

Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

9-2021

A Real-time Algorithm to Achieve Precise Coordinated Arrival Times in a Time-variant Environment

Shawn S. Stephens

Follow this and additional works at: <https://scholar.afit.edu/etd>



Part of the [Aerospace Engineering Commons](#)

Recommended Citation

Stephens, Shawn S., "A Real-time Algorithm to Achieve Precise Coordinated Arrival Times in a Time-variant Environment" (2021). *Theses and Dissertations*. 5093.

<https://scholar.afit.edu/etd/5093>

This Dissertation is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact AFIT.ENWL.Repository@us.af.mil.



**A REAL-TIME ALGORITHM TO ACHIEVE
PRECISE COORDINATED ARRIVAL TIMES
IN A TIME-VARIANT ENVIRONMENT**

DISSERTATION

Shawn S. Stephens, Major, USAF

AFIT-ENY-DS-21-S-109

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-DS-21-S-109

A REAL-TIME ALGORITHM TO ACHIEVE PRECISE
COORDINATED ARRIVAL TIMES IN A TIME-VARIANT ENVIRONMENT

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Aeronautical Engineering

Shawn S. Stephens, B.S.A.E., M.S.E.

Major, USAF

16 September 2021

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENY-DS-21-S-109

A REAL-TIME ALGORITHM TO ACHIEVE PRECISE
COORDINATED ARRIVAL TIMES IN A TIME-VARIANT ENVIRONMENT

DISSERTATION

Shawn S. Stephens, B.S.A.E., M.S.E.
Major, USAF

Committee Membership:

Dr. D. R. Kunz
Chairman

Dr. W. P. Baker
Member

Maj J. A. Hess, PhD
Member

Dr. S. R. Graham
Dean's Representative

Abstract

The coordinated arrival time problem seeks to control a vehicle’s trajectory to achieve some pre-defined final state at a desired arrival time. The persistent monitoring problem is a type of coordinated arrival problem where a stationary or moving ground target must be constantly observed by a group of aircraft. This research simplifies the problem to overfly the target at specific time intervals relative to the other aircraft in the group while increasing the difficulty of the problem by considering aircraft traveling at vastly different airspeeds. Previously, the problem has been solved with “plan-and-follow” algorithms or real-time guidance algorithms. However, the “plan-and-follow” method must recalculate paths when the scenario changes and real-time guidance algorithms do not incorporate keep-out zone constraints well. This research combines the benefits of both methods. The result of this research is a real time guidance algorithm which accurately guides a group of airspeed heterogeneous aircraft to achieve a desired relative arrival time, predictable flight path, and accounts for varying wind conditions, keep-out zones, and non-cooperative targets. A wide variety of scenarios were evaluated with aircraft ranging in speed from 174 knots to 511 knots. However, a notable scenario utilizing a nonlinear fighter aircraft model demonstrated the algorithm’s ability to guide a group of four aircraft to achieve the desired time spacing to within 0.1 seconds in a scenario with 160 knot winds, a target moving at 102 knots, keep-out zones, and moderate wind turbulence.

I want to thank my wife for keeping me focused on the big picture and supporting me through these fun three years. She's always there to lend an ear while I work out a problem or offer an outside viewpoint which has been the key to solving a problem on more than one occasion.

Acknowledgements

I'd like to thank my advisor, Dr. Kunz, for pointing me in the right directions and keeping me on track whenever my work drifted away from end goal. I appreciate the help in of my committee, Dr. Baker and Maj Hess, for all the time invested in reviewing papers, tackling math problems, and providing the critical insight to ensure that I understood what I was doing and how I was doing it.

Finally, this work would not have been possible if not for the support from the AFRL Control Systems Center. Dr. Casbeer, Dr. Tran, Dr. Doman, and Dr. Bolender have all been a significant part of this work whether it was pointing me in the right direction to formulating a problem or providing a fresh look at solving the problem. I appreciate all their time and effort which helped to complete this research.

Shawn S. Stephens

Table of Contents

	Page
Abstract	iv
Acknowledgements	vi
List of Figures	xii
List of Tables	xvi
List of Symbols	xxiii
List of Abbreviations	xxiv
I. Introduction	1
1.1 Motivation	2
1.2 Research Questions, Tasks, and Scope	5
Research Questions	5
Research Scope	5
Research Tasks	6
1.3 Assumptions and Limitations	7
1.4 Research Methodology	7
1.5 Expected Contributions	9
1.6 Document Outline	10
II. Literature Review	11
2.1 Introduction	11
2.2 Path Planning	11
2.3 Plan-and-Follow	12
Path Planning Methods	12
Path Following Guidance Laws	27
Arrival Time Control	31
Plan-and-Follow Summary	32
2.4 Real-Time Trajectory Guidance	33
Proportional Navigation Guidance Law	33
Virtual Target	34
Optimal Control Guidance	35
Real-Time Trajectory Guidance Summary	36
2.5 Methods to Solve Optimal Control Problems	37
Numerical Methods to Solve Optimal Control Problems	37
Indirect Methods to Solve Optimal Control Problems	39
2.6 String Stability	41
String Stability Example	42

	Page
String Stability Literature Review	44
2.7 Summary	46
III. Aircraft and Wind Models	48
3.1 Introduction	48
3.2 Simplified Aircraft Model	48
State Assumptions	51
Control Assumptions	53
Simplified Model Summary	56
Aircraft Model Configurations	59
Simplified Model Bank Angle Command System	60
3.3 Nonlinear Aircraft Model	61
Nonlinear Aircraft Model Control Systems	62
Circular Path Tracking Control System	67
Summary	69
3.4 Wind Turbulence Effects Model	70
3.5 Wind Triangle	73
3.6 Wind Measurement Model	74
3.7 Aircraft and Environmental Models Summary	75
IV. Optimal Lead Turn Problem	76
4.1 Optimal Lead Turn Problem Definition	78
Cost Functional	79
Terminal Constraints	79
Mathematical Definition	81
Numerical Solutions	82
4.2 Near-Optimal Lead Turn Problem	83
4.3 Near-Optimal Lead Turn Solution	85
Minimum Time-to-Bank Solution	85
Near-Optimal Lead Turn Solution	96
4.4 Near-Optimality of the Solution	103
4.5 Lead Turn with Existing Guidance Laws	105
Guidance Law Integration	105
Guidance Law Scenarios	112
4.6 Guidance Law Integration Performance Analysis	114
Comparison Metric	114
Results	114
4.7 Near-Optimal Lead Turn Conclusions	117

	Page
V. Racetrack Algorithm Development	119
5.1 Introduction	119
5.2 Primary Controls	119
5.3 Racetrack Reference Frame	120
5.4 Arrival Time Estimation	124
Straight Section Time Estimation	124
Curved Section Time Estimation	125
Path Acquisition	135
5.5 Arrival Time Control	136
Arrival Time Control Via Airspeed	137
Arrival Time Control Via Path Length	139
5.6 Path Tracking Effect on Arrival Time	141
Straight Segments	142
Curved Segments	142
5.7 Model Predictive Control Path Follower	144
Arrival Time Control Features	146
Model Predictive Control Parameters and Performance	147
5.8 Racetrack Manager	149
Arrival Time Communication	150
Enabling Airspeed Control	150
Endpoint Estimation	150
Racetrack Reset Trigger	152
Keep-Out Zone Determination	153
Quadrant Control	158
VI. Racetrack Algorithm Stability Analysis	165
6.1 Definitions	165
6.2 Arrival Time Problem Dynamics	167
Wind and Airspeed Relationship	168
Wind Assumptions	169
Error Definitions	171
Virtual Target Control	173
Control Bounds	181
Control Selection	182
6.3 String Stability Analysis	183
Equilibrium Point	186
Simplifying the Time Variant System	186
Single Vehicle Stability	188
String Stability Analysis	190
6.4 Stability Analysis Summary	193

	Page
VII. Simulation Results and Performance Characterization	194
7.1 Performance Metrics and Variables	194
7.2 Solver Configurations	196
7.3 Initial Controller Performance Comparison	198
Controller Configurations	198
Static Target Simulations	199
Moving Target Simulations	202
Controller Selection	206
7.4 Arrival Time Stability Analysis	206
String Stability	207
Single Aircraft Nominal Performance	209
Saturated Airspeed and Turbulence	212
Effect of Path Tracking Error on Arrival Time	213
7.5 Final Controller Configurations	215
7.6 Final Controller Performance Characterization	217
Static Target Simulations	217
Moving Target Simulations	220
Single Vehicle Analysis	230
7.7 Cooperative Control Simulations	232
Static Target with Keep-Out Zones	233
Moving Target with Keep-Out Zones	235
Non-Cooperative Target with Keep-Out Zones	238
7.8 Nonlinear Aircraft Model Simulations	240
Comparison to Simplified Model	241
Nonlinear Cooperative Control Scenarios	243
7.9 Results Summary	250
VIII. Conclusion	251
8.1 Research Questions and Contributions	251
Optimal Lead Turn	253
Arrival Time and Airspeed Control	255
Keep-Out Zone Constraints	257
Arrival Time Control Algorithm Performance	258
Summary	262
8.2 Future Work and Recommendations	262
Flight Test	262
Racetrack Algorithm Research	263
Optimal Lead Turn Research	264
Appendix A. Near-Optimal Lead Turn Solution Code	266

	Page
Appendix B. Proofs	270
B.1 Bounds on $s_i(t)$	270
Bibliography	273
Vita.....	281

List of Figures

Figure	Page
1	Dubins Paths 14
2	Linear and Nonlinear Dynamic Inversion Guidance Comparison 28
3	String Stable Example 43
4	String Unstable Examples 43
5	Linear Quadratic Regulator Bank Angle Command System 61
6	Altitude-Hold Autopilot Overview 63
7	Bank-Hold Autopilot Overview 67
8	Bank-angle Command Climb Adjustment 69
9	Outer-loop Autopilot Performance 70
10	Moderate Turbulence Model Output with Two Standard Deviations (Dashed Lines) 72
11	Wind Triangle 73
12	Dubins Path Following Example 77
13	Optimal Lead Turn Problem Setup 79
14	GPOPS-II Numeric Solution to the Optimal Lead Turn Problem 82
15	Minimum Time-to-Bank Solution Comparison, States 96
16	Minimum Time-to-Bank Solution Comparison, Costates 97
17	Nonlinear Guidance Law Parameters; Ref. [1] 107
18	Course Command to Bank Angle Command 109
19	Synthetic Waypoint Guidance Circular Orbit Error 112
20	“Lead/Optimal” Scenario Example, Look-ahead Point Identified as Star 113

Figure	Page
21	Nonlinear Guidance Law (NGL) Example: Light Utility Aircraft, Light Turbulence, with Wind 115
22	ℓ^2 -norm Percent Difference from Control Scenario 116
23	Synthetic Waypoint GuidanceError Comparison 117
24	Effect of Adjusting a on the Racetrack Shape 120
25	Left-Hand Racetrack Definition 121
26	Left-Hand Racetrack Definition 122
27	Reference Frame Rotation 123
28	Straight Racetrack Section, Quadrant IV 125
29	Curved Racetrack Section, Quadrant I 127
30	Average Airspeed Quadrature Estimate Comparison 130
31	Constant Bank Angle Turn Ground Path 131
32	Air Mass Turn Calculations 132
33	Velocity used in Time Estimation Example 138
34	Effect of Adjusting a on the Racetrack Shape 139
35	Straight Racetrack Section, Quadrant IV 142
36	Circular Segment Error 143
37	Path Follower Variables 145
38	Heavy Cargo Aircraft Path Following Performance Example 148
39	Fighter Aircraft Arrival Time Control Feature Example 149
40	Method 1: Constant Target Speed, Fixed End Point 151
41	Method 2: Constant Target Speed, Moving End Point 152
42	Change in Superellipsoids with ε_2 155
43	Left-Hand Racetrack Definition 156

Figure	Page
44	Left-Hand Racetrack Definition 159
45	Ground-fixed Path Nominal State Machine 159
46	Wind Direction Effect on Ground Track 160
47	Constant Bank Angle Quadrant Determination 162
48	Target Stop at time $t = t_1$ 163
49	Full Quadrant State Machine 164
50	Wind Triangle Variables 169
51	Virtual Target Variables 174
52	Rate-Limiting and Saturated Sets 185
53	Initial Comparison Static Endpoint Arrival Time Error (s) 200
54	Ground Track Comparison, ‘×’ Markers are Equivalent Time 201
55	Static Endpoint Arrival Time Errors (s) v. Wind Direction (deg) 202
56	Initial Comparison Multi-pass Average Arrival Time Error (s) 204
57	Initial Comparison Non-cooperative Target Average Magnitude of Arrival Time Error (s) 205
58	Twenty Aircraft, String Stable with Wind and Turbulence 208
59	Single Aircraft with Turbulence and Wind 210
60	Airspeed Saturated Aircraft with Turbulence; $T_{e_{i-1}}(t) > 5$ 212
61	Final Algorithm Static Arrival Time Error 219
62	Final Algorithm Static Arrival Airspeed Error 220
63	Multi-Pass Setup Example 221

