

COBRA-DDP: Trajectory Generation and Collision Avoidance Augmentations for eVTOL Vehicles

Matthew D. Houghton^{*}, Michael J. Acheson[†], Andrew Patterson[‡], and Irene M. Gregory[§]
NASA Langley Research Center, Hampton, VA, 23681

This paper presents a receding horizon model predictive control variation of the combined Bernstein polynomial optimal reciprocal collision avoidance (ORCA) differential dynamic programming (COBRA-DDP) algorithm for advanced air mobility (AAM) vehicles. Collision avoidance in combination with effective trajectory replanning are expected to be core components of AAM vehicles operating within a crowded airspace. This environment necessitates the use of real-time trajectory planning algorithms that are capable of planning around large numbers of stationary and moving obstacles. Previous work on COBRA-DDP demonstrated the capability of the algorithm to produce dynamically feasible trajectories for AAM vehicles and general collision avoidance. This paper improves upon the previous work by increasing the number of stationary and moving obstacles, while implementing a variation of COBRA-DDP that lends itself to real-time application. These advancements are demonstrated on an electric vertical takeoff and landing (eVTOL) vehicle simulation with highly nonlinear vehicle dynamics.

I. Introduction

The capability to autonomously plan trajectories, perform trajectory re-planning (as required) and safely execute the desired trajectories are foundational to autonomous execution of the Advanced Air Mobility mission. A wide array of trajectory plans are required such as: original flight plan, contingency plans (e.g., alternate field divert or weather and no fly area avoidance), intruder aircraft collision avoidance, and emergency plans due to system faults or failures. Autonomously handling inflight contingencies, such as system failures, necessitates the execution of the classic priorities taught to human pilots: Aviate, Navigate and Communicate. Thus, we require the ability to autonomously maintain control of the vehicle (Aviate) while planning (and executing) a corresponding safe trajectory (Navigate). Once these time critical tasks are underway, then non-time critical communication to the outside world improves system wide situational awareness. This wide array of trajectory plans requires a corresponding spectrum of timescales and dynamic complexity, which is depicted in Fig. 1. For example, a pre-flight flight plan (i.e., low complexity) is a simple trajectory with some core waypoints that satisfies both a low-fidelity fuel consumption model and the published flight departure, cruise, and terminal area flight rules. Conversely, an example trajectory requiring a high degree of dynamic complexity is a dynamically feasible reference trajectory provided to a tracking controller that provides performance guarantees under uncertainty to perform a specific highly dynamic flight maneuver. Clearly, the timescales and modeling complexity needs of these two trajectories differ substantially.

This paper addresses the first two priorities (Aviate and Navigate) of executing a desired trajectory using a model predictive control variant of differential dynamic programming (DDP) coupled together with a trajectory replanner that is a blend of Bernstein polynomials (BP) with the Optimal Reciprocal Collision Avoidance (ORCA) algorithm. The combination of DDP with Bernstein polynomials and ORCA, combined Bernstein polynomial optimal reciprocal collision avoidance differential dynamic programming (COBRA-DDP), was introduced in our previous work [1]. That work detailed the collision avoidant trajectory planner COBRA-DDP and assessed its ability to quickly plan dynamically feasible trajectories for electric vertical takeoff and landing (eVTOL) vehicles. This prior work highlighted potential adjustments to COBRA-DDP that would increase its effectiveness for eVTOL applications. The previous implementation proved beneficial by demonstrating a trajectory planning approach with the capability to provide dynamically feasible collision avoidant trajectories and warm start other state-constrained algorithms. However, it was limited in its real-time applicability by planning the entire trajectory in one large optimization (which in many cases is intractable given the time constraints) and only demonstrates single obstacle avoidance. In response, the focus

^{*}Research Engineer, Dynamic Systems and Control Branch, Matthew.D.Houghton@nasa.gov.

[†]Senior Research Engineer, Dynamic Systems and Control Branch, Michael.J.Acheson@nasa.gov, Member AIAA.

[‡]Research Engineer, Dynamic Systems and Control Branch, Andrew.Patterson@nasa.gov.

[§]NASA Senior Technologist for Advanced Control Theory and Application, Irene.M.Gregory@nasa.gov, Fellow AIAA.

