AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-2760

A Leader-Follower Control Strategy Built and Refined using Relational Maneuver Primitives for Approximating Optimal Trajectories in Real-Time

Carl A. Gotwald* and Michael D. Zollars†
Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 45433

Jonah A. Reeger‡
Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 45433

Isaac E. Weintraub§
Aerospace Systems Directorate, Wright-Patterson Air Force Base, OH, 45433

The work herein presents a new approach to aircraft control within a leader-follower formation, and the refinement process used in its development. During autonomous flight, a follower aircraft is tasked to rejoin and maintain a designated formation position, which is defined dynamically in reference to a maneuvering leader aircraft. Simple control algorithms, named Relational Maneuver Primitives, are intelligently combined into a control strategy such that the autonomous follower aircraft's response approximates the optimal trajectory. This research specifically addresses the control strategy refinement process where a genetic algorithm is implemented to determine the appropriate selection of Relational Maneuver Primitives based on the Time-To-Go between the follower aircraft and the desired goal position. The control strategy takes the form of a lookup table database based on the current state of the follower aircraft which provides the autopilot with the appropriate Relational Maneuver Primitive to use over the next time step. Two test cases are presented which demonstrate a 36% and 48% reduction in error between the follower response trajectory and the follower optimal trajectory when compared to using a single Relational Maneuver Primitive.

I. Introduction

Manned-Unmanned Teaming (MUM-T) is used to frame the interactions between manned weapon systems and unmanned support systems. This concept was first used by the army in the Apache AH-64D which is discussed in detail by Van Riper [1] and Durbin [2]. The Air Force has expressed its implementation of the MUM-T construct with the Loyal Wingman concept. The Loyal Wingman concept consists of an autonomous wingman supporting a manned leader aircraft. A thorough discussion of optimal control trajectories of an uninhabited Loyal Wingman is covered in [3] with further discussion in [4, 5]. This paper explores the Loyal Wingman concept using a leader-follower construct, and attempts to approximate the optimal trajectory with a control strategy that can run in real-time.

In this work the leader aircraft flies a known trajectory, while the autonomous follower aircraft is guided via the derived control strategy. The control strategy is a decision engine which determines which Relational Maneuver Primitive (RMP) to employ at a given time step. The RMPs were designed to be simple control laws which generate a total acceleration command based on the current follower aircraft state in relation to the leader aircraft. The control strategy was created using a heuristic refinement process which determines the best combination of a set of 5 RMPs to employ for a given Time-To-Go (TTG) from the follower aircraft's current state to the designated goal position.

II. Background

The background section begins with a description of the individual RMPs and their underlying principles based on a Park Guidance control scheme [6]. Then a description of the control simulation environment is given which was used to

*Maj, USAF, PhD Student, Department of Aeronautics and Astronautics
†Lt Col, USAF, PhD, Assistant Professor, Department of Aeronautics and Astronautics
‡PhD, Assistant Professor, Department of Mathematics
§PhD, Electronics Engineer, Control Science Center, Air Force Research Laboratory, AIAA Senior Member

simulate the response of the follower aircraft. Next, a discussion of the actual implementation of the control strategy is provided, including the structure and how TTG was determined for a given time step. A summary of how the optimal trajectory was determined is provided, which is based on previous work seen in [7]. Lastly, an overview of genetic algorithms is provided as the heuristic technique used in the control strategy refinement process. The focus of this paper is the process used to refine the control strategy based on a given leader trajectory and initial condition such that the follower response approximates the optimal trajectory.

A. Park Guidance

Park guidance approximates traditional Proportional-Derivative (PD) controller guidance, but adds an element of anticipatory control which enables tight tracking when following curved paths as documented in [6]. An overview of the Park guidance geometry can be seen in Fig. 1. The control guidance selects a lead point ahead of the goal position

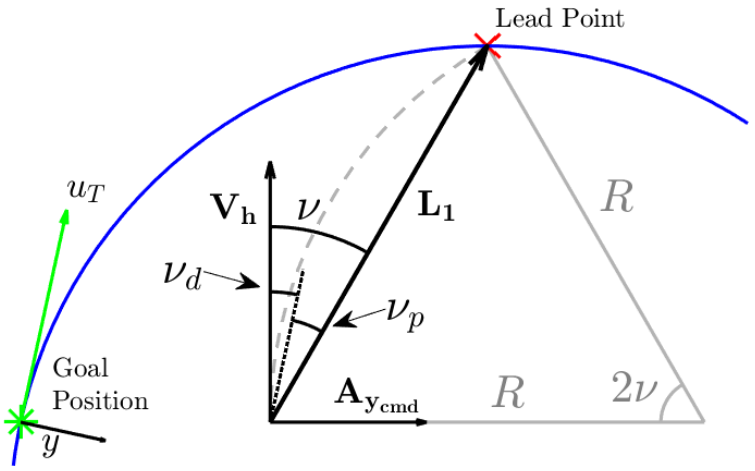


Fig. 1 Overview of Park Guidance Geometry

along the desired trajectory based on the intersection between the desired trajectory in blue, and a circle with the user selected radius L_1 and centered at the follower aircraft's location. The park guidance controller then generates a lateral command that is equivalent to the centripetal acceleration required to follow the instantaneous circular segment defined between the follower aircraft's velocity vector and the lead point as depicted by the gray dashed line. This acceleration command is given by Eq. (1).

$$A_{y_{cmd}} = 2\frac{||\mathbf{V_h}||^2}{L_1}\sin(\nu) \tag{1}$$

Here L_1 represents the magnitude of the vector $\mathbf{L_1}$ which points from the follower aircraft to the lead point, and ν is the angle between the velocity vector and $\mathbf{L_1}$. $\mathbf{V_h}$ is the projection of the follower aircraft's velocity vector into the 2-D plane of motion described by the leader aircraft's velocity and acceleration vectors. For this research the desired path is the circle described by the goal position and the unit tangent vector u_T. u_T is aligned with the leader aircraft's velocity vector. A desirable characteristic of this guidance law is its tendency to make a large intercept angle when the vehicle is far away from the desired path and a small intercept angle when the vehicle is close to the desired path. This controller was shown to have superior performance in a flight test as compared to a traditional PD and Proportional-Integral-Derivative (PID) controller for an unmanned aerial vehicle following a curved trajectory [6].

The Park guidance can be cast into a second order, linear, time-invariant ordinary differential equation using some reasonable assumptions. The full derivation can be seen in [8], with a summary provided here. First, the commanded turn angle ν is decomposed into Eq. (2) based on Fig. 1.

$$\nu = \nu_p + \nu_d \tag{2}$$

ν_p is the angle between u_T and the lead vector $\mathbf{L_1}$. The angle ν_d lies between $\mathbf{V_h}$ and u_T. Define Δy as the distance along the y-axis, indicated in the lower left of Fig. 1, between a line parallel with u_T and bisecting ν, and the tip of the vector $\mathbf{L_1}$. Note that the bisecting line is indicated in Fig. 1 by a black dotted line. Also, define $\Delta\dot{y}$ as the projection of

$\mathbf{V_h}$ onto the y-axis. Using the small angle assumption for both ν_p and ν_d leads to Eqs. (3) and (4).

$$\sin(\nu_p) \approx \nu_p = -\frac{\Delta y}{L_1} \tag{3}$$

$$\sin(\nu_d) \approx \nu_d = -\frac{\Delta \dot{y}}{||\mathbf{V}_h||} \tag{4}$$

Note the negative signs arises from the angles being defined positive in a clockwise direction. The small angle assumption also leads to Eq. (5)

$$\sin(\nu) = \sin(\nu_p + \nu_d) = \sin(\nu_p)\cos(\nu_d) + \cos(\nu_p)\sin(\nu_d) \approx \nu_p + \nu_d = -\frac{\Delta y}{L_1} - \frac{\Delta \dot{y}}{||\mathbf{V}_h||} \tag{5}$$

Substituting Eq. (5) into Eq. (1) yields

$$A_{y_{cmd}} \approx \Delta \ddot{y} = 2\frac{||\mathbf{V}_h||^2}{L_1}\left(-\frac{\Delta y}{L_1} - \frac{\Delta \dot{y}}{||\mathbf{V}_h||}\right) \tag{6}$$

By defining $L_1 = \left(\frac{2\zeta}{\omega_n}\right)||\mathbf{V}_h||$ and selecting $\zeta = \frac{1}{\sqrt{2}}$, the system can be written as a second order ordinary differential equation, as seen in Eq. (7) with $\omega_n = 2\zeta\frac{||\mathbf{V}_h||}{L_1}$.

$$\Delta \ddot{y} + 2\zeta\left(2\zeta\frac{||\mathbf{V}_h||}{L_1}\right)\Delta \dot{y} + 4\zeta^2\frac{||\mathbf{V}_h||^2}{L_1^2}\Delta y = 0 \tag{7}$$

This allows the park guidance controller to be defined in terms of a desired natural frequency in the definition of L_1 (Ref. [8]). Once a desired natural frequency is selected L_1 is determined, which along with the follower aircraft's position and turn circle of the goal position allows for the calculation of the lead point and the vector $\mathbf{L_1}$. Then ν can be calculated and $A_{y_{cmd}}$ is determined from Eq. 1.

