The heading frame is the

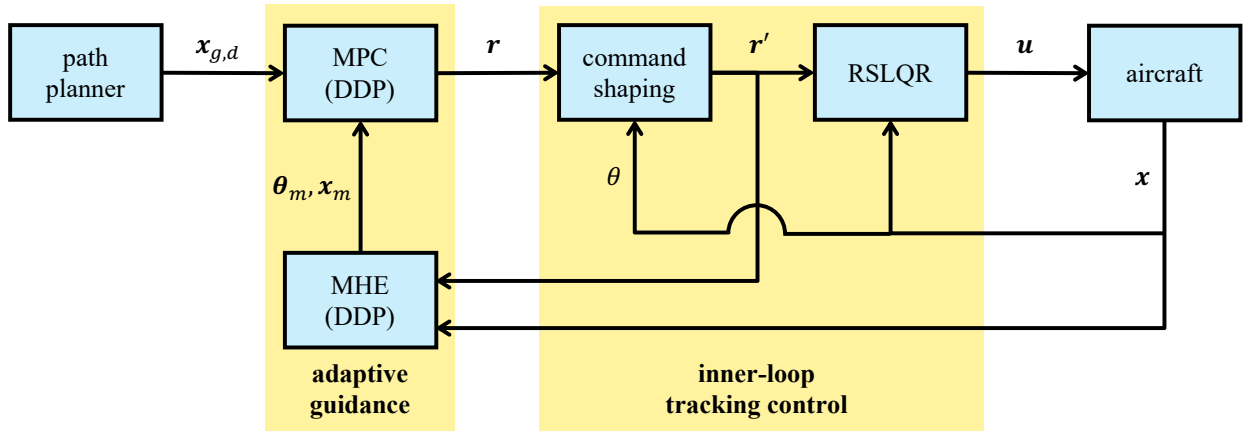


Fig. 4 Overview of the guidance and control architecture for the L+C example problem. MPC and MHE problems are based on kinematic guidance and inner-loop performance models. The RSLQR control law is described in Section IV.C, and the command shaping for turn coordination in Section IV.D. The path planner is not modeled in this example; see [19] for an example of how DDP can be combined with a high-level path planner in the context of AAM.

inertial frame rotated about the z -axis by ψ . Next let $\mathbf{p} = (p_n \ p_e \ p_d)^T$ be the vehicle's inertial north, east, and down position, $\mathbf{v}_h = (u_h \ v_h \ w_h)^T$ be the heading-frame velocity, and $\boldsymbol{\omega} = (p \ q \ r)^T$ be the vehicle's angular velocity with respect to the inertial frame, expressed in the body frame. The state dynamics are given by

$$\begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}}_h \\ \dot{\boldsymbol{\eta}} \\ \mathbf{J}\dot{\boldsymbol{\omega}} \end{pmatrix} = \begin{pmatrix} \mathcal{R}_{h \rightarrow i}(\psi) \mathbf{v}_h \\ -\dot{\psi} \mathbf{e}_3 \times \mathbf{v}_h + g \mathbf{e}_3 + \frac{1}{m} \mathbf{F}_h(\mathbf{v}_h, \boldsymbol{\eta}, \boldsymbol{\omega}, \mathbf{u}) \\ \mathbf{S}(\boldsymbol{\eta}) \boldsymbol{\omega} \\ -\boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega} + \mathbf{M}(\mathbf{v}_h, \boldsymbol{\eta}, \boldsymbol{\omega}, \mathbf{u}) \end{pmatrix}, \quad (20)$$

where g is gravitational acceleration, m is the vehicle's mass, $\mathbf{J}$ is its inertia matrix, $\mathbf{u} \in \mathbb{R}^{14}$ is the vector of control inputs[†], $\mathbf{e}_3 := (0 \ 0 \ 1)^T$, and

$$\mathcal{R}_{h \rightarrow i}(\psi) := \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{S}(\boldsymbol{\eta}) := \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{pmatrix}. \quad (21)$$

The functions $\mathbf{F}_h, \mathbf{M} : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^{14} \rightarrow \mathbb{R}^3$ are aero-propulsive models for the forces (expressed in the heading frame) and moments acting on the vehicle. The models used in this work are described in [18].

B. Kinematic Guidance Model

In this section we present the reduced-order kinematic model which we use for guidance in this work. The model state and reference commands are a subset of the full vehicle state described in Section IV.A, namely

$$\mathbf{x}_g = (p_n \ p_e \ h \ \psi)^T, \quad \mathbf{u}_g = (u_{h,c} \ v_{h,c} \ w_{h,c} \ \dot{\psi}_c)^T, \quad (22)$$

where $h = -p_d$ denotes altitude and a subscript “ c ” denotes a commanded state. At this abstraction level, we assume that heading frame velocities and yaw rate change instantaneously. When the inner-loop tracking performance is good, this is a reasonable approximation as the inner loop will operate at a faster timescale than the guidance algorithm.

Assuming that there is no wind, and that $u_h = u_{h,c}$, $v_h = v_{h,c}$, $w_h = w_{h,c}$, and $\dot{\psi} = \dot{\psi}_c$, Eq. (20) reduces to

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{h} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} u_{h,c} \cos \psi - v_{h,c} \sin \psi \\ u_{h,c} \sin \psi + v_{h,c} \cos \psi \\ -w_{h,c} \\ \dot{\psi}_c \end{pmatrix}. \quad (23)$$

We note that when $v_{h,c} = 0$, Eq. (23) is equivalent to a commonly used kinematic model of fixed-wing aircraft in coordinated flight [7, Ch. 9.2]. However, the model we use here is also applicable at low airspeeds and allows for lateral translational maneuvers which may be needed in some operations.

C. RSLQR Inner-loop Design and Performance Model

Here we provide an overview of the inner-loop state-feedback RSLQR design used in this study. A general description of the RSLQR design process can be found in, e.g., [8]. This control law is not meant to represent a complete or optimal design—many design choices have been made for simplicity, rather than performance, and the control system is not finely tuned. Rather, our use of a basic controller allows us to demonstrate the efficacy of the proposed estimation and guidance method. Indeed, we contend that working with a roughly designed inner-loop controller actually better serves to highlight the efficacy of the outer-loop algorithms.

Assume that the 6DoF dynamics are trimmed in wings-level flight with airspeed $\bar{u}_h$, pitch angle $\bar{\theta}$, and control inputs $\bar{\mathbf{u}}$. Define $\Delta u_{h,c} := u_{h,c} - \bar{u}_h$, $\Delta \theta := \theta - \bar{\theta}$, and $\Delta \mathbf{u} = \mathbf{u} - \bar{\mathbf{u}}$. Note that other states are zero at wings-level trim. The inner-loop control objective is to track reference command inputs,

$$\mathbf{r} = (\Delta u_{h,c} \ v_{h,c} \ w_{h,c} \ \dot{\psi}_c)^T, \quad (24)$$

[†]These include 8 vertical rotors, a pusher propeller, two ailerons, two elevators, and a rudder.

and the controller state is defined as a subset of the full vehicle state and integrator states for each reference input:

$$\mathbf{z} = \begin{pmatrix} \Delta u_h & v_h & w_h & \phi & \Delta\theta & p & q & r & e_{I,u} & e_{I,v} & e_{I,w} & e_{I,\dot{\psi}} \end{pmatrix}^T \in \mathbb{R}^{12}. \quad (25)$$

The integrator states evolve according to

$$\begin{pmatrix} \dot{e}_{I,u} \\ \dot{e}_{I,v} \\ \dot{e}_{I,w} \\ \dot{e}_{I,\dot{\psi}} \end{pmatrix} = \begin{pmatrix} u_h - u_{h,c} \\ v_h - v_{h,c} \\ w_h - w_{h,c} \\ \dot{\psi} - \dot{\psi}_c \end{pmatrix}. \quad (26)$$

Now let $\mathbf{A}_{ol} \in \mathbb{R}^{8 \times 8}$ and $\mathbf{B}_{ol} \in \mathbb{R}^{8 \times 14}$ be the open-loop state and control Jacobian matrices, respectively, and define

$$\mathbf{C}_{ol} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/\cos \bar{\theta} \end{pmatrix} \in \mathbb{R}^{4 \times 8} \quad \Rightarrow \quad \begin{pmatrix} \mathbf{C}_{ol} & \mathbf{0}_{4 \times 4} \end{pmatrix} \mathbf{z} \approx \begin{pmatrix} \Delta u_h \\ v_h \\ w_h \\ \dot{\psi} \end{pmatrix}. \quad (27)$$

Then the linearized dynamics are

$$\dot{\mathbf{z}} = \mathbf{A}_{lqr} \mathbf{z} + \mathbf{B}_{lqr} \Delta \mathbf{u} + \mathbf{B}_r \mathbf{r}, \quad (28)$$

for

$$\mathbf{A}_{lqr} := \begin{pmatrix} \mathbf{A}_{ol} & \mathbf{0}_{8 \times 4} \\ \mathbf{C}_{ol} & \mathbf{0}_{4 \times 4} \end{pmatrix}, \quad \mathbf{B}_{lqr} := \begin{pmatrix} \mathbf{B}_{ol} \\ \mathbf{0}_{4 \times 14} \end{pmatrix}, \quad \mathbf{B}_r := \begin{pmatrix} \mathbf{0}_{8 \times 4} \\ -\mathbf{I}_{4 \times 4} \end{pmatrix}. \quad (29)$$

Notice that for constant $\mathbf{r}$ we have $\dot{\mathbf{z}} = \mathbf{A}_{lqr} \dot{\mathbf{z}} + \mathbf{B}_{lqr} \dot{\mathbf{u}}$. Thus, given cost matrices $\mathbf{Q}_{lqr} \in \mathbb{R}^{12 \times 12}$ and $\mathbf{R}_{lqr} \in \mathbb{R}^{14 \times 14}$, we can compute the LQR feedback gain matrix, $\mathbf{K}_{lqr} \in \mathbb{R}^{14 \times 12}$, and get

$$\Delta \dot{\mathbf{u}} = -\mathbf{K}_{lqr} \dot{\mathbf{z}} \quad \Rightarrow \quad \Delta \mathbf{u} = -\mathbf{K}_{lqr} \mathbf{z} = -\begin{pmatrix} \mathbf{K}_P & \mathbf{K}_I \end{pmatrix} \mathbf{z}, \quad (30)$$

where $\mathbf{K}_P \in \mathbb{R}^{14 \times 8}$, $\mathbf{K}_I \in \mathbb{R}^{14 \times 4}$ are the proportional and integral gains, respectively.