Trajectory Generation

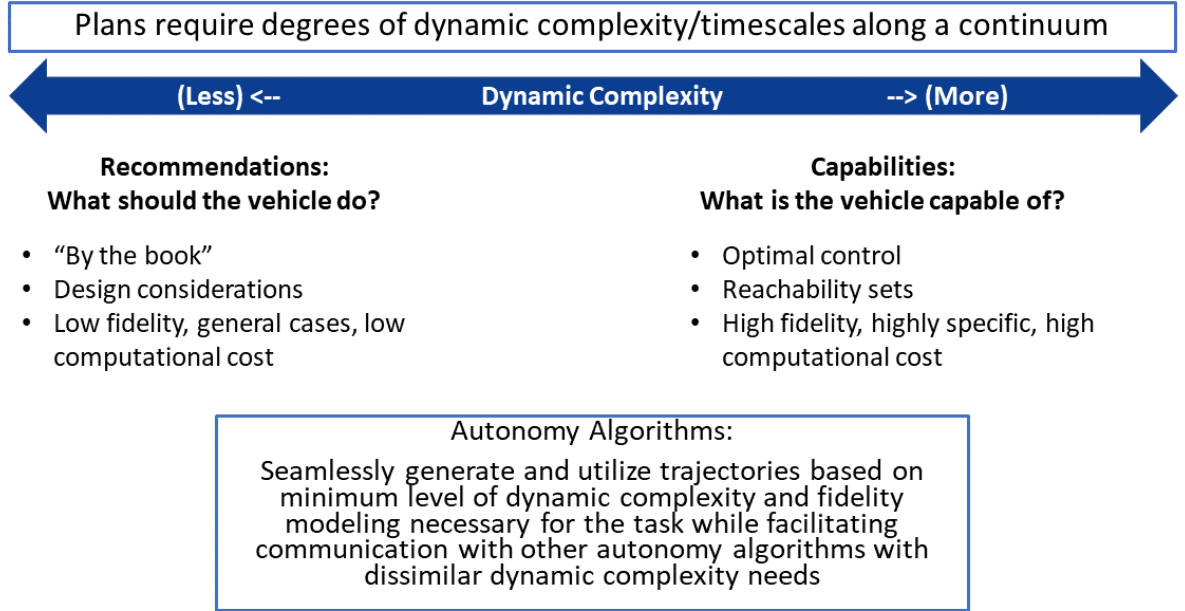


Fig. 1 Trajectory Dynamic Complexity Continuum

of this work is to introduce a receding horizon model predictive control (MPC) variant of COBRA-DDP, which we demonstrated in a simulated crowded airspace environment.

The paper is organized as follows: Section II will provide an overview of the COBRA-DDP algorithm and its components. Section III describes how to implement the receding horizon MPC variation of the algorithm, Section IV explains the experiments and results of COBRA-DDP, and finally Section V discusses the effectiveness of the MPC variant and future work.

II. COBRA-DDP

COBRA-DDP is a collision avoidance trajectory replanning algorithm that provides immediate collision avoidance recognition across the entire trajectory while considering the dynamic constraints of the vehicle. The implementation introduced in [1] describes the full trajectory planning variation as well as the three core components of the algorithm, ORCA, BP curves, and DDP. ORCA is a collision avoidance algorithm used to detect collisions along the trajectory and employs simplified dynamics to calculate a new trajectory that is guaranteed to be safe (under the linearity assumptions of ORCA) [2], using the velocity obstacle representation from [3] to do so. The use of simplified dynamics allows the algorithm to accommodate a large number of moving and stationary obstacles when planning trajectories. optimal reciprocal collision avoidance (ORCA) has the capability for planning around both cooperative and non-cooperative obstacles making it ideal for trajectory planning within a crowded airspace.

DDP, originally developed by Jacobson and Mayne [4, 5], is an optimal control approach that has been shown as an effective trajectory planner for advanced air mobility (AAM) vehicle configurations [6, 7] due to its ability to generate dynamically feasible trajectories for vehicles with highly nonlinear dynamics. The algorithm and its variants enable this type of planning by using second-order approximations along a nominal trajectory and display quadratic convergence to a local solution.

Bernstein polynomial (BP) curves are used to represent the vehicle trajectories calculated by ORCA and DDP and serve as the means to transfer information between them. Piece-wise BP curves provide fast computation, compact trajectory representation, and seamless integration of position, velocity, and acceleration. This polynomial representation [8] has two critical properties: BPs have closed form derivatives and have efficient algorithms for computing bounds on their curves from the parameters. These properties allow the curves to exactly meet terminal constraints on their

derivatives [9] and their intermediate values are bounded. The ORCA algorithm outputs a collision avoidant trajectory using position and velocity values. Using the BP framework, this future position and velocity can be used to construct a continuous collision avoidance maneuver. The bounds improve the dynamic feasibility of the trajectory before use in DDP for optimization; a useful benefit as DDP is a local trajectory optimizer that improves significantly when warm started with a reasonable nominal trajectory. A description on computing bounds on this class of curves is given in [10].

A. ORCA & Bernstein Polynomial Interaction

In this section, the combination of ORCA and Bernstein polynomials (also known as Bézier curves) is detailed. As a collision avoidance algorithm, ORCA can handle both stationary and moving intruder objects, cooperative (joint) and non-cooperative collision avoidance, a variety of convex intruder object shapes, accommodation any number of intruder objects, and specification of preferred avoidance direction. ORCA is very computationally efficient and its computational cost scales nearly linearly with the number of obstacles and obstacle complexity, where each obstacle shape is combined to create a single convex hull velocity object. The main drawback of ORCA is that it assumes instantaneous response (e.g., instantaneous turn rate) and as such, the resulting avoidance manner is often not dynamically feasible. Similarly, Bernstein polynomials have a number of advantages that make them suitable for trajectory specification. Bernstein polynomials are very compact representations which can be efficiently (computationally) interrogated at any point along the specified trajectory. Bernstein polynomials are smooth and continuously differentiable for the number of specified derivatives, and they provide mathematical assurance that the resulting curves are contained within the convex hull delineated by their control points.

Our prior work blending ORCA and Bernstein polynomials required knowledge of a known ownship Bernstein polynomial trajectory representation and estimated intruder's Bernstein polynomial trajectories. The ownship and intruder trajectories were sampled periodically in the future, to supply respective positions and velocities to the ORCA algorithm. If a pending proximity collision detection occurs, then the ORCA algorithm provided the point in space that is a safe point consisting of a spatial position and time combination that is guaranteed to provide collision free linear transit from the initial sampling point/time to the ORCA output safe point (position/time). The safe point is simultaneously closest to the boundary of the collision volume while also nearest in alignment with the user specified preferred ownship velocity direction. The ownship position and velocity information at the trajectory sampled point in conjunction with the same information at the ORCA safe point, was sufficient to uniquely specify Bernstein polynomial waypoints. These new waypoints were then incorporated into the existing ownship trajectory polynomial. Additional sampling/ORCA pass(es) were conducted to ensure that the curved polynomial transition from the sampling to safe point was also collision free and/or to add additional waypoints to start the avoidance maneuver earlier to ensure the polynomial curvature was both (heuristically) dynamically feasible and collision free.