Figure	Page
64	Repeat Pass Comparison to Static Target 222
65	Multi-pass Average Arrival Time Error (s), Varying Wind Speed and Target Speed 223
66	Multi-pass Average Arrival Time Error (s), Varying Target Speed and Heading 224
67	Changing Target Heading for a Counterclockwise Orbit 225
68	Multi-pass Average Arrival Time Error (s), Varying Wind Direction and Target Direction 225
69	Final Algorithm Non-cooperative Arrival Airspeed Error 227
70	Non-Cooperative Target Speed Ratio v. Target Stop Time Ratio 228
71	Non-Cooperative Target Stop Time Ratio v. Target Relative Heading 229
72	58 ft/s Target stopping with aircraft in first turn 231
73	Sample Scenario Time History 232
74	Scenario 1: Static with Keep Out Zone 235
75	Scenario 2: Moving Target with Keep Out Zone 237
76	Scenario 3: Non-Cooperative Target with Keep Out Zone 240
77	Simplified Model and Nonlinear Model Repeat Pass Comparison 242
78	Nonlinear Model: Static Target with Keep Out Zone 245
79	Nonlinear Model: Moving Target with Keep Out Zone 247
80	Nonlinear Model: Stopping Target 249

List of Tables

Table	Page
1	Aircraft Class Parameters 59
2	Global Wind Turbulence Parameters from Ref. [2] 71
3	Aircraft Dependent Wind Turbulence Parameters 71
4	Turbulence Model Two Standard Deviations Value (Light/Moderate) 72
5	Taylor Polynomial Approximation Solution (TPAS) Optimality Metric 105
6	Aircraft Class Parameters; reproduced from Table 1 106
7	Vector Field Follower Parameters 108
8	Vector Field Follower Course Hold Parameters 109
9	Nonlinear Dynamic Inversion Follower Parameters, all aircraft 110
10	Synthetic Waypoint Guidance Course Hold Parameters 111
11	Model Predictive Control Aircraft Parameters 148
12	Nominal Arrival Error and Turbulence Effects 211
13	Zero steady wind, Moderate Turbulence, 500 ft path error Comparison 214
14	Aircraft Dependent Parameters 216
15	Simplified Model Static Results at Orbit Completion 234
16	Simplified Model Multi-Pass Results at Orbit Completion 236
17	Simplified Model Non-Cooperative Results at Orbit Completion 239
18	Nonlinear Model Static Results at Orbit Completion 245
19	Nonlinear Model Multi-Pass Results at Orbit Completion 247

Table		Page
20	Nonlinear Model Non-Cooperative Results at Orbit Completion	249

List of Symbols

Symbols

- a** Acceleration
- a Racetrack semi-major axis size
- $\mathcal{A}/\mathcal{B}_i$ Non-dimensional upper/lower airspeed rate limit for aircraft i
- A_i/B_i Upper/lower airspeed rate limit for aircraft i , (ft/s)
- a_c Centripetal acceleration
- a_s Lateral Acceleration Command; Ref. [3]
- $\mathbf{A}(t)$ Linear Time-variant system \mathbf{A} matrix, Chapter VI
- b Racetrack Turn Radius
- $\mathbf{C}_{r/q}$ Rotation matrix from reference frame q to r
- c Cosine
- d Distance
- D_T Turn Direction Variable
- \mathbf{E} $2 \times n$ vector of n keep-out zone “query” point coordinates
- \mathbf{e}_3 Unit vector along the third cartesian axis
- e Error
- F Superquadric “in-out” function
- G Groundspeed
- g Acceleration due to Gravity; Positive
- \mathcal{H} Hamiltonian
- h Height
- J Cost Functional
- J_x Moment of Inertia about the X-axis
- J_{xz} Product of Inertia about the X-axis and Z-axis
- J_y Moment of Inertia about the Y-axis

J_z	Moment of Inertia about the Z-axis
k_{ARI}	Aileron-rudder interconnect gain
k_I	Integral Gain
k_{div}	Vector Field Follower Gain
k_{line}	Vector Field Follower Straight Path Gain
k_{orbit}	Vector Field Follower Circular Path Gain
k_P	Proportional Gain
k_T	Arrival Time Error Gain
k_V	Non-dimensionalized Airspeed Error Gain
k_δ	MPC heading error gain
ℓ	Applied Roll Moment, Chapter III
L_1	Constant value of costate 1
L_D	Look-ahead distance for the Non-linear Guidance Law; Ref. [3]
m	Mass
$\mathbf{M}(\theta)$	Clockwise Axis Rotation Matrix for Angle θ
m	Applied Pitch Moment, Chapter III
m_i	Multiplicative wind parameter for aircraft i , Chapter VI-VII
\mathcal{M}	Set of reachable airspeeds
N	Load Factor
n	Applied Yaw Moment, Chapter III
P	Body-Axis Roll Rate
p	Position
Q	Body-Axis Pitch Rate
q	Arbitrary Variable
Q_{I-IV}	Racetrack Quadrant: I, II, III, or IV
R	Body-Axis Yaw Rate

R_T	Circular Path Turn Radius
s	Sine
s_0	Initial arc length
S_i	Arrival time spacing for aircraft i
s_i	Airspeed to groundspeed ratio for aircraft i , Chapter VI
T	Arrival Time
U	X-Body-Axis Velocity, Chapter III
u_V	Control
V	Y-Body-Axis Velocity, Chapter III
\mathcal{V}_i	Non-dimensional airspeed for aircraft i
V_q	Airspeed identified by subscript q
V_s	Airspeed projected along groundspeed vector
V_T	True airspeed
\mathbf{w}	System disturbance vector, Chapter VI
W	Z-Body-Axis Velocity, Chapter III
W_χ	Course Controller Design Parameter
W_f	Filtered Windspeed
w_i	Additive wind parameter for aircraft i , Chapter VI-VII
w_i	Quadrature weight i , Chapter V
W_q	Wind component velocity along the arbitrary axis q
W_s	Windspeed
W_t	Unmeasured or turbulent Windspeed
X	X-Body-Axis Force
x	x-axis position
\mathbf{x}	State variable vector, Chapter VI
X_T, Y_T, Z_T	Force due to Thrust

Y	Y-Body-Axis Force
y	y-axis position
s_0	Initial y-position
Z	Z-Body-Axis Force

Greek Symbols

α	Angle of Attack
β	Angle of Sideslip
γ	Flight-path Angle
δx_{turn}	Incremental change in $x^{(r)}$ during a turn
δ	Actuator Deflection, Chapter III
δ	Bounded variable, Chapter VI
δ	MPC Heading Error, Chapter V
$\Delta\chi$	Vector Field Follower Course Angle Error from Path
$\varepsilon_{1,2}$	Superquadric shape variable along axis 1 or 2, controls “roundness”
ζ_{ndi}	Desired Damping Ratio of the NDI Controller
ζ_χ	Desired Damping of Course Controller
η	Line of Sight Angle;Ref. [3]
θ	Pitch Angle, Chapter III). Arbitrary angle (elsewhere)
κ	Curvature
$\lambda_{1,2}$	Costate 1 or 2, Chapter IV
ρ	Bounded variable, Chapter VI
σ	Standard Deviation, Chapter III
σ_i	Variable projecting true airspeed along ground path, Chapter VI
τ	Time Constant
ϕ	Roll Angle
ϕ^\dagger	Solution with zero initial bank angle, Chapter IV

ϕ'	Referring to the actual bank angle, Chapter IV
ϕ_{cmd}	Bank Angle Command
ϕ_m	Constant bank angle solved for in MPC algorithm, Chapter V
χ	Course Angle
χ_f	Final Desired Ground Course
$\chi_{cmd_{orbit}}$	Vector Field Follower Orbit Course Command
χ_{far}	Vector Field Follower Orbit Far Field Course Command
$\chi_{cmd_{line}}$	Vector Field Follower Straight Line Course Command
ψ	Heading Angle
ω	Angular Rate
$\omega_{n_{ndi}}$	Desired Natural Frequency of the NDI Controller
ω_{n_χ}	Desired Natural Frequency of Course Controller
$\omega_{P,Q,R}$	Turbulence in the P , Q , or R axis

Superscripts

'	Rotated axis variable, Chapter V
'	Velocity relative to the wind, Chapter III
*	Optimal solution, Chapter IV
+/-	Variable upper/lower command limit, Chapter VI
(a)	Moving Air-mass Reference Frame
(k)	Keep-out zone Superellipsoid Reference Frame
(e)	Referenced to the Earth-Fixed Reference Frame
(r)	Referenced to the Racetrack Reference Frame

Subscripts

A	Force due to Aerodynamics
ap	Function approximation
avg_t	Average over time

avg_w Weighted average
avg_θ Average over angle
c Center of the Racetrack
f Endpoint of the Racetrack
cmd Commanded Value
cross Crosswind Value
D Down
E East
e_i Error for aircraft *i*
eq System equilibrium point
frd Forward-right-down
m_i Measured term for aircraft *i*
min/max Minimum or maximum value
N North
ned North-east-down
P Roll Axis
q pitch Axis
r yaw Axis
ref Reference or desired value
rem Time Remaining
tgt Referring to the overfly target
u_i Unmeasured term for aircraft *i*
V along velocity axis
x Along x-axis
y Along y-axis
Z Z-axis

List of Abbreviations

AIAA	American Institute of Aeronautics and Astronautics
ARI	aileron-rudder interconnect
CDA	continuous descent arrival
DCASS	Dayton-Cincinnati Aerospace Sciences Symposium
DCM	direction cosine matrix
DoD	Department of Defense
ILS	instrument landing system
ISR	intelligence, surveillance, and reconnaissance
JGCD	Journal of Guidance, Control, and Dynamics
LDI	linear differential inclusion
LQR	linear quadratic regulator
LTV	linear time variant
MPC	model predictive control
NDI	nonlinear dynamic inversion
NGL	nonlinear guidance law
NLP	non-linear program
PH	pythagorean hodograph
PI	proportional integral
PID	proportional integral derivative
SWG	synthetic waypoint guidance
TPAS	Taylor polynomial approximation solution
VFF	vector field follower

A REAL-TIME ALGORITHM TO ACHIEVE PRECISE
COORDINATED ARRIVAL TIMES IN A TIME-VARIANT ENVIRONMENT

I. Introduction

The Merriam-Webster dictionary defines automation as the “automatically controlled operation of an apparatus, process, or system by mechanical or electronic devices that take the place of human labor” [4]. Humans have been automating apparatus, processes, and systems since the first tools were invented. Recently, there has been growth in the field of aviation automation. The primary goal of automation in aviation over the last century has been to reduce pilot workload by taking pilot performed tasks and having some other system perform the tasks. One of the first tasks to be automated was the direct control of aircraft; automated with the invention of the autopilot [5].

The first mechanical autopilot was demonstrated in public on a Curtiss C-2 biplane in 1914 [5]. This autopilot was only able to control heading and attitude but represented a major accomplishment since the first powered airplane, the Wright Flyer, had flown only 11 years earlier in 1903 [5, 6]. The development of the autopilot continued at a steady pace during World War I and World War II resulting in the first transatlantic flight under the control of an autopilot to include takeoff and landing [6]. Aircraft continually flew faster and higher eventually requiring power-boostered flight controls to provide enough power to move the control surfaces. The need for boostered controls led to the development of automatic stability augmentation which was notably demonstrated on the X-15 rocket plane in the 1950s and 1960s [6].

The next major advancement in automatic control was the development of the

fly-by-wire aircraft. The first digital fly-by-wire (DFBW) aircraft, an F-8 Crusader modified by NASA, took flight in 1972 [7]. The technology demonstrated by the DFBW test aircraft eventually supported the development of the F-16 and the Space Shuttle [6].

The combination of digital systems with the concept and operation of the autopilot has further expanded the capabilities of the autopilot. Today autopilots are able to maintain altitude, change to an assigned altitude, intercept courses, and even guide an aircraft along a pre-programmed path of waypoints [8]. However, the overall purpose of the autopilot remains unchanged: to help “the pilot focus on the overall status of the aircraft and flight” [8].

1.1 Motivation

The “Department of Defense (DoD) Unmanned Systems Integrated Roadmap: 2017-2042” calls for “technological advancements that increase the efficiency and effectiveness of unmanned systems” [9]. These types of advancements will allow unmanned systems to perform a broader range of tasks (routine navigation, continuous monitoring, arrival time coordination) with reduced input from human operators. Subsequently, operators will be able to shift their focus from controlling a single unmanned system to managing an entire mission made up of multiple manned and unmanned systems. One of the next steps in achieving these advancements is the automation of tasks which do not require advanced logic such as artificial intelligence.

Coordinating the arrival time of aircraft is a task which is ready to be automated. This task requires an aircraft to arrive at a desired state (position, airspeed, etc.) at some specified time. The specified arrival time may be prescribed such that a group of vehicles arrives at their desired states at the same time however, simultaneous arrival is not the only possibility.

In the case of commercial aircraft arriving at an airport, a set of aircraft should arrive at specific but separate times in order to allow a plane to land and exit the runway before the next aircraft is in a position to land. In the commercial landing scenario, the aircraft may be assigned "landing times" to target which creates multiple problems which are not linked together.

Time coordination is also an important task to the United States Air Force because it enables time synchronization. Time synchronization is key to two of the nine United States Principles of Joint Operations: mass and surprise [10]. The two principles depend on the ability to control the time at which a military operation occurs. Furthermore, automating this task supports the "mission command" tenet of airpower. Automating this task allows the "centralized command, distributed control, and decentralized execution," a key component of mission command, of the arrival time coordination task by moving the moving the execution and management of the time coordination task from an operator to the automated aircraft system [10].

Additionally, the commercial aircraft landing scenario may be changed such that the aircraft must land a set time apart and now the multiple problems are linked by their landing times. This change represents a variation of the continuous monitoring problem. The continuous monitoring problem is a term used in intelligence, surveillance, and reconnaissance (ISR) to describe a scenario where a stationary or moving target must continuously have a sensor observing it. The problem is a type of coordinated arrival problem because the start of an observation segment may be viewed as the specified arrival time. Subsequently, the next vehicle is prescribed to arrive at its endpoint when the previous vehicle finishes its observation. In this scenario, the end of observation of the first vehicle is now linked to the start of observation of the second vehicle.

