SAND2022-16713C

Improving the Maneuver Automaton with Maneuver Interruption

Kevin S. Choi*, Zachary C. Goddard† and Samuel J. Deal ‡
Georgia Institute of Technology, Atlanta, GA, 30332, USA

Kyle Williams §
Sandia National Laboratories, Albuquerque, NM, 87123, USA

Anirban Mazumdar¶
Georgia Institute of Technology, Atlanta, GA, 30332, USA

This paper introduces a method for improving the re-planning ability of the Maneuver Automaton. The maneuver-to-maneuver transition, referred to as "Maneuver Interruption", is enabled by identifying near-identical dynamic states between different maneuvers. Naïve search first identifies viable maneuver connections within the motion primitive library, then the identified connections are pruned using Monte-Carlo simulations. Maneuver Interruption is tested on obstacle evasion tasks using a nonlinear ZOHD Drift model with a motion primitive library generated using reinforcement learning. Through Monte-Carlo simulations, the connection library was pruned to 1.5% of the size of the full library with minimal losses in performance. On randomly generated obstacle fields, Maneuver Interruption has enabled longer collision-free flights at the cost of some trajectory-tracking performance.

I. Introduction

Motion primitive-based planning is an effective method for rapidly generating kinodynamic motion plans for autonomous aircraft. Motion primitives specify feasible sub-trajectories that the aircraft may take. These primitives can be generated off-line, analyzed rapidly, and concatenated to form motion plans that inherently respect the vehicle dynamics and constraints. The Maneuver Automaton (MA) by Frazzoli [1, 2] is a framework for organizing motion primitives in a manner suitable for motion planning. Frazzoli identifies two classes of motion primitives: trims, which define steady-state aircraft trajectories, and maneuvers, which are finite-time trajectories that are compatible with trim states at their beginning and end. The Maneuver Automaton generates motion plans by treating itself as a finite state machine, with trims as nodes and maneuvers as edges. The Maneuver Automaton has been demonstrated to be effective in unmanned aerial vehicles [3–5]. However, a general weakness of motion planning in this manner is the fact that the MA cannot re-plan during its maneuver states and has to wait until the maneuver has ended. For a sizeable portion of the flight time, the aircraft motion is prescribed and cannot be updated based on new information. The ability to respond rapidly to external changes is an attractive trait for many types of autonomous aircraft. Specifically, aircraft operating in dynamic environments or with limited sensing may need to update the motion plan mid-maneuver. The inability to replan can degrade performance and potentially cause catastrophic failure.

Furthermore, unmanned aerial vehicles (UAVs) often operate with limited sensing capability in cluttered environments. Such environments can be indoors or very close to the ground. Given limited sensing in a cluttered environment, a UAV must often make rapid, last-minute changes to its trajectories to evade obstacles that appear in its sensing horizon. We seek to modify the Maneuver Automaton framework in order to enable re-planning while maintaining the many positive attributes of the approach.

This paper proposes a new method of enabling maneuver interruption for an autonomous aircraft planner using the MA framework. Viable mid-maneuver connections are identified using an Euclidean distance of select aircraft states, with individual states weighted by the difficulty of its transition. A closed-loop controller on both aircraft position

*Graduate Student, Woodruff School of Mechanical Engineering.
†Graduate Student, Woodruff School of Mechanical Engineering, AIAA Student Member
‡Graduate Student, Woodruff School of Mechanical Engineering.
§Pathfinder Technologies
¶Assistant Professor, Woodruff School of Mechanical Engineering, AIAA Member.

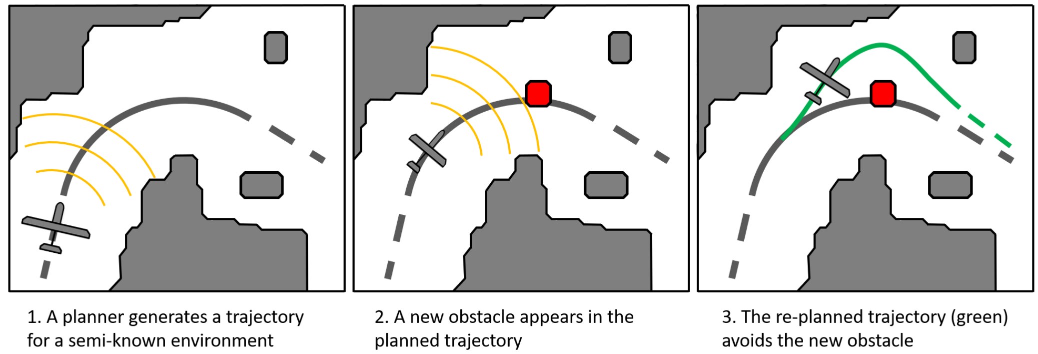


Fig. 1 The ability to re-plan from a maneuver trajectory is critical to avoiding collisions. Here, re-planning is required in order to avoid a collision with a newly detected obstacle (red).

and state enables robustness in the primitive-based controller and handles the variances in state caused by maneuver interruption. In this work, we focus on small unmanned aerial vehicles. We conclude the paper by evaluating the effectiveness of our proposed methods using simulations within a randomly generated obstacle field.

The contributions of this paper are as follows:

1) The development of a new method of re-planning called "Maneuver Interruption."
2) Evaluation of maneuver connection pruning based on Monte-Carlo simulations.
3) Demonstration of maneuver interruption on an arbitrary set of pilot-generated maneuvers.
4) A case-study example of the improvement in performance for a vehicle with limited sensing range.

The paper begins (Section II) with a literature review of relevant literature on the Maneuver Automaton and other motion primitive-based planners. Section III describes the identification of dynamically feasible maneuver connections, pruning of the identified connections, and closed-loop control for Maneuver Interruption. Section IV details the application of Maneuver Interruption on a model of the Drift. The section includes a description of the nonlinear flight dynamics model (FDM) and the motion primitive library used for validation, a demonstration of Maneuver Interruption on well-known maneuvers, and an implementation of Maneuver Interruption on a Hybrid A* planner exploring a randomly generated obstacle field.