As this algorithm has no knowledge of actual vehicle dynamics, the best it could do was provide a Bernstein polynomial trajectory that should be generically dynamically feasible (e.g., limit turn rates, bank angles, and acceleration) for typical aerospace vehicles (e.g., specify maximum as standard turn rate). The results of the combined Bernstein polynomial/ORCA collision avoidance algorithm could be considered a 3-DOF representation of a dynamic trajectory that any generic aerospace vehicle should be able to track successfully. This resulting medium complexity trajectory satisfied the continuum needs for low fidelity requirements such as meeting collision avoidance criteria and providing the ability to specify rules-of-the-road behaviors (e.g., right to right passes). It also provided the high complexity ability to interrogate trajectory points for position, velocity, accelerations, etc. Thus, to ensure that the replanned trajectory was dynamically feasible for an eVTOL vehicle, the incorporation of DDP was required.

B. Bernstein Polynomial & DDP Interactions

DDP is an optimal control and trajectory generation algorithm that simultaneously solves for an optimal trajectory and the associated optimal controller for a given cost function (i.e., running and terminal costs). In its simplest forms, DDP solves the unconstrained state problem with constrained control effectors. However, a wide variety of DDP algorithms exist, including state constrained variations (i.e., that enforce obstacle avoidance) but at a substantial computational cost. In this work, the computationally efficient unconstrained state variation (i.e., DDP does not check for collision avoidance) is employed as simply a trajectory tracking controller, and the BP ORCA algorithm is used to enforce obstacle avoidance.

DDP optimizes a trajectory for highly nonlinear vehicle dynamics by using second-order approximations along a nominal trajectory to achieve quadratic convergence to a local minimum. While DDP convergence is guaranteed under mild assumptions, because DDP is iterative, the number of iterations required for convergence are influenced by the

initial state and control trajectories provided to the algorithm. Thus, in order to reduce the computational effort to arrive at a solution, work is done to provide DDP with a reasonable set of warm start control actions. This can, at times, be a complicated process; however, an advantage of the COBRA-DDP algorithm is that this information is readily available due to the properties of a BP curve. Once the simplified ORCA trajectory has been converted into a BP curve with general dynamic considerations, the BP curve and its derivatives (which are also BP curves) can be sampled at any time or arbitrary rate. DDP leverages this information to generate trajectory states and/or controls at the sampling rate necessary to optimize for a dynamically feasible trajectory.

For the specific case of the NASA Lift+Cruise eVTOL vehicle [11], DDP requires the following states to generate a feasible trajectory: inertial frame position (x, y, z) , body velocities (u, v, w) , Euler angles (ϕ, θ, ψ) , and body rates (p, q, r) . For controls, DDP applies the speed of the pusher propeller, the speed of the eight lifting rotors, and the deflection angle of the elevator, aileron, and rudder. The position information can be converted from the BP parameterized positions, $\zeta_{x,y,z}$, while the body velocities and Euler angles are analytically converted from the BP curve's velocity, ζ_{v_x, v_y, v_z} at each timestep. For this specific simulated vehicle, linearly interpolated values from a set of trim conditions based on the heading frame forward and vertical velocities, u and v are used to fill in the remaining states and control values. While these values are not necessarily dynamically feasible, they serve as a strong initial set of states and controls for DDP to optimize over to reach a feasible collision avoidant trajectory for the eVTOL vehicle. Figure 2 depicts the conversion from BP curve information to DDP states and controls. Further detail of this process and the conversion back to BP waypoints is laid out in [1].

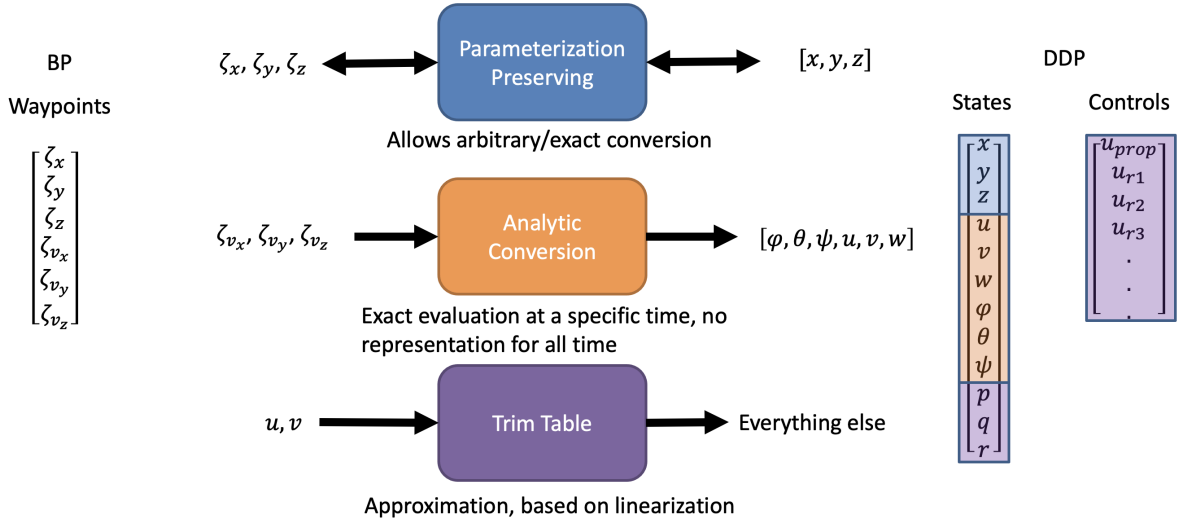


Fig. 2 Bernstein Polynomial and DDP State Information Transfer Process

The combination of ORCA collision detection and DDP dynamic feasibility planning form the core of the COBRA-DDP algorithm, with BP curves serving as the ideal information transfer between the two. The purpose of this structure is to incorporate as much relevant information from the different dynamic complexity levels, retain dynamic feasibility for the trajectory, and do so in a way that is as computationally efficient as possible. Converting waypoint information into controls for a dynamically feasible trajectory that is capable of collision avoidance, following of predefined rules-of-the-road, and taking into account highly nonlinear vehicle dynamics provides a strong start to tackling autonomous flight. A visual of the trajectory-based COBRA-DDP algorithm progression is shown in Fig. 3.