Although the coordinated arrival time task is clearly useful to military operations,

it is not unique to the military. This task is performed many times every day by pilots and air traffic controllers as aircraft are routed in and out of airports while maintaining a safe but efficient physical separation. A goal of the Next Generation Air Transportation System is to rely on a time-based air traffic management system in the 2026-2030 time frame [11].

One of the capabilities of such a system will be to “accurately predict the aircraft’s 4-D trajectory” in order support to higher throughput; a “4-D trajectory” includes time in addition to the three spatial dimensions. Currently, air traffic controllers issue speed, altitude, and heading commands to aircraft in order to achieve time coordination. The next step in air traffic automation is for simple arrival time commands to be sent from the ground controllers to the aircraft. The aircraft will then be responsible for meeting the specified arrival time while avoiding any airspace restrictions and providing a predictable flight path. A predictable flight path allows air traffic controllers and other aircraft to identify and address any potential conflicts, such as a mid-air collision with another aircraft, which may occur along the aircraft’s flight path.

In order to automate the coordinated arrival time task, the goal of this research is to develop a practical control algorithm which solves the coordinated arrival time problem in real time while considering environmental effects, airspace, and maintaining a predictable flight path. The goal of this research is to allow scenarios similar to the continuous monitoring problem where the arrival time of an aircraft may either be fixed or dependent on a preceding vehicle.

1.2 Research Questions, Tasks, and Scope

Research Questions.

Hypothesis: A control algorithm exists which may predictably guide an aircraft through a constrained environment to achieve a desired final position, heading, and velocity at a specified arrival time by adjusting both the speed and flight path of the aircraft

The research questions related to this hypothesis are:

1. What is the time optimal lead turn time for an aircraft, controlled by roll rate, to closely follow a Dubins path?
2. What technique is best suited to control the arrival time and velocity of an aircraft in near real time?
3. Is a keep-out zone avoidance feature able to be integrated into the arrival time control algorithm?
4. How does an arrival time control algorithm perform in the presence of environmental anomalies and time dependent constraints?

Research Scope.

The research is focused on developing predictable methods to autonomously control the arrival time and arrival velocity of aircraft flying through a realistic wind environment while adhering to predefined keep-out zone constraints. There are many options to solve this problem which involve preplanning the path, selecting waypoints, and then controlling velocity along the defined path. This area has been well studied therefore, the author believes the real time control of the path and velocity simultaneously deserves study.

The primary result of this work is the development of a stable multi-vehicle arrival time control algorithm and the characterization of the control algorithm across a wide range of realistic situations. In this work, stable refers to the property where the response of the dynamic system (i.e., the closed loop control algorithm) is guaranteed to remain bounded. The simulation scenarios are focused on single-vehicle situations which replicate individual components of a multi-vehicle scenario. However, a small number of multi-vehicle scenarios are studied to ensure the controller behaves as predicted by the single vehicle characterization and determine any the limitations of the controller.

Research Tasks.

A number of tasks are accomplished which address the above research questions.

1. Question 1: Solve the minimum time to bank optimal control problem
2. Question 1: Solve the optimal lead turn time problem
3. Question 2: Investigate methods to control arrival time using the aircraft's flight path.
4. Question 2/4: Develop a practical arrival time control algorithm that is capable of providing precise arrival time and velocity control in the presence of environmental effects and time dependent constraints.
5. Question 3: Investigate and select a method to incorporate keep-out zone constraints into an arrival time control algorithm.
6. Question 4: Perform single-vehicle and multi-vehicle simulations with stationary/moving targets, single/multiple target visits, environmental effects, and keep-out zones.

1.3 Assumptions and Limitations

- The research focuses on fixed wing aircraft flying at constant altitude. This simulates the aircraft being assigned a fixed altitude in an operational environment.
- A nonlinear aircraft model is used to evaluate the arrival time control algorithm for a small subset of simulations. These simulations are used to ensure the algorithm behaves as predicted when integrated into a realistic aircraft model. The nonlinear aircraft model is the F-16 model from Ref. [6]. A simple altitude hold and bank angle control system are developed to accept inputs from the arrival time control algorithm.
- It is assumed the arrival time control algorithm has knowledge of its position relative to the desired arrival point, true airspeed, wind speed, heading, and bank angle.
- The research only addresses predefined keep-out zones and not “pop-up” or time-dependent keep-out zones.

1.4 Research Methodology

The dissertation research is divided into four parts. The first part defines the dynamics models used in this research. A majority of the research utilizes a simplified dynamics model which is derived from the nonlinear equations of motion. A number of assumptions are applied to the nonlinear equations of motion to reduce the number of state variables in the model while accurately modeling fixed wing dynamics. The primary benefit of the model is the ability to solve optimal control problems, rapidly perform simulations, and change the aircraft characteristics with a few simple

parameters. Four types of aircraft models are defined used to determine how aircraft performance affects the performance of the arrival time algorithm.

A second, nonlinear, dynamics model is used to verify the results of the simplified model hold for a nonlinear model. The model used is the F-16 dynamics model from Stevens and Lewis [6]. However, this model represents just the aircraft dynamics where the inputs are the control surface deflections. Therefore, a control system which accepts the command inputs from the arrival time control is also developed. Finally, the environmental model used to generate non-steady wind turbulence is introduced. The Dryden Wind Turbulence model in MATLAB[®] is used due to its ease of integration into Simulink.

The second part of the research focuses on solving the optimal lead turn problem. This optimal control problem addresses a common issue observed when following paths defined by straight lines and circular arcs, commonly referred to as Dubins paths. These paths are commonly used in aircraft path planning due to their simplicity. The transition between the straight and curved path segments would require an instantaneous change in bank angle to be followed precisely by an aircraft; which is clearly not feasible. This problem is applicable to this research due to the assumed path shape used by the arrival time control algorithm, a racetrack.

The goal of the optimal lead turn time problem is to find the control and initial time/position for an aircraft to transition from the straight path segment to the curved path segment in minimum time, or vice versa. The dynamics utilize the simplified model which expands upon Dubins vehicle dynamics by including roll rate and roll acceleration to model the aircraft roll mode. Determining the optimal lead turn time for an aircraft to follow a Dubins path will allow the arrival time control algorithm to better follow Dubins paths.

The third part of the research develops and analyzes an algorithm which is able

to solve the arrival time problem in real time. The first portion of this phase defines the path being followed, a racetrack, and the various algorithm subsystems. These subsystems determine the racetrack size, enforce keep-out zones, and guide the aircraft to and along the path. The second portion of this phase derives a method to ensure the stability of the algorithm when used in a multi-vehicle scenario.

Finally, the fourth part of the research consists of testing the capabilities of the control algorithm in a simulation environment. Simulations are primarily performed on a simplified aircraft model to more efficiently study variables which affect the performance of the algorithm. The simulations include a wide range of wind environments, arrival time constraints, and moving target conditions. The simulation difficulty is then increased by studying the effects of changing the desired endpoint and desired arrival time after the simulation has begun. Multiple arrivals, simulating a go-around or continuous monitoring, are also studied.

Multi-vehicle scenarios are also studied where the spacing between arrival times is the priority and the desired arrival time multiple vehicles will depend on the preceding vehicle. These scenarios directly simulate the continuous monitoring problem and coordinated arrival time problem. Finally, the algorithm is tested using the non-linear model on the multi-vehicle scenarios to study the performance of the algorithm in a more realistic environment.

1.5 Expected Contributions

Specific contributions to the control of autonomous vehicles are made in this research:

- A near-optimal solution to the optimal lead turn time problem
- A framework for a real-time arrival time controller which compensates for

steady-state winds, is robust to random turbulence, allows for repeat visits, and operates within keep-out zone constraints

- A method to guarantee the string stability of a group of airspeed heterogeneous, nonlinear aircraft with a cascaded interconnection utilizing the developed arrival time algorithm.
- A performance characterization of the real-time arrival time controller across a range of wind conditions and arrival requirements
- A demonstration of the arrival time controller in single and multi-vehicle scenario with cooperative and non-cooperative targets

1.6 Document Outline

The research hypothesis of this paper along with the related questions and tasks have been presented in this chapter. Chapter II will present a survey of existing literature related to the arrival time problem. Chapter III derives and defines the aircraft and environmental models this research uses to solve the arrival time problem. Chapter IV introduces and solves the optimal lead turn problem then analyzes the solutions effect on a variety of guidance laws. Chapter V develops the racetrack algorithm and its various subsystems. Chapter VI analyzes the stability of the developed algorithm. Chapter VII presents the results of the extensive simulation effort to verify the stability of the algorithm and characterize its performance across a wide range of scenarios to include multi-vehicle simulations. Finally, Chapter VIII summarizes the conclusions of this research and makes recommendations for the continuation of the research.

II. Literature Review

2.1 Introduction

The coordinated arrival time problem is a specialized type of path planning problem. The purpose of the general path planning problem is to generate a trajectory or guidance commands which safely transfer a vehicle from an initial state (position, speed, attitude, etc.) to a final state while minimizing some prescribed cost functional and adhering to any imposed constraints. As an example, a commonly used cost function is the length of the path or time to travel the path.

This literature review begins with a brief overview of path planning focusing on the distinction between local and global path planning. Next, an overview of two types of local path planning is given along with examples from recent research: “Plan-and-Follow” methods and real-time trajectory guidance. Finally, an overview of string stability theory is presented to provide insight on a method to analytically determine the stability of a group of time coordinated vehicles.

2.2 Path Planning

Path planning scenarios may be grouped into two areas: global and local path planning. Global path planning, also referred to as route planning, desires to find a route through a set of points of interest. These routes are typically modeled using straight lines between waypoints and optimized using a wide variety of optimization methods [12]. Subsequently, a local path planner refines the routes to generate flyable trajectories. Local path planners generate a flyable trajectory as opposed to a route through waypoints.

Global path planners are a powerful approach to path planning especially when the problem contains multiple fixed obstacles such as threats, keep-out zones, or

buildings. However, the plan and refine approach of global path planners is, at best, a near real-time method due to the required route optimization. Since the goal of this research is to develop a real-time guidance algorithm, only local path planning methods are reviewed.

2.3 Plan-and-Follow

The plan-and-follow method is straightforward in its methodology: plan a flyable path based on some assumed environment and follow it in real time. This method allows for complex, iterative optimization techniques to be used in calculating the path. Subsequently, a real-time path follower allows for the vehicle to account for minor changes in the environment while still following the planned path. However, the resulting paths are optimal only for the underlying environmental assumptions. Major changes to these assumptions may require the path to be re-planned or else the path being followed may no longer be optimal. This method is broken down into three distinct components: planning, following, and arrival time control methods.

Path Planning Methods.

This section reviews four methods used to produce optimal paths: Dubins paths, Pythagorean Hodographs, differential flatness, and optimal control path planning.

Dubins Paths.

One of the most basic path planning problems is finding the optimal path between two points. With no constraints the solution to this problem is a straight line. A version of this problem which is more applicable to aircraft is to find the shortest path between two points with bounded curvature. The solutions to this problem are referred to as the Dubins paths; named for L.E. Dubins who first proved the solution

to the problem [13].

The problem is to find the shortest path between two poses (position and heading) subject to a constraint on the curvature of the path. Curvature describes how fast a path turns. A path with zero curvature is a straight line while the curvature along a circular path is constant but not-zero; note that curvature is defined as the inverse of the circle's radius. The inclusion of curvature as a problem constraint forces the solution to curve as opposed to instantaneously changing direction.

In the late 1950s, Dubins provided a mathematical proof of the solution to the shortest path between two points with bounded curvature [13]. Dubins proved that the solution to the aforementioned problem is a set of no more than three line-segments. The first and third segment are constant curvature turns. The second segment may either be a straight line, or another constant curvature turn.

The six possible types of Dubins paths are presented in Fig. 1 where the paths are labeled according to the turn order; left (L), straight (S), or right (R). As an example, a LSR path contains segments in the order of Left, Straight, and then Right. A benefit of the calculation of these paths is that the length of the path is easily obtained since the paths are composed of circular arcs and straight line-segments. The solution to the minimum time problem is then to calculate each possible type of path, at most six, and select the shortest.

Dubins paths have been further studied by Shkel where the set of possible Dubins paths was further constrained based on the problem setup [14]. The benefit of this approach is that each type of path does not need to be calculated. Rather, a small set of parameters are calculated based on the initial and final poses of the problem. Shkel characterized the values of each parameter which relate to the different Dubins path types.

Subsequently, the term Dubins vehicle was coined to describe the dynamics of a

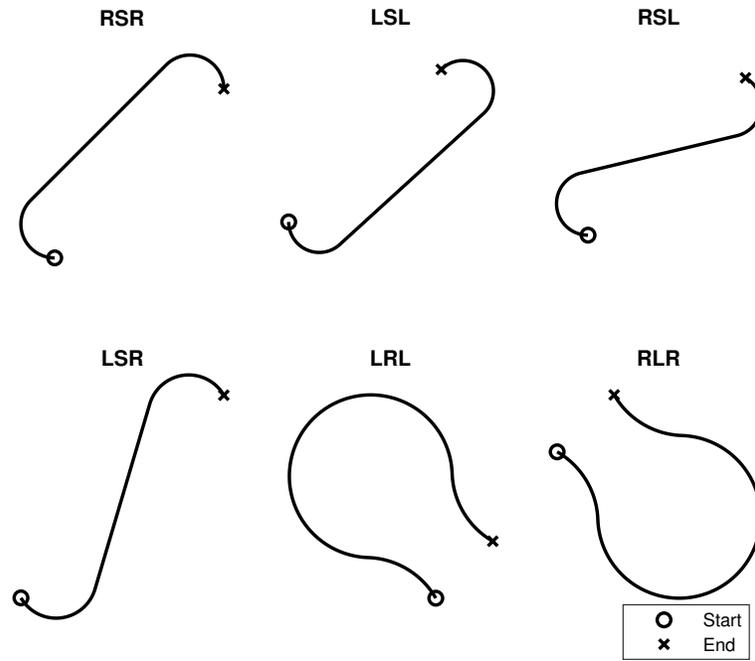


Figure 1. Dubins Paths

vehicle whose trajectories are Dubins paths. This type of vehicle travels at constant speed and may instantaneously change the curvature of its current trajectory [12]. Dubins vehicle dynamics have been widely used in aeronautical research because Dubins paths model aircraft dynamics well and are computationally simple.