A 1-D variation of the Park guidance controller was created to allow a similar controller to be applied in the vertical when decoupled from the horizontal control [8]. The follower aircraft's position and velocity are first projected into a plane of motion defined by the leader aircraft's velocity and acceleration vector. The z-axis is normal to this plane and points in the direction produced by the cross product from the leader aircraft's velocity vector to the leader aircraft's acceleration vector. An overview of the geometry of the 1-D park guidance can be seen in Fig. 2. The magnitude

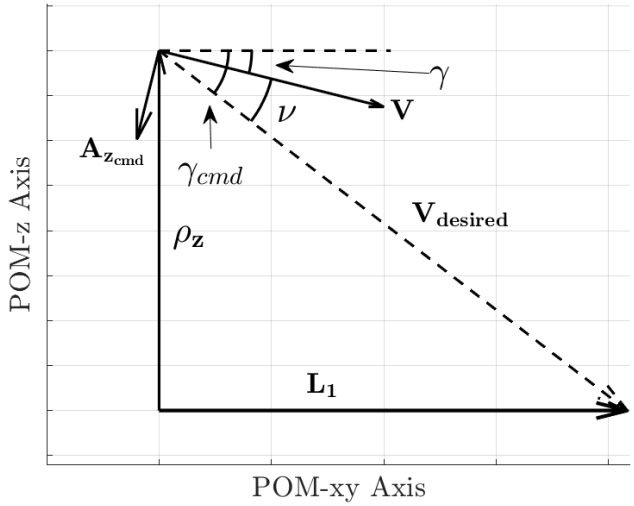


Fig. 2 Overview of 1-D Park Guidance Geometry

L_1 is still determined from a selected natural frequency as discussed in the previous paragraph, but the $\mathbf{L_1}$ vector is restricted to operating in the POM-xy axis. Define ρ_z as the POM-z axis component of the follower aircraft's position. The hypotenuse of ρ_z and $\mathbf{L_1}$ defines a desired velocity vector, $\mathbf{V_{desired}}$. The commanded flight path angle (FPA) of

$\mathbf{V_{desired}}$ is γ_{cmd}. The current FPA, γ, in the POM-z axis is defined by the follower aircraft's velocity vector. The angle command, ν, is defined as the difference between the command FPA and the current FPA. These three angles are defined in Eqs. 8-10. Then $A_{z_{cmd}}$ is calculated with Eq. 1 as before.

$$\gamma = -\tan^{-1}\left(\frac{\mathbf{V}_z}{||\mathbf{V}||}\right) \tag{8}$$

$$\gamma_{cmd} = \tan^{-1}\left(\frac{\rho_z}{L_1}\right) \tag{9}$$

$$\nu = \gamma_{cmd} - \gamma \tag{10}$$

In order to extend the park guidance to 3-D, L_1 is used to define a sphere. Thus, to find the lead point for a given lead radius L_1 requires the solution of a sphere-circle intercept problem, where the lead radius defines the sphere, and the leader aircraft's turn circle defines the circle. This problem reduces to a circle-circle intersection problem, the same as the 2-D problem, by finding the circular intersection of the sphere and the leader aircraft's plane of motion. The control guidance is then developed in the same manner as the 2-D guidance with a modification to the 3-D acceleration command, which becomes Eq. (11) where $\mathbf{u}_{VL}$ is used to ensure the acceleration command acts normal to the velocity vector [8] and within the leader aircraft's plane of motion.

$$\mathbf{A}_{y_{cmd}} = \frac{4\zeta^2||\mathbf{V}||^2}{L_1}(\sin(\nu))\mathbf{u}_{VL} \tag{11}$$

B. Relational Maneuver Primitives

The goal of the RMPs was to create simple control units that define predictable flight behaviors. Instead of creating a control strategy with non-deterministic behavior, the RMPs produce a known response even if the control strategy may select amongst them based on the current flight conditions. Each RMP was designed to accomplish one unique goal, and the group of RMPs were selected to provide a variety of control laws for the control strategy to choose amongst. This research specifically used RMPs which are designed to provide relational goal seeking behaviors to control one aircraft relative to another. The control strategy can select from five RMPs, which were developed by Air Force Research Labs

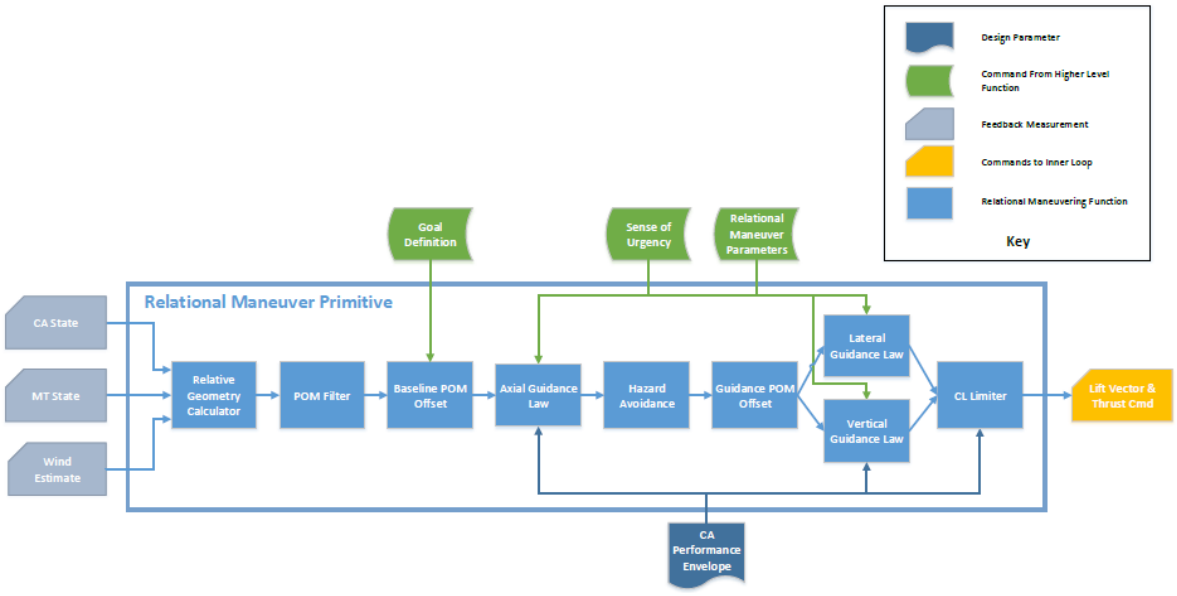


Fig. 3 RMP Functional Architecture [8]

(AFRL) and Rockwell Collins and are documented in [8]. A summary of each RMP is provided. It should be noted that any control law that can ingest the follower aircraft relative state information and produce a control in the form of total acceleration and true airspeed (TAS) could be included, allowing for a wider variety of behavior and options for the control strategy. A detailed diagram of the RMP architecture can be seen in Fig. 3.

1. Far Field Pursuit (FFP), RMP-1

The FFP RMP attempts to point the follower aircraft's velocity vector to an intercept point based on the line of sight to the leader aircraft. This RMP is used at long range since it is unlikely a good resolution of the leaders plane of motion could be estimated. This leads to the use of decoupled horizontal and vertical guidance laws.

The horizontal guidance is determined using a fixed angle lead pursuit controller. For this research, the fixed angle was selected at 20 degrees, and a constant TAS of 650 knots was used. These values were selected in an attempt to approximate the bang-bang behavior expected with a minimum time optimal control trajectory while still allowing margin for overshoots. One advantage of this RMP is that only rough estimates of the leader's velocity and acceleration vectors are required. Also, if an accurate estimate of the sign of the line of sight rate of the leader aircraft is available, the RMP is robust to fairly large errors [8].

The vertical guidance used for the far field pursuit RMP is the 1-D Park guidance with parameterized constraints on FPA. The vertical guidance attempts to match the estimated altitude of the leader aircraft, but limits on FPA are necessary due to the potential for large error in the range estimates of the leader aircraft. Additionally, an altitude offset can be set if matching the leader's altitude is not a desirable behavior based on the mission of the follower aircraft.

2. Planned Trajectory Rendezvous (PTR), RMP-2

The PTR RMP was designed based on the assumption of having access to an increased knowledge of the leader aircraft's trajectory, and is therefore designed to be used at medium ranges. Based upon this greater confidence a rendezvous trajectory can be used to better determine a control input for the follower aircraft. However, at medium range it is still assumed that large changes in the leader's trajectory may occur prior to rendezvous, and thus the PTR RMP aims to use iterative planning while still maintaining an achievable trajectory at all planning time steps. The PTR RMP also uses decoupled horizontal and vertical guidance.

In the vertical plane, the 1-D Park guidance is used to match the follower and leader aircraft's altitude. In the horizontal plane, a Dubin's intercept strategy is used [9], which begins based on the follower aircraft's current position and velocity. The path terminates at a forward time projection of the leader aircraft's position and velocity vectors based on an estimated TTG. The 2-D Dubin's path ensures at least one achievable trajectory is always viable as documented in [9]. Then Park guidance is used to track this trajectory until the next update. In order to iterate on the Dubin's path solution, Algorithm 1 is implemented to determine the rendezvous trajectory at each time step. This algorithm uses a constant speed assumption when calculating TTG along the Dubin's path.