To derive a closed-loop performance model for DDP, we substitute Eq. (30) back into Eq. (28). This yields the ideal inner-loop dynamics

$$\begin{cases} \dot{\mathbf{z}}_{id} = \mathbf{A} \mathbf{z}_{id} + \mathbf{B}_r \mathbf{r} \\ \Delta \mathbf{u}_{id} = \mathbf{C} \mathbf{z}_{id} \end{cases} \quad \mathbf{A} := \begin{pmatrix} \mathbf{A}_{ol} - \mathbf{B}_{ol} \mathbf{K}_P & -\mathbf{B}_{ol} \mathbf{K}_I \\ \mathbf{C}_{ol} & \mathbf{0}_{4 \times 4} \end{pmatrix}, \quad \mathbf{C} := \begin{pmatrix} \mathbf{C}_{ol} & \mathbf{0}_{4 \times 4} \end{pmatrix}. \quad (31)$$

One can verify that the ideal closed-loop system (Eq. (31)) has unity DC gain, meaning $\Delta \mathbf{u}_{id} \rightarrow \mathbf{r}$ for constant reference inputs. These dynamics are valid when $\mathbf{z}$ is small and when $\mathbf{A}_{ol}$ and $\mathbf{B}_{ol}$ accurately approximate the open-loop Jacobian matrices. When the aircraft is far from trim, control effectors saturate or fail, or the aircraft experiences other major disturbances, Eq. (31) no longer accurately models the aircraft's closed-loop response to reference commands. Therefore we define

$$\dot{\mathbf{z}} = \left(\mathbf{A} + \hat{\mathbf{A}}_{\Delta} \right) \mathbf{z} + \mathbf{B}_r \mathbf{r} + \boldsymbol{\sigma}, \quad \hat{\mathbf{u}}_g = \mathbf{C} \mathbf{z} + \bar{\mathbf{u}}, \quad \mathbf{z}(0) = \mathbf{z}_0. \quad (32)$$

Combining Eq. (32) with the guidance model (Eq. (23)) gives us a closed-loop performance model in the form of Eq. (19). Note that we do not include a gain multiplier, $\hat{\omega}$, because the $\mathbf{B}_r \mathbf{r}$ term in Eq. (32) only affects the RSLQR integrator dynamics, which are exactly known. More complicated inner-loop control systems could still benefit from inclusion of a $\hat{\omega}$ parameter.

D. Lateral Velocity Command Shaping for Turn Coordination

Unfortunately, regulating heading-frame lateral velocity does not enforce turn coordination, which is desirable at higher airspeeds. To this end, we introduce command shaping for the lateral heading velocity command, $v_{h,c}$, such that body-frame lateral velocity, v_b , goes to zero.

To derive the modified command, we recall that v_b satisfies

$$v_b = u_h \sin \phi \sin \theta + v_h \cos \phi + w_h \sin \phi \cos \theta. \quad (33)$$

Setting $v_b = 0$ and solving for v_h gives

$$v_{h,v_b=0} = -\tan \phi (u_h \sin \theta + w_h \cos \theta). \quad (34)$$

We set the bank angle ϕ to satisfy the coordinated turn equation [7, Chs. 5, 9],

$$\tan \phi = \frac{V}{g} \dot{\psi}, \quad V = \text{sign}(u_h) \|\mathbf{v}_h\|_2, \quad (35)$$

and assume that $u_h = u_{h,c}$, $w_h = w_{h,c}$, and $\dot{\psi} = \dot{\psi}_c$. Then the reference command passed to the controller is modified to

$$v'_{h,c}(\mathbf{r}, \theta) := v_{h,c} + v_{h,v_b=0} = v_{h,c} - \frac{V_c}{g} \dot{\psi}_c (u_{h,c} \sin \theta + w_{h,c} \cos \theta), \quad (36)$$

with $V_c \approx \text{sign}(u_{h,c}) \|\mathbf{v}_{h,c}\|_2$. Hence the reference command passed to the inner-loop RSLQR and controller performance model is modified to

$$\mathbf{r}'(\mathbf{r}, \theta) := \begin{pmatrix} \Delta u_{h,c} & v'_{h,c} & w_{h,c} & \dot{\psi}_c \end{pmatrix}^T. \quad (37)$$

By construction, when $v_{h,c} = 0$, this command leads to $v_b \rightarrow 0$ in steady state. Meanwhile, if $\dot{\psi}_c = 0$ it is clear that $v_h \rightarrow v'_{h,c} = v_{h,c}$ in steady state, including for non-zero $v_{h,c}$. Unfortunately, non-zero $v_{h,c}$ will not be achieved when we also command $\dot{\psi}_c \neq 0$. However, this is deemed acceptable since at higher airspeeds we will usually only command $\dot{\psi}_c$ (analogously to commanding bank for fixed wing aircraft).

E. Simulation Results

In this section, we present sample simulation results to provide a sense of the framework's potential and the importance of its various components. To focus on the main ideas, we limit the scope of the study to a single maneuver in transition flight ($\bar{u}_h = 100$ ft/s, all rotors turned on). We repeat the maneuver twice for each model configuration: once with nominal conditions, and once where we fail rotors 5 and 6 (setting their output to 0 RPM); these are the two inboard engines on the right wing (see [1]). In each simulation, the true vehicle state evolves according to the full nonlinear 6DoF dynamics with RSLQR to track guidance commands provided by DDP.

The maneuver selected for this study is an aggressive lateral course change and recovery. This can be thought of as a collision avoidance maneuver provided by a higher level path planner (see e.g., [19]). Specifically, we ask for a $\psi_d = 60^\circ$ (right) heading for 20 seconds, then $\psi_d = -60^\circ$ (left) for 20 seconds, followed by $\psi_d = 0^\circ$ to return to the original trajectory. This is depicted in Fig. 5. Importantly, we specify the maneuver using only waypoints and straight line segments, so without DDP guidance, it is not dynamically feasible nor directly trackable by the RSLQR inner loop. The DDP MPC problem is formulated as a position-tracking problem with a simple quadratic cost metric:

$$\left\{ \begin{array}{l} \min_{\mathbf{r}(\cdot)} \int_{t_0}^{t_0+T_{\text{mpc}}} \left[\left(\frac{p_n(t) - p_{n,d}(t)}{100} \right)^2 + \left(\frac{p_e(t) - p_{e,d}(t)}{100} \right)^2 + \left(\frac{h(t) - h_d(t)}{50} \right)^2 + \left(\frac{\psi(t) - \psi_d(t)}{15^\circ} \right)^2 \right. \\ \quad \left. + \left(\frac{u_{h,c}(t) - \bar{u}_h}{5} \right)^2 + (10v_{h,c}(t))^2 + \left(\frac{w_{h,c}(t)}{5} \right)^2 + \left(\frac{\dot{\psi}_c(t)}{12^\circ} \right)^2 \right] dt, \\ \text{s.t. } \dot{\mathbf{x}}_m = \mathbf{f}_m(\mathbf{x}_m, \mathbf{r}; \boldsymbol{\theta}_m) \\ \quad \mathbf{x}_m(t_0) = \mathbf{x}_{m,0} \\ \quad 20 \leq u_{h,c} \leq 160 \\ \quad -10 \leq v_{h,c} \leq 10 \\ \quad -10 \leq w_{h,c} \leq 10 \\ \quad -12^\circ/\text{s} \leq \dot{\psi}_c \leq 12^\circ/\text{s} \end{array} \right. \quad (38)$$

We set the control horizon to $T_{\text{mpc}} = 10$ s. DDP is applied to the discretized problem with a timestep of $\Delta t = 0.05$ s. The guidance commands are updated at a rate of 4 Hz with a *single* DDP iteration. The inner-loop RSLQR tracks the first 0.25s of the commanded trajectory until new commands are computed, with inner-loop controls applied at 100 Hz. Actuator dynamics are ignored in this simulation.

Several configurations of guidance model states and dynamics, $\mathbf{x}_m$ and $\mathbf{f}_m(\cdot)$, are defined as follows:

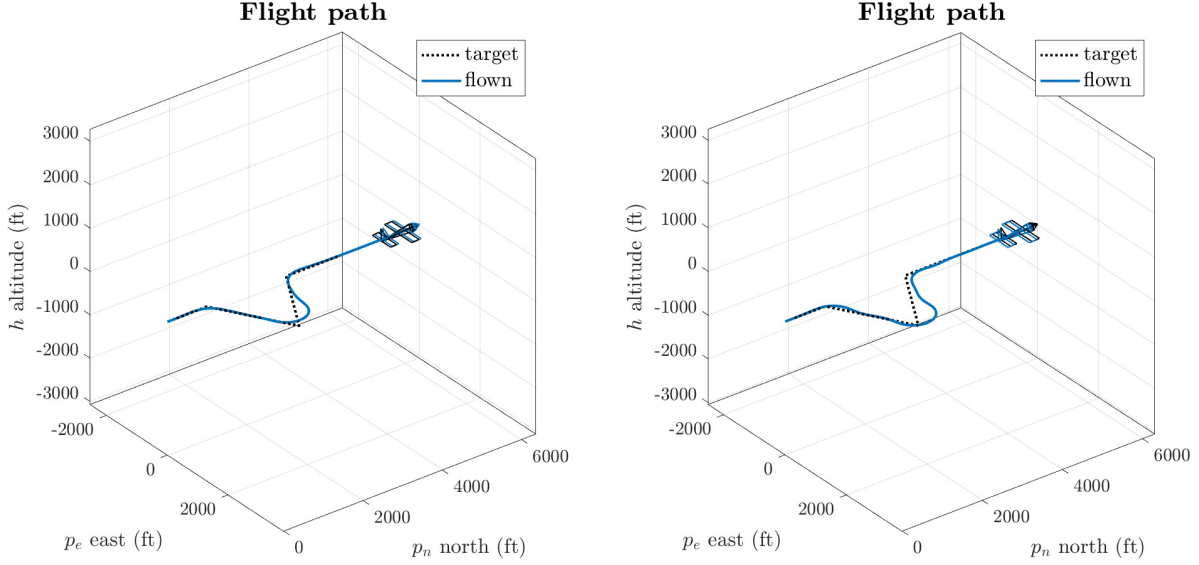


Fig. 5 Collision avoidance maneuver in transition. Left: adaptive guidance based on kinematic guidance model together with the RSLQR performance model with online parameter estimation. Right: guidance based on the kinematic guidance model only.