III. Receding Horizon Model Predictive Control Implementation

This section will discuss the changes made from the original full trajectory COBRA-DDP algorithm implementation necessary for receding horizon (RH) MPC. In a dynamically changing environment, a previously optimal trajectory may no longer be a viable solution as obstacle and state information are updated. This requires consistent replanning of the trajectory as the environment changes. Unfortunately, it is typically intractable for a fully nonlinear dynamically feasible trajectory for the entire flight path to be replanned with each environmental update. Therefore, a receding

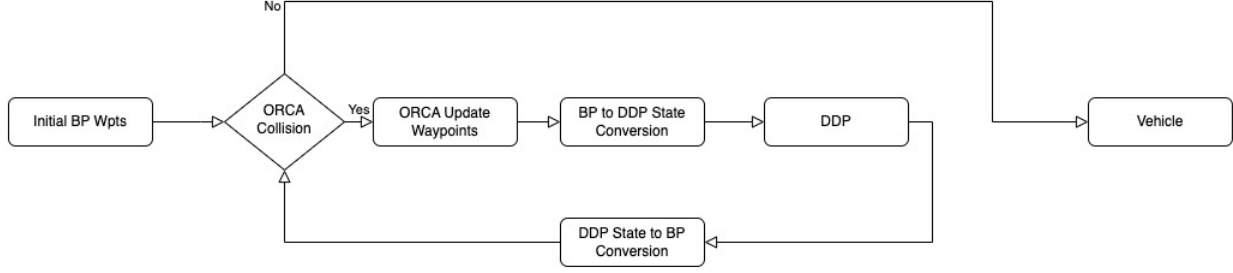


Fig. 3 Trajectory based COBRA-DDP Flow Chart

horizon model predictive control approach is implemented where the immediate subsection of the full trajectory is planned and updated at a specified time interval. This subtrajectory is small enough to be optimized within the specified time. It is important to note here that DDP has been selected by the authors due to the strength and speed of the algorithm for handling highly nonlinear dynamics as well as its useful variations. For both the previous work [1] and the current receding horizon (RH) MPC implementation of combined Bernstein polynomial optimal reciprocal collision avoidance (COBRA), many other optimal control algorithms are capable of replacing DDP in this work, the implementation should have an iterative variation. Readers are welcome to swap DDP for their preferred optimal control algorithm when considering the COBRA algorithm.

DDP has a RH MPC implementation that will serve as the general example for this formulation [12]. The ORCA implementation speed is such that much larger sections of trajectory are able to be assessed for collision and replanning than DDP can optimize. Therefore, asynchronous updates will be made by the DDP and ORCA components of COBRA-DDP. Adjustments necessary for each section of the algorithm will be discussed further in the section.

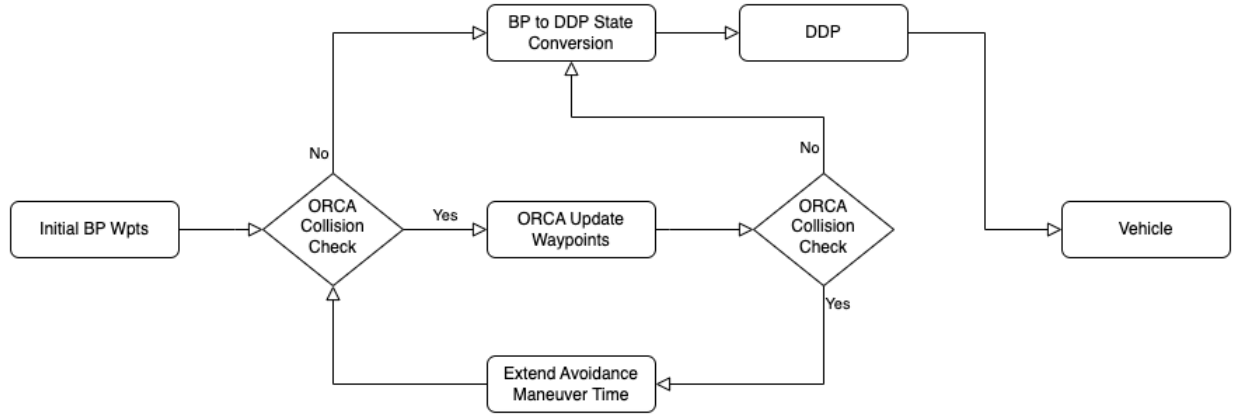


Fig. 4 MPC Trajectory based COBRA-DDP Flow Chart

This work extends COBRA-DDP with a receding horizon formulation with the algorithm depicted in Fig. 4. Here, the time horizon of DDP has been decreased from the long-term trajectory to a subtrajectory that is able to be optimized given realistic time constraints. As such, DDP has been moved in the flow chart and optimizes over the immediate short time horizon to provide the control actions to the vehicle. The extending of avoidance maneuver time block occurs after ORCA collision checks have corrected for a potential collision, but the dynamic constraints applied by the BP curve still results in a collision. For further information on this process, see Appendix A of [1]. The benefits of this extension are the typical benefits of receding horizon control: that the computation burden of the optimization problem is reduced, and that new information is incorporated with each updated optimization horizon which leads to implicit feedback compensation that can help reduce online tracking errors. In addition, the COBRA component helps address some of the pitfalls of receding horizon algorithms, where the length of the prediction horizon can drastically change performance in certain applications. Combined with the COBRA algorithm, future information is provided to the tracking controller in the form of a reference input that is designed to address long term planning concerns. The

