Robust Post-Stall Motion Planning with Fixed-Wing UAVs

Joseph Moore
Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland 20723
Email: {joseph.moore}@jhuapl.edu

I. Introduction

Fixed-wing unmanned aerial vehicles (UAVs) offer significant performance advantages over rotary-wing UAVs in terms of endurance and efficiency. However, traditionally these vehicles have been severely limited with regards to maneueverability. In this paper, we present a motion planning approach for enabling highly maneuverable fixed-wing UAVs in complex environments. Our approach generates collision-free trajectories in real-time for a fixed-wing aircraft while also accounting for the uncertainty associated with post-stall aerodynamics through adaptation. We believe that our approach has the potential to dramatically increase the range and endurance of unmanned systems in constrained subterranean and urban environments which may span for many miles.

II. Related Work

In comparison to rotorcraft UAVs [8], relatively little work has been done to explore the use of fixed-wing UAVs as a platform for mobile roboitcs. This is primarily because fixed-wing UAVs, in traditional, low angle-of-attack flight regimes, have limited the maneuverability when compared with their quadcopter counterparts. However, fixed-wing vehicles have the potential to be as maneuverable as rotorcraft, if the vehicle is able to leave low angle-of-attack domains. Some of the earliest and most notable work in post-stall maneuvers with fixed-wing UAVs is presented in [17], where researchers explored transition to-and-from a prop-hang configuration. This was preceded by work in [11], which sought to generate real-time trajectories for a planar ducted-fan UAV with aerodynamic surfaces. The control of fixed-wing UAVs in post-stall flight was also advanced through the exploration of post-stall perching [13], where a library of trajectories was used to enable robust perching performance. Motion planning for fixed-wing UAVs was also explored in [9], where a library of funnels exploited invariance of initial conditions to navigate safely through an obstacle field in real-time. The initial work was limited to simple planar model representations, but then was extended to higher dimensional models which could exploit the full angle-of-attack envelope. More recently, researchers have explored real-time trajectory motion-planning of fixed-wing UAVs using simplified models restricted to low angle-of-attack domains [1]. Researchers have also explored methods for offline motion planning, using a combination of randomized motion planning and trajectory optimization [7].

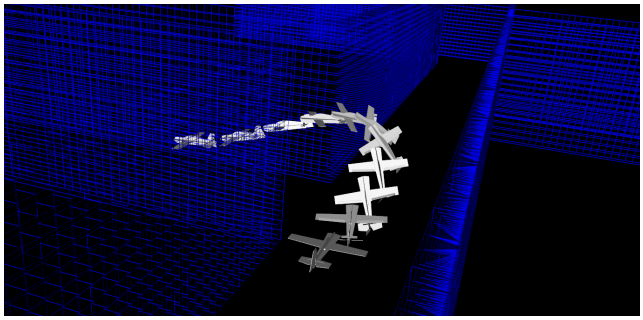

Fig. 1. Results of post-stall motion planning with fixed-wing UAV. RRT is used to generate a collision-free seed for the trajectory optimizer, which is able to exploit the post-stall aerodynamics to execute a collision-free 90 degree Herbst-like maneuver. Blue 3D grid represents voxelized map.

III. Approach

We believe that to navigate in complex environments which are unknown a-priori, we must be able to generate sophisticated trajectories in real-time that exploit the full flight envelope for fixed-wing vehicles and achieve "supermaneuverability" [3]. Trajectory libraries, while commonly used to reduce online computational burdens, often become computationally expensive and result in sub-optimal solutions when the vehicle state space is large (>6) and the environment is complex [10]. In addition, our approach must be able to handle complicated physics models, with complex aerodynamics, which are, in general, not differentially flat.

In this paper, we present a method which consists of randomized motion planning, direct trajectory optimization, and indirect trajectory optimization to enable post-stall motion planning with an update rate of at least 50Hz. We explore a receding horizon control strategy, rather than a real-time two-degree-of-freedom design [14]. Using a two-degree-of-freedom design requires methods for guaranteeing trajectory tracking over some finite horizon (e.g. tube MPC), and therefore adds an additional layer of computational complexity. We hypothesize that if we can replan trajectories fast enough and account for uncertainty, this may be sufficient to achieve robust performance.