Multiple geometry-based methods to calculate the Dubins path are presented by Tsourdos in Ref. [12]. The methods are not iterative and are computationally inexpensive. Thus, a typical formulation to solve the minimum time path planning problem calculates every possible type of Dubins path, then outputs the shortest path as the solution.

In Ref. [15] Dubins paths were used to find the shortest path from an arbitrary initial position to a circle of known size. The problem is challenging because the final pose of the path is not specific. Rather, a set of final poses is acceptable with regard to the problem being solved. The solution was obtained analytically by utilizing the

initial pose compared to the position of the circle.

Dubins paths have also been used in the field of artificial intelligence. In Ref. [16], Dubins vehicle dynamics are used to train a neural network to solve a Tail-Chase problem where a pursuer unmanned aircraft is attempting to reach a “reward region” around a moving target while avoiding a “penalty region” which is also near the target. The output of the neural network is one of two general guidance schemes; either navigate to the target or avoid the “penalty region.”

In Ref. [17], the problem considered was to find the optimal paths for three vehicles through a set of waypoints, a type of traveling salesman problem. The objective of the optimization problem was to minimize the total time to visit all waypoints subject to constraints on the arrival time and arrival heading at a subset of the waypoints. The solution utilized a genetic algorithm which generated an initial set of solutions and then used multiple heuristic methods to improve the “best” solution with regard to the original problem. The results demonstrated the computational efficiency of Dubins paths where the solution algorithm calculated over 11,000 Dubins paths per second on a typical desktop computer.

The effects of the environment have also been considered with regard to the minimum time Dubins vehicle problem. Namely, what is the minimum time path to travel between two poses in a constant wind field. This problem increases the realism of the Dubins problem solution with regard to aircraft. One solution to this problem calculated a Dubins path through the moving air mass then compared the result to the desired final pose [18, 19]. This approach required multiple iterations which was possible since basic Dubins paths are computationally inexpensive.

A second approach attempted to find an analytic solution to the constant wind Dubins problem. Techy was able to obtain analytic solutions for Left-Straight-Left and Right-Straight-Right types of paths [20]. However, the other types of Dubins

paths required numerical solutions. Additionally, Ayhan was able to compute turn-turn type Dubins paths through a constant wind field which were of a specified length [21].

Time synchronization has also been well-studied utilizing Dubins paths. Reference [22] used 3-D Dubins paths to plan the simultaneous arrival of multiple unmanned aircraft. The work further constrains the problem by ensuring the paths are “safe to fly,” defined as the curvature of the paths are bounded by a prescribed maximum curvature, a minimum distance between unmanned aircraft at all times is enforced, and the paths do not intersect at points of equal length.

Dubins paths have been well-researched and are useful in the area of aircraft path planning. However, the primary limitation of Dubins paths is that they contain instantaneous changes in heading rate. Instantaneous changes in heading rate would correspond to instantaneous changes in bank angle for an aircraft which is clearly not possible. This difference typically creates a small error when an aircraft attempts to follow a Dubins path which is investigated in Chapter IV.

Pythagorean Hodographs.

Pythagorean hodographs (PH) are a specific type of path parameterization. Path parameterization optimizes the path by assuming some class of basis functions which are assembled in a linear combination. If twice-differentiable continuous functions are used as the basis functions, the resulting path and heading will also be defined by a continuous function. Thus, the Dubins path limitation of discontinuous heading rate may be overcome at the cost of optimality and computational efficiency which is discussed.

The type of parameterization reviewed here is the PH Bézier curve. Bézier curves are polynomial curves created from linear combinations of Bernstein basis polyno-

mials. The basis polynomials are defined to be functions of a variable, q , which is defined to be zero at the beginning of the path and one at the end of the path.

The PH was first introduced in 1990 by Farouki and Sakkalis [23]. A PH exists for a parametric curve if the sum of squares of its parametric derivatives is equal to a square number [12]. This special formulation is particularly useful since the length of a parametric curve may be obtained by integrating the magnitude of the velocity vector with respect q .

The magnitude of the velocity vector is often difficult to work with mathematically since it involves taking a square root. However, the radicand of a PH curve is, by definition, a square and the square root disappears. Thus, the length of a PH curve is simply the integral of a rational polynomial.

The next step is determining the polynomials for the parametric curve which satisfy the conditions of a PH. The typical choice of curve is the Bézier curve which is a parametric curve whose path is defined by the Bernstein basis polynomials. Farouki provides an extensive review of the Bernstein polynomial basis in Ref. [24]. However, a subset of numerical properties [24] of Bézier curves and Bernstein polynomials which are useful to path planning are listed here:

- Bernstein polynomials curves contain $N + 1$ control points where N is the degree of the Bernstein basis polynomial.
- The integral of a Bernstein polynomial is the sum of the control points divided by the degree plus one.
- The derivative and arithmetic operations of Bernstein polynomials may be calculated only using the control points and the result is also a Bernstein polynomial.
- An algorithm called de Casteljau's algorithm separates a Bézier curve into two

Bézier curves of equal degree using only the control points of the original curve.

- A Bézier curve lies within the convex hull of its control points.
- The second and second-to-last control points, along with the first and last control points, define the derivative or heading of the Bézier curve at the terminal points.

The primary strength of these attributes lies in the ability to derive a continuous path of known length and manipulate the path using only the control points. This results in fewer degrees of freedom in the optimization problem as opposed to discretizing the entire path to determine the solution. However, the resultant paths are optimal only in the sense that they are the optimal Bézier curve as opposed to the globally optimal path. Note that a Dubins path may not be represented as a Bézier curve due to the discontinuous curvature of a Dubins path.

The flyability of the path may be analyzed by utilizing the curvature of the path which is also a Bézier curve if the path is a PH [12]. Using the convex hull property of Bézier curves, the maximum possible curvature is calculated using the control points of the curvature function. However, this may be excessively conservative as the control points may not be near the actual maximum value of the curve. This may be remedied with the application of de Casteljau's algorithm.

De Casteljau's algorithm is used to split a single curve into multiple curves of the same degree. The benefit is that the control points approach the curve as the curve is split into smaller and smaller sub-curves. Thus, a set number of iterations of de Casteljau's algorithm will allow flyability to be enforced while not being overly conservative.

An excellent example of implementing PH paths was presented by Choe [25]. Choe's research describes a general framework for generating time coordinated paths

using PH Bézier curves. The research also allowed for the inclusion of path separation constraints which impose a minimum separation distance between paths. The research also enables greater control over timing by calculating a speed profile in addition to the path trajectory. The calculated speed profile is also a Bézier curve.

Although the method requires an optimization problem to be solved, the strength of the method is the reduced size of the optimization problem due to the use of quintic PH Bézier curves. Applying the PH condition causes the two-dimensional Bézier curve path to be defined by two second order Bézier curves for a total of 6 path variables per vehicle. Additionally, a timing law is used to control the arrival time of the vehicles. The integral of the inverse of the timing law is the arrival time of the vehicle. The timing law is assumed to be a second-order Bézier curve which only has three control points.

The path and speed profile may be combined to calculate both the curvature of the path and lateral acceleration of the vehicle. Since the curve is a PH curve, the curvature is a rational Bézier curve and the acceleration is also a Bézier curve. De Casteljau's algorithm may be applied to both of these types of curves which quickly determines the global maximum or minimum value of the curve. Thus only 4 constraints are imposed for the aircraft dynamics: minimum velocity, maximum velocity, maximum curvature, and maximum acceleration.

The constraints on the terminal position and heading are applied by simply calculating these values from the design variables describing the path. The distance between paths is also a Bézier curve and de Casteljau's algorithm is applied to quickly find the minimum separation to enforce the spatial path deconfliction constraint.

Finally, the arrival time is calculated using the integral of the inverse timing law, as previously mentioned. The timing law is a quadratic Bézier curve in this research. The arrival time constraint may be imposed as a specific arrival time, simultaneous

arrival time for all aircraft, or a set of arrival times which are equally spaced.

To summarize the method in Ref. [25], the coordinated arrival time problem is a constrained optimization problem using PH Bézier curves. The resulting paths are sub-optimal when attempting to find a minimum-time trajectory because of their assumed shape. However, the problem being solved focuses on meeting the arrival time constraints. The primary strength of this formulation is the reduction of the problem size, only nine variables, to describe the continuous path and velocity of each vehicle. Each vehicle also has four dynamics inequality constraints and 6 terminal equality constraints. Additional timing constraints are applied depending on the desired arrival specifications.

The paper also considered and imposed path deconfliction constraints in the case of multiple vehicles. The method still requires a constrained optimization to be solved which may not allow for a real-time implementation. However, the calculation of both path and velocity profiles allows greater flexibility in arrival time than if velocity were held constant. The method used in Ref. [25] was later extended to three-dimensions in Ref. [26]. An extensive review of PH curves, Bézier curves, and their use in cooperative trajectory generation may be found in Ref. [27].

The use of Pythagorean hodographs shows the power of parameterization and the ability to calculate a path utilizing a small subset of control points. However, as previously mentioned, this method forces the path to a specific type of shape which may not be globally optimal. Additionally, the path still requires optimization which may not be able to occur in real time.

Differential Flatness.

The next “plan-and-follow” method to be discussed is differential flatness. The method is similar to the previous path parameterization methods and is primarily

used for trajectory generation. However, not every dynamic system is differentially flat which limits the application of this method. The concept of differential flatness was introduced by Fliess in the early 1990s and a summary of the theory may be found in Ref. [28].

First a basic definition of differential flatness is presented. Assume there is a system where the derivatives of the state variables are functions of the states and the inputs. The system is differentially flat if there exist flat outputs, equal to the number of inputs, which are functions of the state and control derivatives such that the states and control can be defined as functions of the flat outputs and their derivatives.

The definition of differential flatness is best shown in equation form. Nieuwstadt gives an excellent definition from Ref. [29]:

A nonlinear system

$$\begin{aligned} \dot{x} &= f(x, u), & x &\in \mathbb{R}^n, u \in \mathbb{R}^m \\ y &= h(x), & y &\in \mathbb{R}^m \end{aligned} \quad (1)$$

is differentially flat if we can find flat outputs $z \in \mathbb{R}^m$ of the form

$$z = \zeta(x, u, \dot{u}, \dots, u^{(l)}) \quad (2)$$

such that

$$\begin{aligned} x &= x(z, \dot{z}, \dots, z^{(l)}) =: x(\bar{z}) \\ u &= u(z, \dot{z}, \dots, z^{(l)}) =: u(\bar{z}). \end{aligned} \quad (3)$$

The benefit of this formulation in solving a dynamic system, linear or nonlinear, is that the problem may be solved in the space of the flat outputs which may be smaller than the space of the states. In the context of trajectory generation, the flat outputs and their derivatives are often parameterized by B-splines which are defined by their control points [29]; similar to PH Bézier curves.

Nieuwstadt provides some useful background and examples with regards to tra-

jectory generation in Ref. [30, 29]. He also provides a framework for “real-time” trajectory generation but notes that the definition of “real-time” is ill-defined. The definition used in the research of a “real-time computation” is a computation “that can be performed faster than the reference update.”

Nieuwstadt creates two algorithms which attempt to achieve this real-time trajectory generation. The first is effectively a trajectory tracker which controls the system to achieve a desired state in a receding horizon fashion. A time delay is chosen which allows an optimization routine to solve for the flat outputs which connect the initial state of the current iteration to the desired state at the next iteration. A point along the flat path is chosen and used as the command to the system.

The research notes that there is a trade-off determined by the choice of the time delay. A shorter time delay results in better performance but higher control magnitudes which could decrease stability. The opposite is true for higher time delays. Additionally, the research proves the algorithm achieves “real-time asymptotic trajectory generation” with the correct choice of points along the flat path used to derive the input command. A proof is provided in Ref. [29] which shows the steady state output error is zero.

The second algorithm attempts to incorporate some cost function into the problem formulation rather than simply following a reference trajectory. The method is similar to the first algorithm except that the feasible trajectory from the first algorithm is optimized based on the cost function. The algorithm preemptively stops the optimization in order to honor the delay time. Thus, at worst case, the second algorithm outputs the result from the first algorithm. Similarly, the second algorithm can be shown to eventually converge the error to zero. However, the research shows the second algorithm is too slow by a factor of ten to support realistic time delays.

In summary, differential flatness is a path parameterization method which may be

formulated to generate trajectories in real time; depending on the definition of real time. The method has the potential to reduce the size of an optimization problem via the correct selection of the flat variables. However, the method is only applicable to systems which may be written in a very specific form.

Optimal Control Path Planning.