- 6DoF + RSLQR: The full nonlinear 6DoF L+C dynamics (Eq. (20)) with RSLQR and command shaping (Eqs. (26), (30) and (37)). $\mathbf{x}_m = \begin{pmatrix} \mathbf{p}^T & \mathbf{v}_h^T & \boldsymbol{\eta}^T & \boldsymbol{\omega}^T & \mathbf{e}_I^T \end{pmatrix}^T \in \mathbb{R}^{16}$.
 - This configuration performs MHE to estimate parameters which multiply effector inputs, allowing DDP to recognize precisely which effector fails.
- Kinematic + RSLQR: Combined kinematic guidance and RSLQR performance models (Eqs. (23) and (32)), including command shaping (Eq. (37)). $\mathbf{x}_m = \begin{pmatrix} \mathbf{x}_g^T & \mathbf{z}^T \end{pmatrix}^T \in \mathbb{R}^{16}$.
 - $\hat{\mathbf{A}}_\Delta$ estimation on/off. If on, estimate model parameters $\hat{\mathbf{A}}_\Delta$. If off, fix $\hat{\mathbf{A}}_\Delta \equiv \mathbf{0}$.
 - $\boldsymbol{\sigma}$ estimation on/off. If on, estimate additive uncertainty $\boldsymbol{\sigma}$ for the inner loop *and* the kinematic model. If off, fix $\boldsymbol{\sigma} \equiv \mathbf{0}$.
- Kinematic only: Only the kinematic guidance model (Eq. (23)). $\mathbf{x}_m = \mathbf{x}_g \in \mathbb{R}^4$.
 - $\boldsymbol{\sigma}$ estimation on/off. If on, estimate additive uncertainty $\boldsymbol{\sigma}$. If off, fix $\boldsymbol{\sigma} \equiv \mathbf{0}$.

MHE cost functions are constructed in the form of Eq. (7), with weights selected based on performance in nominal conditions and in single-rotor-failure cases. Cost functions are made as similar as possible across guidance model configurations.

Total cost (evaluated over the 80s simulation timespan) and CPU clock time for each configuration are reported in Tables 1 and 2. Plots of selected simulations for nominal conditions (no rotor failures) are presented in Figs. 6 to 10; selected simulations of failure cases are shown in Figs. 11 to 15.

Detail-oriented readers may be skeptical of some high-frequency oscillations in the heading rate commands, $\dot{\psi}_c$. These can be eliminated by increasing the guidance update rate, further tuning the MPC and MHE problem formulations, and/or inserting a filter in the command channel. They were ignored for this preliminary study, as we did not observe a significant impact on the flown trajectories.

1. Nominal Case

We first discuss the nominal (no rotor failure) simulations (see Table 1). We find that, under ideal conditions, the inner-loop RSLQR performs robustly even during aggressive maneuvers like this, enabling excellent position-tracking as seen in the baseline 6DoF simulation (see Fig. 6).

In this case, the kinematic + RSLQR model enabled virtually identical performance to the full 6DoF + RSLQR

model	est. $\hat{\mathbf{A}}_{\Delta}$	est. σ	MPC $\frac{\text{CPU time}}{\text{DDP iter.}}$	MHE $\frac{\text{CPU time}}{\text{DDP iter.}}$	total cost
6DoF + RSLQR	-	-	1.847s	0.038s	220
kinematic + RSLQR	n	n	0.022s	-	281
kinematic + RSLQR	n	y	0.024s	0.009s	231
kinematic + RSLQR	y	n	0.022s	0.009s	253
kinematic + RSLQR	y	y	0.023s	0.010s	212
kinematic only	-	n	0.009s	-	530
kinematic only	-	y	0.009s	0.001s	605

Table 1 Performance and computational cost of various configurations for a collision avoidance maneuver in transition. Total cost (evaluated over the 80s simulation timespan) provides a high-level evaluation of the configuration’s position-tracking performance.

model	est. $\hat{\mathbf{A}}_{\Delta}$	est. σ	MPC $\frac{\text{CPU time}}{\text{DDP iter.}}$ (s)	MHE $\frac{\text{CPU time}}{\text{DDP iter.}}$ (s)	total cost
6DoF + RSLQR	-	-	2.852s	0.044s	232
kinematic + RSLQR	n	n	0.022s	-	303
kinematic + RSLQR	n	y	0.023s	0.009s	267
kinematic + RSLQR	y	n	0.022s	0.009s	287
kinematic + RSLQR	y	y	0.024s	0.010s	253
kinematic only	-	n	0.008s	-	891
kinematic only	-	y	0.009s	0.001s	∞

Table 2 Performance and computational cost of various configurations for a collision avoidance maneuver in transition with failure of rotors 5 and 6. Infinite total cost represents a loss of control (see Fig. 14).

model, as evidenced by the similar trajectories and total cost metrics (compare Fig. 6 to Figs. 7 and 8). Furthermore, the adaptive guidance approach offered better performance than the static guidance model, as online parameter estimates improved the linear performance model during the aggressive turns. Best performance was obtained by estimating both $\hat{\mathbf{A}}_{\Delta}$ and σ , though some benefits were observed from each individually.

Meanwhile, the lowest-fidelity kinematic-only model led to significantly reduced performance (total cost 530 compared to 220 for the 6DoF + RSLQR model). To understand why, we consider Fig. 9, in which we immediately see large oscillations in the heading rate and angle. This is because DDP computes highly aggressive guidance commands, since the kinematic-only model assumes instantaneous command responses. Specifically, DDP waits to initiate the first turn as long as possible, then commands the maximum allowable heading rate. Unaware of the lag in the response, it turns around too late, and this oscillatory pattern repeats throughout the maneuver. The kinematic-only model does perform adequately for more benign maneuvers, but we intended this example to push the limits of the guidance and control system.

Surprisingly, combining the kinematic-only model with additive uncertainty estimation led to worse oscillations. Furthermore, we encountered considerable difficulties in tuning this configuration: many alternative cost parameters, guidance update rates, and other numerical tricks were tested, but none led to substantially better outcomes. In the rotor failure case, this lack of robustness led to instability.

2. Rotor Failure Case

We now turn to the multiple rotor failure simulations (see Table 2). On its own, the RSLQR designed for this study does not fully compensate for multiple failures, resulting in degraded command-tracking performance and stability. As expected, this resulted in increased total cost for each configuration compared to the nominal case.

First, comparing Figs. 9 and 14, large, lightly-damped oscillations clearly illustrate the degraded stability of the kinematic-only configuration in the presence of rotor failures. Stability of the kinematic model with additive uncertainty was degraded to the extent that the aircraft entered a sustained Dutch roll (see Fig. 15).

In contrast, both the 6DoF + RSLQR and kinematic + RSLQR configurations were largely able to recover their

nominal performance. The adaptive guidance configurations, especially the model which estimated both $\hat{\mathbf{A}}_\Delta$ and σ , continued to outperform the static inner-loop model. This suggests that knowledge of the inner-loop dynamics, especially when updated online, is critical for successful use of reduced-order guidance models at the edges of the flight envelope and/or in the presence of major disturbances. More benign maneuvers and disturbances may not require the inner-loop model.

These simulations illustrate the usefulness of adaptive guidance in scenarios where inner-loop dynamics differ significantly from design conditions. Furthermore, the proposed linear performance model appears to be sufficient for characterizing the RSLQR-in-the-loop dynamics in this context, though more work will need to be done to evaluate the model systematically.

3. Computational Efficiency

Simulations were implemented in MATLAB R2024a and run on a 2023 MacBook Pro with M2 Pro chip and 16GB RAM. Efforts were made to write efficient code, but the implementation should not be considered optimized. Therefore, reported CPU clock times give a general idea of the computational feasibility and relative efficiency of each guidance model.

In preliminary testing, it was found that a sufficiently long time horizon ($T_{\text{mpc}} = 10\text{s}$) was required for good performance. Shorter time horizons often resulted in commands which were too aggressive, while longer horizons caused convergence difficulties for DDP. In instances where MPC-DDP iterations struggled to converge, we warm-started the computation by running DDP on the MPC problem with the reduced kinematic guidance model.

For the 6DoF + RSLQR dynamics, each iteration of DDP for the MPC problem took about 2 or 3 seconds. In the simulation, guidance was updated at 4 Hz, hence this model would not be feasible for real-time implementation without improvements. Meanwhile, the kinematic + RSLQR model took about 0.03s per DDP iteration. This is about 100 times faster than the full 6DoF model, and sufficiently fast for the simulated 4 Hz guidance update rate.