A. Dynamics Model

Building off the work in [13], we use a three-dimensional model of a fixed-wing UAV with a propeller in a "puller" configuration. We define our state as $\mathbf{x} = \{x_r, y_r, z_r, \phi, \theta, \psi, \delta_1, \delta_2, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z\}$. Here $\mathbf{r} = \begin{bmatrix} x_r, y_r, z_r \end{bmatrix}^T$ represents the position of the center of mass

in the world frame $O_{x_r y_r z_r}$, $\boldsymbol{\theta} = \begin{bmatrix}\phi, \theta, \psi\end{bmatrix}^T$ represents the set of z-y-x euler angles, $\boldsymbol{\delta} = \begin{bmatrix}\delta_1, \delta_2\end{bmatrix}^T$ are control surface deflections due to the right and left elevons, $\mathbf{v} = \begin{bmatrix}v_x, v_y, v_z\end{bmatrix}^T$ is the velocity of the center of mass in the *body* fixed frame O_{xyz}, $\boldsymbol{\omega} = \begin{bmatrix}\omega_x, \omega_y, \omega_z\end{bmatrix}^T$ represents the angular velocity of the body the body-fixed frame. We can then write $\mathbf{x} = \{\mathbf{r}^T, \boldsymbol{\theta}^T, \boldsymbol{\delta}^T, \mathbf{v}^T, \boldsymbol{\omega}^T\}^T$.

The equations of motion then become

$$\begin{aligned}
\dot{\mathbf{r}} &= \mathbf{R}_b^r \mathbf{v} \\
\dot{\boldsymbol{\theta}} &= \mathbf{R}_\omega^{-1} \boldsymbol{\omega} \\
\dot{\boldsymbol{\delta}} &= \mathbf{u}_{cs} \\
\dot{\mathbf{v}} &= \mathbf{f}/m - \boldsymbol{\omega} \times \mathbf{v} \\
\dot{\boldsymbol{\omega}} &= \mathbf{J}^{-1}(\mathbf{m} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega})
\end{aligned} \tag{1}$$

Here m is vehicle mass, $\mathbf{J}$ is the vehicle's inertia tensor with respect to the center of mass, $\mathbf{f}$ is the total force applied to the vehicle in body-fixed coordinates, $\mathbf{m}$ are the moments applied about the vehicle's center of mass in body-fixed coordinates, $\mathbf{u}_{cs}$ are the angular velocity inputs of the control surfaces. $\mathbf{R}_b^r$ denotes the rotation from the body-fixed frame to the world frame, and $\mathbf{R}_{\boldsymbol{\omega}}$ is the rotation which maps the euler angle rates to an angular velocity in body-fixed frame.

The forces acting on the vehicle can be written as

$$\mathbf{f} = \mathbf{f}_w + \sum_i \mathbf{R}_{\delta_i}^b \mathbf{f}_{\delta_i} + \mathbf{R}_b^{rT} g \mathbf{e}_{z_r} + \sum_i \mathbf{R}_{t_i}^b \mathbf{f}_{t_i} \tag{2}$$

where $\mathbf{f}_w$ represents the force due to the wing, $\mathbf{f}_{\delta_i}$ represent the forces due to the control surfaces, $\mathbf{f}_{t_i}$ represent the forces due to the thrust sources. $\mathbf{R}_{t_i}^b$ is the rotation matrix that defines the orientation of the thrust source with respect to the body fixed frame. $\mathbf{R}_{\delta_i}^b$ is the rotation matrix that defines the control surface reference frame with respect to the body fixed frame.

The forces acting on the wing are given as

$$\mathbf{f}_w = f_{n,w}\mathbf{e}_y = \frac{1}{2} C_{n,w} \rho |\mathbf{v}_w|^2 S_w \mathbf{e}_y \tag{3}$$

where $C_{n,w}$ comes from the flat plate model in [4] and is given as $C_{n,fp} = 2\sin\alpha_w$. Here $\alpha_w = \arctan\frac{v_{w,z}}{v_{w,x}}$ and $\mathbf{v}_w = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_w$, where $\mathbf{r}_w$ is the distance from the center of gravity to the wing centroid.