II. Related Works

Kinodynamic motion planners are designed to produce trajectories that respect the kinematic and dynamic constraints of the system [6]. This method of planning is relevant to systems that must satisfy kinematic constraints, such as goal-following or obstacle evasion while having nonlinear and complex dynamics. Computing feasible trajectories that satisfy both constraints is often time-consuming, making rapid planning difficult. Some methods include Closed-Loop Rapidly Exploring Random Trees (CL-RRT), B-splines, and Stable Sparse RRT (SST) [7–9]. However, these involve repeatedly numerically solving the equations of motion. These forward simulations can be time-consuming. In contrast, the Maneuver Automaton simplifies kinodynamic planning by specifying a collection of sub-trajectories known to be dynamically feasible, leaving only the kinematic problem to be solved. The remaining kinematic problem can be solved using RRT, Hybrid A*, and other well-known planning algorithms [10, 11]. As a result, motion primitives used within the Maneuver Automaton enable rapid kinodynamic planning at the expense of optimality.

Motion primitives allow complicated flight dynamics to be condensed into a form that is easily handled by an overseeing planner. Many existing works have utilized motion primitives to simplify kinodynamic motion planning. For example, previous work [5] enabled the use of aggressive, nonlinear turn-around maneuvers through the use of motion primitives. Another work [12] utilized an approximation where a set of commanded accelerations serve as the motion primitive for the cave-exploring quadcopter.

The MA framework is a popular and successful method of enabling agile flight in small autonomous drones [3–5]. The MA provides a way of concatenating motion primitives by specifying two classes of compatible motion primitives, trims and maneuvers, which combine to form a finite state machine. Trim primitives describe the vehicle in steady-state where control inputs are kept constant. Maneuver primitives (which will be referred to as simply "maneuvers" throughout this paper) are finite-time trajectories that are compatible with the end of one trim state and the beginning of another.

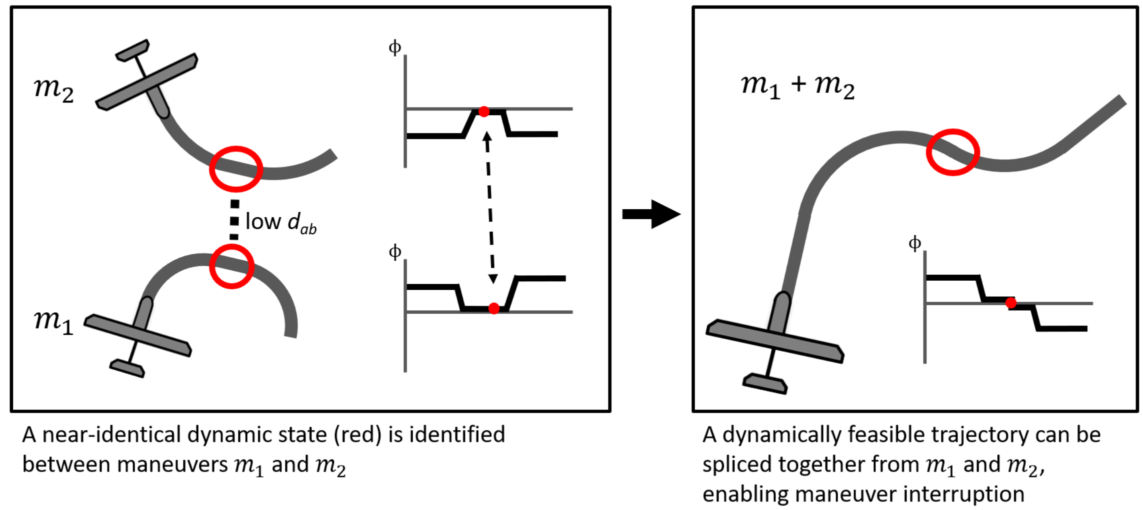


Fig. 2 Identification of connection points for maneuver interruption.

The motion plan is created by concatenating trims with maneuvers. Robustness of the motion plan is achieved using an underlying controller that ensures that the vehicle remains "close" to the specified trajectory [1, 2].

Formally, the Maneuver Automaton is a Motion Description Language that describes motion in terms of two classes of motion primitives: trim primitives and maneuvers. Trim primitives are steady-state motions where controls are kept constant, most commonly found in equilibrium flight conditions. Maneuvers are nontrivial primitives that are compatible with trim primitives at its beginning and end. These motion primitives are invariant to a symmetry group G, where the action of G on the state commutes with the state flow. Examples of symmetry groups for aircraft are lateral position, altitude, and heading. The Maneuver Automaton is a hybrid system, meaning that the planning state space includes both continuous and discrete components [2].

Several planning methods can utilize the MA to create kinodynamically viable motion plans. For example, Hybrid A* generates a feasible trajectory by performing a search on the motion primitive library. In general, the motion plan generated using a MA-based planner consists of trim primitives held for some time, punctuated by maneuvers that allow aggressive maneuvering and transitioning between different trim primitives.

Funnel libraries guarantee robustness of a funnel trajectory to points within the funnel as long as the initial state is within the inlet set of that funnel. These funnels can then be combined in sequence to create a motion plan that explicitly takes into account the effects of uncertainty [13]. Similar to the MA with maneuver interruption, a funnel-based planner is able to re-plan from any state to a funnel as long as the current state is within the inlet set of that funnel. The primary difference between the MA and the funnel-based planner is the introduction of purposeful quantization in the design of the control system, reducing the complexity of the planning task [14].

III. Maneuver Interruption

Despite the many positive attributes of motion primitives and the Maneuver Automaton, an enduring issue is the need to be in a trim state before choosing a new maneuver. This means that if new information emerges, the vehicle cannot respond to it until the current maneuver is complete. This can be especially problematic when the primitive library contains lengthy maneuvers. Therefore, we desire a way to enable re-planning even when the aircraft is not necessarily within a trim primitive. Our method of re-planning, called "maneuver interruption", enables the MA-based planner to switch between different maneuvers. Maneuver interruption is achieved by identifying near-identical points in aircraft dynamics between different maneuver trajectories. The system dynamics are invariant with respect to symmetry group variables (e.g. spatial coordinates), so these variables between the two trajectories can be spliced together to make the two trajectories spatially continuous. These connection points can be identified offline and collected as a library for rapid use during online planning. An illustration of Maneuver Interruption is shown in Fig. 2.