Optimal control theory, in general, seeks to find the optimal control input to a set of dynamics which minimizes a desired cost function [31]. The computed trajectories may be truly optimal if a basis function is not assumed for any variables. Thereby overcoming a limitation of some path parameterization methods. A primary use of this theory is the development of an optimal open-loop guidance law which will produce a control which minimizes some prescribed cost functional over time for any initial state. This section focuses on the calculation of paths using optimal control. The development of real-time guidance laws using optimal control is addressed in Section 2.3 and methods to solve optimal control problems will be discussed in Sec. 2.5.

An optimal control problem, if successfully solved, outputs both the optimal control and resulting optimal states. If the problem were successfully solved, the optimal states adhere to the imposed dynamic constraints of the problem. In applying this method, an optimal path may be derived which inherently satisfies the dynamic constraints of the aircraft as opposed to applying geometric constraints to a path optimization as in Sec. 2.3. First, the optimal control formulation of the shortest path between two points with bounded curvature is reviewed. Recall the proven solution to this problem are Dubins paths which were discussed in Sec. 2.3.

Boissonnat formulated the shortest path between two points with bounded curvature problem as an optimal control problem in Ref. [32]. In this formulation, a vehicle travels at a constant velocity and is controlled by a heading rate of change

input. Utilizing optimal control theory Boissonnat solved the problem, and the resultant optimal trajectories were Dubins paths.

A benefit of the optimal control formulation, with regards to cooperative arrival time problem, is that time is a required component of the problem formulation. In Boissonnat's formulation, the end time of the trajectory is a free variable, and the objective is to find the minimum time path. Optimal control problems are also formulated as a fixed final time problem allowing for the final time to be specified or in relation to another vehicle.

A limitation of the optimal control method is that an analytical solution to the posed problems may only be found in a small subset of circumstances. Typically, these are problems with two or fewer states and no constraints besides the dynamics of the system. Thus, a numerical method must often be used to determine the solution to an optimal control problem.

Optimal Control Path Planning Strategies.

This section presents two examples of optimal control theory being used to generate an optimal trajectory. The first is detailed in Ref. [33]. The research paper addresses the computation of the optimal landing paths for large passenger aircraft; specifically, a Boeing 737-500 and Boeing 767-400.

The continuous descent arrival (CDA) procedure is used as the basis for the problem. This aeronautical procedure was intended as a means to reduce flight time, reduce fuel used, and reduce noise perceived on the ground. The procedure achieves these benefits by minimizing engine thrust for as much of the profile as safely possible. However, differences in these idle thrust trajectories between aircraft have limited the use of the procedures due to a required increase in the safety separation distances between aircraft. This increase in separation decreases rate which aircraft may fly into

an airport.

Park proposed a solution to this problem which consisted of numerically calculating an optimal solution to the problem then proposing a simple, sub-optimal solution which may be readily calculated without the use of optimal control software [33]. This problem addressed only the vertical maneuvering plane and the states used consisted of distance from the instrument landing system (ILS), height, and true airspeed. The control inputs were engine control and flight path angle.

A challenge of this problem was the required use of multiple flap settings by large commercial aircraft. Aside from changing the overall aerodynamics of the aircraft, flap position also has an associated minimum and maximum safe airspeed which must be honored. The requirements of this problem drove the use of a multi-phase optimal control approach where each flap setting was a different phase.

The optimal flight paths were computed using GPOPS-II [34] and analyzed to determine a simple analogue to the optimal flight path. The optimal solution typically resulted in a variable air speed which may be difficult to follow precisely. Therefore, the research proposed a sub-optimal solution which utilized a constant airspeed for each segment. The adjusted airspeeds were calculated as a single variable optimization problem to minimize fuel burn during that portion of the flight. Subsequently, it was shown that the adjusted airspeeds were a quadratic function of the wind speed allowing for a simple calculation of sub-optimal airspeed by aircraft and air traffic controllers.

The optimization method in Ref. [33] highlights the ability to develop a near-optimal but computationally-efficient trajectory from a numerically-computed optimal trajectory. As a comparison, the fuel optimal trajectory required 53 seconds to compute using GPOPS-II while the sub-optimal trajectory required only two seconds to determine the optimal descent speed. Further details regarding this formulation

or the CDA problem may be found in Ref. [35].

The second example of optimal control path planning is detailed in Ref. [36]. This research computed terrain-following and terrain-avoidance trajectories using optimal control theory. The problem was to obtain a flyable trajectory which navigates between terminal conditions across a mountain range while maintaining a minimum height above the terrain. In this research the terrain elevation is obtained via a table lookup of tabulated data based on the vehicle position. The problem formulation uses a simplified set of aircraft dynamics where load factor rate and roll acceleration are the only control inputs into the problem.

The problem was formulated as both a minimum time (terrain-avoidance) and minimum height above terrain (terrain-following) problem. The minimum time problem allows the final time to be free while the minimum height above terrain problem fixes the final time at some specified value. This feature of optimal control problems is valuable to coordinated arrival problems as the arrival time may simply be specified as an inherent constraint of the problem.

However, the terrain-following problem required over an hour of computation time on a desktop computer to solve a trajectory which was approximately 60 nautical miles long and had a flight time of 530 seconds. The terrain-following problem was formulated to activate the terrain path constraint as much as possible which significantly affected the computational efficiency of the algorithm.

In summary, optimal control path planning is a powerful trajectory generation method which inherently creates trajectories which are flyable. However, numerical optimization methods are typically required to solve realistic problems and it is infeasible to generate optimal trajectories in real time. Although, the optimal trajectories may be computationally inefficient, Ref. [33] showed the ability to formulate a simple near-optimal solution which was able to be determined in real time.

Path Following Guidance Laws.

Once a path has been planned, the next step is to fly the path. In general, the goal of a path following guidance law is to drive the error between the vehicle's position and the path to zero. Path following is an entire field unto itself, and a short summary of some useful guidance techniques is provided. The review discusses linear guidance and non-linear dynamic inversion with respect to path following based on the overview given in Ref. [12]. Additionally, a brief overview of optimal control guidance is presented.

Simple Guidance Law Algorithms.

Linear guidance is a simple form of path-following guidance since the control is linear with respect to the error. The method initially specifies the feedback algorithm to be used as a linear combination of variables with an associated gain. The gains are chosen through an iterative process to achieve some desired path-following performance.

An example is given for following a straight line where a vehicle is traveling at constant velocity and has control over the rate of change of its velocity vector, a Dubins vehicle. A guidance law for this example would output the rate of change of the velocity vector as a function of the distance from the line to be followed. Linear guidance methods are simple in their implementation but typically exhibit error when tracking curved trajectories.

Non-linear dynamic inversion (NDI) is a type of model reference control. This method specifies the desired response of the error signal, as a second order differential equation where the damping ratio and natural frequency are specified for example. The dynamics of the system are then derived and substituted into the desired dynamics to derive the guidance law as a function of the specified dynamics. The derivation

results in the controller having knowledge of the turn radius of the reference path in addition to the difference between the vehicle heading and the path heading at the closest point. This extra information allows the NDI algorithm to perform better than the simple linear guidance. The NDI method results in a more complicated guidance law but often follows both straight and curved paths better than simple linear guidance.

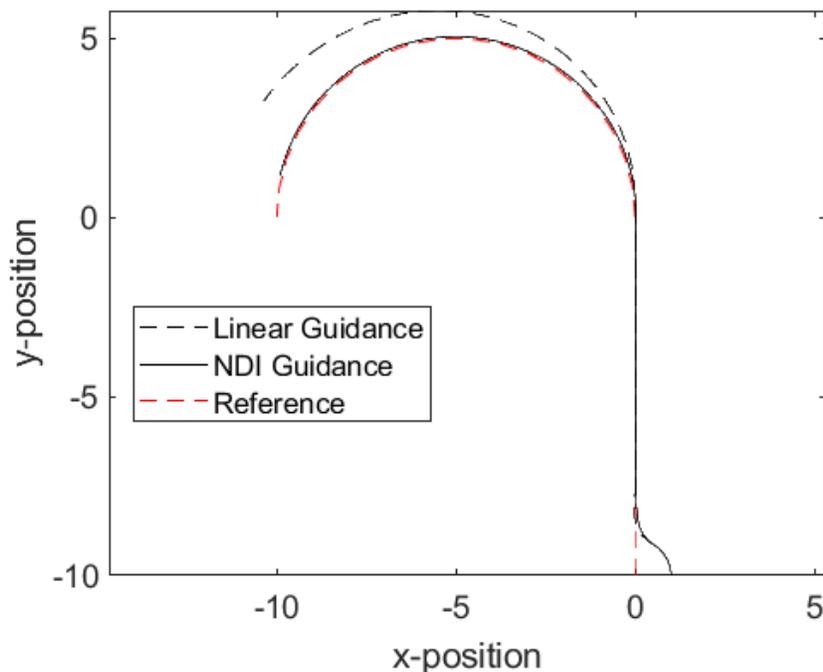


Figure 2. Linear and Nonlinear Dynamic Inversion Guidance Comparison

A comparison of the linear and NDI guidance presented in Ref. [12] is given in Fig. 2. The path-following trajectories are attempting to follow the same reference path. Dubins vehicle dynamics are assumed with constant velocity and the same saturation limit on heading rate of change is applied to both algorithms. Both guidance laws acquire and follow the straight path segment well, but the NDI guidance law clearly tracks the curved section of the reference path better than the linear guidance law.

Advanced Guidance Law Algorithms.

A more advanced path guidance method is to use optimal control theory to develop a path-following algorithm. The optimization problem to be solved, typically, is to find the control which minimizes the distance from a prescribed path. An analytic solution for the optimal control must be obtained in order to use the method to create a path-following algorithm. The control is often obtained using the necessary conditions for optimal control and Pontryagin's Minimum Principle in the case of bounded control [31].

A special class of controller exists if the problem may be formulated such that the cost functional is quadratic and the dynamics are a linear combination of the states and the control. If a solution may be found, it is termed the linear quadratic regulator (LQR) controller. The derivation of the LQR controller and requirements for the existence of a solution may be found in Ref. [31]. The benefit of this formulation is that the controller is guaranteed to be stable if a solution may be found. However, the controller requires full knowledge of all the state variables. If the state information is not available then a linear quadratic estimator, also referred to as a Kalman filter, may be used to estimate the required states from the available states.

Another application of optimal control theory is the generalized method of developing a three-dimensional trajectory tracker found in Ref. [37]. The \mathcal{H}_∞ method of control synthesis, an optimal control method, is demonstrated in Ref. [37]. The method generates a control law by finding the control which minimizes the infinity norm of the closed loop system. The research then demonstrates the controller performance tracking lines and helices with and without a constant wind field.

The formulation of the algorithm allows the path being followed to be defined by waypoints or as a continuous function. The primary advantage of this method is that the only inputs to the trajectory tracker are the current position of the point along

the trajectory being followed and the heading rate of change of the trajectory at that point.

Another guidance law was developed as a combination of a NDI and adaptive guidance law. This guidance law addresses the challenge of following a Dubins path to include the effects of wind [38]. In this work, the path error is modeled as a second-order damped oscillation. Choosing the natural frequency and damping ratio of the error oscillation determines the response of the controller. A limitation of this controller was observed that a set of parameters, natural frequency and damping ratio, would only achieve satisfactory performance with either a large or small initial error but not both. An adaptive controller was developed to overcome this limitation where the natural frequency was adjusted to achieve the desired performance of a controller which worked well both near and far from the path. Simulations demonstrated reasonable performance with and without wind.

The challenge of following Dubins paths is increased with the assumption of non-zero wind. The challenge in following a Dubins path in wind is that the earth-fixed turn radius of the aircraft is affected by both the magnitude and direction of the wind. Wolek [39] addresses this problem by computing the minimum turn radius for a given upper bound on the wind speed. The result is a path flyable by a vehicle with the specified heading rate constraint which is longer than the original Dubins path. The worst case turn radius would only be observed if the wind vector is in the same direction as the vehicle's velocity vector at some point along the path. Thus, the method is conservative but provides a feasible solution dependent on the wind magnitude.

There are many methods to path following which vary in their performance depending on the environment and path being followed. In the "plan-and-follow" the path follower is part of an upper-level open loop control in that the path follower is

simply following the pre-planned path. In order to affect the arrival time, either the path must be recalculated, or the airspeed of the aircraft changed.

Arrival Time Control.

Arrival time control, in the “plan-and-follow” method, is primarily achieved in the “plan” portion of the method. The previous sections have discussed several methods where a path of a desired time or length is calculated, and a specified arrival time is achieved by successfully flying the path. The methods have assumed both constant and variable velocity. In either case the ability to achieve the desired arrival time depends on the ability aircraft to follow the path and velocity profile.

An example of a “plan-and-follow” formulation utilizing arrival time control may be found in Ref. [40]. The path may be discretized from any of the previously defined methods or may be generated from a waypoint path planner such as those described in Ref. [41, 42]. Arrival time control is achieved via control of the aircraft velocity.

In Ref. [40], the problem is to achieve either simultaneous arrival or spaced arrival. With a defined path, Nelson implements a cooperative timing function which maps the range of arrival times for a vehicle to an “energy” cost. The energy cost is effectively total fuel used; however, the paper is focused on battery powered vehicles. The cost is calculated assuming constant velocity and straight-line distance between waypoints. Minimum energy times are selected which achieve the desired timing.

Next, the waypoint-defined path is followed using a vector field path-following algorithm described in Ref. [43]. The arrival time is estimated using straight line distances between waypoints and constant current velocity. A simple velocity controller is then used to adjust the current velocity and minimize the error between the desired arrival time and estimated arrival time.