V. Conclusions and Future Work

In this paper, we have explored the interaction between inner-loop feedback control laws with outer-loop guidance algorithms based on optimal control and estimation. In order to limit complexity, it is common practice to use reduced-order models in the guidance loop. With this in mind, we introduced adaptive performance models with linear sub-components to represent the tracking performance of the inner-loop controller. This is a natural formulation for adaptive flight control systems which track a linear reference model.

Focusing on the L+C concept aircraft, we designed a RSLQR to track commands in the heading frame, and formed the accompanying closed-loop performance model. We wrapped an adaptive guidance loop around this, using DDP to solve the MPC and MHE problems with various guidance models. We compared the robust tracking performance of these configurations with an illustrative aggressive position tracking problem in the presence of multiple rotor failures.

In this study, the proposed kinematic + RSLQR guidance model recovered the performance and robustness of the full 6DoF + RSLQR model, at a small fraction of the computational cost. The adaptive guidance model further improved performance. This is a promising indication for potential future implementation in real-time guidance systems. On the other hand, using the kinematic model alone (without knowledge of the inner-loop dynamics) resulted in poor stability and performance during the aggressive test maneuver and especially when effector failures were introduced.

It is worth emphasizing that the proposed algorithm is not a panacea. Specifically, if the inner-loop controller cannot possibly stabilize the aircraft, this adaptive guidance algorithm will not stabilize it either. Rather, it attempts to provide guidance commands which account for underlying dynamics, and modify the performance model as necessary to match observed responses.

With this in mind, there are a number of areas in which the algorithm might be improved. First, tuning cost function weights and other MHE problem parameters is not always straightforward, and some time must be invested to obtain robust performance. An important line of future work is to investigate efficient strategies for formulating the MHE problem appropriately, and/or modifying the structure of the inner-loop performance model.

Next, using constant estimates of σ for planning may not be effective for long time-horizon trajectory optimization. For this reason, we would like to explore alternative formulations with time and/or state dependence. That is, $\sigma \in \mathbb{R}^{n_z}$ is replaced by a term like $\beta\sigma(t, \mathbf{z})$, where $\sigma : [0, \infty) \times \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_\sigma}$ is now a vector of time- and state-dependent basis functions and $\beta \in \mathbb{R}^{n_x \times n_\sigma}$ is a matrix of coefficients. In addition, we would like to consider avenues for analytical mathematical understanding of the approximation properties and limitations of the proposed inner-loop performance models described in Section III.

Finally, based on promising initial results with a basic inner-loop control law, we are interested in combining the adaptive guidance algorithm with a more advanced controller, such as an $\mathcal{L}_1$ adaptive augmentation of a RSLQR [15]. We expect that such a control law could stabilize the aircraft at faster timescales than the guidance system can possibly operate, and thus their combination could improve performance in the context of a much wider range of uncertainties.

Appendix

A. Simulation Plots: Nominal Case

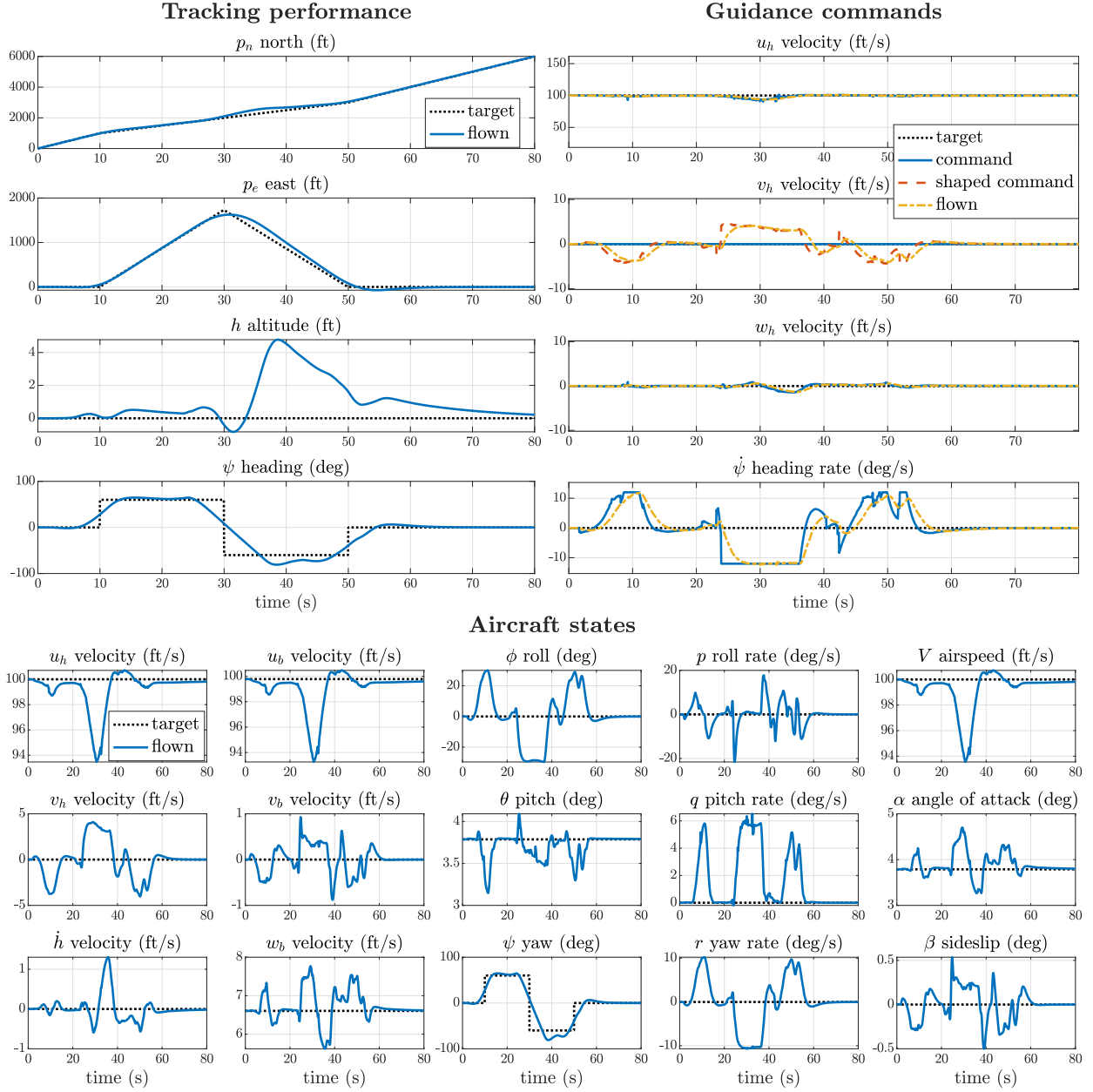


Fig. 6 Collision avoidance maneuver in transition. DDP guidance is based on the full 6DoF + RSLQR model. This simulation represents a best-case performance baseline in nominal conditions.

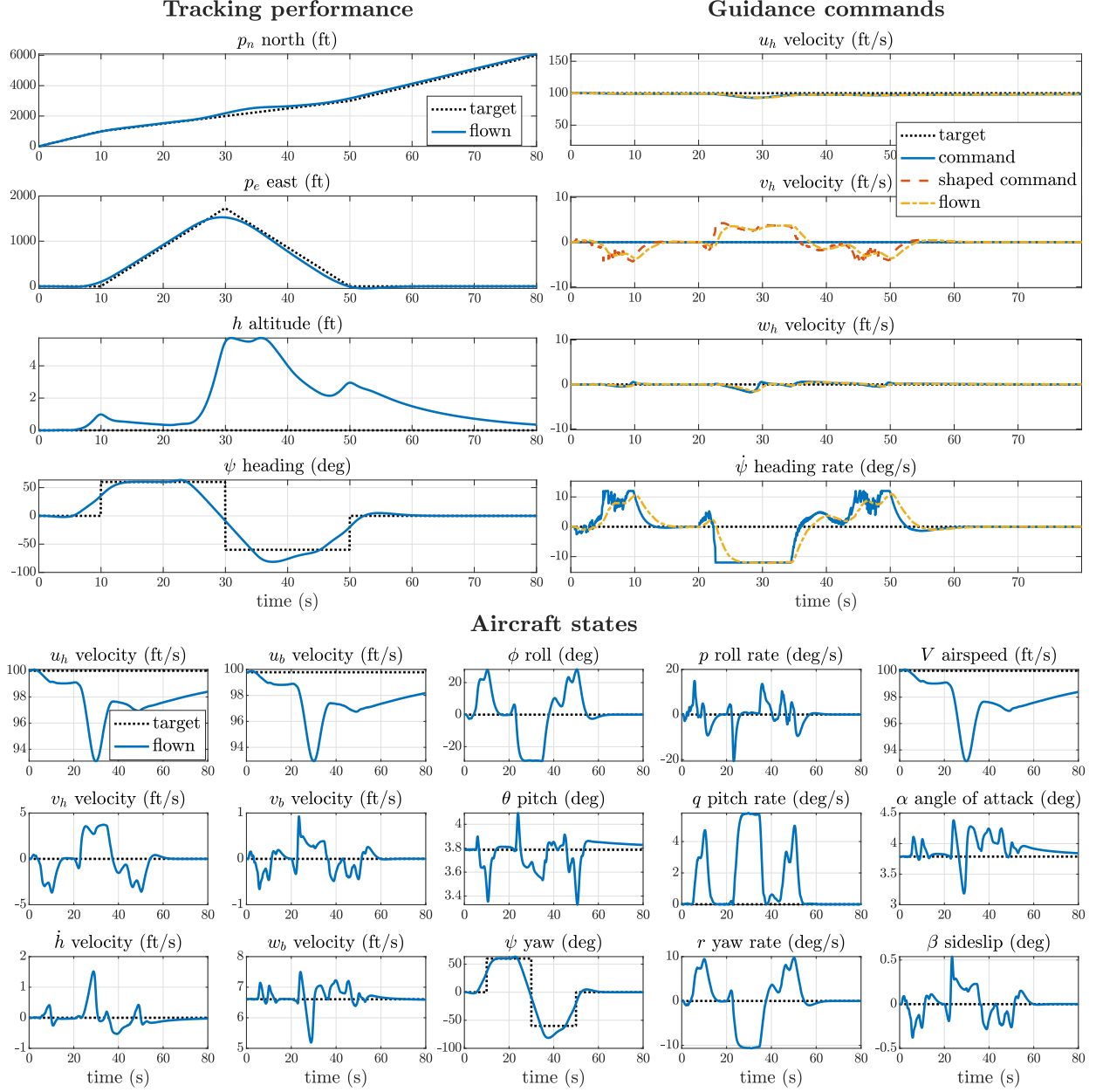


Fig. 7 Collision avoidance maneuver in transition. DDP guidance is based on the kinematic + RSLQR performance model. Note the similarity to the 6DoF baseline (Fig. 6).