Several challenges exist in enabling maneuver interruption. Firstly, maneuver interruption increases planning complexity by adding many more discrete choices. Secondly, any small discontinuities introduced to the state trajectory through maneuver interruption must not propagate into larger errors. The following sections address the two challenges in the following ways: 1. identifying a concise set of points suitable for effective re-planning by first finding all dynamically feasible points through naïve search, then pruning the connection points through Monte-Carlo simulation, and 2. using closed-loop control on the aircraft state in conjunction with spatial trajectory control to ensure stability during maneuver transitions.

A. Identifying Maneuver Connections

Maneuver interruption is enabled by identifying near-identical non-invariant state points between two maneuver trajectories. The identification of "close enough" trajectories is approximated by a weighted euclidean distance between two aircraft states. Each state is weighted by its contribution to the distinctness between two non-invariant states. In this paper, this weight is given by $w = \frac{t}{x_0}$, where x_0 is a nominal offset state and t is the time required to settle from the nominal offset state to an equilibrium state.

$$\begin{aligned} x &= [\alpha, \beta, \phi, \theta, \psi, u, p, q, r] \\ w &= \left[\frac{t_{\alpha_0}}{\alpha_0}, \frac{t_{\beta_0}}{\beta_0}, ..., \frac{t_{r_0}}{r_0}\right] \\ t_{ab} &= \| w \odot (x_a - x_b) \| \end{aligned} \tag{1}$$

The quantity t_{ab} in eq. 1 represents the distance between two non-invariant aircraft states. Non-invariant state variables include angle of attack α, sideslip angle β, forward speed, angular velocities, and aircraft attitude. The constants t_0 are the times required to transition from level flight to a nominal change in each state variable x_0. Note that the quantity t_{ab} has a unit of time, and that t_{ab} roughly represents the time required to settle to a new state following a discontinuity in the state trajectory. To identify dynamically feasible re-planning points in a maneuver, the distance t_{ab} is evaluated at regular intervals against other maneuvers in the motion primitive library. If t_{ab} is smaller than a conservatively chosen nominal value, then it is added as a dynamically feasible re-planning point.

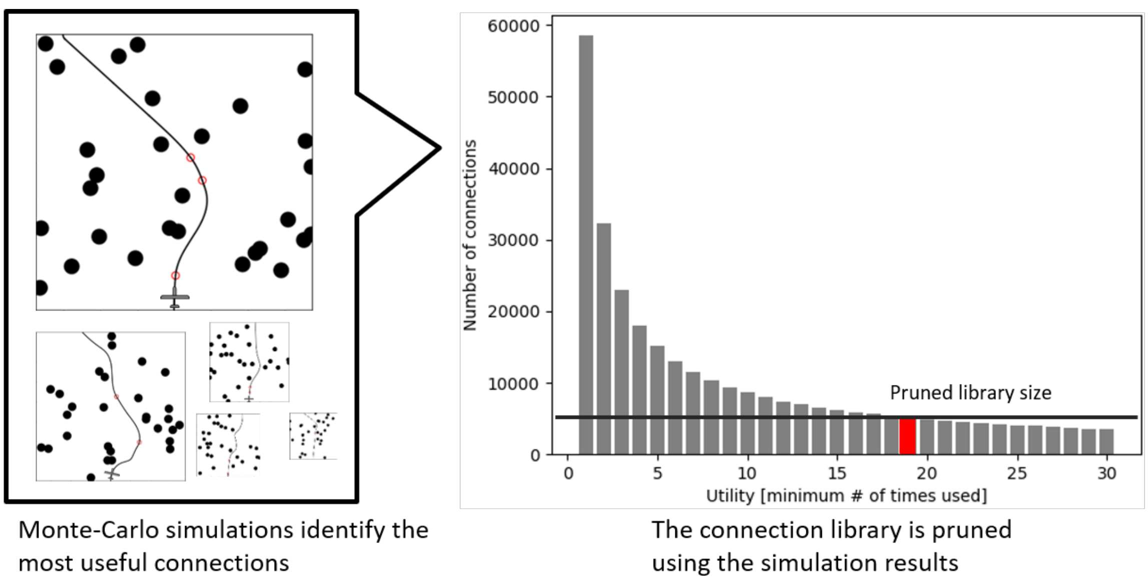


Fig. 3 The Monte Carlo simulation. Maneuver interruption is circled in the example obstacle fields.

The naïve approach of finding all re-planning points admissible by t_{ab} generates a large number of connections that are not particularly useful to re-planning. For example, connections occur extremely frequently near trim primitives due to the fact that the MA requires maneuvers to be left- and right-compatible with trim primitives. These connections do not add much to re-planning ability as re-planning is already possible within the nearby trim primitives.

Reduction of planning complexity through discretization of the search space is one of the advantages of the MA. In order to keep the planning complexity low, it is desired to keep only the connections that contribute significantly to re-planning performance. Monte Carlo methods are able to model complex, analytically intractable functions through random sampling. Here, Monte Carlo methods can be used to measure the usefulness of a large library of maneuver connections given an arbitrary planning algorithm. The utility of each maneuver connection is measured by tallying the number of times that maneuver connection is used in a large number randomly generated path-planning simulations (Figure 3, left). The connection library is then pruned using this information (Fig. 3, right).

B. Closed-Loop Control

Maneuver interruption introduces discontinuities in the dynamic trajectory. In order to ensure that these offsets do not propagate into larger errors, closed-loop control on the aircraft state is necessary. Because these discontinuities are expected to be small and occur near trajectories already known to be dynamically feasible, simple linear controllers are generally sufficient for closed-loop tracking of motion primitives. For this paper, controller design will focus on the feedback control of non-invariant states (e.g. attitude and airspeed) and anticipatory control of the invariant states (e.g. position, heading, and altitude) to eliminate the accumulation of positional errors.

Motion primitives from the MA describe a dynamically feasible state trajectory and predict a motion path. The state feedback controller acts on the former, ensuring that the aircraft correctly follows the state trajectory described by the motion primitive. Specifically, it is a linear feedback controller acting on the aircraft attitude $[\phi, \theta, \psi]$ and airspeed u_∞. The controller is shown in eq. 2.

$$u_s = \begin{bmatrix} p_c \\ q_c \\ r_c \\ u_c \end{bmatrix} = K_p \begin{bmatrix} 1 & 0 & -s(\theta) & 0 \\ 0 & c(\phi) & s(\phi)c(\theta) & 0 \\ 0 & -s(\phi) & c(\phi)c(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_d - \phi \\ \theta_d - \theta \\ \psi_d - \psi \\ u_{\infty,d} - u_\infty \end{bmatrix} \tag{2}$$

The motion path consisting of the invariant state variables $[x, y, z, \phi]$, also requires a controller to prevent the aircraft from drifting away from the motion trajectory predicted by the motion primitive. A position feedback controller is proposed in [15]. The controller offers tight tracking of curved flight paths by using an anticipatory control element. By looking ahead by distance L_1 and deriving the necessary lateral acceleration needed to converge to the future point, the position controller offers better tracking of curved trajectories over a conventional PID controller.