Nelson demonstrated this method with multiple flight tests in windy conditions

where the wind was between 30 and 60 percent of the vehicle airspeed. The simultaneous arrival experiments achieved an average arrival time between the first and last vehicle of 1.6 seconds. The spaced arrival experiments achieved an average error of 0.6 seconds from the desired arrival time. The method provides a notable example of achieving real-time arrival time control using only velocity control once a time-feasible path had been selected. However, wind is not considered in the path planning algorithm in this research. In the event the velocity control is saturated and the desired arrival time is no longer achievable, a new feasible waypoint-defined-path is calculated.

Plan-and-Follow Summary.

This section reviewed various methods to “plan-and-follow” a path. A number of these methods showed examples where the vehicle arrival time could be planned and subsequently achieved by a real-time path follower. The wide variety of methods allows many types of problems to be addressed; general trajectory, minimum-fuel descents, terrain-following trajectories, and optimal paths in the presence of wind were given as examples. The strength of this method is that an optimal trajectory may be found, and that the trajectory is pre-defined. This allows for the inclusion of keep-out zones in the trajectory generation and identifies the path to be flown; a key to predictable path-following behavior.

The primary limitation, regarding cooperative timing, is that the path being followed may not be adjusted in real time. This limits the ability to achieve a desired arrival time when the scenario conditions (endpoint, wind, etc.) are allowed to change.

2.4 Real-Time Trajectory Guidance

The next set of research focuses on real-time control of an aircraft's trajectory to affect the arrival time of the aircraft. These methods are similar to the path-following methods discussed in Sec. 2.3. However, the error being minimized is no longer the error between vehicle position and desired path. Rather it is the error between the desired arrival time and the estimated arrival time. These methods have a potential advantage over the "plan-and-follow" approach because they may adjust both the path length and velocity allowing a wider range of arrival times to be achieved in real time.

Estimating the arrival is as simple as dividing the remaining distance to the final position by the velocity of the vehicle. However, the calculation of the distance remaining is not trivial and additional constraints may quickly complicate this estimate. This section will review proportional navigation, virtual target, and optimal control guidance focusing on arrival time performance and path predictability achieved by each method.

Proportional Navigation Guidance Law.

Proportional navigation is similar to the linear guidance method discussed in Sec. 2.3. A simple, constant gain is chosen which is applied to some combination of state variables to produce the control. The first problem to be addressed is to find the guidance law which maneuvers a vehicle to a terminal point, at a prescribed terminal heading, at a specified arrival time.

Saleem develops a proportional navigation solution to this problem in Ref. [44]. This method assumes Dubins vehicle dynamics and control over lateral acceleration which is analogous to turn rate. The feedback gain is calculated based on the initial geometry of a specific scenario. The method uses several Taylor series expansions to

develop a closed-form estimate of the final arrival time by integrating the closed-loop dynamics forward in time. The estimated arrival time is also the minimum arrival time for a given state so long as the guidance law is followed to the end of the scenario. This method is shown to accurately estimate the final arrival time when following the prescribed guidance law.

With an accurate estimate of arrival time, any desired arrival time greater than the minimum time may be achieved. Saleem accomplishes this by commanding a heading rate in the opposite direction of the nominal guidance law command. This command increases the estimated arrival time and is performed until the estimated arrival time meets the desired arrival time where the original guidance law is followed.

To summarize Saleem's work in Ref. [44], a time estimate is obtained by assuming a guidance law and making several approximations which allow the closed-loop dynamics to be integrated forward in time. The method then commands the vehicle away from the target until the estimated arrival time matches the desired arrival time and the original guidance law is followed.

Proportional navigation is also used to avoid obstacles in Ref. [12]. The method uses geometry to determine if the vehicle path will overfly some keep out area. If the keep out area will be violated, then a simple proportional navigation controller takes over from the path follower and avoids the obstacle. At some prescribed point the path follower retakes control. This method requires a means to determine if the current path will intersect a keep-out zone which is complicated in the case of real-time trajectory guidance where the path is constantly changing.

Virtual Target.

Next, a method which adds a constraint on arrival velocity to the previous problem is reviewed. Bélanger solves this problem using a virtual target to guide the actual

vehicle towards the target in Ref. [45]. The virtual vehicle moves along a pre-defined path at the desired final velocity which is set up to coincide with the desired final position at the desired final time and heading. The solution is achieved by minimizing the error between the actual vehicle and the virtual vehicle. When the position error between the vehicle and virtual target is consistently zero, the vehicle will arrive at the actual target with the prescribed heading, velocity, and arrival time.

This method is similar to the path planning method for arrival time control discussed in Sec. 2.3. The method uses a simple predefined path for the virtual vehicle which eliminates “planning” phase of the algorithm. The method is successful if a sufficient amount of time is given for the vehicle to catch the virtual vehicle.

The algorithm was demonstrated in a hardware-in-the-loop simulation utilizing an autopilot and six-degree-of-freedom nonlinear unmanned aircraft model with a static endpoint. The performance of the algorithm was shown to degrade in the case of moving endpoints. Cooperative moving targets were simply treated as a static target since the target position at the arrival time was known.

Optimal Control Guidance.

The general formulation of optimal control guidance law development was discussed in Sec. 2.3 and methods to solve optimal control problems will be discussed in Sec. 2.5. When utilizing optimal control to follow a predefined path, the cost function being minimized is typically the distance from the path. This section addresses a formulation which minimizes the difference of the final state (position, time, and velocity) from a desired final state.

An unbounded missile guidance law was developed which achieves a desired final time, position, and velocity [46]. In this work the equations of motion are a simple double integrator which include the effect of gravity. The derivative of vectorized

velocity is equal to the control plus gravity. Thus, the control assumes direct control over the vehicle acceleration along each inertial axis.

The problem is then formulated as a fixed final time optimal control problem with initial and final position and velocity vectors specified. The cost function takes on an interesting form in that it seeks minimize the integral of the control divided by the time remaining. The cost function effectively rewards control when the time remaining is high and penalizes control when the time remaining is low. This behavior is desirable as it tends to “stabilize” the control near the target. The unbounded optimal control problem is solved using typical optimal control methods. The resultant control is then a function of the time remaining which needs to be estimated.

The arrival time is predicted by numerically integrating the closed-loop dynamics. The dynamics are integrated until the resulting integrated range rate to the target indicates the target has been passed. The operations involve simple arithmetic and are computationally inexpensive. Multiple additional guidance modes were developed to either increase, decrease, or maintain the predicted arrival time similar to Ref. [44].

The aforementioned guidance law, developed by Harrison, performed well in simulations but focused on short-duration missile-engagement scenarios. The guidance law also performed well as a cooperative controller to achieve simultaneous and spaced arrival. The spaced arrival times were based upon the estimated arrival time of a preceding missile. A downside of this problem formulation is that there are no means to enforce keep-out zones to prevent the flight path from crossing a specified area. Additionally, the flight path may be predicted but only through numerical integration.

Real-Time Trajectory Guidance Summary.

This section addressed real-time trajectory guidance methods which were able to achieve a specified arrival time. All methods focused on dynamics which were

simplified from the full non-linear dynamics of an aircraft. The primary benefit of this method was a wide range of arrival times may be achieved. The methods were similar in that multiple modes were required to extend the path to achieve a desired arrival time which was longer than the minimum time.

However, the limitation of this method is that keep-out zones were not included and in fact no bounds on the path trajectory were enforced. This prevents a predictable path from being developed. Obstacle avoidance controllers were discussed in Sec. 2.3 but would likely require a mode-switching algorithm to include with a real-time trajectory guidance algorithm which achieves arrival time control.

2.5 Methods to Solve Optimal Control Problems

Optimal control problems are typically solved with one of two methods, direct or indirect. Direct methods numerically solve the optimal control problem by transcribing the equations of motion of the optimal control problem into a series of constraints at each time step. The transcription process effectively turns the optimal control problem into a static optimization problem where the design variables are the control and/or state at every time step. Indirect methods analytically solve optimal control problems using the calculus of variations and Pontryagin’s Minimum Principle (PMP) [31].

Numerical Methods to Solve Optimal Control Problems.

There are a variety of methods to numerically solve optimal control problems and an in-depth comparison may be found in Ref. [47]. Examples of some methods are Single Shooting, Multiple Shooting, and the Pseudospectral Method. The first two will not be addressed in detail as they are typically computationally inefficient [47].

The term shooting refers to the “shooting” of the dynamics based upon a guessed

control input. Thus, in the shooting methods, the control at the discretized points in time is the primary design variable. This can be a difficult optimization problem to solve because every instance of control may have an effect over the terminal state. It is often seen that the shooting methods provide an accurate solution but have trouble converging unless the initial guess provided to the solver is near to the optimal solution.

Another numerical method is the pseudospectral method. In the pseudospectral method, constraints are also enforced at the discretization points which are referred to as collocation points. However, the state in addition to the control are the design variables. Interpolating polynomials are used to approximate state and control between the collocation points. A property of these polynomials is that their derivative may be calculated by a simple matrix multiplication. Thus, the derivative of the states at every collocation point may be set equal to the dynamics equations (functions of the state and control) using equality constraints.

Finally, the states may be propagated forward in time and any cost function inside the integral may be calculated using Gaussian quadrature. Gaussian quadrature is a mathematically simple method to evaluate the definite integral of a polynomial where a linear combination of weights and the value of the polynomial at pre-defined collocation points approximates the value of the integral.

The accuracy of this method depends on the number and spacing of the collocation points. The collocation points are often spaced over intervals according to the roots of Legendre polynomials [34] but the roots of other polynomials may be used [48]. The roots of these polynomials are more densely spaced near the edges of the interval over which the polynomial is defined. This spacing increases the accuracy of the method by concentrating collocation points near the edges of an interval where integration error is often observed.

The spacing and number of these points, often termed the “mesh,” may be further adjusted through multiple iterations. The iterative method allows for a coarse initial guess to be solved quickly and the resulting solution used as the initial guess to further iterations with a finer, and more accurate, mesh. The pseudospectral formulation also results in sparse derivative matrices (both the Jacobian and Hessian) which may be used by advanced non-linear program (NLP) solvers such as IPOPT [49] and SNOPT [50] to rapidly solve the optimization problem.

Finally, complicated problems may also be formulated as multi-phase problems. In multi-phase formulations, each phase may be viewed as a small optimization problem. The different phases are tied to each with constraints on the time and states at the phase boundaries. A common use of phases may be to optimize a trajectory where the aircraft aerodynamics change, such as lowering the landing gear. The pseudospectral method has been used in the generalized optimal control problem solver GPOPS-II and a description of the specific pseudospectral method used by this software may be found in Ref. [34]. Several useful examples are given on the GPOPS-II website and in Ref. [47, 33, 36].

Indirect Methods to Solve Optimal Control Problems.

In the indirect method, several necessary conditions for optimality may be developed from the problem statement which aid in the solution of the problem. A brief overview of the indirect method will be given based on Ref. [31].

Take a general optimal control problem defined by

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t) dt \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t) \end{aligned} \tag{4}$$

where \mathbf{x} is the $n \times 1$ state vector, \mathbf{u} is the $m \times 1$ control vector, \mathbf{a} is an $n \times 1$ vector of

the equations of motion, and t_f is the final time. The cost functional is comprised of a terminal cost h and a running cost g . The terminal cost h is dependent on the final state and the final time. The running cost g is dependent on the state, control, and time. Equation (4) is used to create the Hamiltonian, \mathcal{H} , defined by Eq. (5) where the variable $\boldsymbol{\lambda}$ represents the costates of the problem. The costates do not typically relate to a physical aspect of the problem but are used to determine the solution.

$$\mathcal{H}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) \triangleq g(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^T(t)[\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)] \quad (5)$$

A key idea of the indirect method is Pontryagin’s Minimum Principle (PMP) [31]. The principle states that a control which minimizes the Hamiltonian will also minimize the cost functional. In minimum time problems, this principle often results in “bang-bang” control where the value of the control is always the upper or lower control limit set by the problem statement. Subsequently, applying PMP to an optimal control problem often results in the sign of the control being related to the costates of the problem.

The necessary conditions for optimality used in the indirect method are derived using the calculus of variations in Ref. [31]. The resulting necessary conditions are defined by Eq. (6) to Eq. (9). The conditions in Eq. (6) to Eq. (8) are applicable to any type of optimal control problem. The condition defined by Eq. (9), termed the transversality condition, must always be true for an optimal solution but the variables change depending on the type of the problem. The variable $\delta\mathbf{x}_f$, the variation in final state, is zero for a fixed final state problem and δt_f , the variation in final time, is zero

for a fixed final time problem.

$$\dot{\mathbf{x}}^*(t) = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) \quad (6)$$

$$\dot{\boldsymbol{\lambda}}^*(t) = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) \quad (7)$$

$$\mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \leq \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t) \text{ for all admissible } \mathbf{u}(t) \text{ (PMP)} \quad (8)$$

$$0 = \left[\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}^*(t_f), t_f) - \boldsymbol{\lambda}^*(t_f) \right]^T \delta \mathbf{x}_f \\ + \left[\mathcal{H}(\mathbf{x}^*(t_f), \mathbf{u}^*(t_f), \boldsymbol{\lambda}^*(t_f), t_f) + \frac{\partial h}{\partial t}(\mathbf{x}^*(t_f), t_f) \right] \delta t_f \quad (9)$$

Two other conditions may be derived if the Hamiltonian does not explicitly depend on time as typically seen in the case of minimum time problems; $\mathcal{H} = f(\mathbf{x}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t))$. If the problem has a fixed final-time and the Hamiltonian is not an explicit function of time, then

$$\mathcal{H} = f(\mathbf{x}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t)) = c_1 \quad (10)$$

where c_1 is some constant value. If the problem has a free final-time and the Hamiltonian is not an explicit function of time, then

$$\mathcal{H} = f(\mathbf{x}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t)) = 0 \quad (11)$$

The conditions defined by Eq. (6) thru Eq. (11) may be combined with the problem boundary conditions to find a potentially optimal solution.