Algorithm 1 Algorithm for Dubins path solution iteration

$TimeToGo \leftarrow TimeToGoPrevious$
$FinalState \leftarrow$ ProjectMTState$(MTState, TimeToGo)$
$Path \leftarrow$ SolveDubins$(CurrentState, FinalState)$
$TimeToGoActual \leftarrow$ EvaluateTTG$(Path)$
$TTGError \leftarrow TimeToGo - TimeToGoActual$
$TimeToGo \leftarrow TimeToGo - \lambda_1 * TTGError - \lambda_2 * TimeToGo$
return $Path, TimeToGo$

First, a guess of the TTG to rendezvous is created, and the moving target is projected forward along its estimated turn circle to find an estimated rendezvous point. Then, the Dubin's path is calculated and the TTG along the Dubin's path is determined. The actual TTG of the follower aircraft to track this Dubin's path solution is then compared to the initial estimate to create the TTG error. The TTG is then updated with modification based on the user selected convergence parameters λ_1 and λ_2. These parameters where tuned during the creation of the control simulation and were provided by Rockwell Collins.

When determining the Dubin's path, only the four solution types consisting of turn-straight-turn legs are included since they are sufficient to cover all initial and final condition pairs. This also avoids the three turn solutions which require 2 max roll rate roll reversals. While not all solution trajectories will be feasible for every initial and final condition pair, there will always be at least one feasible trajectory as proven in [9]. When multiple feasible solutions exist, the one with the shortest path length is selected.

Due to the nature of having multiple feasible solutions available, consideration was given to the potential for rapid changes in TTG if different paths are selected in subsequent time steps. Since the Dubin's path solution is being tracked by the Park guidance controller, the ultimate result of a large change in TTG is that the lateral acceleration command will

change drastically at each time step. In order to minimize the impact of this behavior, a simple first order filter is applied to the lateral acceleration commanded output. The filter takes the form of Eq. (12) where τ is a user selected time delay and Δt is the control simulation time step of 0.05 seconds. This filter was developed by Rockwell Collins along with the control simulation and the selection of the time delay. This filter was shown to have adequate performance over multiple initial conditions in [8].

$$\mathbf{A}^{n+1}_{y_{cmdFiltered}} = -(\frac{\Delta t - 2\tau}{\Delta t + 2\tau})\mathbf{A}^{n}_{y_{cmdFiltered}} + \frac{\Delta t}{\Delta t + 2\tau}\mathbf{A}^{n+1}_{y_{cmd}} + \frac{\Delta t}{\Delta t + 2\tau}\mathbf{A}^{n}_{y_{cmd}} \tag{12}$$

3. In-Plane Pursuit (IPP), RMP-3, Out of Plane Pursuit (OPP), RMP-4, and Local Level Pursuit (LLP), RMP-5

The last three RMPs assume a higher degree of accuracy in the knowledge of the leader aircraft's current trajectory and future maneuvering. This allows these RMPs to operate directly on the error to the goal position. While all three RMPs operate directly on this error, they differ in the coordinate frame used for guidance calculations and whether they used decoupled horizontal and vertical guidance or if they used the 3-D Park guidance algorithm.

These RMPs calculate an axial guidance which determines an airspeed command to the goal position. This airspeed command is determined using a PID controller which calculates a desired overtake rate. This overtake rate is then implemented in three parts: an airspeed command, a lateral offset command, or a vertical offset command. The details of this calculation can be found in [8]. These offsets are then applied to create the reference trajectory of the goal position. For the LLP and IPP, horizontal and vertical guidance are decoupled and use the applicable Park guidance laws. The OPP RMP uses the 3-D Park guidance.

Using the local level reference frame or allowing the follower aircraft to maneuver out of plane by using 3-D Park guidance allows the aircraft to converge faster to the goal position if operating under the assumption that the leader aircraft is relatively stable. Forcing the follower aircraft to prioritize completing the rejoin to the goal position by staying in plane of the leader aircraft's plane of motion tends to more effectively deal with uncertainty in the leader aircraft's trajectory.

C. Control Simulation

The control simulation environment was used to estimate the response of the follower aircraft throughout this research. The following sections detail the required inputs for the control simulation, the architecture and individual components, and finally the control simulation output.

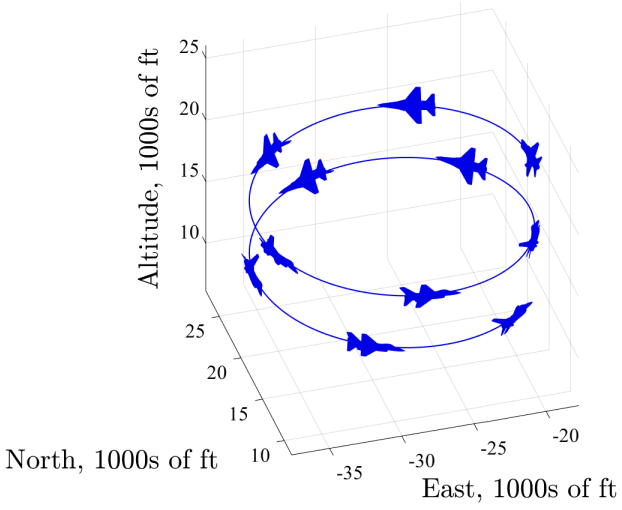


Fig. 4 Descending Spiral Leader Trajectory

1. Input-Leader Trajectory Generation

The first required input for the control simulation is the leader aircraft's trajectory. Each trajectory was built independent from the control simulation, and is represented as a 3-D position, velocity, acceleration, and jerk vector. The trajectories include all state information every twentieth of a second. This research used a descending spiral trajectory, seen in Fig. 4 which represents a typical maneuver used in training environments. This leader aircraft trajectory was provided along with the initial version of the control simulation developed by Rockwell Collins and AFRL and is documented in [8].

2. Input-Initial Condition Definition

The second required input for the control simulation is a number of follower parameters which are specified and provided as an input to the control simulation. These parameters include: Position as a latitude, longitude, and altitude above the 1984 World Geodetic System (WGS-84) ellipsoid [10], TAS, heading, roll angle, and FPA. The follower aircraft is assumed to start in a trimmed condition at these parameters. These same initial conditions are also input into the optimal trajectory tool discussed in Section II.D, and the control strategy refinement process itself.

3. Control Simulation Architecture

The architecture of the control simulation can be seen in Fig. 5. The Azimuth, Elevation, and Range (AER) Sensor Model and Track Estimator in the upper left corner of the model was bypassed for this research. The leader aircraft's trajectory, here referred to as the maneuvering target is fed directly to the RMP. The RMP also ingests the Relational Maneuver Parameters, Goal Definition, and Sense of Urgency as inputs, which together make up the control strategy and is discussed further in Section II.C.5. The RMPs operate with these inputs along with the follower aircraft's position to create a total acceleration command and TAS command output. The control simulation then uses a basic inner loop autopilot which converts the control output from the RMP into roll rate (p), pitch rate (q), yaw rate (r), and throttle position commands usable by the Basic Aircraft Model (BAM). The simulation then uses the BAM to estimate the follower aircraft's response to a given control input over each time step. This estimated response is then fed back as the follower aircraft's state for the next time step. The control simulation outputs a time history of the follower aircraft's states, leader aircraft's states, and control response for the duration of the simulation.

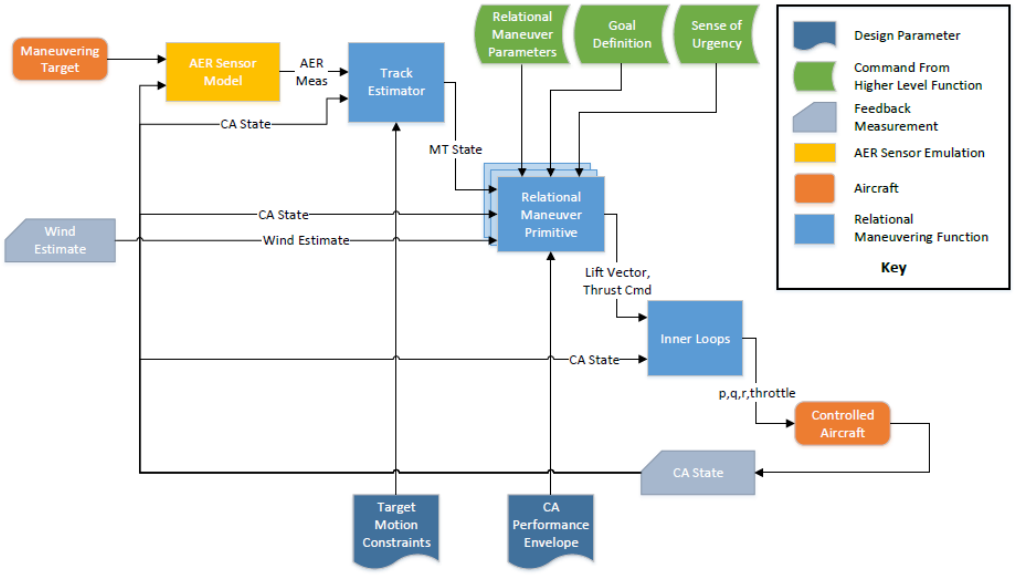


Fig. 5 Control Simulation Architecture [8]