To model forces our control surfaces, we use

$$\mathbf{f}_{\delta_i} = f_{n,\delta_i}\mathbf{e}_{y_\delta} = \frac{1}{2} C_{n,fp} \rho_{\delta_i} |\mathbf{v}_{\delta_i}|^2 S_e \mathbf{e}_{y_\delta} \tag{4}$$

where $\mathbf{v}_{\delta_i}$ is the velocity of the i^{th} control surface in the control surface reference frame given as

$$\mathbf{v}_{\delta_i} = \mathbf{R}_{\delta_i}(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_h + \mathbf{v}_{bw}) + (\mathbf{R}_{\delta_i}\boldsymbol{\omega} + \boldsymbol{\omega}_{\delta_i}) \times \mathbf{r}_{\delta_i}. \tag{5}$$

Here $\mathbf{r}_h$ represents the displacement from the vehicle center of mass to the control surface "hinge" point, $\mathbf{R}_{\delta_i}$ represents the rotation matrix that transforms vectors in the body-fixed frame into the control surface frame, $\boldsymbol{\omega}_{\delta_i}$ is the simple rotation rate of the control surface in the control surface frame.

$\mathbf{v}_{bw}$ is the velocity due to the backwash of the propeller. It can be approximated using momentum theory as,

$$\mathbf{v}_{bw} = v_{bw}\mathbf{e}_x = \sqrt{\|\mathbf{v}_p\|_2^2 + \frac{2u_t}{\rho_t S_{disk}}} - \|\mathbf{v}_p\|_2. \tag{6}$$

where u_t is the thrust input, S_{disk} is the area of the actuator disk, ρ is the density of air and $\mathbf{v}_p$ is the velocity at the propeller. For a given control surfaces, the angle of attack becomes

$$\alpha_\delta = \arctan\frac{v_{\delta,z}}{v_{\delta,x}} \tag{7}$$

$\mathbf{m}$ in the body fixed frame can be given as

$$\mathbf{m} = \mathbf{r}_w \times \mathbf{f}_w + \sum_i (\mathbf{r}_{\delta_i} \times \mathbf{R}_\delta \mathbf{f}_{\delta_i}) \tag{8}$$

Here $\mathbf{r}_w = c_w\mathbf{e}_x$ and $\mathbf{r}_{\delta_i} = l_h\mathbf{e}_x + \mathbf{R}_\delta(l_\delta \mathbf{e}_{x_\delta})$.

B. Motion Planning

To achieve real-time planning for fixed-wing vehicles across the entire flight envelope, we utilize a three-stage, hierarchical motion-planning strategy consisting of a randomized motion planner, a direct, and an indirect trajectory optimizer. We attempt to exploit the advantages of all three trajectory generation techniques to achieve effective receding horizon control in the presence of obstacles.

1) Randomized Motion Planning: Randomized motion planning has proved to be an effective strategy for generating motion plans in complex environments and in the presence of local minima. Rapildy-exploring Random Trees (RRTs) have been a particularly powerful tool for solving path planning problems. However, RRTs often exhibit poor performance when required to reason about dynamical systems with large state-spaces. Here, we use a simplified three-state model to plan a collision-free path with a rapidly exploring random tree in three dimensions. OctoMap [5], is used represent the environment and provide information about obstacles. While the trajectory generated by this process will not be a feasible path for our aircraft to follow, it will provide a seed to the next stage of the planner, which will enable the trajectory optimizer to avoid local minima. Here, we study the motion planning problem apart from the perception problem and assume that we have a full map of the environment. However, the nature of our approach (i.e., receding horizon control), could easily extend to the case where the field-of-view is limited and trajectories must be re-planned when provided with new information about the environment.