2.6 String Stability

The previous sections focused on various methods to control the arrival time of a single vehicle. Logically, the methods may be extended to multiple vehicles by

deriving the desired arrival time of a follower vehicle from the states of the preceding vehicles. A simple example would be for a group of vehicles to arrive ten seconds apart based on the estimated arrival time of the preceding vehicle. This introduces a stability concern as the new system of systems must be robust to perturbations which may travel through the system of vehicles.

String stability refers to a property of an interconnected dynamic system where, if stable, the response of the entire system may be bounded regardless of the number of subsystems [51, 52]. The response is often analyzed based on the error of a system from some reference value since the desired origin of such a system is zero. To allow any number of subsystems, the determination of string stability typically revolves around ensuring that any perturbation of the string is attenuated as the error propagates to other subsystems. The concept of string stability is best demonstrated with an example.

String Stability Example.

In a cascaded system, information is transmitted sequentially and in a single direction (i.e., a vehicle only has information about itself and the immediately preceding vehicle). As an example, take a cascaded system of 10 cars where each vehicle desires to be 10 ft behind the preceding vehicle; thus, there is zero error when this condition is achieved. First, Fig. 3 shows the expected behavior of a string stable system. The maximum error of each subsequent vehicle is less than the preceding vehicle and stabilizes at zero, the desired position. At the end of the simulation, all vehicle errors are zero indicating that all vehicles are spaced 10 ft apart.

Figure 4a shows a case where the system is string unstable because the error parameter clearly grows unbounded. However, nonlinear features such as saturation functions can dramatically change the behavior of a string unstable system. In Fig. 4b

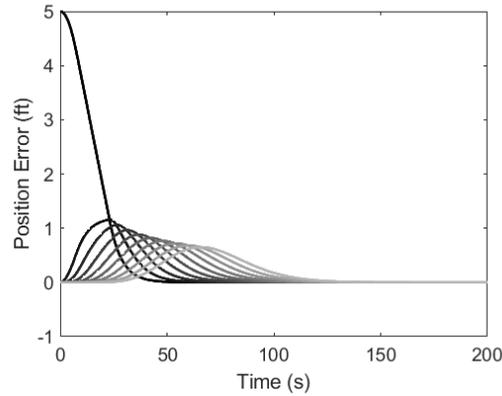


Figure 3. String Stable Example

multiple nested saturation functions were added to the system in Fig. 4a. Although the bounds never exceed the initial error, the peak error still grows with each vehicle down the string. The bound on the maximum error of the system is not independent of the number of vehicles, therefore the system is also string unstable.

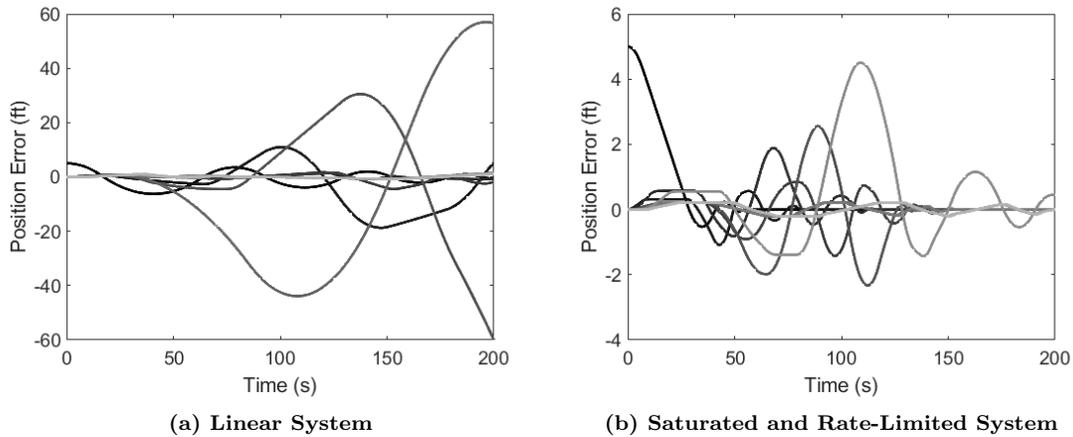


Figure 4. String Unstable Examples

This example presents string stability using position as the error variable. The concept of string stability is the same when applied to arrival time. The state variable used to calculate the error is now the estimated arrival time of the vehicle and the reference is based off the arrival time of the preceding vehicle. The difference is in the dynamics of the state variables.

String Stability Literature Review.

The theory of string stability was introduced to analyze the performance of interconnected dynamic systems. Chu took the approach to treat any string of a finite number of vehicles as a small segment of some infinite string. Chu then provided a definition of string stability based on the relative position of each vehicle to its neighbors [51].

Swaroop generalized the problem of determining string stability for a system in terms of Lyapunov stability allowing for linear and non-linear systems to be analyzed [52, 53]. Subsequently, Swaroop showed that groups of vehicles spaced with constant distance must have inter-vehicle communication to guarantee string stability. However, a string with constant time headway can be shown to be weakly string stable with only on-board information [54].

More recently, the idea of disturbance string stability was introduced in Ref. [55] which addresses the effects of external disturbances on the stability of the interconnected system. However, much of this research focused on strings of vehicles which were homogeneous with regards to their dynamic capabilities.

One of the first studies on the effects of heterogeneity in vehicle strings investigated wheeled vehicles which could generate the same amount of force but carried different masses while traveling up a hill [56]. Sheikholeslam later showed that a string of nonlinear heterogeneous vehicles could be string stable in position with the appropriate choice of control law but provided limited simulations [57, 58]. Subsequently, Shaw provided a more generalized analysis of heterogeneous vehicle strings and showed that any heterogeneous string could be shown to have bounded position-spacing errors if each vehicle had information of the preceding and the lead vehicle in the string [59].

The aforementioned research is primarily focused on relative position spacing of

vehicles in close proximity to each other. However, there is often a need for vehicles to be spaced in time while not being near to one another. Time spacing is more applicable than distance spacing when a group of aircraft are capable of traveling at a wider range of speeds. Furthermore, it may even be desirable for a group of aircraft to change speeds during a given scenario while maintaining a specified time spacing. This occurs many times a day in commercial aviation when aircraft begin their descent towards an airport at a faster speed than the end of the descent at the final approach to landing [60].

The requirements for an automated aircraft spacing system have recently been highlighted as the Federal Aviation Administration moves to implement its Interval Management concept [61, 62]. A variety of control laws have been designed to space heavy commercial aircraft with both constant speed and constant acceleration profiles [63, 60, 64, 65]. Additionally, an algorithm to calculate trajectories and guarantee safety from collision was analyzed for a group of homogeneous aircraft [66].

Furthermore, NASA developed a spacing algorithm and flight tested a prototype implementation on medium passenger aircraft (Boeing 757 and Boeing 737) as a part of NASA's Air Traffic Management Technology Demonstration (ATD-1) [67, 68]. Notably, a common attribute of these aircraft control laws is that only position and speed information may be passed between aircraft based on existing and planned aircraft hardware [61].

Due to the emphasis on commercial aviation, the existing aircraft string stability research has focused on a linear analysis of homogeneous passenger aircraft. However, there is a lack of research into the string stability of airspeed heterogeneous, nonlinear aircraft. The scenario is applicable to both current military aviation and future commercial aviation as smaller unmanned vehicles are integrated into the National Airspace System. This dissertation aims to expand this area of research.

2.7 Summary

The coordinated arrival time problem has been well-researched with researchers primarily solving the problem using a “plan-and-follow” or a real-time guidance method. The “plan-and-follow” method is able to achieve coordinated arrival time across a wide range of environmental constraints to include wind and keep-out zones. The resulting paths are calculated for a given environment and then followed by a path follower. The path followers reviewed are able to closely follow a path which allows the predicted arrival time, from the planning phase, to be achieved with little change in velocity.

However, the “plan-and-follow” method is not robust to changes in arrival time and a new path is typically required to be computed. This is the primary limitation of the “plan-and-follow” method in that path calculation is typically not achievable in real time depending on the definition of real time. If a reference update rate of less than a second is assumed, then the path calculation is not achievable.

Real-time trajectory guidance is, by definition, able to control the trajectory in real time. The ability to control trajectory and velocity increases control over arrival time which is determined by the remaining distance divided by velocity. The method is also robust to changes in the environment and desired arrival time which are typically inputs into the guidance law itself.

However, when utilizing real-time trajectory guidance, the path to be flown is not specified ahead of time which does not allow for a simple prediction of where the vehicle will maneuver. Subsequently, it is difficult to incorporate keep-out zones because it is difficult to predict when the path will intersect a keep-out zone.

This research combines “plan-and-follow” with real-time guidance to achieve real-time arrival time control while including keep-out zone constraints and providing a predictable path. This is primarily accomplished by separating the path following

control loop from the arrival time control loop.

First, a simple and easily adjustable path shape is chosen which may be defined by one or two variables. The path is assumed to be at constant altitude and comprised of straight and circular path segments; the same as the path segments of a Dubins path. The simple shape allows for keep-out zones to be avoided by determining if the assumed path will intersect the keep-out zone. A path follower is then selected which simply follows the calculated desired path.

An airspeed controller will then be designed to finely control the arrival time in the final portions of the path when the path size may not be adjusted. String stability theory is applied in the design and analysis of the algorithm to ensure a cooperative timing scenario is robust to changes in arrival time and environment. First, the dynamics models used in the various portions of this research are derived.

III. Aircraft and Wind Models

3.1 Introduction

Two aircraft flight dynamics models are used in this research. Each model consists of basic flight dynamics and a limited flight control law. It is assumed the flight control law accepts airspeed and bank angle commands from the arrival time algorithm developed in this research. Furthermore, the bank angle commands are assumed to be calculated to follow straight and circular path segments at a constant altitude.

First, a simplified dynamics and control model is derived which will aid in the initial design and analysis of the arrival time control algorithm. This type of simplified dynamics model was common to nearly all research reviewed in Chapter II. The second model is comprised of a six degree of freedom nonlinear F-16 model provided in Ref. [6] along with an altitude-hold, bank-hold, and airspeed-hold autopilot which are also designed in this chapter.

Finally, the effects of wind are significant aspects of this research. The wind is modeled as a steady state component summed with a noisy component representing turbulence. The turbulence model outputs the effect of turbulence onto the translational and rotational rates of the aircraft which will drive how the wind effects are included into portions of the model. The Dryden Turbulence model used in this research is discussed at the end of this chapter.

3.2 Simplified Aircraft Model

The benefit of simplified models in the initial design of a system is a reduction in the number of variables in the system at the cost of a small amount of error. This section will develop the model used in this research from the basic aircraft equations of motion. The secondary benefit of this model is a typical aircraft dynamic response

without knowledge of aircraft forces and moments.

The derivation begins with the flat-earth and body reference frame six degree of freedom equations of motion taken from Stevens and Lewis [6] presented in Eq. (12) to Eq. (23); note that sine, cosine, and tangent have been abbreviated s , c , and t for brevity.

Force equations:

$$\dot{U} = RV - QW - gs\theta + (X_A + X_T)/m \quad (12)$$

$$\dot{V} = -RU + PW + gs\phi c\theta + (Y_A + Y_T)/m \quad (13)$$

$$\dot{W} = QU - PV + gc\phi c\theta + (Z_A + Z_T)/m \quad (14)$$

Kinematic equations:

$$\dot{\phi} = P + t\theta(Qs\phi + Rc\phi) \quad (15)$$

$$\dot{\theta} = Qc\phi - Rs\phi \quad (16)$$

$$\dot{\psi} = (Qs\phi + Rc\phi)/c\theta \quad (17)$$

Moment equations:

$$(J_x J_z - J_{xz}^2)\dot{P} = J_{xz}[J_x - J_y + J_z]PQ - [J_z(J_z - J_y) + J_{xz}^2]QR + J_z\ell + J_{xz}n \quad (18)$$

$$J_y\dot{Q} = [J_z - J_x]PR - J_{xz}(P^2 - R^2) + m \quad (19)$$

$$(J_x J_z - J_{xz}^2)\dot{R} = [(J_x - J_y)J_x + J_{xz}^2]PQ - J_{xz}[J_x - J_y + J_z]QR + J_{xz}\ell + J_x n \quad (20)$$

Navigation equations:

$$\dot{p}_N = U c \theta c \psi + V(-c \phi s \psi + s \phi s \theta c \psi) + W(s \phi s \psi + c \phi s \theta c \psi) \quad (21)$$

$$\dot{p}_E = U c \theta s \psi + V(c \phi s \psi + s \phi s \theta s \psi) + W(-s \phi c \psi + c \phi s \theta s \psi) \quad (22)$$

$$\dot{h} = U s \theta - V s \phi c \theta - W c \phi c \theta \quad (23)$$

The control input into these equations of motion is through the aerodynamic forces (X_A, Y_A, Z_A) , thrust forces (X_T, Y_T, Z_T) , and moments (ℓ, m, n) . The forces and moments are calculated with detailed knowledge of the aircraft aerodynamics and engine performance. Additionally, the mass properties of the aircraft must be known to determine the mass (m) , moments of inertia (J_x, J_y, J_z) and product of inertia (J_{xz})

The effects of wind on the dynamics are characterized with an axes transformation. First, the direction cosine matrix (DCM) which transfers the forward-right-down body reference frame to the north-east-down inertial reference frame is defined as

$$\mathbf{C}_{frd/ned} = \begin{bmatrix} c \theta c \psi & c \theta s \psi & -s \theta \\ (-c \phi s \psi + s \phi s \theta c \psi) & (c \phi c \psi + s \phi s \theta s \psi) & s \phi c \theta \\ (s \phi s \psi + c \phi s \theta c \psi) & (-s \phi c \psi + c \phi s \theta s \psi) & c \phi c \theta \end{bmatrix}. \quad (24)$$

As an example, this matrix is also used to derive the navigation equations of motion by the relation,

$$\begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ \dot{p}_D \end{bmatrix} = \mathbf{C}_{frd/ned}^T \begin{bmatrix} U \\ V \\ W \end{bmatrix}. \quad (25)$$

Note that D represents the down direction while the equations of motion use h where $-\dot{p}_D = \dot{h}$. The DCM is used to relate wind, typically defined in the north-east-down reference frame, to the body reference frame by the relation,

$$\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} - \mathbf{C}_{frd/ned} \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix} \quad (26)$$

where the body reference frame wind relative velocities are U' , V' , and W' and the inertial frame wind velocities are W_N , W_E , and W_D . The inertial frame wind velocities are defined as the velocity along the North, East, and Down axis respectively.