separation of timescales allows the collision avoidance and planning problems to be addressed independently of the DDP-based tracking controller. The planning problem solution is required to be in the form of a Bernstein polynomial that can provide future information to DDP as a reference trajectory, then DDP is used to ensure feasibility optimizing in the neighborhood of a safe, well-planned long-term trajectory. Ideally, DDP would be able to find a feasible trajectory that is locally optimal within that same neighborhood. The size of this neighborhood, that contains both the low-fidelity plan and a locally optimal trajectory, could be used to inform the higher-level planner for the creation of a safety radius. Computing this neighborhood a priori is an open problem, and the current implementation relies on a posterior verification of the safety constraints, with replanning as needed. It is also worth noting that shortening the time horizon of DDP also required a reworking of the approach of the cost function. Instead of optimizing toward a specific long-term maneuver or target state, the cost function focused on maintaining the BP sub-trajectory for the given time horizon. The cost function therefore has to be capable of optimizing regardless of the long-term planned maneuver. Demonstrating that a single cost function was suitable for all stages of flight, vertical takeoff, transition, and fixed wing cruise, simplifies the process for tuning experiments with these vehicles.

The previous architecture used DDP as a long term planer; therefore, it was important that DDP trajectory be reconverted to Bernstein polynomial form and checked for collision using ORCA. This conversion process would reduce some of the benefit of DDP by converting a high-fidelity feasible trajectory into a low-fidelity trajectory that may have tracking errors. By implementing DDP in a RH fashion, collision checking becomes less important since we are only optimizing over a small section of a desired trajectory. The deviation between the ORCA safe trajectory and the DDP trajectory is expected to be small, since the BP trajectory from ORCA is chosen to be heuristically feasible. If we have a bound on the tracking performance of DDP, the safety radius can be increased to compensate.

IV. Experiment Results

In this section, the basic operation of the combined ORCA and piece-wise Bernstein polynomial algorithm is demonstrated with an example trajectory. The experiments were performed in a full non-linear 6-DOF, Matlab® * and Simulink® environment that contains a medium fidelity aero-propulsive model [13] of the NASA's Revolutionary Vertical Lift Technologies (RVLTL) Lift+Cruise UAM vehicle [11]. Figure 5a shows a typical transition vehicle pre-flight (planned) trajectory. The thirteen minute flight consists of a hover climb, a vehicle transition to wing-borne flight during the departure procedure, a short duration cruise, followed by a cruise descent and transition to hover for a landing. Unbeknownst during the pre-flight planning, two intruder aircraft traveling opposite to ownship will be encountered during cruise, and a large stationary avoidance area (e.g., no fly area, localized weather, or noise abatement area) will be encountered during descent/approach. These pending/potential contingencies will be denoted as obstacles 1-3 in the sequel. For each obstacle and ownship, a safety sphere is defined. The safety radii for ownship and the obstacles are provided to ORCA for use in generation of the collision velocity obstacles. In this example the generated safe trajectories will ensure that ownship and obstacle(s) safety spheres do not coincide at any time. The ownship safety radius is 100 ft., and the obstacle radii are 300, 400 and 5000 ft. respectively.

Figure 5b shows that if the original trajectory were flown exactly, then minimum separation distances (sum of ownship and respective obstacle safety radii) would be violated for both the first and third obstacles (not the second obstacle). The ORCA Bernstein combination does not have or require full knowledge of the obstacles a priori. Instead, the algorithm utilizes a nominal look ahead time (15 seconds for this example), which equates to the algorithm having knowledge of obstacle position and velocity within a realistic perception time frame. In this example, the obstacle trajectories intercede with the ownship trajectory at different times. This results in the algorithm replanning a new trajectory based on a different current trajectory in a sequential manner. For example, the current trajectory starts with the pre-flight plan, but upon encountering obstacle 1, a trajectory replan is required (referred to as maneuver 1 trajectory) that not only avoids obstacle 1 but also completes the rest of the future original trajectory waypoints. When obstacle 2 is encountered, the maneuver 1 trajectory (not the original pre-flight trajectory) is being flown and used as the basis to determine if a pending safety violation exists, which requires trajectory replanning.

Next, we examine each obstacle encounter and any associated trajectory replanning in detail as shown in Fig. 6. Maneuver 1 is shown in Figs. 6a and 6b. Figure 6a shows that the maneuver 1 replan trajectory occurs early in cruise and consists of a benign (level) right hand turn to enforce a left-to-left pass for opposing (head-on) aircraft. Figure 6b shows that minimum separation violations (less than 400 feet) would occur at ≈ 197 seconds for the pre-flight trajectory and that maneuver 1 ensures adequate minimum separation is achieved.