4. BAM

The BAM was provided by AFRL as a compiled digital library, and works as a black box model within the simulation code. The state vector that operates within the BAM consists of **X** a position vector, **V** a velocity vector, **Q** a quaternion

defining the aircraft rotation, $\mathbf{X}_{rates}$ the body axis rates, $\mathbf{W}$ the wind velocity, and m mass. The model was built from a nonlinear 6 Degree-of-Freedom rigid body model, and improved through various testing and refinement. The BAM

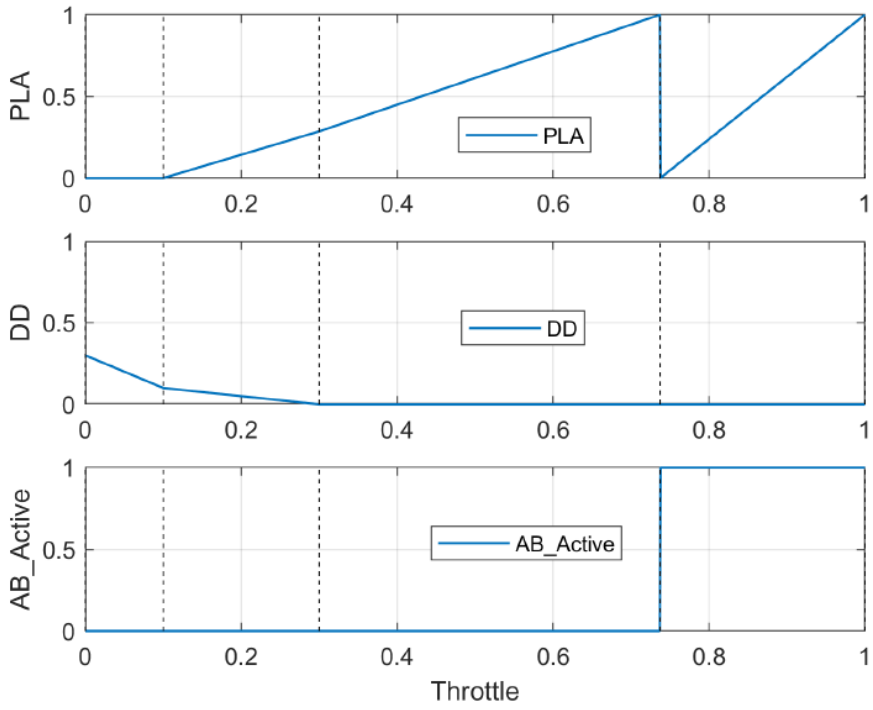


Fig. 6 Mapping from Idealized Throttle Position to PLA, DD, and Afterburner [8]

ingests a command vector of $[p, q, r, T]$, where T is the idealized throttle command. T is then decomposed with the mapping in Fig. 6 into a power lever angle, drag device, and afterburner position. The BAM is then provided with the duration of the current time step and it returns the new estimated values of the state vector at the end of the current time step. The BAM currently exists as a complied C code digital link library. It is initialized, loaded, and called within the control simulation environment. Similar to the inner loop autopilot model, the BAM itself is not critical to this research and was implemented with the ability to be replaced by any other sufficiently detailed model which can accept the same control inputs and can output the same state vector.

5. Control Strategy

The control strategy consists of the three green blocks in the upper right of Fig. 5. The goal position is a point in space the RMP will attempt to intercept and maintain. The sense of urgency is a user defined parameter between 0 and 1 which linearly scales a number of gain and "aggressiveness" parameters such as $N_{z_{max}}$ and α_{max}. Since the scenario presented in this research is focused on a minimum time to rejoin, the Sense of Urgency parameter remains fixed at 1.

The relational maneuver parameters contain both the control strategy decision engine as well as the constraints dictated by the follower aircraft's performance envelope such as max airspeed and structural limits. In the simplest form, the control strategy decision engine simply enforces a user selected RMP throughout the duration of the simulation. In its fully implemented form, the control strategy decision engine uses a database lookup based on the follower aircraft's current state information, and an estimate of the current TTG to the goal position to determine which RMP to implement over the next time step. An example of a Control Strategy Element (CSE) can be seen in Table 1. The control strategy decision engine references to a specific CSE, which is associated with the appropriate follower aircraft state information, and inputs the current TTG which then returns the correct RMP. The TTG estimate is determined using the same method as discussed in Section II.B.2. The terminal criteria assumes that the desired goal position will end with the follower aircraft maintaining a similar path as the leader aircraft, but could be adjusted based on expected intercept geometry for other scenarios. Once the TTG estimate is determined, the correct RMP can be selected and executed.

Table 1 CSE Example

Time-to-Go	RMP
$-\infty < TTG < 10$	FFP
$10 \leq TTG < 20$	LLP
$20 \leq TTG < 30$	OPP
$30 \leq TTG < 40$	OPP
$40 \leq TTG < 50$	PTR
$50 \leq TTG < \infty$	FFP

6. Inner Loop Autopilot Model

The inner loop autopilot model is not the focus of this research, but is necessary to translate the RMP acceleration vector and airspeed command into a p, q, r, and throttle command for the BAM. The architecture of the inner loop autopilot is designed with the idea that it could be replaced by any aircraft autopilot which can accept the RMP output. A detailed discussion of the specific inner loop autopilot model used in this research can be found in [8], with an overview provided here for reference. The inner loop autopilot architecture can be seen in Fig. 7. The inner loop autopilot first decomposes the acceleration vector into a tangent acceleration command for the throttle loop, a load factor command which feeds the pitch rate loop, and a lift orientation command that goes to the roll/yaw rate loop. The output from the inner loop autopilot model is a pitch rate, roll rate, yaw rate, and idealized throttle position command.

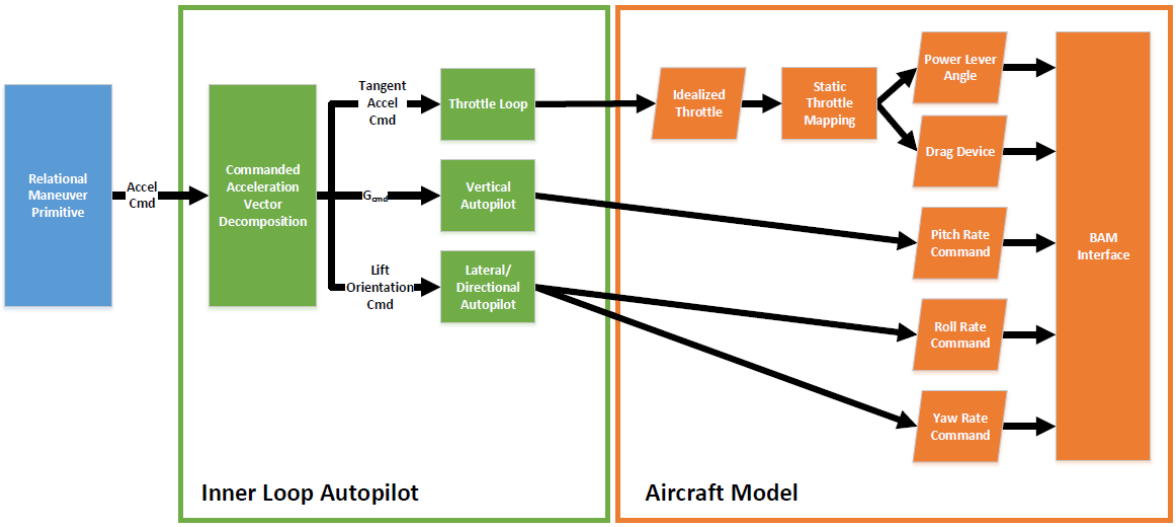


Fig. 7 Inner Loop Autopilot Interface [8]

7. Control Simulation Software

The control simulation is built as .m files using Matlab® R2023a for execution. All computations were conducted on a 2018 Dell Alienware R17 Laptop which includes an Intel Core i7-9750H processor, and 16 GB of Ram. A simple timing command is used to monitor the run time of the simulation. This is then compared to the simulation time and reported as a multiplier of simulation time.

8. Output-Follower Response Trajectory

The output of the control simulation is the state history of the follower aircraft, including position, velocity, and acceleration vectors in the inertial reference frame. This response trajectory is recorded at the same rate of the simulation, with a time interval of 0.05 seconds. This output is then used to feed both the optimal trajectory tool and the error analysis portion of the control strategy refinement process.