The overall controller, shown in Figure 4, combines the state-following controller with the navigational controller proposed in [15]. The state feedback controller, shown in the upper left, is used when the aircraft is already tightly tracking the positional trajectory, and the position controller, shown in the lower left, is used as a corrective measure when positional error increases. The two controllers are weighted by a continuous function that favors the non-invariant state variable trajectory when the positional error is minimal and favors the navigational controller when the positional error is large.

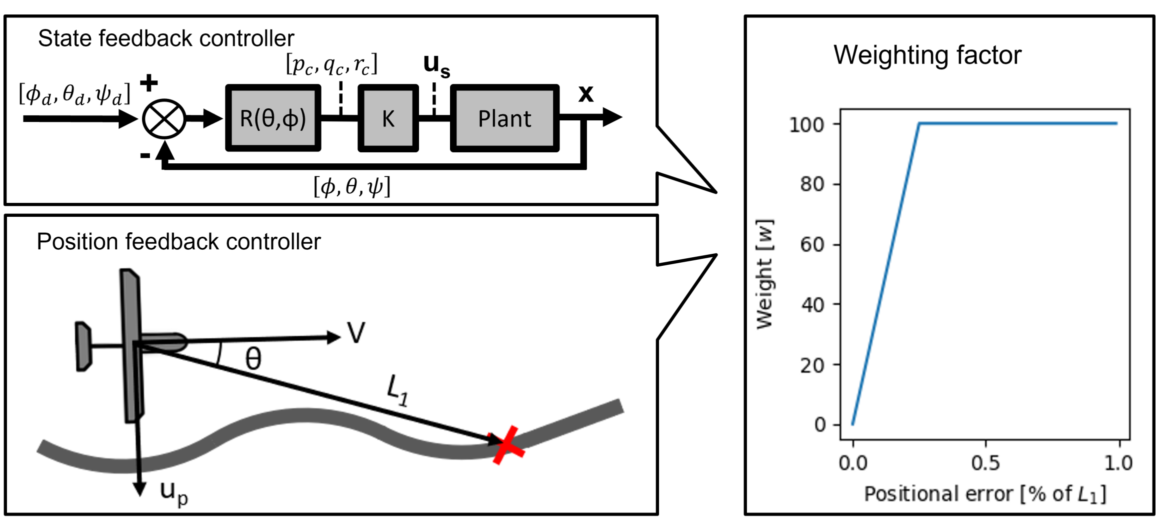


Fig. 4 Closed-loop controller for tracking motion primitives.

C. Planning with Maneuver Interruption

Maneuver Interruption improves the re-planning ability of the Maneuver Automaton. Planning with Maneuver Interruption consists of a motion planner implemented with the Maneuver Automaton, which is then augmented with re-planning with Maneuver Interruption. Many existing planners, with some modifications, are suitable for use with the Maneuver Automaton: RRT, Hybrid A*, and PRMs. For this work, a Hybrid A* planner will be used to demonstrate the re-planning ability of the Maneuver Automaton.

The Hybrid A* algorithm extends the A* algorithm to create feasible vehicle trajectories [11]. The original Hybrid A* algorithm used a range of curves that represent feasible paths for a car. Applied with the MA, the Hybrid A* algorithm searches over motion primitives to create a feasible path for an aircraft [16]. The nodes of the Hybrid A* algorithm are in the form (x, y, z, ψ, q_t), where (x, y, z) represent the position, ψ the heading, and q_t the trim primitive. The edges of the Hybrid A* algorithm include maneuvers as well as trim primitives fixed to some coasting time.

Algorithm 1 Maneuver Interruption

Inputs: the aircraft **a**, explored robot c-space **O**, maneuver plan **currPlan**, and current plan index **i**
a = a.step(currPlan[i]) ▷ 1 step in the motion plan
O = O.update(a) ▷ Update with new obstacles
t = a.timeSinceLastSwitch() ▷ Time since last maneuver switch
if $(S.trajectory \cap O) \neq \emptyset$ and $t > t_h$ **then**
    PQ = currPlan[i].getConnections() ▷ PQ is a priority queue of connections
    while $PQ \neq \emptyset$ **do**
        C = PQ.pop()
        newPlan, success = HybridA*(start = C) ▷ Generate a new motion plan starting from the connected maneuver C
        if success **then**
            return newPlan
        end if
    end while
end if
return currPlan

Planning with Maneuver Interruption utilizes the underlying motion planner until a new obstacle appears in the existing maneuver trajectory. Maneuver Interruption allows the aircraft to escape a maneuver when the current maneuver is projected to hit an obstacle. Such a scenario might occur in the context of moving obstacles, or in a semi-known environment where the locations of some obstacles are unknown (illustrated in Fig. 1). If a collision is detected, the switching algorithm attempts to generate new plans from the list of available connections from the aircraft's current state. If a new collision plan is successfully generated, then the current plan is replaced with the new plan and maneuver interruption is completed.

IV. Maneuver Interruption on a UAV

To demonstrate Maneuver Interruption on a physical system, a flight dynamics model (FDM) of the ZOHD Drift, a fixed-wing remote-control aircraft, is used. For motion planning, a Maneuver Automaton compliant motion primitive library is generated through reinforcement learning. The following sections will provide an overview of the FDM and the motion primitive library, a demonstration of Maneuver Interruption between some basic fighter maneuvers (BFMs), and evaluation of path-planning performance on an obstacle field given limited knowledge of the aircraft's surroundings.

A. FDM and Motion Primitive Library

1. Aircraft Platform

Maneuver Interruption is tested with a flight dynamics model of the ZOHD Drift. The Drift is a powered fixed-wing UAV platform with a wingspan of 34.25in, flight speed of 22m/s, weight of 250g, and a thrust-to-weight ratio of 0.3. The aircraft is pictured in Figure 5 along with the simulation rendering in FlightGear.