*The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

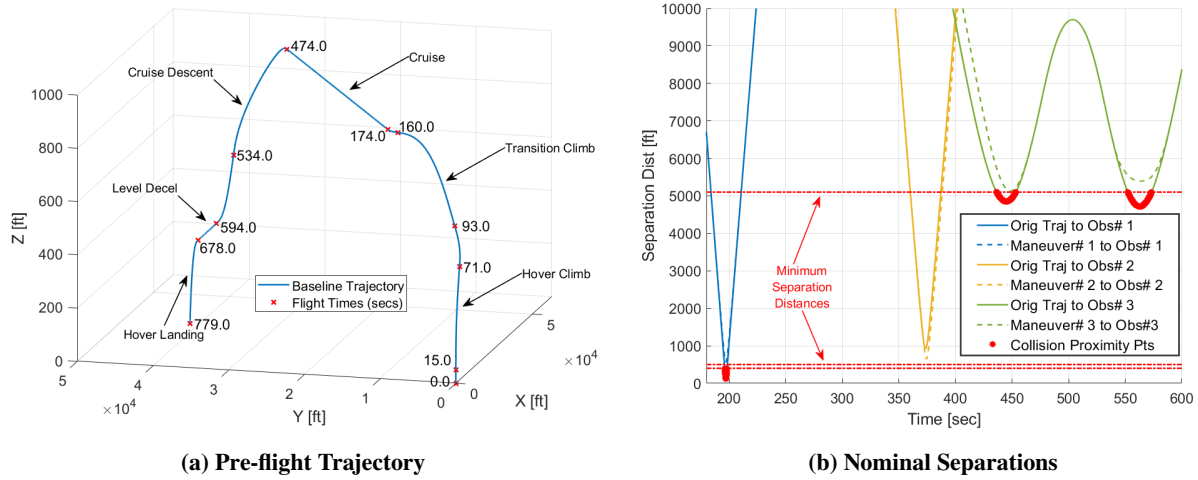


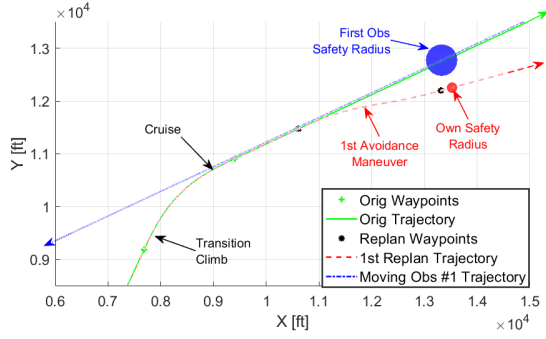
Fig. 5 Nominal Trajectory Separation Distances

Figure 6c shows that when the UAM vehicle flies maneuver 1 trajectory, minimum separation violations occur (near the end of the cruise segment) which necessitates a trajectory replanning yielding maneuver 2 trajectory which is a level right hand turn resulting in another left-to-left pass. Note that Fig. 6d shows that if the pre-flight (Original trajectory) was flown then no collision conflict would have occurred. However, since the maneuver 1 trajectory is being flown (based on the previous avoidance maneuver) then obstacle 2 becomes a concern based on the planning response to obstacle 1. Figure 6d shows that adequate separation is achieved with a second avoidance maneuver using ORCA with BP.

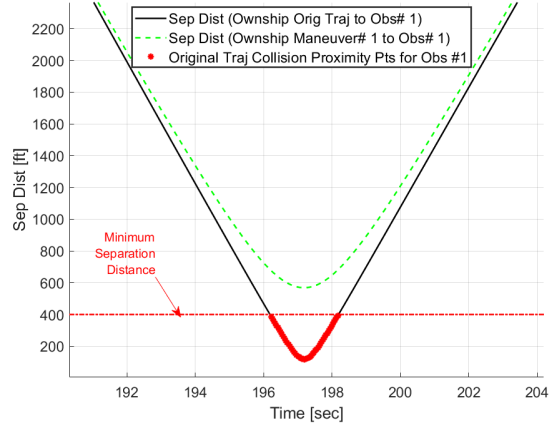
Obstacle 3 is encountered during the descent and maneuver to the terminal area. Recalling from Fig. 5b that the original trajectory minimum distance violations would occur as ownship approached the stationary obstacle and again as it was passing the stationary obstacle a second time. However, as the maneuver 2 trajectory is being flown, then Fig. 6e shows that no avoidance maneuver is now required when approaching obstacle 3, but that a maneuver is required on the second approach to obstacle 3 during maneuvering to the terminal approach. Figure 6f confirms that the maneuver 2 trajectory does indeed resolve the first conflict at ≈ 445 seconds, but that a substantial conflict resolution maneuver is required for the second approach at time ≈ 563 seconds. Again, a nearly level right-hand turn resolves the minimum separation situation. For this realistic example, four separate (sequential) trajectories were required using a nominal perception time of 15 seconds to safely navigate the dynamic situation based on ownship and obstacle trajectories but also based on ownship dynamic replanning (i.e., some future conflicts resolved but new ones were created).

Utilizing the trajectory information from ORCA and BP as the experiment proceeds, MPC DDP converts each subtrajectory into a dynamically feasible set of states and controls to implement. The resulting position, body velocity, and Euler angles for the entire experiment are depicted in Fig. 7. DDP's optimization allows for the vehicle's implemented trajectory to closely match the desired states of the collision free ORCA Bernstein trajectory. The average position error for the entire flight is 0.293 meters. While not a substantial difference from the collision free trajectory, this information can be used to increase the safety radius considerations of the ownship in order to more accurately consider how the interaction between ORCA-Bez and DDP trajectories affects collision avoidance guarantees. Further, we can note that the rate and velocity time histories deviate from the reference trajectory to compensate for uncertainties and inadequacies of the trim table during flight. These deviations are all of appropriate scale and frequency, though they may appear fast due to the length of the time history shown.

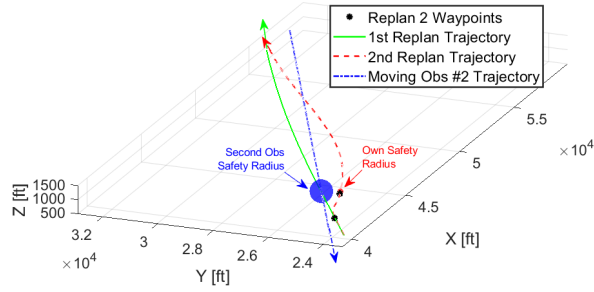
While these experiments were not conducted on dedicated real-time hardware, it should be noted that the ORCA and BP algorithm was run with a one second update rate for own-ship trajectory and only examined a 15 second look ahead window, which can readily be performed in real-time. The process to obtain DDP required states from BP uses extremely fast BP interrogation routines, and the DDP initial control estimates were obtained using a gain scheduled controller table lookup. The MPC and DDP combination only performed one iteration of DDP for a short 2 second duration trajectory and utilized the unconstrained (very computationally efficient) baseline formulation of DDP. Thus, the MPC and DDP combination is also readily attainable in real-time.