D. Optimal Trajectory

The goal of this research is to control the follower aircraft to approximate the optimal trajectory. The optimal trajectory tool solves the optimal control problem such that given a leader trajectory and initial condition, the follower trajectory rejoins to the goal position in minimum time and then minimizes deviation from the goal position until the end of the simulation time. The optimal trajectory is used to determine the performance of the control strategy. The optimal trajectory tool uses a direct collocation method to find the follower optimal trajectory for that specific scenario. The following sections detail the required assumptions, models, solvers, and how the initial guess is generated.

1. Inputs-Leader Trajectory, Initial Condition, Follower Response Trajectory

The optimal trajectory tool requires the leader trajectory and initial conditions (ICs) which are described in Section II.C.2and II.C.1. It also requires a follower response trajectory, discussed in Section II.C.8, that is output from the control simulation which is used as the initial guess discussed in Section II.D.6.

2. Assumptions

A number of simplifying assumptions were used in the creation of the optimal trajectory tool. The follower aircraft was modeled with triple integrator dynamics, with control applied as a 3-D jerk vector. This model assumed perfect knowledge of the aerodynamic forces and thrust interactions. The entire leader trajectory was treated as a set of known parameters to the solver. No exogenous inputs were considered within the tool. Therefore, deterministic knowledge of all the follower states was assumed, which included 3-D position, velocity, and acceleration.

Both the leader and follower aircraft were modeled as point masses about their center of gravity. In formulating the optimal control problem, the existence of an optimal solution was assumed. A scenario without a solution could be easily crafted, such as the simple case where a trailing follower aircraft cannot accelerate fast enough to catch a higher performance leader aircraft in the two-minute simulation time. This assumption was satisfied within this research by choosing appropriate ICs and aircraft performance envelopes. Additionally, the test cases presented leader aircraft trajectories such that the opportunity existed for an optimal rejoin and formation position maintenance.

Lastly, the aircraft performance envelope was modeled with a minimum velocity and acceleration to prevent aerodynamic stall, a maximum velocity to limit maximum dynamic pressure forces, and a maximum acceleration to limit maximum structural forces.

3. Direct Collocation Solver

This problem was solved with the direct orthogonal collocation solver GPOPS-II discussed in [11]. GPOPS-II uses a polynomial approximation of the state and control vectors. The solver transcribed the continuous optimal control problem by interpolating these approximations at Legendre-Gauss-Radau based collocation points as discussed in [12]. Gaussian quadrature was used to approximate the integral of the cost function, and a differentiation matrix constructed by differentiating the Lagrange basis constructed at the set of collocation points, and then evaluating these derivatives at each collocation point in turn. Additionally, the boundary constraints were enforced at the associated point along the trajectory and the path constraints were enforced at each collocation point. GPOPS-II then formatted the problem for hand-off to a nonlinear program (NLP) solver. The NLP solver used both the gradient and Hessian of the transcribed problem with respect to the coordinates of the state and control parameterization. The NLP Solver Interior Point OPTimizer (IPOPT) was used as provided with GPOPS-II. IPOPT used an interior point boundary value method which was described in [13]. A tolerance of 10^{-5} was used for the determination of convergence within IPOPT. GPOPS-II iteratively refines the solution by creating piecewise polynomials of varying degree to achieve a prescribed tolerance on an error estimate based on how well the solution approximates the dynamics and constraints.

Scaling was used to aid in solver convergence, which simply consisted of normalizing to 100s of meters of distance. The NLP solver requires bounds to constrain the feasible region for all variables. The position state was bounded to the maximum distance the follower aircraft could travel straight along the coordinate direction at maximum velocity over the two-minute simulation time. The velocity and acceleration components were bounded by $\pm V_{max}$ and $\pm A_{max}$ respectively. An acceleration rate limit of ±5 g/s was used to bound the control components. Unless specified elsewhere, default settings were used as provided with the software. GPOPS-II reported the final solution with values of the states, controls, and costate estimates at each collocation point and one interpolated point at the end of each of the two phases.

4. Aircraft Dynamics Model

The aircraft dynamics within the optimal trajectory tool were modeled with the triple integrator equations. These states are constrained by the aircraft's flight envelope by limiting velocity and acceleration. The state vector of the follower aircraft consists of the position, velocity, and an acceleration vector. The rate of acceleration, or jerk, vector is the independent control. The triple integrator dynamics can be seen in Eq. (13).

$$\dot{p} = v, \quad \dot{v} = a, \quad \dot{a} = u \tag{13}$$

5. Mathematical Formulation

The goal of the optimal trajectory tool is to determine the optimal path for a follower aircraft to rejoin to a formation position, defined in reference to the leader aircraft, in minimum time and subsequently maintain the formation position with the least deviation until a fixed final time. This problem was crafted into a two-phase dynamic optimal control problem based on the GPOPS-II standard problem formulation [14]. A summary of the problem's mathematical formulation can be seen on the following page.

The cost function for Phase 1 dictates a minimum time solution to execute the initial rejoin from a predefined starting location to the goal position, taking the form $t_f^{(1)}$. The scenario begins at time zero with a known initial state of both the leader and follower aircraft. Note that $t_f^{(i)}$ indicates the final time of phase i. Phase 1 included the terminal condition requiring the follower to attain a formation position at $t_f^{(1)}$ which was enforced by Eq. (14) and described in [15]:

$$\begin{aligned} f_1(p_d) &= (p_{d_x} - [R_L^i c^L]_x)^2 \\ f_2(p_d) &= (p_{d_y} - [R_L^i c^L]_y)^2 + (p_{d_z} - [R_L^i c^L]_z)^2 - R_{ring}^2 \end{aligned} \tag{14}$$

It should be noted the Eq. (14) assumes the goal position is defined as a ring of points. For this research the ring radius was set to zero reducing the formation position to a single point, which yields Eq. (15).

$$\begin{aligned} f_1(p_d) &= (p_{d_x} - [R_L^i c^L]_x)^2 \\ f_2(p_d) &= (p_{d_y} - [R_L^i c^L]_y)^2 + (p_{d_z} - [R_L^i c^L]_z)^2 \end{aligned} \tag{15}$$

These equations provided a way to characterize the distance from the follower aircraft to the formation ring or goal position. A visual depiction of the formation ring can be seen in Fig. 8 reproduced from [15]. Again note that this research shrank the ring radius to zero yielding a single formation point. The parameter $f_1(p_d(t_f^{(1)}))$ ensured the follower aircraft attained the same x^L-coordinate in the leader aircraft's reference frame as the tip of the vector c^L, which was used to define the center point of the formation ring. The parameter $f_2(p_d(t_f^{(1)}))$ was used to ensure that the y^L and z^L coordinates of the follower aircraft in relation to the leader aircraft were at the same coordinates as the c^L vector. Simply put, when $f_1 = f_2 = 0$ the follower aircraft was established at the goal position defined by the vector c^L. Lastly, constraints were enforced on the magnitude of the follower aircraft's velocity and acceleration vectors for the duration of the Phase. The magnitude of the follower aircraft's velocity vector was constrained by a minimum and maximum velocity, as dictated by the aircraft's stall speed and maximum speed. The magnitude of the acceleration vector was constrained by a minimum acceleration to avoid zero-g flight, and a maximum acceleration to prevent overloading the aircraft structure. Both of these constraints were enforced as path constraints throughout the duration of Phase 1.

Minimizing the cost function of Phase 2 required the follower aircraft to minimize the deviation from the goal position for the duration of the phase. This deviation was characterized by squaring and summing f_1 and f_2 as defined during Phase 1, and integrating the resulting values over the duration of the phase as can be seen in the cost function of Phase 2 below. Phase 2 used the same follower aircraft dynamics along with the same state and control constraints. The second Phase also contained the requirement for the position and velocity to match across phases, ensuring a continuous trajectory throughout the entire scenario. Additionally, the condition for the free final time of Phase 1 to match the initial time of Phase 2 was enforced. The final time of Phase 2 was fixed, and was chosen to ensure that the follower aircraft could attain the goal position during Phase 1 before the fixed final time of Phase 2.

Since the optimization software required one cost function for the overall problem, the cost functions from both Phase 1 and Phase 2 were summed to produce a cumulative cost function. In order to account for a difference in the magnitudes between the cost functions of each Phase, a weighting parameter β was included in the summation. The weighting parameter β was chosen through experimentation, and found to be 0.4 which yielded consistent solution convergence across all tested initial conditions.