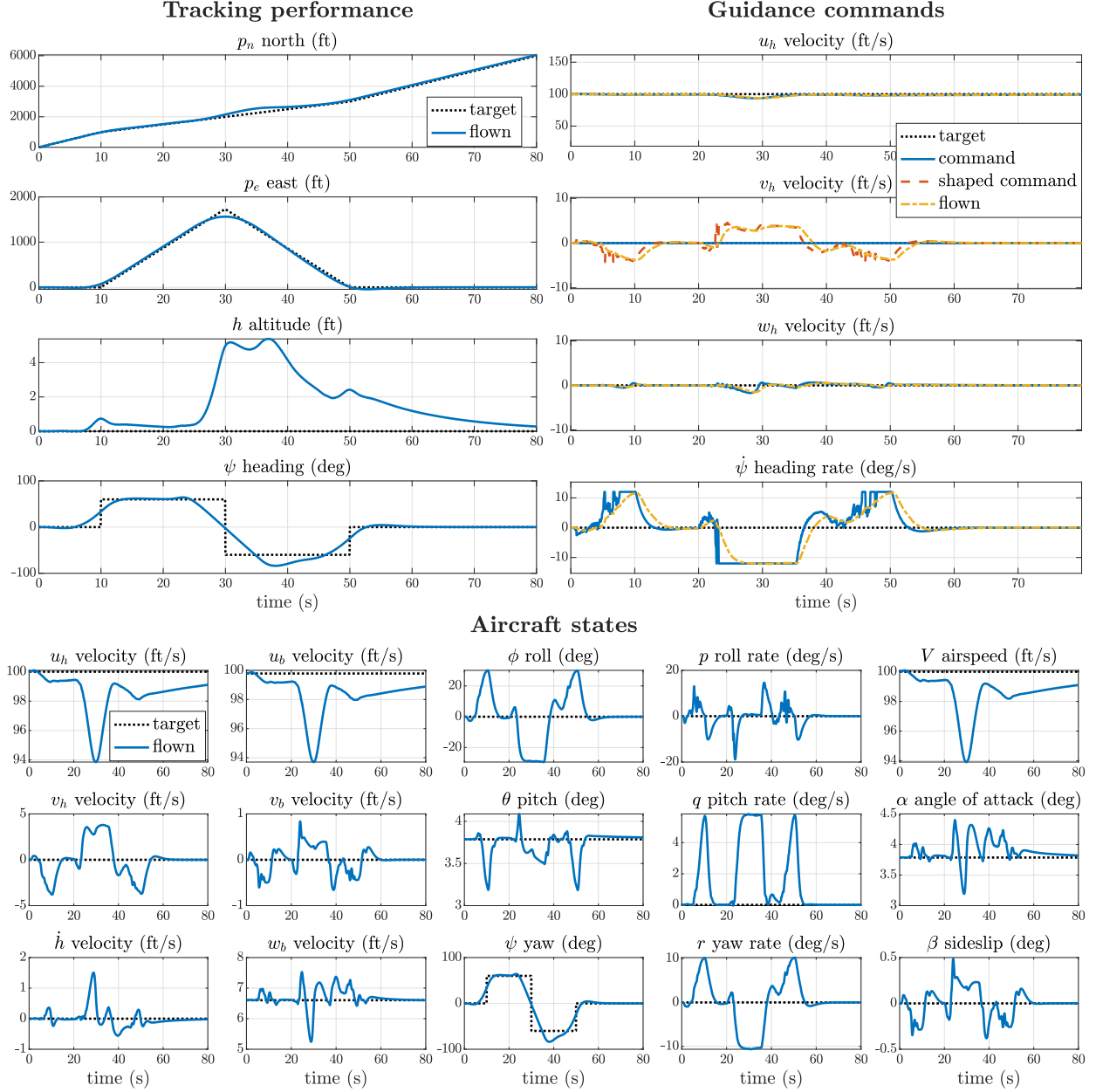


Fig. 8 Collision avoidance maneuver in transition. DDP guidance is based on the kinematic + RSLQR performance model with estimation of model parameters $\hat{\mathbf{A}}_\Delta$ and σ . Note the similarity to the 6DoF baseline (Fig. 6), as well as modestly-improved position tracking compared to non-adaptive configuration (Fig. 7).

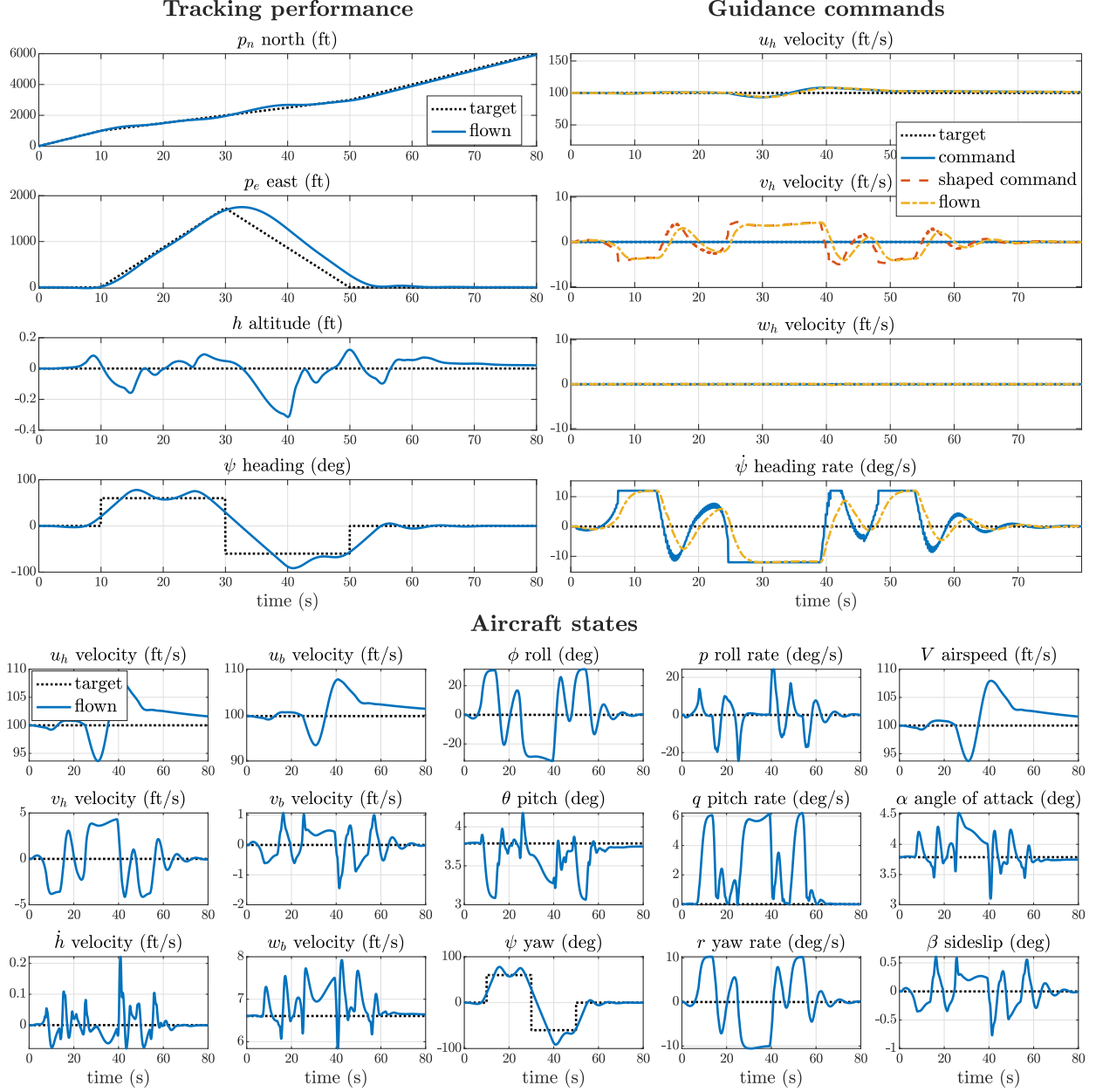


Fig. 9 Collision avoidance maneuver in transition. DDP guidance is based on the kinematic model only. Note the large oscillations in heading and heading rate, compared to the 6DoF baseline (Fig. 6).