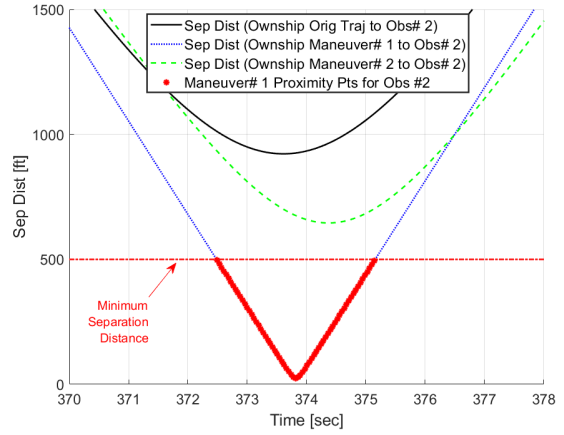
(a) Avoidance Maneuver 1



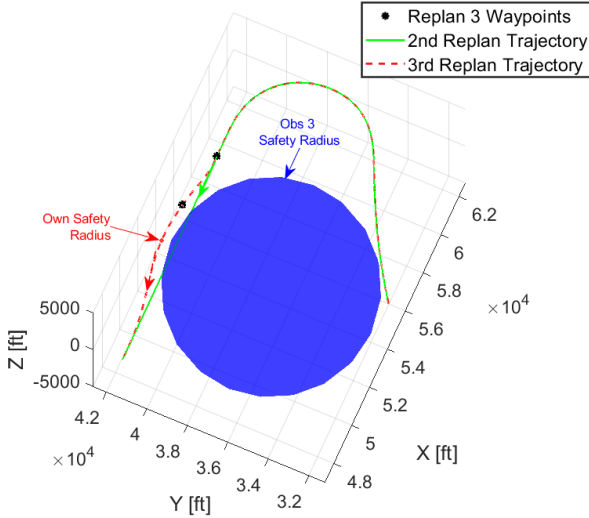
(b) Maneuver 1 Separation



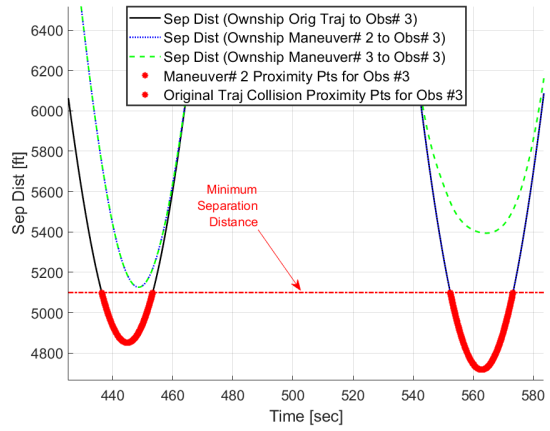
(c) Avoidance Maneuver 2



(d) Maneuver 2 Separation



(e) Avoidance Maneuver 3



(f) Maneuver 3 Separation

Fig. 6 Maneuver Effects on Separation Distances

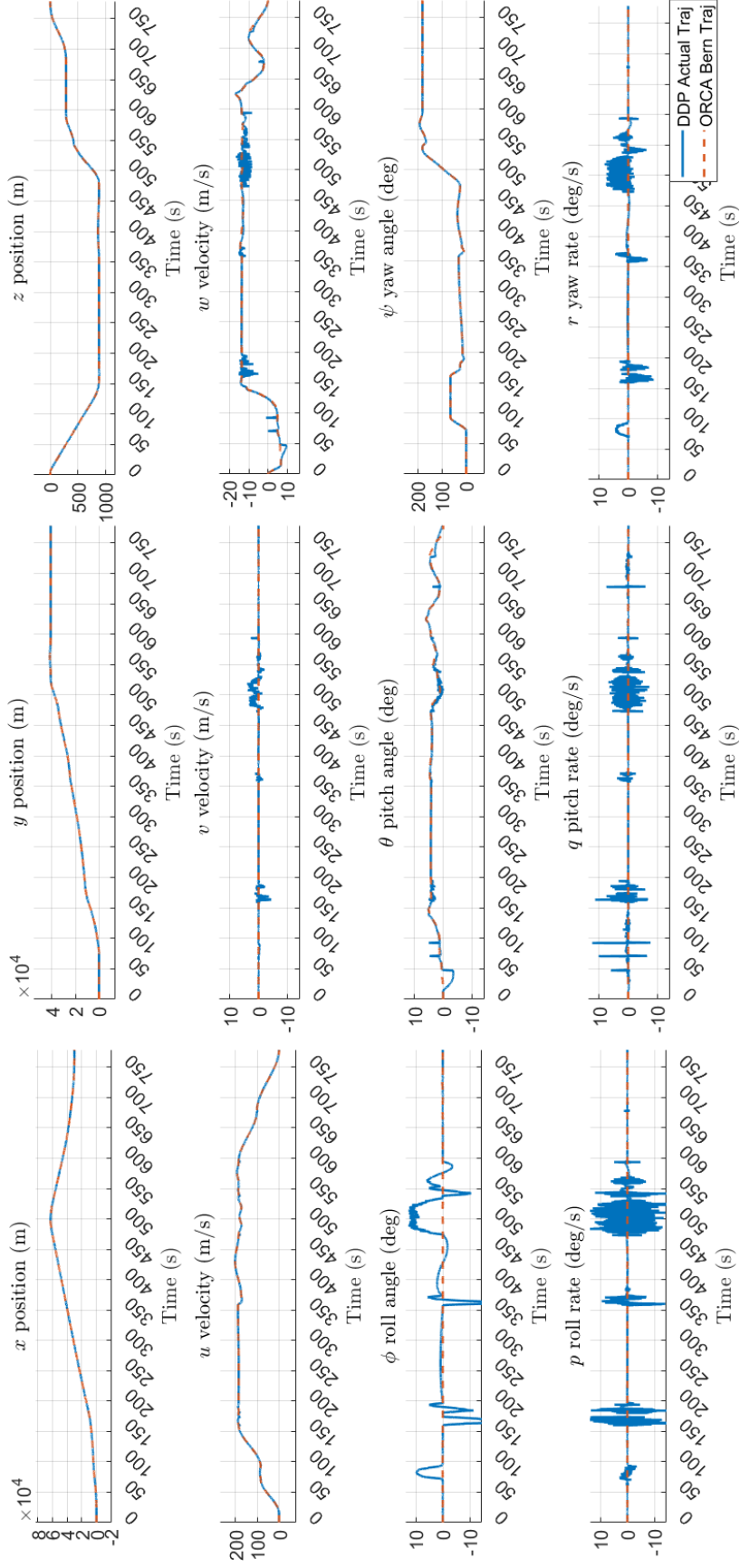


Fig. 7 MPC DDP state plots for Experiment Case

V. Discussion and Future Work

This work demonstrated the capabilities of a receding horizon model predictive control COBRA-DDP implementation for AAM eVTOL vehicles. By adjusting the formulation of the original algorithm from longer scale trajectory generation to an iterative shorter scale time horizon, this variation of COBRA-DDP is more suited to handle real-time trajectory planning and replanning scenarios. The primary experiment above demonstrated the effectiveness of warm starting the DDP optimization process with collision free BP transformed controls and trajectories. By shortening the time horizon of DDP, this also required a reworking of the approach of the cost function. Instead of optimizing toward a specific long term maneuver or target state, the cost function focused on maintaining the BP subtrajectory for the given time horizon. The cost function, therefore, has to be capable of optimizing regardless of the long-term planned maneuver. Demonstrating that a single cost function was suitable for all stages of flight, vertical takeoff, transition, and fixed wing cruise, simplifies the process for tuning experiments with these vehicles.

While this work's experiments demonstrated the capability of avoiding collisions with multiple obstacles, both stationary and moving, the primary benefit of ORCA and BP is more effectively demonstrated when encountering a substantial number of obstacles. Future work is planned to demonstrate these advantages in relevant locations by having simulations incorporate trajectory planning and replanning with obstacle maps of urban environments as well as moving obstacles.

Further considerations must be taken in account when determining how an AAM vehicle trajectory is generated. While a starting and goal waypoints are required, there may be trajectory replanning that requires higher level decision making (e.g., rules of the road, traffic pattern entry) that will necessitate a dynamic update to the ORCA preferred velocity collision avoidance decision and/or Bernstein polynomial selection. Additionally, for a given obstacle landscape, there are often multiple trajectory solutions that achieve desired separation, where one solution is preferred for other considerations. Future work will demonstrate examining multiple Bernstein polynomial trajectory options concurrently while seeking to enforce higher level objectives.