Overall: $\min_{u(t)} \; J = \beta J_1 + (1-\beta) J_2$ ..... Cost Function

Phase 1: $\min_{u(t)} \; J_1 = t_f^{(1)}$ ..... Cost Function

S.T. $\dot{p_f} = v_f$ ..... Dynamics

$\dot{v_f} = a_f$ ..... Dynamics

$\dot{a_f} = u$ ..... Dynamics

$t_0 = 0,\ p_f(t_0) = p_{f_0},\ v_f(t_0) = v_{f_0},\ a_f(t_0) = a_{f_0}$ ..... BC

$f_1(p_d(t_f^{(1)})) = f_2(p_d(t_f^{(1)})) = 0$ ..... BC

$|v_f| \le v_{f_{max}},\ |a_f| \le a_{f_{max}}$ ..... Constraints

Phase 2: $\min_{u(t)} \; J_2 = \int_{t_0^{(2)}}^{t_f^{(2)}} \{f_1(p_d(t))^2 + f_2(p_d(t))^2\} dt$ ..... Cost Function

S.T. $\dot{p_f} = v_f$ ..... Dynamics

$\dot{v_f} = a_f$ ..... Dynamics

$\dot{a_f} = u$ ..... Dynamics

$t_f^{(2)} = 120$ ..... BC

$t_o^{(2)} = t_f^{(1)},\ p_f(t_0^{(2)}) = p_f(t_f^{(1)})$ ..... Phase Continuity

$v_f(t_0^{(2)}) = v_f(t_f^{(1)}),\ a_f(t_0^{(2)}) = a_f(t_f^{(1)})$ ..... Phase Continuity

$|v_f| \le v_{f_{max}},\ |a_f| \le a_{f_{max}}$ ..... Constraints

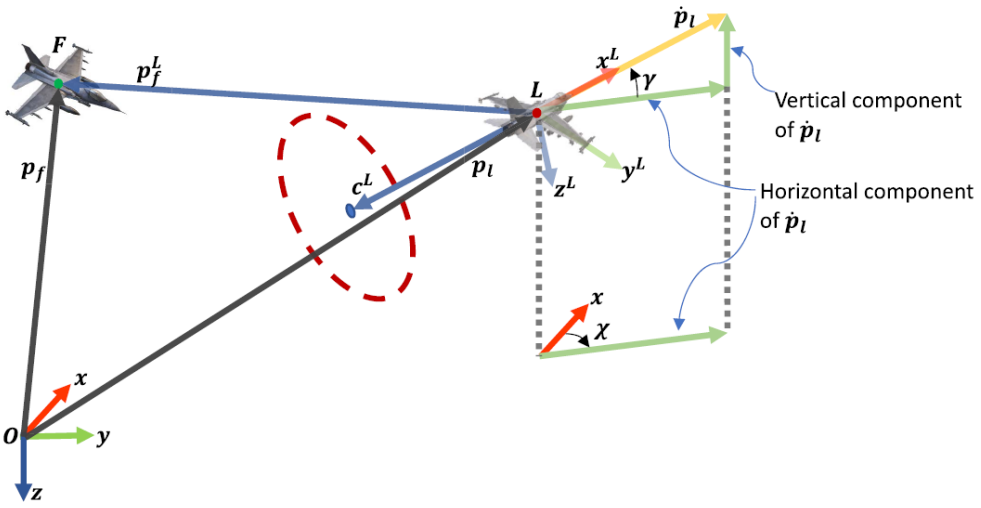


Fig. 8 Depiction of Formation Ring [15]

6. Initial Guess Generation

GPOPS-II requires an initial guess for the states and controls. The initial guess of the follower aircraft's states and controls were generated from the follower response trajectory input to the optimal trajectory tool. The initial guess was generated by creating a CSE which was set to output the OPP RMP for all TTGs as discussed in Section II.C.5. Since

the output trajectory does not contain a jerk vector, the initial control guess was determined by Eq. (16) to first order.

$$\mathbf{J}_{n+1} = \frac{\mathbf{A}_{n+1} - \mathbf{A}_n}{\Delta t}, \mathbf{J}_1 = \mathbf{0} \tag{16}$$

The follower response trajectory provided a sufficient initial guess such that the optimal trajectory tool was able to converge to a solution in all cases.

7. Optimal Trajectory Tool Software

The commercial Matlab® based tool GPOPS-II is used to determine the optimal trajectory using the same hardware and software as the control simulation as discussed in Section II.C.7.

8. Output-Follower Optimal Trajectory

The follower optimal trajectory was output in the same format as the follower response trajectory of Section II.C.8. Additionally, the follower optimal trajectory includes the jerk control vector required to follow the optimal state trajectory.

E. Genetic Algorithms (GAs)

GAs are an evolutionary approach which attempts to emulate evolutionary process in genetics to determine an optimal solution to an optimization problem [16]. GAs contain five components: encoding, fitness, selection, crossover, and mutation. The encoding mechanism simply exists to represent the problem variables in appropriate mathematical terms. The fitness function determines the quality of each population member, and is often the objective function of the optimization problem. A selection mechanism that operates on a survival-of-the-fittest concept is used to determine which solutions have the best fitness and will continue to the next generation. The crossover mechanism is used to select certain population members and mutate them, such that the resulting member is different than its parent member while progressing to a more optimal solution [17]. Genetic algorithms in general are robust, provide optimization over a large state-space, and are not affected due to slight changes in inputs or the presence of noise like some traditional artificial intelligence algorithms [18].

III. Control Strategy Refinement Methodology

The control strategy refinement process is used to determine a CSE for a given set of ICs such that the follower aircraft response produced from a given control simulation more closely aligns with the follower optimal trajectory. A genetic algorithm is implemented to search the space of possible discrete RMP selections within the CSE with the aim to improve the response to an acceptable level. It should be noted that this may not be the most efficient or fastest search method, but it was chosen for the simplicity, ease of implementation, and its ability to handle discrete solution spaces. For this research, the variables are integer values between 1 and 5 which represent which RMP is implemented in-between each of the switching times associated with the CSE. The algorithm implementation was built from the work contained in [19] and [20].

A. Inputs-Leader Trajectory, Initial Condition, Follower Optimal Trajectory

The CSE refinement process requires both the leader trajectory and ICs as specified in Sections II.C.2 and II.C.1. Additionally, the follower optimal trajectory associated with the same leader trajectory and initial condition produced from the optimal trajectory tool are required for fitness calculations, and is created in accordance with Section II.D

B. Genetic Algorithm Overview

The genetic algorithm provided with the Matlab® software was used for the control strategy refinement process. The help documentation provided with Matlab® 2023a details the implementation of the genetic algorithm. However, an overview is provided here which includes the specifics of how the algorithm was executed for this specific research.

C. Initial Population Creation

The Matlab® genetic algorithm creates a random initial population while enforcing integer values within the supplied bounds for each variable. An example population of 5 members can be seen in Table 2 which is translated within the

control simulation such that the RMPs associated with the numbers listed are implemented based on the current TTG. Thus, each population member has 6 options of which RMP is implemented for a given IC. In the actual implementation the population size was 200 members, which is the default setting when each member has greater than 5 associated decision variables.

Table 2 Initial Population Example

Member 1	Member 2	Member 3	Member 4	Member 5
1	5	3	5	5
3	3	2	1	5
2	5	4	3	2
5	4	3	3	3
1	5	1	2	3
1	1	3	2	1

D. Fitness Calculation

In order to determine the fitness of each population member, the associated RMP selections are first executed in the control simulation as a single CSE to generate an associated follower response trajectory. To compare the follower response trajectories that the control simulations produced against the optimal trajectory, the optimal trajectory is first interpolated with a polynomial at each collocation point in time to correspond with the time vector of the simulation. Then the 2-norm distance between the paths at corresponding time steps is determined, and stored in an error vector. The fitness of the population member is equal to the sum of the entries of the error vector. The Matlab® algorithm executes its internal scaling on this fitness value to create the associated expectation values. Of note, the control simulation environment is vectorized to the maximum extent which allows for an entire population to be simulated at the same time.

E. Children Generation

Population members created after the initial population are known as children of the previous population. Matlab®'s genetic algorithm creates these children members primarily through the processes of mutation and crossover. The foundation for these processes are found in [20]. For this work all settings related to children generation were left at their default values which are specific to integer optimization problems. Subsequent populations are created until achieving one of the termination criteria listed in Section III.F.

F. Termination Criteria

Two main criteria are used to determine when the algorithm terminates. The $MaxStallGenerations$ determines when the algorithm has not found a member with a lower fitness value in the predetermined number of generations. This was set to 15 for this research. The second criteria is the $FunctionTolerance$ which was set to the default value of 10^{-6}. Thus, the best solution is produced when the algorithm exits and indicates the average change in the fitness value is less than the $FunctionTolerance$ over the number of $MaxStallGenerations$.

G. Output-Refined CSE

The output from the control strategy refinement process is a refined CSE which approximates the optimal trajectory. The follower response trajectory associated with the refined CSE can be reproduced using the control simulation for further analysis.