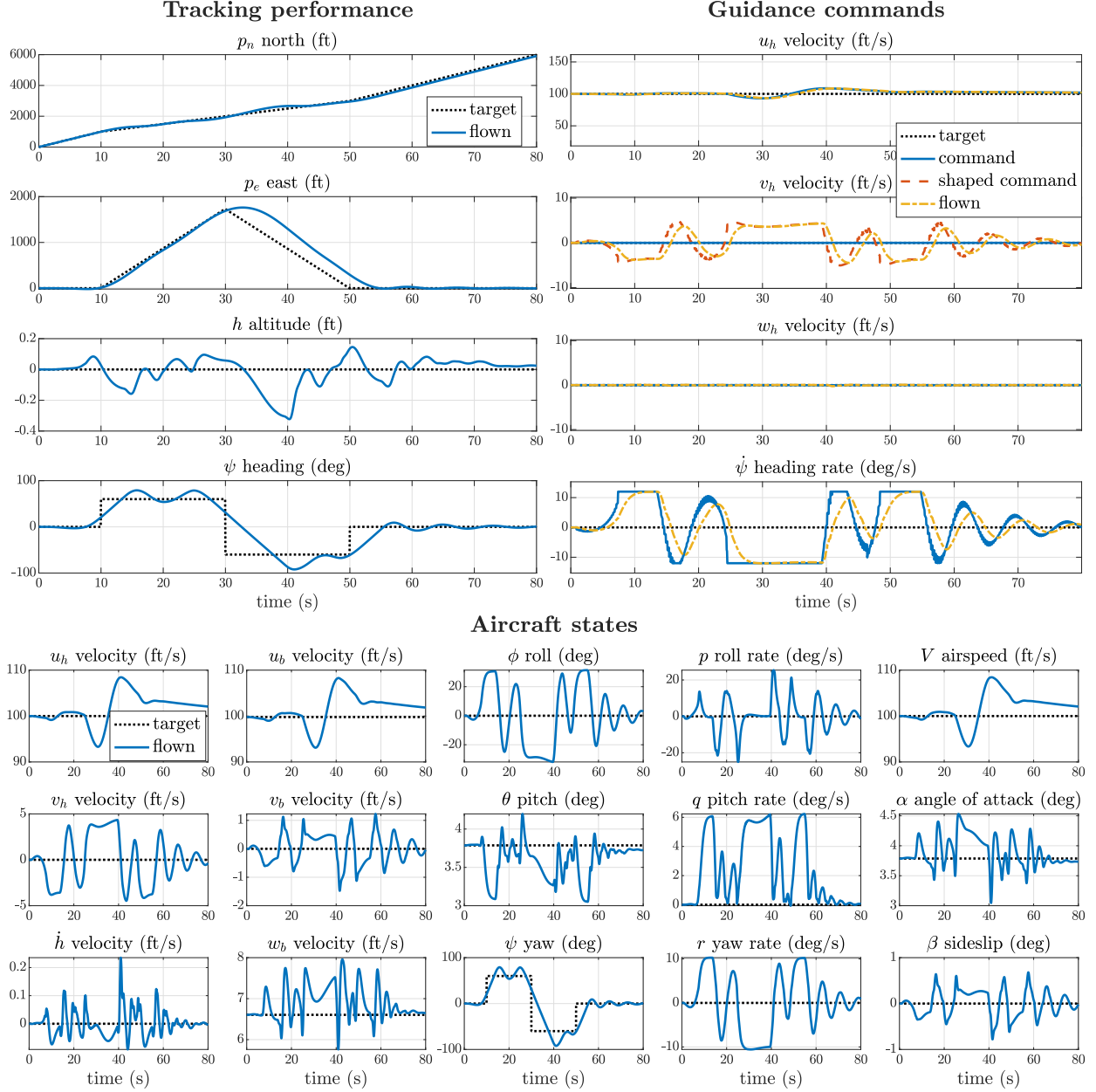


Fig. 10 Collision avoidance maneuver in transition. DDP guidance is based on the kinematic model only, together with additive uncertainty (σ) estimation. Note the large, lightly-damped oscillations in heading and heading rate.

B. Simulation Plots: Rotor Failure Case

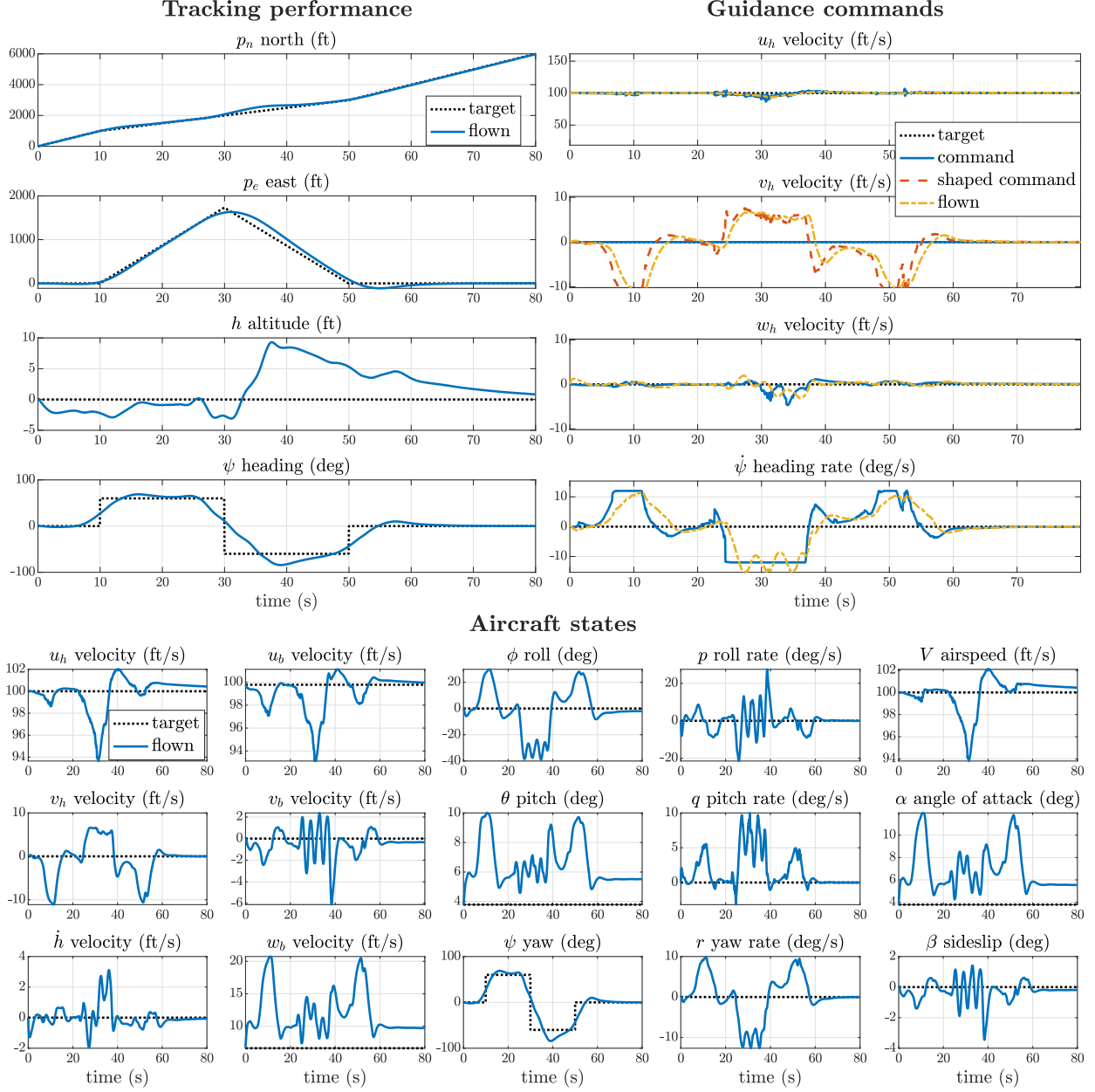


Fig. 11 Collision avoidance maneuver in transition with rotors 5 and 6 failed. DDP guidance is based on the full 6DoF + RSLQR model. With complete dynamics information, DDP is largely able to identify the failed rotors and modify guidance commands to follow the desired flight path with only modest deviation from the nominal case (Fig. 6). This simulation represents a best-case performance baseline in this failure scenario.