The FDM utilizes aerodynamic coefficients found empirically through wind tunnel testing to compute aerodynamic forces and a computationally efficient rigid-body model to convert said forces into rigid-body motion. A linearized

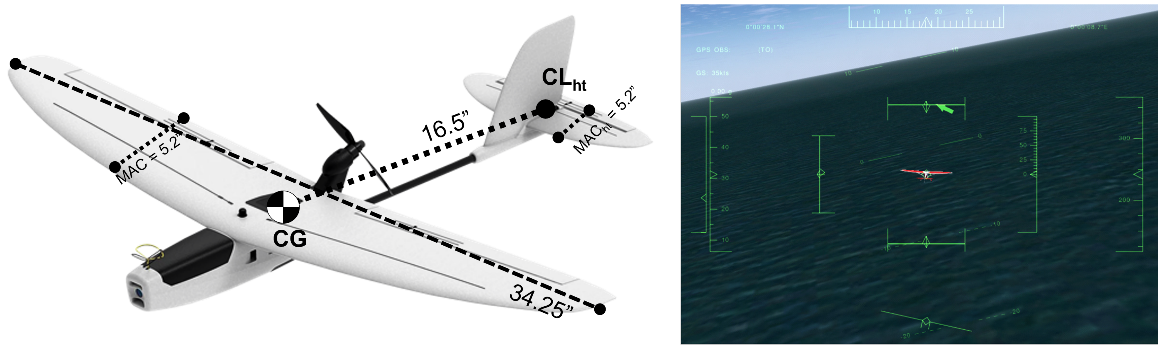


Fig. 5 The ZOHD Drift with dimensions (left) and its rendering in FlightGear (right).

model of the ZOHD Drift FDM is provided in eq. 3 and eq. 4 for the reader's reference. The FDM outputs are fed into FlightGear, which provides valuable visualization of aircraft flight.

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.183 & 0.460 & -0.033 & -9.81 \\ 1.717 & 53.35 & 12.15 & 0.00 \\ 0.720 & 28.50 & -16.39 & 0.00 \\ 0.000 & 0.000 & 1.000 & 0.00 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.238 & 2.839 \\ 43.29 & 0.533 \\ 210.1 & -0.126 \\ 1.196 & 0.000 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_T \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -4.587 & 1.761 & -17.86 & 9.81 \\ -5.657 & 55.47 & 2.626 & 0.00 \\ -13.77 & 19.50 & 6.624 & 0.00 \\ 0.000 & 0.000 & 1.000 & 0.00 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} -39.45 & 29.69 \\ 1135 & -37.30 \\ 411.6 & -90.77 \\ 6.789 & -.2346 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{4}$$

2. Sensor Model

Much work has been done for obstacle detection in UAVs. Obstacle detection often utilize multiple sensors, such as LiDAR, cameras, and other range-finding sensors to provide a reliable estimate of obstacles that lie ahead [17–19]. Mapping techniques allow the aircraft to remember the location of obstacles after the obstacles leave the aircraft's field of view. This work assumes a sensor suite capable of detecting obstacles within a 100-meter radius of the aircraft.

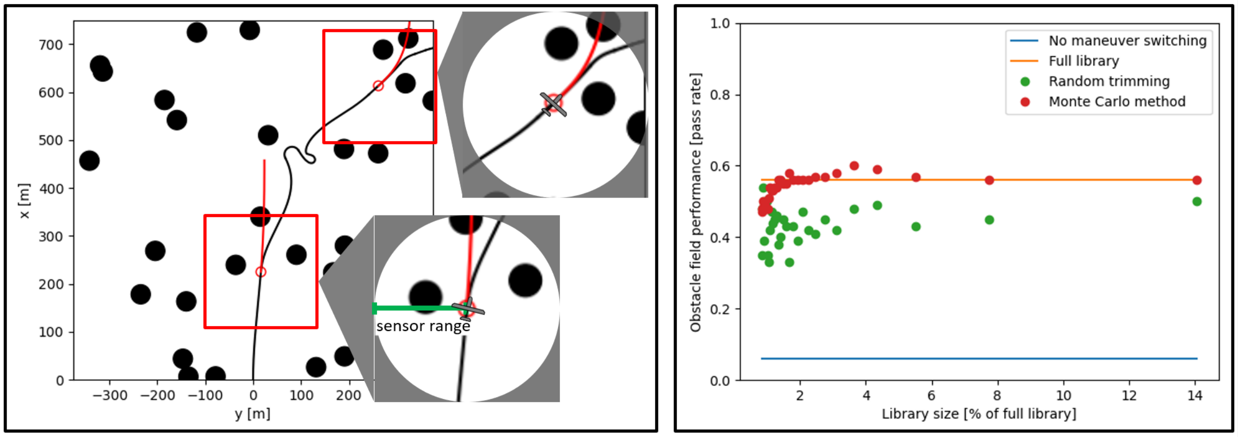
B. Motion Primitive Library

Motion primitive libraries can be generated using a range of methods including expert knowledge, optimal control, and machine learning. Previous work has shown how reinforcement learning can generate motion primitive libraries with minimal human input [16]. Specifically, the Soft Actor-Critic (SAC) algorithm is used to generate trajectories that attempt to minimize the time to the goal. Trims and maneuver primitives are then extracted from the trajectories produced by the reinforcement learning agent. A motion primitive library of 11 trims and 255 maneuvers was created for the ZOHD Drift.

C. Generating Maneuver Connections

The maneuver interruption paradigm was applied to the ZOHD Drift's motion primitive library. Dynamically feasible maneuver connections were identified through naïve search. With the maximum admissible distance between maneuver connections set at a nominal value of $t_{ab} < 2.5s$, a total of 339969 maneuver connections were identified.

The total number of maneuver connections can be drastically reduced by pruning the connections with respect to their utility in re-planning. The utility of each maneuver connection is measured with Monte-Carlo simulations. The Monte Carlo simulation introduces randomly generated obstacles in the aircraft's sensing horizon, forcing the plane to re-plan its trajectory around newly detected obstacles (Fig. 6, left). In the simulation, the aircraft uses a planning policy based on the Hybrid A* algorithm with the cost function favoring forward traversal. Due to the high computational costs associated with Monte-Carlo simulations, the aircraft model is not explicitly used in the simulation. Instead, the



The aircraft is forced to divert from the original maneuver (red) when new obstacles appear in its sensing horizon.