2) Direct Trajectory Optimization: Direct trajectory optimization formulates the nonlinear optimization problem by including both inputs and states as decision variables [18]. A collocation or transcription of the dynamics is then used to constrain the variables to ensure dynamic feasibility. For this reason, direct trajectory optimizers are well suited to accept a set of state variable initializations and are are often more robust to local minima then indirect methods [18]. Here, we use a direct transcription formulation of Simpson's integration rule as described in [15]. We use a simplified model for

our dynamics, where we assume we have direct control over the vehicle's angular acceleration and the effects of control surface deflections and velocities are negligible. This allows us to reduce our model from 16 states to 12 states. To solve our direct trajectory optimization problem, we employ the Sparse Nonlinear Optimizer (SNOPT) [2]. Results can be seen in Figure 1. Collision avoidance is formulated as a non-penetration constraint on the occupancy grid.

C. Indirect Trajectory Optimization

Indirect trajectory optimization approaches, such as back-propagation-through-time (BPTT), real-time recurrent learning (RTRL), and iterative LQR (iLQR), are typically very susceptible to local minima, but can often be executed much more rapidly. Our approach takes advantage of this speed advantage and uses indirect trajectory optimization (BPTT) to quickly refine the solution from the direct trajectory optimization routine so that the complete dynamics and constraints are satisfied. The key to using indirect trajectory optimization effectively is providing the optimizer with seed values close to the optimum. To do this, we must find initial values for the system inputs. The prior optimization step provides us with the angular accelerations of the vehicle. From these angular accelerations, the required velocities and positions of the control surfaces can be solved for analytically.

For a fixed-wing aircraft with a rudder, elevator, and ailerons, the equations for angular acceleration are given as

$$\begin{aligned}&\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \mathbf{r}_w \times \mathbf{f}_w = \\ &(\mathbf{r}_{\delta_{a1}} \times \mathbf{R}_{\delta_{a1}}\mathbf{f}_{\delta_{a1}}) + (\mathbf{r}_{\delta_{a2}} \times \mathbf{R}_{\delta_{a2}}\mathbf{f}_{\delta_{a2}}) \\ &+ (\mathbf{r}_{\delta_e} \times \mathbf{R}_{\delta_e}\mathbf{f}_{\delta_e}) + (\mathbf{r}_{\delta_r} \times \mathbf{R}_{\delta_r}\mathbf{f}_{\delta_r}) \end{aligned} \tag{9}$$

If we assume that $\delta_{a1} = \delta_{a2}$, and that $\dot{\delta}_{a1} = \dot{\delta}_{a2}$, we have

$$\begin{aligned}&\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \mathbf{r}_w \times \mathbf{f}_w \\ &- \begin{bmatrix} 0 \\ (f_{n,\delta_{a1}} + f_{n,\delta_{a2}})(-l_{ha,x}c_{\delta_{a2}} + l_{\delta_a}) - f_{n,\delta_r}l_{h,z}s_{\delta_r} \\ (f_{n,\delta_{a1}} + f_{n,\delta_{a2}})l_{ha,y}s_{\delta_{a2}} \end{bmatrix} \\ &= \begin{bmatrix} (f_{n,\delta_{a1}} - f_{n,\delta_{a2}})l_{ha,y}c_{\delta_{a2}} \\ f_{n,\delta_e}(-l_{he,x}c_{\delta_e} + l_{\delta_e}) \\ f_{n,\delta_r}(l_{h,x}c_{\delta_r} + l_{\delta_r}) \end{bmatrix} \end{aligned} \tag{10}$$

The above exhibits a hierarchal structure, where the aileron inputs can be solved for first, followed by the rudder input, followed by elevator input given the state of the aircraft body and the angular accelerations. In fact, the control surface velocities can be solved for analytically by finding the roots of a sixth order polynomial in the case of roll, and a fourth order polynomial in the case of pitch and yaw.

This can be illustrated by considering the roll dimension.