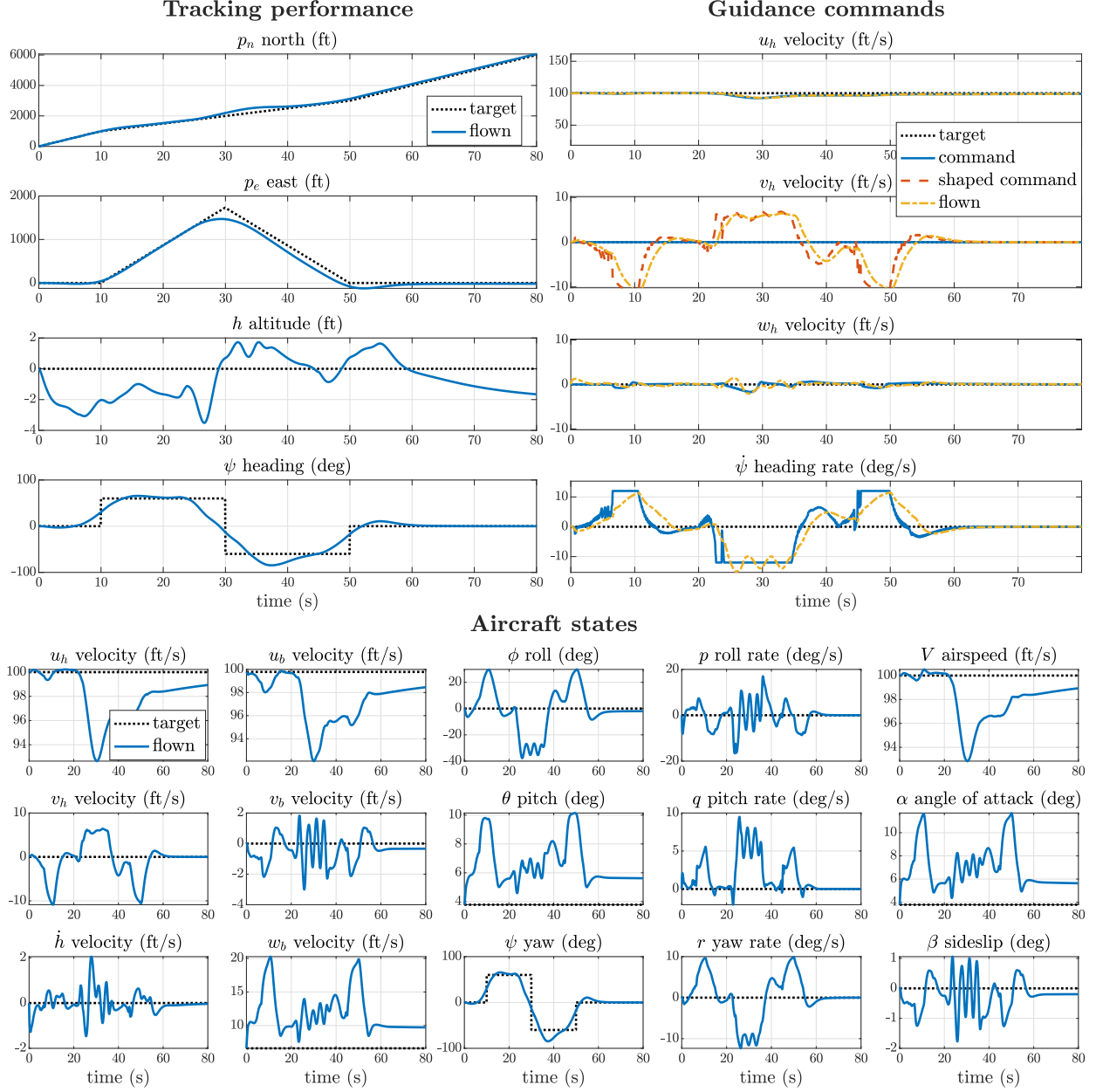


Fig. 12 Collision avoidance maneuver in transition with rotors 5 and 6 failed. DDP guidance is based on the kinematic + RSLQR performance model. Note the similarity to the 6DoF baseline (Fig. 11).

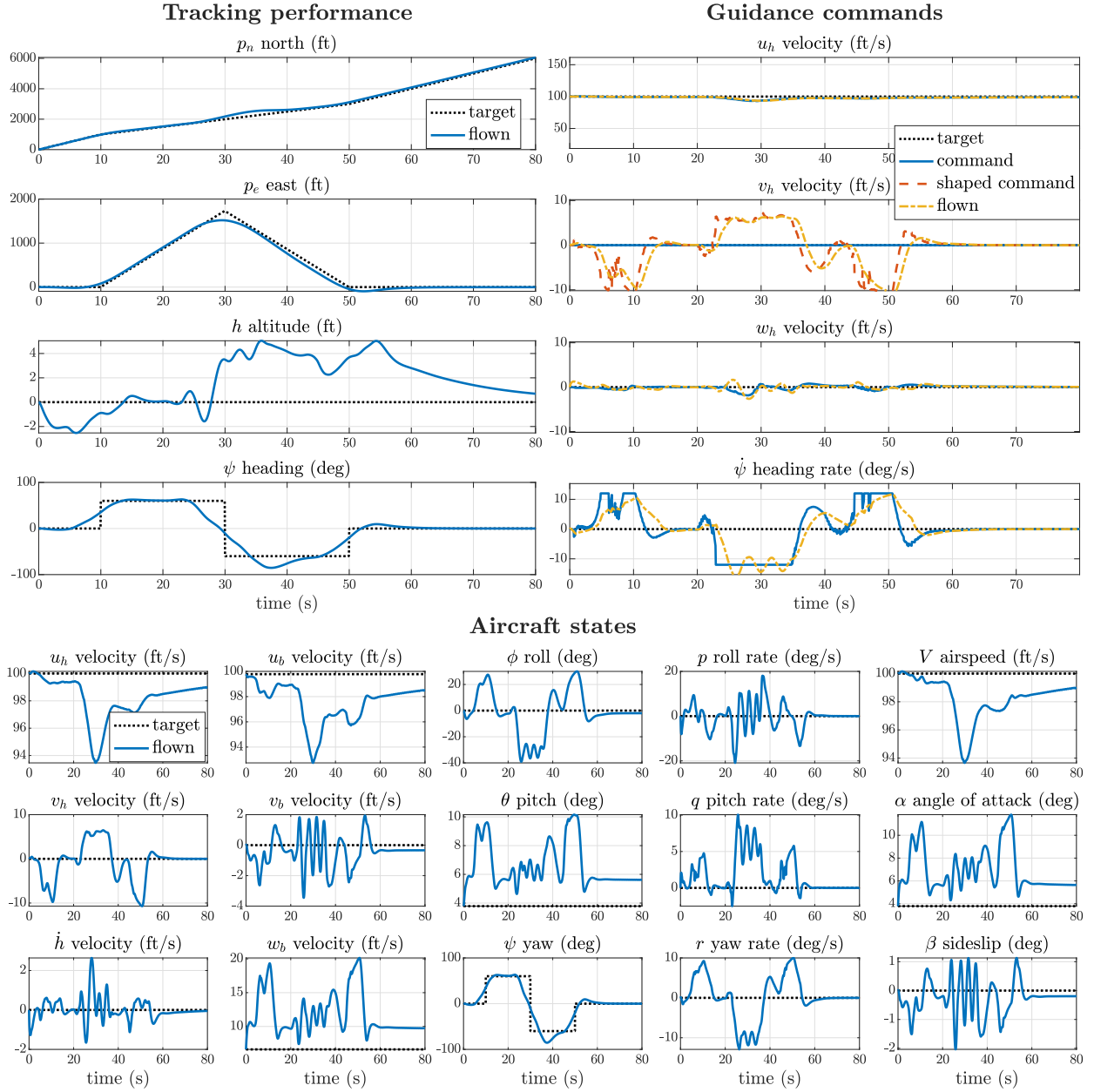


Fig. 13 Collision avoidance maneuver in transition with rotors 5 and 6 failed. DDP guidance is based on the kinematic + RSLQR performance model with estimation of model parameters $\hat{\mathbf{A}}_\Delta$ and σ . Note the similarity to the 6DoF baseline (Fig. 11), as well as modestly-improved position tracking compared to non-adaptive configuration (Fig. 12).

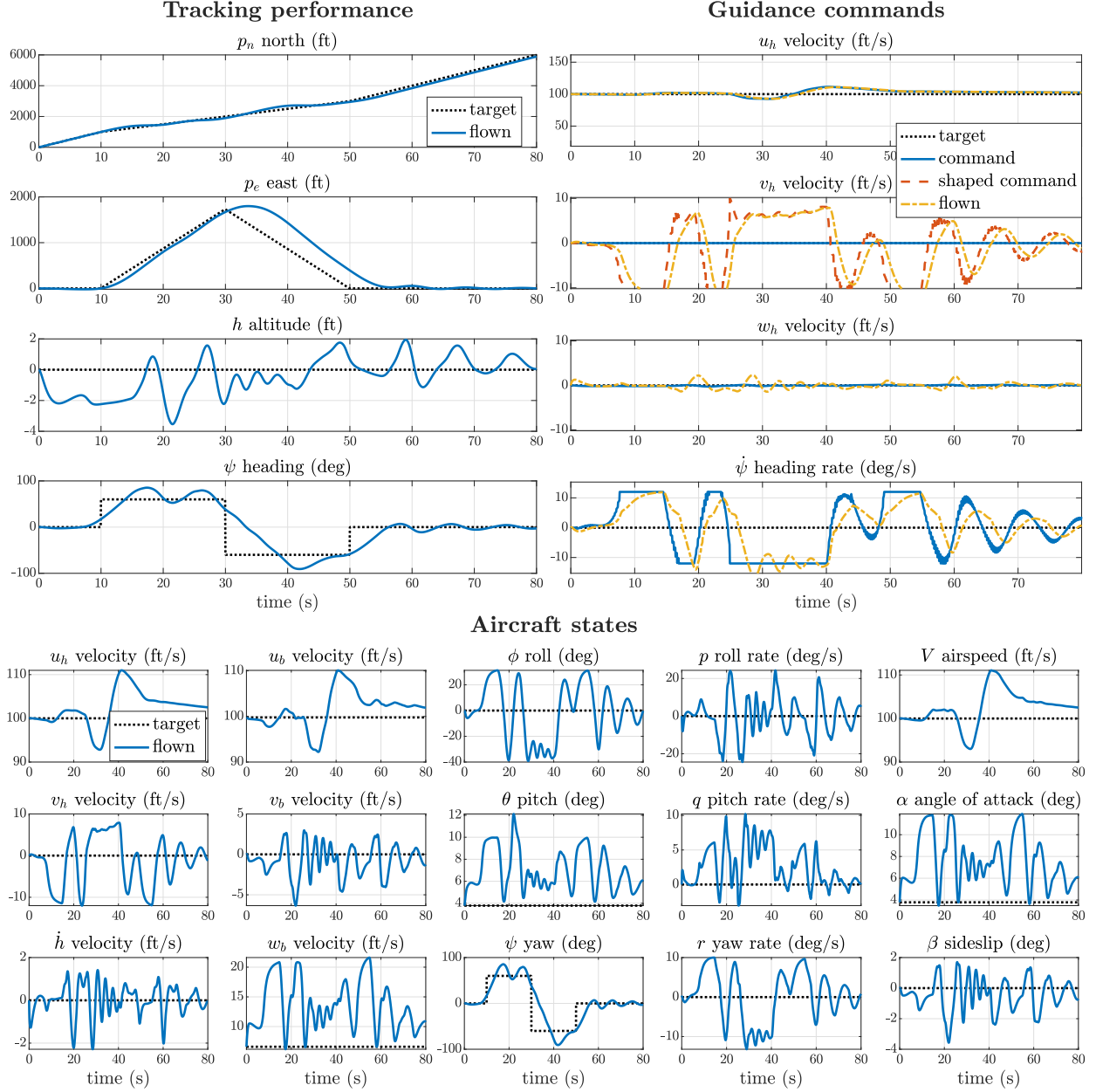


Fig. 14 Collision avoidance maneuver in transition with rotors 5 and 6 failed. DDP guidance is based on the kinematic model only. Note the large oscillations in heading and heading rate, compared to the 6DoF baseline (Fig. 11) and the nominal case (Fig. 9).