IV. Control Strategy Refinement Test Scenarios

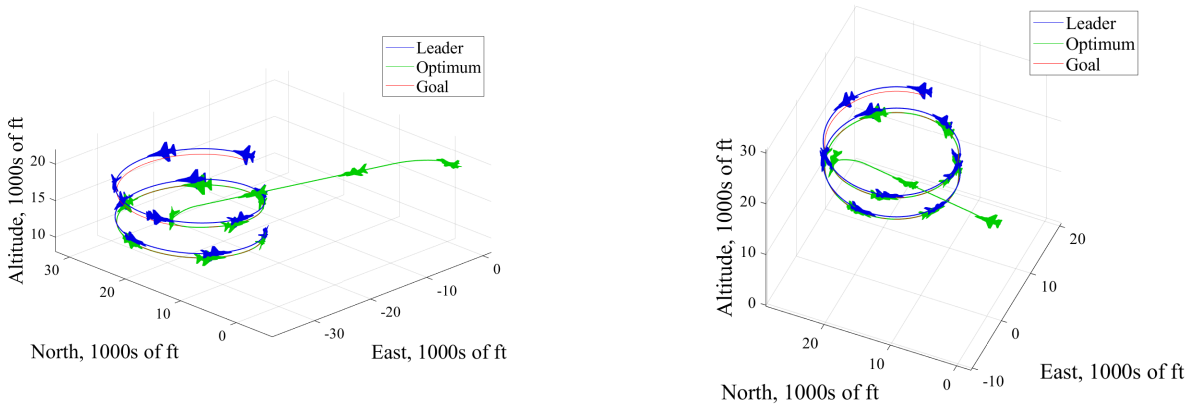
In order to test the control strategy refinement process two sets of initial conditions were selected. Each initial condition had the same velocity, heading, roll and FPA. The only parameter varied for this initial research was the location of the follower aircraft. The first scenario had the follower aircraft located 3 NM aft and 3 NM east of the leader aircraft and 1,000 ft below. The second scenario had the follower aircraft 3 NM aft and 3 NM west of the leader at 1,000 ft below. A summary of the initial conditions can be seen in Table 3. Note that HAE is height above the ellipsoid, and

the goal position is defined in the leader aircraft's body reference frame.

Table 3 Tested ICs

	IC-1	IC-2		IC-1	IC-2
$\begin{bmatrix} \text{Lat} \\ \text{Lon} \\ \text{HAE} \end{bmatrix}$	$\begin{bmatrix} -2.98° \\ 2.91° \\ 18,984 \text{ ft} \end{bmatrix}$	$\begin{bmatrix} -2.98° \\ -2.81° \\ 18,984 \text{ ft} \end{bmatrix}$	Goal	$\begin{bmatrix} -707 \\ -707 \\ 0 \end{bmatrix}$ ft	$\begin{bmatrix} -707 \\ -707 \\ 0 \end{bmatrix}$ ft
Velocity	450 kts	450 kts	Heading	0°	0°
Roll	0°	0°	FPA	0°	0°

Using the descending spiral leader trajectory discussed in Section II.C.1 and the ICs listed, the optimal trajectory tool produced the two follower optimal trajectories seen in Fig. 9. The blue aircraft represents the leader, and the red



(a) IC-1 (b) IC-2

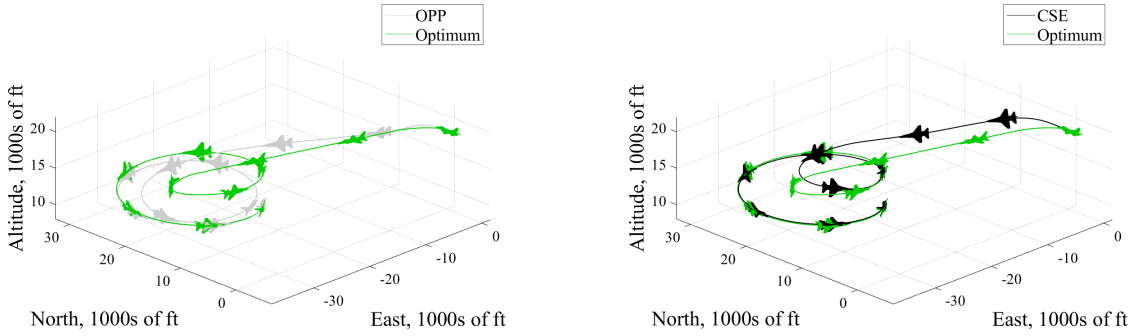
Fig. 9 Follower Optimal Trajectories as Determined by Optimal Trajectory Tool

line represents the goal position which is defined aft and off the leader's left wing. The green aircraft represents the optimal trajectory. For both initial conditions the optimal trajectory can be seen to take a direct course to rejoin to the goal position followed by a maximum control turn within the prescribed velocity and acceleration limits to align with the leaders trajectory. This is exactly the behavior expected from a minimum time optimal trajectory.

V. Results

The goal of the control strategy refinement process is to determine the selection of RMPs for a CSE such that the follower response trajectory aligns as closely as possible to the follower optimal trajectory, for a given initial condition. Figure 10 shows a comparison of the OPP RMP follower response trajectory to the refined CSE response trajectory. The OPP trajectory was selected since it yielded the least error of the 5 RMPs when used individually. The CSE response is able to begin the turn to rejoin to the goal position earlier in the trajectory, and attains the goal position faster than the OPP trajectory.

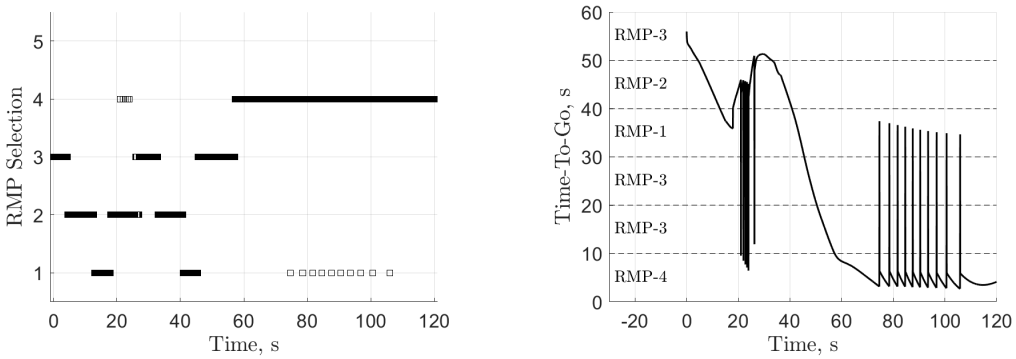
Figure 11 shows how the control strategy selected the RMPs and the associated TTGs. The left side of the second plot shows which RMPs were associated with which blocks of time, with the progression between RMPs seen in the first plot. As expected there is some oscillation in the TTG when the aircraft closely approaches the leader aircraft, yet no significant deviation from the goal position is observed, demonstrating the effectiveness of filtering the commanded acceleration. For this IC the LLP RMP was not needed and the control strategy favored the FFP, PTR, and IPP RMPs when the aircraft was far away from the goal position. The control strategy favored the OPP RMP for station keeping once the goal position was attained. This is in alignment with the design of the RMPs as previously discussed.



(a) OPP RMP Response

(b) Refined CSE Response

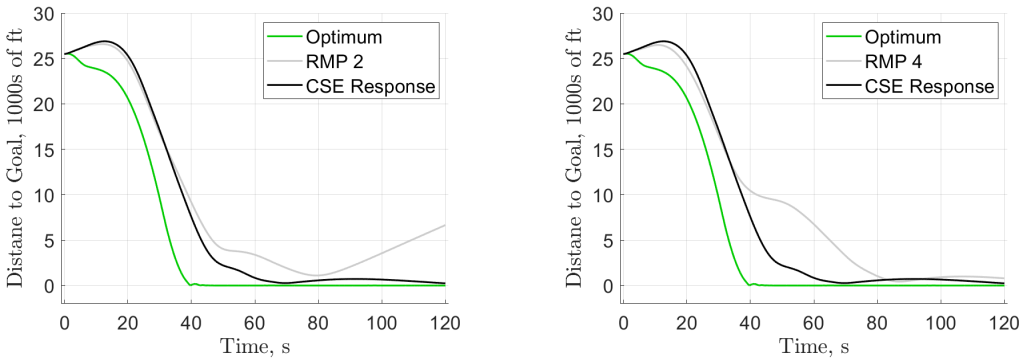
Fig. 10 Comparison of OPP RMP and Refined CSE Response for IC-1



(a) RMP Selection

(b) Time-To-Go

Fig. 11 Control Strategy Execution for IC-1



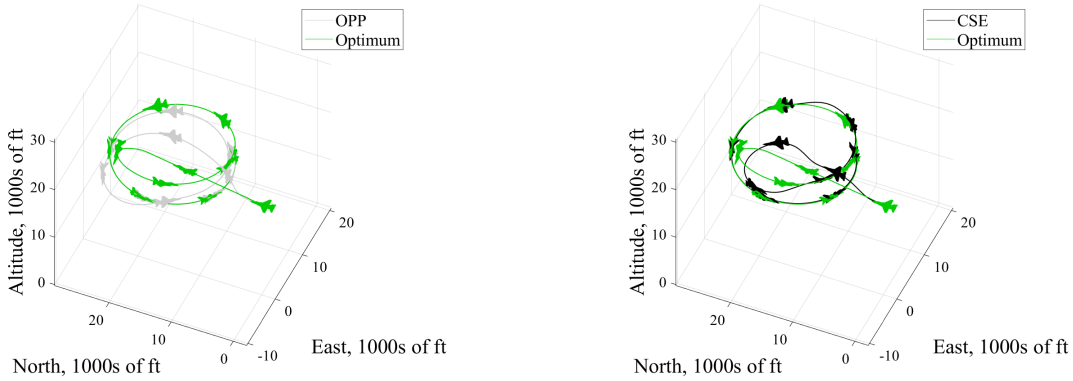
(a) PTR RMP

(b) OPP RMP

Fig. 12 Comparison of the Distance to the Goal Position over Time for the Optimal, CSE, and Single RMP Follower Aircraft Responses for IC-1

A result of the control strategy refinement process was seen in Fig. 12. The control strategy primarily selected the PTR RMP during the first 60 seconds of the trajectory when it showed the best performance compared to the optimal trajectory, and switched to the OPP RMP which showed the best performance for the second half of the trajectory. The ability of the heuristic optimization algorithm to find and combine the best pieces of each RMPs response is an excellent demonstration of the power of the control strategy refinement process.