The performance of the Monte-Carlo connection library is comparable to that of the full library at a fraction of its size

Fig. 6 Monte-Carlo simulations (left) and its performance after pruning (right).

aircraft is assumed to have reasonable tracking of the position trajectories specified by its motion primitives, and a minimum "bumper distance" is added to the obstacles to account for any positional errors. The utility of each maneuver connection is measured by the number of times a particular connection is selected to successfully re-plan the aircraft trajectory. The Monte Carlo simulations were repeated for 400,000 re-planning decisions.

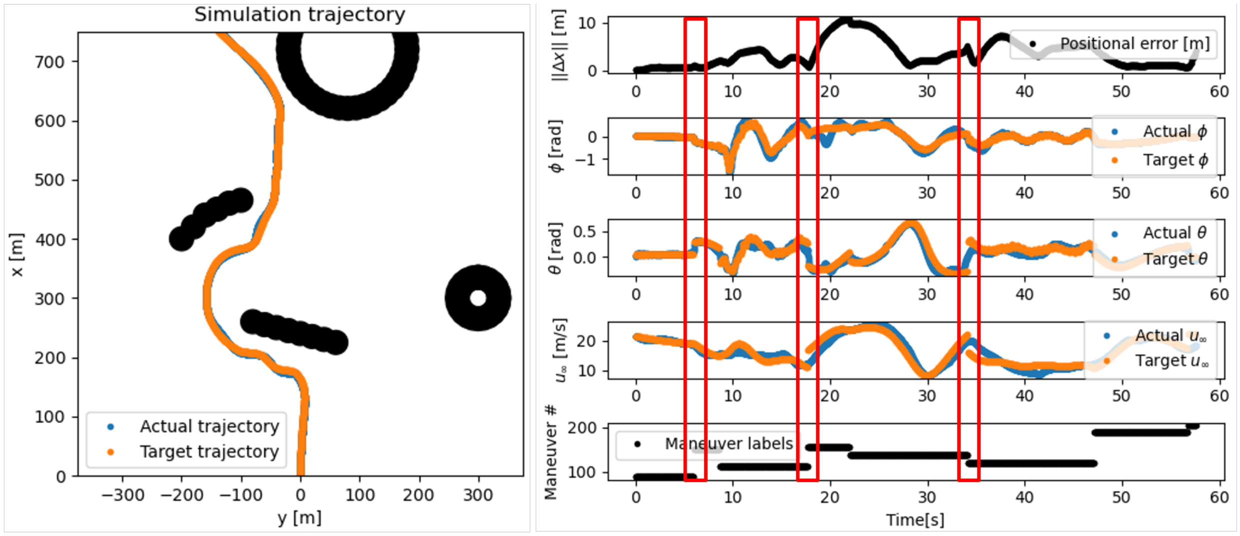
Re-planning performance is measured by the planner's ability to traverse a 750m-long obstacle field without collisions using Hybrid A*. To evaluate the effectiveness of pruning with Monte-Carlo simulations, the pruned connection library is compared to the performance of the full library, a randomly pruned connection library, and no library (no Maneuver Interruption) in Fig. 6. The pass rate is measured across 100 sets of 750m long randomly generated obstacle fields for all 4 cases. For each of the trimmed libraries, the test was performed across 30 different library sizes, ranging from 0.83% of the full library size to 14%.

The library pruned with Monte-Carlo methods had performance comparable to the full library at just 1.5% of the full library size, and success rates drop off rapidly after that point. Random pruning performed better than the library with no connections, but generally worse than the library pruned with Monte-Carlo methods. Overall, the results indicate (a) only a small fraction of the connection library is needed for significant improvements to re-planning performance and (b) Monte-Carlo simulations provide significant improvements to the quality of the pruned connections.

D. Maneuver Interruption in an Obstacle Field

Maneuver Interruption is tested on hand-generated and randomly generated obstacle fields. The motion plan is generated with Hybrid A* with Maneuver Interruption, and the resulting motion plan is tracked with closed-loop state control as well as the trajectory-following controller. Figure 7 shows the motion path generated by Hybrid A* with Maneuver Interruption in orange and the trajectory of the Drift FDM following the motion path in blue. The red boxes highlight the points at which maneuver interruption occur. The discontinuities in the aircraft attitude are clearly visible at these points.

Maneuver Interruption is evaluated on a randomly generated obstacle field. The obstacle field consists of 3 circles and 4 lines with randomized placement. The circles have radii of 30 to 100 meters, while the lines have lengths between 60 to 200 meters. Two hundred obstacles were generated in this manner for testing. The motion path is generated using the Hybrid A* algorithm, with the agent given a sensor model limiting the obstacles it can see to a 100-meter radius. Some of the obstacle fields and the motion paths generated are shown in Fig.8. Three metrics evaluate the performance of the planning algorithm: the ability to cross the 750m-long obstacle field, the average distance travelled before a collision, and the mean position error of the aircraft compared to the positional trajectory predicted by its motion primitives. Across the same set of obstacle fields, the Hybrid A* planner is evaluated with and without Maneuver Interruption. The planner with Maneuver Interruption had a success rate of 53.8%, average distance travelled of 588m, and mean position error of 9.6m. The planner without Maneuver Interruption had a success rate of 15.6%, average distance travelled of 362m, and mean position error of 4.1m. Overall, Maneuver Interruption led to better obstacle-avoiding performance at



Planner trajectory (orange) vs. FDM simulation trajectory (blue)

Motion primitive state trajectory (orange) vs. the FDM simulation trajectory (blue). Maneuver Interruption is highlighted in red boxes.

Fig. 7 FDM simulation following the trajectory produced by the Hybrid A* algorithm with Maneuver Interruption. The agent's vision is limited to a 100 meter radius.

the cost of some trajectory-tracking performance.