We know that

$$f_{n,\delta_a} = \sin\alpha_a \rho |\mathbf{v}_{\delta_a}|^2 S_a = \rho v_{\delta_a,z}|\mathbf{v}_{\delta_a}|S_a \tag{11}$$

and

$$\mathbf{v}_{\delta_a} = \mathbf{R}_{\delta_a}(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_h + \mathbf{v}_{bw}) + \mathbf{R}_{\delta_a}\boldsymbol{\omega} \times \mathbf{r}_{\delta_a} + \omega_{\delta_a} r_{\delta_a}\mathbf{e}_{z_\delta}. \tag{12}$$

To find the angular velocity of the aileron, we must solve

$$c_0^2 = v_{\delta_{a1},z}^2(v_{\delta_{a1},x}^2 + v_{\delta_{a1},y}^2 + (v_{\delta_{a1},z} + r_{\delta_a}\omega_{\delta_a})^2) \tag{13}$$

$$- v_{\delta_{a2},z}^2(v_{\delta_{a2},x}^2 + v_{\delta_{a2},y}^2 + (v_{\delta_{a2},z} + r_{\delta_a}\omega_{\delta_a})^2). \tag{14}$$

D. Local Direct Adaptive Control

In this work, we also present a means of reasoning about unknown dynamics. If we assume that our unknown dynamics can be captured by constant parameters, we can use the notion of adaptive control Lyapunov functions [6] to compute adaptive feedback laws. These feedback laws utilize a direct adaptive control paradigm- that is, they do not attempt to identify unknown parameters. Rather, they attempt to adjust the control laws in the presence of unknown parameters to more effectively achieve the control objective [16].

The adaptive control Lyapunov function formulation is provided in [6]. Given the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{F}(\mathbf{x})\boldsymbol{\theta} + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{15}$$

$V(\mathbf{x}, \boldsymbol{\theta})$ is an adaptive control Lyapunov function, if it is a control Lyapunov function (clf) for the augmented system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{F}(\mathbf{x})\left(\boldsymbol{\theta} + \boldsymbol{\Gamma}\left(\frac{\partial V_a}{\partial \theta}\right)^T\right) + \mathbf{g}(\mathbf{x})\mathbf{u}. \tag{16}$$

While the original treatment of adaptive clfs sought to prove that systems were globally adaptively stabilizable, these adaptive clfs can also be computed for locally to a trajectory. In [12], sum-of-squares programming was used to find locally valid adaptive clfs. However, linear optimal control can also be used to find these local controllers with reduced computational burden. By applying time-varying LQR to the augmented system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{F}(\mathbf{x})\boldsymbol{\theta} + \mathbf{g}(\mathbf{x})\mathbf{u} \tag{17}$$

$$\dot{\boldsymbol{\theta}} = 0 \tag{18}$$

we can find a $V(\mathbf{x}, \boldsymbol{\theta})$ which is valid in the neighborhood of a trajectory.

This provides a local direct adaptive controller which can dramatically improve trajectory following in the presence of unknown dynamics. To test this direct adaptive controller, we explored performance in the longitudinal "perching problem" [13], where a small, unpowered glider in steady-level flight must exploit post-stall aerodynamics to accurately land with low-velocity at a specific point in space. In one simulation-based experiment, we parameterized the flat plate lift, drag, and moment coefficients of a glider with an unknown offset and greatly improved perching performance without multiple iterations (see Figures 2 and 3). Furthermore, because the adaptation law in this paradigm is a function of the system state, we can augment the existing fixed-wing state with the parameter estimates and incorporate parameter adaptation as part of our final indirect optimization stage. This would not only allow for local direct adaptation to unknown parameters, but also for online modification of the nominal trajectory to facilitate faster parameter convergence.

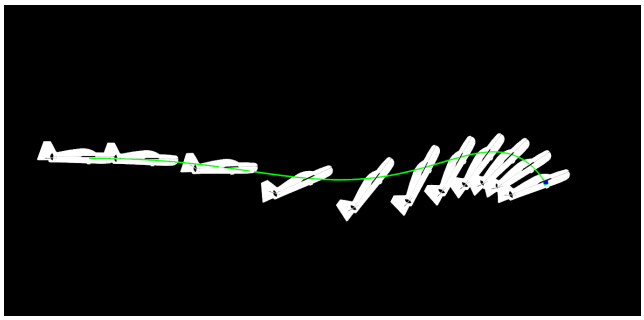

Fig. 2. Perching in the presence of unknown dynamics: With adaptation. The vehicle center of mass (blue dot) still reaches the goal state (green dot)