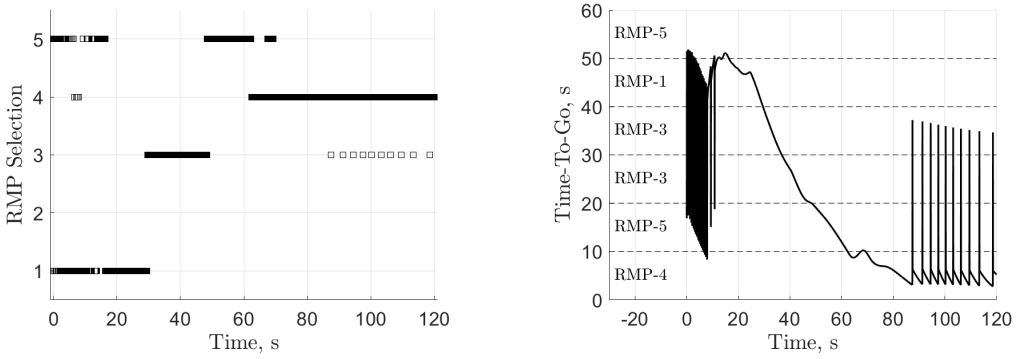
Similar results are seen for the refined CSE produced by the control strategy refinement process for IC-2 in Figs. 13, 14, and 15. Again the refined CSE follower response trajectory does a better job approximating the aggressive turn



(a) OPP RMP Response

(b) Refined CSE Response

Fig. 13 Comparison of OPP RMP and Refined CSE Response for IC-2

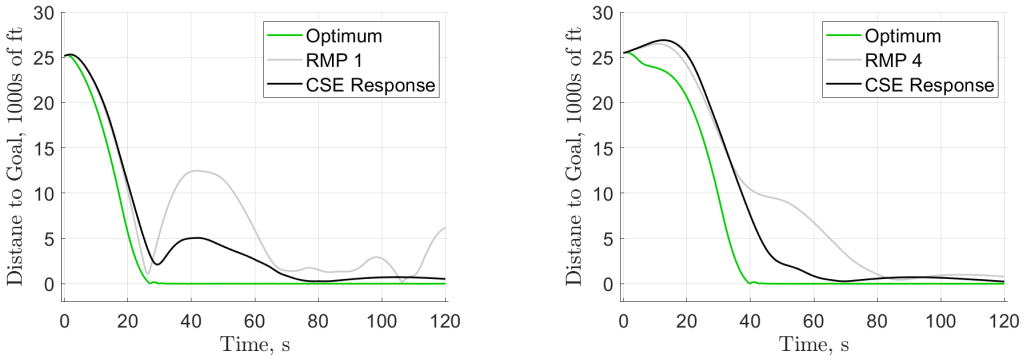


(a) RMP Selection

(b) Time-To-Go

Fig. 14 Control Strategy Execution for IC-2

of the follower optimal trajectory and completes the rejoin in less time. The control strategy refinement process was able to take advantage of the best performance seen in the FFP RMP during the earlier part of the simulation while still settling on the OPP RMP once the rejoin was completed. These two ICs demonstrate the initial capabilities of the control strategy refinement process to create a control strategy built on RMPs which brings the follower response into closer alignment with the follower optimal trajectory.



(a) FFP RMP

(b) OPP RMP

Fig. 15 Comparison of the Distance to the Goal Position over Time for the Optimal, CSE, and Single RMP Follower Aircraft Responses for IC-2

Table 4 shows the fitness for each individual RMP compared to the refined CSE response. These results indicate at least a 36% and 48% error reduction respectively, a significant improvement using the refined CSE when compared to any individual RMP.

Table 4 Fitness of Follower Response Trajectories (10^6 meters)

	IC-1	IC-2
RMP-1	3.90	3.28
RMP-2	3.45	3.69
RMP-3	4.18	3.32
RMP-4	3.51	3.59
RMP-5	4.13	3.73
Refined CSE	2.19	1.71

VI. Analysis & Conclusion

This paper demonstrates the ability to use a genetic algorithm to refine a control strategy based on RMPs to approximate the optimal response of the follower aircraft. Two ICs were tested and each resulted in a refined CSE that produced a response significantly closer to the optimal trajectory than any individual follower response trajectory. Future work is planned to test numerous ICs across a state-space with enough fidelity to allow for implementation for any generic IC. Regions where the same RMP is used will be investigated and combined to produce the most efficient control strategy. Additional leader trajectories are also planned for research. The final goal is a control strategy which can be executed in real-time in a lookup based format for a wide array of initial conditions.

References

[1] Lt Col Van Riper, Steven G., "Apache manned-unmanned teaming capability," Army Magazine, 2014. 64.

[2] Durbin, D. B., and Hicks, J. S., "AH-64D Apache Longbow Aircrew Workload Assessment for Unmanned Aerial System (UAS) Employment," Tech. Rep. ADA494123, Defense Technical Information Center, 2007.

[3] Humphreys, C., "Optimal Control of an Uninhabited Loyal Wingman," Ph.D. thesis, Air Force Institute of Technology, 2016.

[4] Humphreys, C. J., Cobb, R. G., Jacques, D. R., and Reeger, J. A., "A Hybrid Optimization Technique Applied to the Intermediate-Target Optimal Control Problem," *Global Journal of Technology and Optimization*, Vol. 7, No. 2, 2016. Https://doi.org/10.4172/2229-8711.1000200 Issue table of contents in lieu of DOI: https://www.hilarispublisher.com/archive/gjto-volume-7-issue-2-year-2016.html.

[5] Giacomossi, L., Schwanz Dias, S., Brancalion, J. F., and Maximo, M. R. O. A., "Cooperative and Decentralized Decision-Making for Loyal Wingman UAVs," *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*, 2021, pp. 78–83. https://doi.org/10.1109/LARS/SBR/WRE54079.2021.9605468.

[6] Park, S., Deyst, J., and How, J. P., "A New Nonlinear Guidance Logic for Trajectory Tracking," *AIAA*, 2004, pp. 1–16. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2004-4900.

[7] Gotwald, C., Zollars, M., and Weintraub, I., "Determing Follower Aircraft's Optimal Trajectory in Relation to a Dynamic Formation Ring," *IEEE Big Sky 2023*, 2023, pp. 1–11. Prepublication version available: https://doi.org/10.48550/arXiv.2210.01665.

[8] Roup, A., Laird, P., Kirchner, W., and Foreman, M., "Relational Maneuvering Algorithm Description Document," Tech. rep., Air Force Research Laboratory (AFRL), 2020.

[9] Dubins, L. E., "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, Vol. 79, No. 3, 1957, pp. 497–516.

[10] NIMA, "Department of Defense World Geodetic System 1984," Tech. rep, National imagery and Mapping Agency, 2004.

[11] Patterson, M. A., and Rao, A. V., "GPOPS - II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Transactions on Mathematical Software*, Vol. 41, 2014. https://doi.org/10.1145/2558904.

[12] Bogaert, I., "Iteration-Free Computation of Gauss-Legendre Quadrature Nodes and Weights," *SIAM Journal of Scientific Computing*, Vol. 36, No. 3, 2014, pp. A1008–A1026. Available: https://doi.org/10.1137/140954969.

[13] Wachter, A., and Biegler, L. T., "On the immplementation of a primal-duel interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–27. Preprint at http://www.optimization-online.org/DB_HTML/2004/03/836.html.

[14] Patterson, M. A., and Rao, A. V., *GPOPS-II User Guide*, 2nd ed., Dec 2016.

[15] Tran, D., Casbeer, D., Garcia, E., and Weintraub, I., "Ring Formation Maneuver: Double-Integrator Kinematics with Input Saturation," *Journal of Guidance, Control, and Dynamics*, 2021.

[16] Geeks for Geeks, "Genetic Algorithms," Online, Sep 2022. Available: https://www.geeksforgeeks.org/genetic-algorithms/.

[17] Rao, A. V., "A Survey of Numerical Methods for Optimal Control," *Advances in the Astronautical Sciences*, Vol. 135, No. 1, 2010.

[18] Jong, K. A. D., "Analysis of the Behavior of a Class of Genetic Adaptive Systems," Phd thesis, University of Michigan, Aug 1975.

[19] Deb, K., "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2-4, 2000, pp. 311–338.

[20] Deep, Kusum, Singh, K. P., Kansal, M. L., and Mohan, C., "A real coded genetic algorithm for solving integer and mixed integer optimization problems," *Applied Mathematics and Computation*, Vol. 212, No. 2, 2009, pp. 505–518.