E. Maneuver Interruption with Basic Fighter Maneuvers

Maneuver interruption is also relevant to dynamic environments such as aerial combat. For example, expert human pilots are often taught to use a set of maneuvers for engaging adversaries [20]. This aligns well with the idea of motion primitives. However, aerial combat is a dynamic environment, and maneuvers may need to be interrupted when the circumstances change. We show how maneuver interruption can enable re-planning ability by starting from one pilot-generated basic fighter maneuvers (BFM) and switching to a different maneuver, resulting in an entirely different kinematic trajectory than the initial maneuver. In this case we used maneuvers generated by hand designed to replicate three BFMs (not by reinforcement learning). These are the vertical loop, the Immelmann turn, and the barrel roll. First, three common basic fighter maneuvers are generated through human pilot input. Dynamically feasible maneuver connections are identified using techniques described previously, and interesting connection points are selected by hand. Maneuver interruption is tested using the pair of pilot-generated motion primitives along with the state feedback controller. Figure 9 shows two examples of maneuver interruption, where the aircraft begins on one maneuver (highlighted in red) and performs Maneuver Interruption to switch to a portion of the second maneuver (highlighted in blue). The pilot-generated motion trajectories are shown in boxes (1) and (2), while the simulation-generated motion trajectories with maneuver interruption are shown in boxes (3). These initial results illustrate another way where maneuver interruption can increase performance in challenging environments and also create new behaviors.

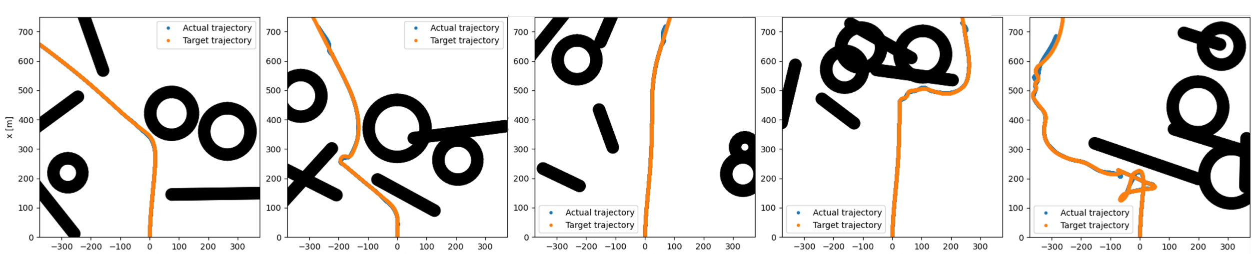


Fig. 8 Randomly generated obstacle fields. The motion path produced by Hybrid A* is in orange, while the FDM simulation trajectory is in blue.

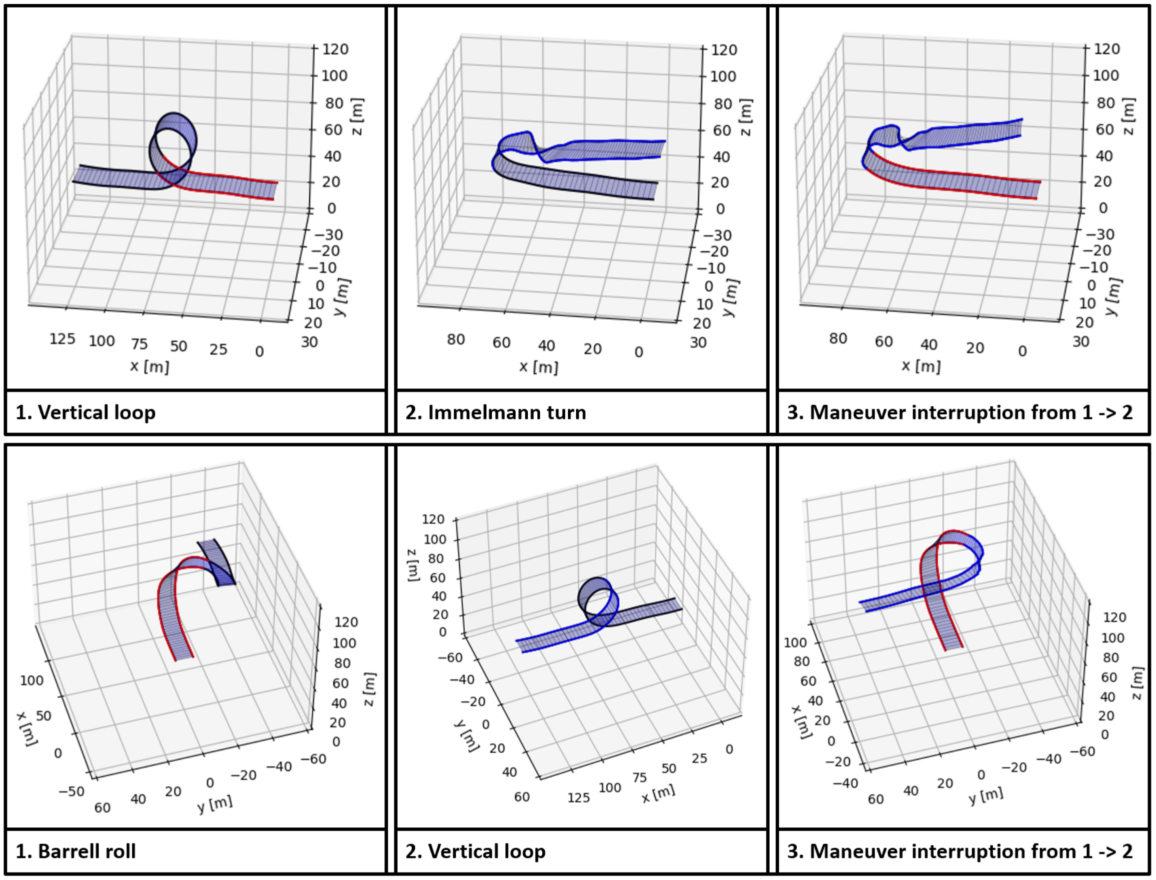


Fig. 9 Maneuver Interruption on pilot-generated maneuvers. The initial maneuver (1, red) is interrupted, and another maneuver (2, blue) begins before completing the first one. The final trajectory is shown in (3).

V. Conclusion

The paper presented a method of improving re-planning for kinodynamic planners using the Maneuver Automaton. Maneuver interruption improves the re-planning performance of planners using the Maneuver Automaton by allowing re-planning to happen during the "maneuver" phase of the Maneuver Automaton, where motion planning is typically fixed to a single trajectory. Maneuver Interruptions allow the planner to escape maneuvers mid-trajectory using dynamic trajectories previously explored in the motion primitive library.