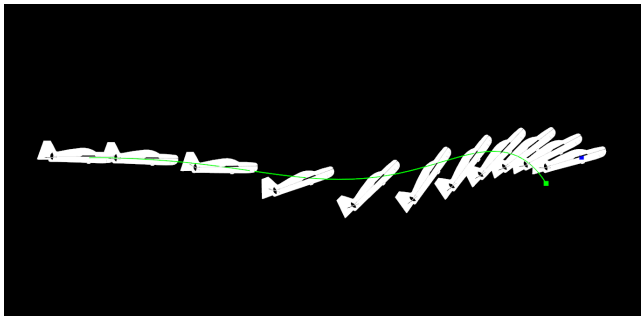

Fig. 3. Perching in the presence of unknown dynamics: Without adaptation. The vehicle center of mass (blue dot) overshoots the goal state (green dot).

IV. Discussion

In this paper, we proposed an approach for real-time receding horizon control with fixed-wing UAVs that can exploit post-stall aerodynamics. We also proposed an approach for direct adaptive control along trajectories to compensate for unmodeled dynamics. Future work will seek to integrate these two approaches to enable robust post-stall motion planning in the presence of unmodeled dynamics and changing environments. We will also begin to explore the use of predictive perception models to allow for planning beyond the line-of-sight. We believe that by enabling fixed-wing UAVs to achieve rotorcraft-like maneuverability in the presence of obstacles, we will dramatically increase the range of unmanned systems in complex and tightly-constrained environments.

References

[1] Hamid Alturbeh and James F Whidborne. Real-time obstacle collision avoidance for fixed wing aircraft using b-splines. In *Control (CONTROL), 2014 UKACC International Conference on*, pages 115–120. IEEE, 2014.

[2] P. Gill, W. Murray, and M. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[3] Wolfgang B Herbst. Supermaneuverability. Technical report, 1984.

[4] Sighard F Hoerner and Henry V Borst. Fluid-dynamic lift: practical information on aerodynamic and hydrodynamic lift. 1985.

[5] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[6] Miroslav Krstić and Peter V Kokotović. Control lyapunov functions for adaptive nonlinear stabilization. *Systems & Control Letters*, 26(1):17–23, 1995.

[7] Joshua M Levin, Aditya Paranjape, and Meyer Nahon. Agile fixed-wing uav motion planning with knife-edge maneuvers. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 114–123. IEEE, 2017.

[8] Chun Fui Liew, Danielle DeLatte, Naoya Takeishi, and Takehisa Yairi. Recent developments in aerial robotics: An survey and prototypes overview. *arXiv preprint arXiv:1711.10085*, 2017.

[9] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *arXiv preprint arXiv:1601.04037*, 2016.

[10] Mark B Milam, Kudah Mushambi, and Richard M Murray. A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, pages 845–851. IEEE, 2000.

[11] Mark B Milam, Ryan Franz, and Richard M Murray. Real-time constrained trajectory generation applied to a flight control experiment. *IFAC Proceedings Volumes*, 35 (1):175–180, 2002.

[12] Joseph Moore and Russ Tedrake. Adaptive control design for underactuated systems using sums-of-squares optimization. In *American Control Conference (ACC), 2014*, pages 721–728. IEEE, 2014.

[13] Joseph Moore, Rick Cory, and Russ Tedrake. Robust post-stall perching with a simple fixed-wing glider using LQR-trees. *Bioinspiration & Biomimetics*, 9(2), 2014.

[14] Richard M Murray. Optimization-based control. *California Institute of Technology, CA*, 2009.

[15] Diego Pardo, Lukas Möller, Michael Neunert, Alexander W Winkler, and Jonas Buchli. Evaluating direct transcription and nonlinear optimization methods for robot motion planning. *IEEE Robotics and Automation Letters*, 1(2):946–953, 2016.

[16] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

[17] Frantisek Sobolic and Jonathan How. Nonlinear agile control test bed for a fixed wing aircraft in a constrained environment. In *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference*, page 1927, 2009.

[18] Russ Tedrake. Underactuated robotics: Learning, planning, and control for efficient and agile machines: Course notes for mit 6.832. *Working draft edition*, page 3, 2009.