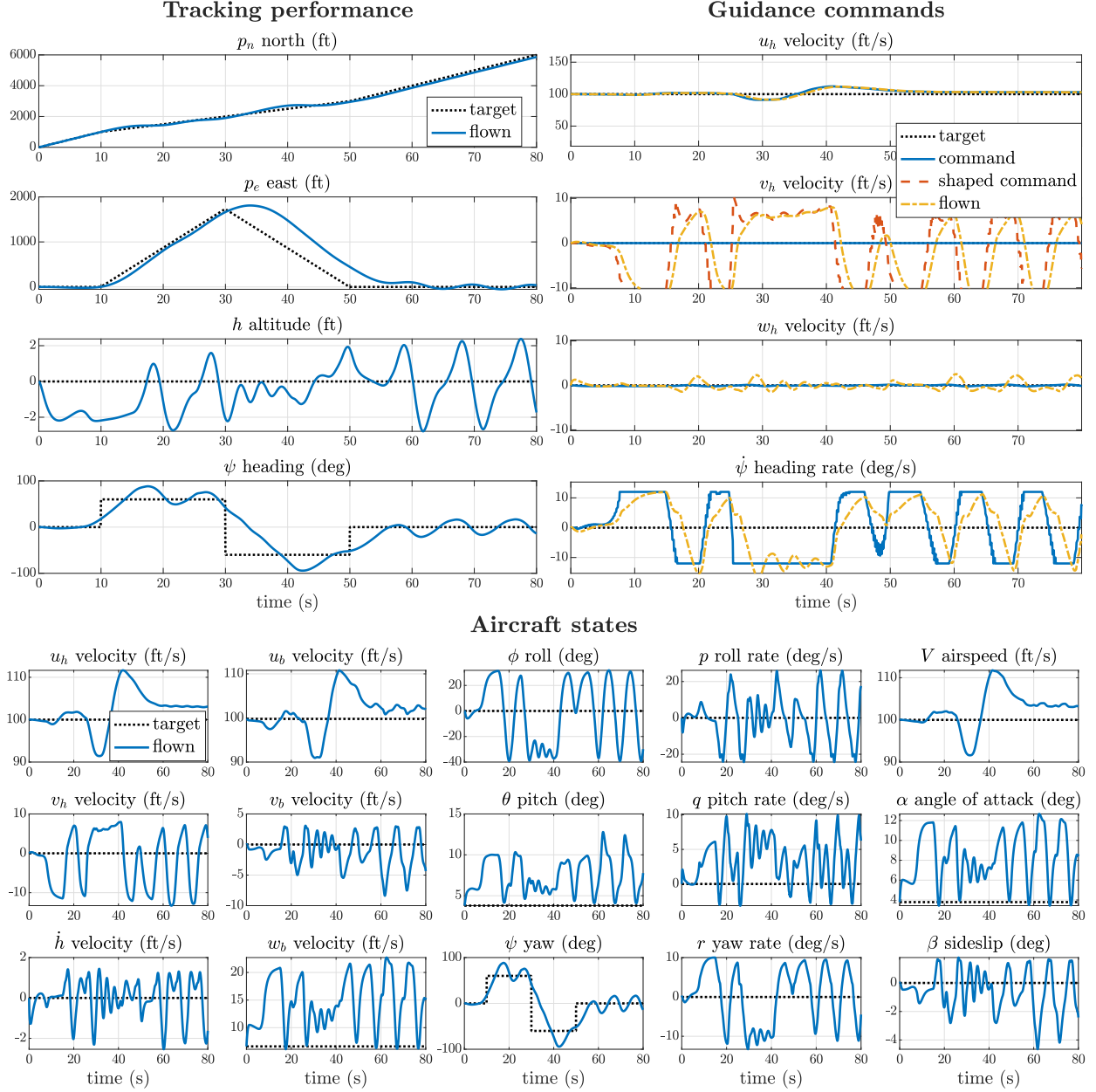


Fig. 15 Collision avoidance maneuver in transition with rotors 5 and 6 failed. DDP guidance is based on the kinematic model only, together with additive uncertainty (σ) estimation. Observe that the poor robustness characteristics of this simple guidance model configuration led to a persistent Dutch roll.

Acknowledgments

The authors would like to thank Matthew Houghton for his work in implementing earlier versions of the DDP code, and for consulting on its use and areas for improvement. We would also like to thank Steven Snyder and Kasey Ackerman for their helpful review of the manuscript.

This research was supported by the NASA Aeronautics Research Mission Directorate (ARMD), Transformative Tools and Technologies (TTT) project, under the Autonomous Systems - Intelligent Contingency Management subproject.

References

- [1] Silva, C., Johnson, W. R., Solis, E., Patterson, M. D., and Antcliff, K. R., “VTOL Urban Air Mobility Concept Vehicles for Technology Development,” *AIAA Aviation Technology, Integration, and Operations Conference*, 2018. <https://doi.org/10.2514/6.2018-3847>.
- [2] Gregory, I. M., Neogi, N. A., Grauer, J. A., Campbell, N. H., Holbrook, J., Bacon, B. J., Murphy, P. C., Moerder, D. D., Simmons, B. M., Acheson, M. J., Britton, T. C., and Cook, J., “Intelligent Contingency Management for Urban Air Mobility,” *AIAA SciTech Forum*, 2021. <https://doi.org/10.2514/6.2021-1000>.
- [3] Jacobson, D. H., and Mayne, D. Q., *Differential Dynamic Programming*, Elsevier, New York, NY, 1970.
- [4] Tassa, Y., Mansard, N., and Todorov, E., “Control-Limited Differential Dynamic Programming,” *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1168–1175. <https://doi.org/10.1109/ICRA.2014.6907001>.
- [5] Houghton, M. D., Oshin, A. B., Acheson, M. J., Theodorou, E. A., and Gregory, I. M., “Path Planning: Differential Dynamic Programming and Model Predictive Path Integral Control on VTOL Aircraft,” *AIAA SciTech Forum*, 2022. <https://doi.org/10.2514/6.2022-0624>.
- [6] Oshin, A., Houghton, M., Acheson, M., Gregory, I., and Theodorou, E., “Parameterized Differential Dynamic Programming,” *Robotics: Science and Systems*, 2022. <https://doi.org/10.15607/RSS.2022.XVIII.046>.
- [7] Beard, R. W., and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, Princeton, NJ, 2012.
- [8] Lavretsky, E., and Wise, K. A., *Robust and Adaptive Control with Aerospace Applications*, 2nd ed., Advanced Textbooks in Control and Signal Processing, Springer, Cham, Switzerland, 2024. <https://doi.org/10.1007/978-3-031-38314-4>.
- [9] Xargay, E., Hovakimyan, N., and Cao, C., “ $\mathcal{L}_1$ Adaptive Controller for Multi-Input Multi-Output Systems in the Presence of Nonlinear Unmatched Uncertainties,” *American Control Conference*, 2010, pp. 874–879. <https://doi.org/10.1109/ACC.2010.5530686>.
- [10] Kobilarov, M., Ta, D.-N., and Dellaert, F., “Differential Dynamic Programming for Optimal Estimation,” *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 863–869. <https://doi.org/10.1109/ICRA.2015.7139279>.
- [11] Betts, J. T., “Survey of Numerical Methods for Trajectory Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. <https://doi.org/10.2514/2.4231>.
- [12] Diehl, M., Ferreau, H. J., and Haverbeke, N., *Nonlinear Model Predictive Control: Towards New Challenging Applications*, Springer Berlin Heidelberg, 2009, Chap. Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, pp. 391–417. https://doi.org/10.1007/978-3-642-01094-1_32.
- [13] Ross, I. M., and Karpenko, M., “A Review of Pseudospectral Optimal Control: From Theory to Flight,” *Annual Reviews in Control*, Vol. 36, No. 2, 2012, pp. 182–197. <https://doi.org/10.1016/j.arcontrol.2012.09.002>.
- [14] Hovakimyan, N., Cao, C., Kharisov, E., Xargay, E., and Gregory, I. M., “ $\mathcal{L}_1$ Adaptive Control for Safety-Critical Systems,” *IEEE Control Systems Magazine*, Vol. 31, No. 5, 2011, pp. 54–104. <https://doi.org/10.1109/MCS.2011.941961>.
- [15] Patterson, A., Ackerman, K. A., Cook, J., Acheson, M. J., and Gregory, I. M., “An $\mathcal{L}_1$ Adaptive Control Augmentation for a Lift-Plus-Cruise Vehicle,” *AIAA SciTech Forum*, 2023. <https://doi.org/10.2514/6.2023-2541>.
- [16] Acheson, M. J., Gregory, I. M., and Cook, J., “Examination of Unified Control Incorporating Generalized Control Allocation,” *AIAA SciTech Forum*, 2021. <https://doi.org/10.2514/6.2021-0999>.
- [17] Cook, J., and Gregory, I. M., “A Robust Uniform Control Approach for VTOL Aircraft,” *VFS Autonomous VTOL Technical Meeting and Electric VTOL Symposium*, 2021.

- [18] Simmons, B. M., Buning, P. G., and Murphy, P. C., “Full-Envelope Aero-Propulsive Model Identification for Lift+Cruise Aircraft Using Computational Experiments,” *AIAA Aviation Forum*, 2021. <https://doi.org/10.2514/6.2021-3170>.
- [19] Houghton, M. D., Acheson, M. J., Patterson, A. P., Oshin, A., and Gregory, I. M., “Combined Bernstein Polynomial Optimal Reciprocal Collision Avoidance Differential Dynamic Programming for Trajectory Replanning and Collision Avoidance for UAM Vehicles,” *AIAA SciTech Forum*, 2023. <https://doi.org/10.2514/6.2023-2544>.