To identify dynamically viable maneuver-to-maneuver connections, near-identical points in the state trajectories are identified within the motion primitive library. In combination with a linear controller that stabilizes the slight differences in the state trajectory, these connections form a network that enable Maneuver Interruption. The maneuver connections are pruned by utility, which is identified through Monte-Carlo simulations. Maneuver Interruption can then augment planning by performing maneuver-to-maneuver switches when necessary.

The paper demonstrated Maneuver Interruption on a flight dynamics model of the Drift, a fixed-wing RC aircraft. Mid-maneuver transitions between three different pilot-generated maneuvers were demonstrated on the Drift FDM. Monte-Carlo simulations proved to be effective in drastically reducing the size of the connection library while having minimal influence on re-planning performance. Lastly, the paper evaluates the performance of a Hybrid A* planner with and without Maneuver Interruption on a large set of randomly generated obstacle fields. The results showed that Maneuver Interruption provides better re-planning performance at the cost of some trajectory-tracking performance.

Acknowledgments

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by the National Technology and Engineering Solutions of

Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

[1] Frazzoli, E., "Maneuver-Based Motion Planning and Coordination for Single and Multiple UAVs," *1st UAV Conference*, Infotech@Aerospace Conferences, American Institute of Aeronautics and Astronautics, 2002. https://doi.org/10.2514/6.2002-3472.

[2] Frazzoli, E., Dahleh, M. A., and Feron, E., "Robust hybrid control for autonomous vehicle motion planning," *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, Vol. 1, 2000, pp. 821–826 vol.1. https://doi.org/10.1109/CDC.2000.912871, iSSN: 0191-2216.

[3] Liao, F., Lai, S., Hu, Y., Cui, J., Wang, J. L., Teo, R., and Lin, F., "3D motion planning for UAVs in GPS-denied unknown forest environment," *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 246–251. https://doi.org/10.1109/IVS.2016.7535393.

[4] Barry, A. J., Florence, P. R., and Tedrake, R., "High-speed autonomous obstacle avoidance with pushbroom stereo," *Journal of Field Robotics*, Vol. 35, No. 1, 2018. https://doi.org/10.1002/rob.21741.

[5] Paranjape, A. A., Meier, K. C., Shi, X., Chung, S.-J., and Hutchinson, S., "Motion primitives and 3D path planning for fast flight through a forest," *The International Journal of Robotics Research*, Vol. 34, No. 3, 2015, pp. 357–377. https://doi.org/10.1177/0278364914558017, publisher: SAGE Publications Ltd STM.

[6] Donald, B., Xavier, P., Canny, J., and Reif, J., "Kinodynamic motion planning," *Journal of the ACM*, Vol. 40, No. 5, 1993, pp. 1048–1066. https://doi.org/10.1145/174147.174150.

[7] LaValle, S. M., and Kuffner, J. J., "Randomized Kinodynamic Planning," *The International Journal of Robotics Research*, Vol. 20, No. 5, 2001, pp. 378–400. https://doi.org/10.1177/02783640122067453, publisher: SAGE Publications Ltd STM.

[8] Lau, B., Sprunk, C., and Burgard, W., "Kinodynamic motion planning for mobile robots using splines," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2427–2433. https://doi.org/10.1109/IROS.2009.5354805, iSSN: 2153-0866.

[9] Li, Y., Littlefield, Z., and Bekris, K. E., "Sparse Methods for Efficient Asymptotically Optimal Kinodynamic Planning," *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*, edited by H. L. Akin, N. M. Amato, V. Isler, and A. F. van der Stappen, Springer Tracts in Advanced Robotics, Springer International Publishing, Cham, 2015, pp. 263–282. https://doi.org/10.1007/978-3-319-16595-0_16.

[10] Karaman, S., and Frazzoli, E., "Incremental Sampling-based Algorithms for Optimal Motion Planning," *Robotics: Science and Systems VI*, 2010. https://doi.org/10.15607/RSS.2010.VI.034.

[11] Dolgov, D., Thrun, S., Montemerlo, M., and Diebel, J., "Practical Search Techniques in Path Planning for Autonomous Driving," *Proccedings of AAAI*, 2008.

[12] Dharmadhikari, M., Dang, T., Solanka, L., Loje, J., Nguyen, H., Khedekar, N., and Alexis, K., "Motion Primitives-based Path Planning for Fast and Agile Exploration using Aerial Robots," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 179–185. https://doi.org/10.1109/ICRA40945.2020.9196964, iSSN: 2577-087X.

[13] Majumdar, A., and Tedrake, R., "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, Vol. 36, No. 8, 2017, pp. 947–982. https://doi.org/10.1177/0278364917712421, publisher: SAGE Publications Ltd STM.

[14] Frazzoli, E., Dahleh, M., and Feron, E., "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, Vol. 21, No. 6, 2005, pp. 1077–1091. https://doi.org/10.1109/TRO.2005.852260.

[15] Park, S., Deyst, J., and How, J., "A New Nonlinear Guidance Logic for Trajectory Tracking," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2004. https://doi.org/10.2514/6.2004-4900, URL https://arc.aiaa.org/doi/10.2514/6.2004-4900.

[16] Goddard, Z. C., Wardlaw, K., Williams, K., Parish, J., and Mazumdar, A., "Utilizing Reinforcement Learning to Continuously Improve a Primitive-Based Motion Planner," *Journal of Aerospace Information Systems*, 2022, pp. 1–12.

[17] Papa, U., Del Core, G., Giordano, G., and Ponte, S., "Obstacle detection and ranging sensor integration for a small unmanned aircraft system," *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2017, pp. 571–577. https://doi.org/10.1109/MetroAeroSpace.2017.7999533.

[18] McGee, T., Sengupta, R., and Hedrick, K., "Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 4679–4684. https://doi.org/10.1109/ROBOT.2005.1570842, iSSN: 1050-4729.

[19] Ramasamy, S., Gardi, A., Liu, J., and Sabatini, R., "A laser obstacle detection and avoidance system for manned and unmanned aircraft applications," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 526–533. https://doi.org/10.1109/ICUAS.2015.7152332.

[20] Shaw, R. L., "Fighter combat," *Tactics and Maneuvering; Naval Institute Press: Annapolis, MD, USA*, 1985.