Acknowledgments

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] Houghton, M. D., Acheson, M. J., Patterson, A. P., Oshin, A., and Gregory, I. M., "Combined Bernstein Polynomial Collision Avoidance Differential Dynamic Programming for Trajectory Replanning and Collision Avoidance for UAM Vehicles," *AIAA SciTech 2023 Forum*, 2023. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2023-2544>.
- [2] van den Berg, J., Guy, S. J., Lin, M., and Manocha, D., "Reciprocal n-body Collision Avoidance," *In Proc. of the IEEE International Conference on Robotics and Automation*, 2010.
- [3] Fiorini, P., and Shiller, Z., "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, 1998.
- [4] Mayne, D., "A Second-Order Gradient Method for Determining Optimal Trajectories of Non-Linear Discrete-Time Systems," *International Journal of Control*, Vol. 3, No. 1, 1966, pp. 85–95.
- [5] Jacobson, D. H., and Mayne, D. Q., *Differential Dynamic Programming*, American Elsevier Pub. Co., 1970.
- [6] 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.
- [7] Oshin, A. B., Houghton, M. D., Acheson, M. J., Gregory, I. M., and Theodorou, E. A., "Parameterized Differential Dynamic Programming," *Robotics: Science and Systems*, 2022.
- [8] Farouki, R. T., "The Bernstein Polynomial Basis: A Centennial Retrospective," *Computer Aided Geometric Design*, Vol. 29, No. 6, 2012, pp. 379–419.
- [9] Patterson, A., MacLin, G. L., Acheson, M. J., Tabasso, C., Cichella, V., and Gregory, I. M., "On Hermite Interpolation using Bernstein Polynomials for Trajectory Generation," Tech. Rep. TM-2023-0013467, National Aeronautics and Space Administration, Hampton, VA, USA, 2023.

- [10] Lakshmanan, A., Patterson, A., Cichella, V., and Hovakimyan, N., “Proximity Queries for Absolutely Continuous Parametric Curves,” *Robotics: Science and Systems XV*, Freiburg im Breisgau, Germany, 2019.
- [11] Silva, C., Johnson, W. R., Solis, E., Patterson, M. D., and Antcliff, K. R., “VTOL urban air mobility concept vehicles for technology development,” *Aviation Technology, Integration, and Operations Conference*, 2018, p. 3847.
- [12] Tassa, Y., Erez, T., and Smart, W., “Receding Horizon Differential Dynamic Programming,” *Advances in Neural Information Processing Systems 20-Proceedings of the 2007 Conference*, 2007.
- [13] Murphy, P. C., Buning, P. G., and Simmons, B. M., “Rapid Aero Modeling for Urban Air Mobility Aircraft in Computational Experiments,” *AIAA SciTech Forum*, January 2021. URL <https://www.doi.org/10.2514/6.2021-1002>.