Finally, the true airspeed, angle of attack, and angle of sideslip are defined by the following equations,

$$V_T = \|[U', V', W']^T\| \quad (27)$$

$$\alpha = \tan^{-1}(W'/U') \quad (28)$$

$$\beta = \sin^{-1}(V'/V_T) \quad (29)$$

Simplifying the full dynamics model will be performed in two sections: assumptions on the state variables and assumptions on the control.

State Assumptions.

Several assumptions are made on the aircraft state variables. First, it is assumed the aircraft is a point mass such that

$$\alpha = \beta = 0. \quad (30)$$

The assumption of zero angle of attack leads to

$$0 = \alpha = \tan^{-1}(W'/U') \quad (31)$$

$$0 = W'. \quad (32)$$

Next, the assumption of zero sideslip leads to

$$0 = \beta = \sin^{-1}(V'/V_T) \quad (33)$$

$$0 = V'. \quad (34)$$

Finally, the derived results of $V' = W' = 0$ lead to

$$V_T = \|[U', V', W']^T\| \quad (35)$$

$$V_T = \sqrt{U'^2 + V'^2 + W'^2} \quad (36)$$

$$V_T = U' \quad (37)$$

The wind reference frame simplifications are then related to the inertial reference frame utilizing the previously defined DCM. The simplification begins with Eq (26) rearranged with the inertial body reference frame quantities on the left and the derived simplifications substituted for U' , V' , and W' ;

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_{frd/nd} \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix}. \quad (38)$$

The simplified body reference frame velocities are rotated back to the inertial

reference frame using Eq (25) which results in,

$$\begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ \dot{p}_D \end{bmatrix} = \mathbf{C}_{frd/ned}^T \left(\begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_{frd/ned} \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix} \right) \quad (39)$$

$$\begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ \dot{p}_D \end{bmatrix} = \mathbf{C}_{frd/ned}^T \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix}. \quad (40)$$

The equations reduce to the simplified navigation equation,

$$\dot{p}_N = V_T \cos \theta \cos \psi + W_N \quad (41)$$

$$\dot{p}_E = V_T \cos \theta \sin \psi + W_E \quad (42)$$

$$\dot{h} = V_T \sin \theta - W_D. \quad (43)$$

Control Assumptions.

Next, assumptions over the rotational and translational control are defined to affect V_T , θ , and ψ in the simplified navigation equations.

Rotational Control.

A common criterion to evaluate an aircraft's roll performance is to compute the equivalent roll-mode time constant [6]. This criterion is often used because the roll mode may be accurately modeled as a first order system when designing a control system. This simplification is used for the model presented in this research. Assuming

a first-order roll mode, the roll acceleration of the model is defined by

$$\dot{P} = -\frac{1}{\tau_P}(P + \omega_P) + \frac{1}{\tau_P}u_P \quad (44)$$

where τ_p is the roll-mode time constant and ω_P is effect of wind turbulence on roll rate.

The pitch and yaw axes will be addressed by utilizing the relationship between centripetal acceleration and turn rate. First, it is assumed that load factor (N_Z) and gravity (g) are the only forces acting upon the aircraft in the Y and Z axes. The acceleration due to gravity is rotated from the inertial frame into the body reference frame using the previously defined DCM, $\mathbf{C}_{frd/ned}$,

$$\mathbf{a}_{frd_g} = g\mathbf{C}_{frd/ned}\mathbf{e}_3 = \begin{bmatrix} -g \sin \theta \\ g \cos \theta \sin \phi \\ g \cos \theta \cos \phi \end{bmatrix}. \quad (45)$$

Next, the body axes accelerations due to load factor and gravity are summed resulting in

$$\mathbf{a}_{frd} = \mathbf{a}_{frd_g} - N_Z\mathbf{e}_3 = \begin{bmatrix} -g \sin \theta \\ g \cos \theta \sin \phi \\ -N_Z + g \cos \theta \cos \phi \end{bmatrix}. \quad (46)$$

Finally, it is assumed there are no other accelerations acting upon the aircraft besides gravity and load factor. The accelerations acting upon the aircraft are transformed into rates using centripetal acceleration as defined by

$$\omega = \frac{a_c}{V_T} \quad (47)$$

where a_c is the acceleration and ω is the angular rate.

Applying this equation to the Y axis and Z axis using the accelerations defined in Eq. (46) results in

$$Q = \frac{g(N_Z - \cos \theta \cos \phi)}{V_T} + \omega_q \quad (48)$$

$$R = \frac{g(\cos \theta \sin \phi)}{V_T} + \omega_r \quad (49)$$

which provides a simplified definition for the pitch and yaw rates including the effects of wind. The simplified angular rates, Q and R , are substituted back into the original Euler angle kinematic equations, Eq (15) to Eq (17), and reduced resulting in

$$\dot{\phi} = P + \frac{N_Z g}{V_T} \tan \theta \sin \phi + \omega_p + \omega_q \sin \phi \tan \theta + \omega_r \cos \phi \tan \theta \quad (50)$$

$$\dot{\theta} = \frac{g}{V_T} (N_Z \cos \phi - \cos \theta) - \omega_q \cos \phi + \omega_r \sin \phi \quad (51)$$

$$\dot{\psi} = \frac{N_Z g \sin \phi}{V_T \cos \theta} + \omega_q \frac{\sin \phi}{\cos \theta} + \omega_r \frac{\cos \phi}{\cos \theta}. \quad (52)$$

Finally, the load factor dynamics are assumed to follow a first-order system of the form

$$\dot{N}_Z = -\frac{1}{\tau_Z} N_Z + \frac{1}{\tau_Z} u_Z \quad (53)$$

where u_Z is the commanded load factor and τ_Z is the time constant of this system. This assumption replicates the typical desired time response of load factor control augmentation systems [6].

Translational Control.

The simplified model assumes the aircraft has adequate control over airspeed such that any commanded airspeed is achievable. The dynamic equation

$$\dot{V}_T = u_V - V_T \quad (54)$$

is used to model the airspeed dynamics of the vehicle where u_V is the commanded airspeed. To increase the realism of the model, an asymmetric airspeed rate limit is implemented to simulate the phenomenon where an aircraft's magnitude of acceleration is typically lower than the magnitude of deceleration. The rate limit is modeled with the saturation function where

$$\underset{a}{\overset{b}{f}}(x) = \begin{cases} b, & x > b \\ x, & b \geq x \geq a \\ a, & x < a \end{cases} \quad (55)$$

Applying the saturation to the airspeed model with some rate-limits $A < B$ results in

$$\dot{V}_T = \underset{A}{\overset{B}{f}}(u_V - V_T) \quad (56)$$

Simplified Model Summary.

This section summarizes the resulting dynamics of the simplified model as a three-dimensional and two-dimensional model.

Simplified Three-Dimensional Model.

The assumptions of the previous derivation are combined and summarized with the following three-dimensional dynamics model.

$$\dot{\phi} = P + \frac{N_Z g}{V_T} \tan \theta \sin \phi + \omega_p + \omega_q \sin \phi \tan \theta + \omega_r \cos \phi \tan \theta \quad (57)$$

$$\dot{\theta} = \frac{g}{V_T} (N_Z \cos \phi - \cos \theta) - \omega_q \cos \phi + \omega_r \sin \phi \quad (58)$$

$$\dot{\psi} = \frac{N_Z g \sin \phi}{V_T \cos \theta} + \omega_q \frac{\sin \phi}{\cos \theta} + \omega_r \frac{\cos \phi}{\cos \theta} \quad (59)$$

$$\dot{p}_N = V_T \cos \theta \cos \psi + W_N \quad (60)$$

$$\dot{p}_E = V_T \cos \theta \sin \psi + W_E \quad (61)$$

$$\dot{h} = V_T \sin \theta - W_D \quad (62)$$

$$\dot{P} = -\frac{1}{\tau_p} (P + \omega_p) + \frac{1}{\tau_p} u_p \quad (63)$$

$$\dot{N}_Z = -\frac{1}{\tau_Z} N_Z + \frac{1}{\tau_Z} u_Z \quad (64)$$

$$\dot{V}_T = \int_A^B (u_V - V_T) \quad (65)$$

Simplified Two-Dimensional Model.

The model is further simplified to a two-dimensional model by assuming constant altitude and zero vertical component of wind speed for all time. These assumptions result in

$$0 = \dot{h}(t) = V_T \sin \theta(t) - 0 \quad (66)$$

$$0 = \theta(t). \quad (67)$$

It follows that if $\theta(t) = 0$ then $\dot{\theta}(t) = 0$. It is assumed that the load factor is precisely controlled to achieve $\dot{\theta}(t) = 0$. This allows for the load factor to be defined by bank

angle, commonly referred to as a level turn [69]. The equation is derived as follows,

$$\dot{\theta} = \frac{g}{V_T}(N_Z \cos \phi - \cos \theta) \quad (68)$$

$$0 = \frac{g}{V_T}(N_Z \cos \phi - 1) \quad (69)$$

$$N_Z = \frac{1}{\cos \phi}. \quad (70)$$

Equation (70) is accurate for steady-state constant-altitude conditions [69]. However, the combination of high bank angles and high roll rates result in N_Z rates of change which may not be assumed to be achievable by a realistic aircraft, violating the assumption that N_Z is precisely followed. This is shown by taking the time derivative of Eq. (70),

$$\dot{N}_Z = \frac{P \tan \phi}{\cos \phi} \quad (71)$$

where increasing the magnitude of P and ϕ affects the magnitude of \dot{N}_Z . The maximum commanded values of ϕ and P are limited to achieve realistic values of \dot{N}_Z ; see Sec. 3.2.

Finally, apply the preceding state, wind, and control assumptions to the six-degree of freedom equations of motion presented at the beginning of this chapter results in

the following dynamics equations which are referred to as the simplified dynamics:

$$\dot{\phi} = P + \omega_P \quad (72)$$

$$\dot{\psi} = \frac{g}{V_T} \tan \phi + \omega_Q \sin \phi + \omega_R \cos \phi \quad (73)$$

$$\dot{p}_N = V_T \cos \psi + W_N \quad (74)$$

$$\dot{p}_E = V_T \sin \psi + W_E \quad (75)$$

$$\dot{P} = -\frac{1}{\tau_P}(P + \omega_P) + \frac{1}{\tau_P}u_P \quad (76)$$

$$\dot{V}_T = \int_A^B (u_V - V_T) \quad (77)$$

Again, this has been derived from the nonlinear equations of motion utilizing common fixed-wing aircraft state and control assumptions [6, 69]. This model is extensively used in the design and simulation of the control algorithm.

Aircraft Model Configurations.

Along with the 2-D dynamic equations, multiple aircraft configurations are defined. Four different classes of aircraft are chosen to mirror the aircraft classes used in flying quality analysis [70, 71]. The parameters which define the aircraft types are presented in Table 1.

Table 1. Aircraft Class Parameters

Aircraft Type	V_n (ft/s)	τ_p (s)	P_{max} (rad/s)	N_{Z_n} (G)	$N_{Z_{max}}$ (G)	$\dot{N}_{Z_{lim}}$ (G/s)	$V_{T_{min/max}}$ (ft/s)	$[A, B]$ (ft/s ²)
Light	293	1.0	1.83	1.15	2.0	3.0	[235,352]	[-10,4]
Medium Cargo	390	1.4	1.54	1.15	2.0	2.0	[313,468]	[-10,4]
Heavy Cargo	544	1.4	0.91	1.15	2.0	4.0	[468,620]	[-10,5]
Fighter	864	1.0	2.75	2.00	4.0	6.0	[771,956]	[-20,10]

The N_Z and \dot{N}_Z limits correspond to typical N_Z capabilities for the corresponding aircraft class and determine the bank angle limits for the aircraft assuming constant

altitude. The roll mode time constant and maximum roll rate were selected to correspond with the Level 1 flying qualities requirements for each aircraft class in the Cat A flight phase [71].

Simplified Model Bank Angle Command System.

The bank angle command drives a linear quadratic regulator (LQR) bank-angle command system. In LQR control, a feedback gain matrix \mathbf{K} is determined which minimizes Eq. (78) subject to the dynamics of the linear system defined by Eq. (79) and Eq. (80). In Eq. (78), \mathbf{Q} is a positive semi-definite matrix which determines the state weight and \mathbf{R} is a positive definite matrix which determines the control weighting. The matrix \mathbf{K} is determined by solving a Riccati equation [72].

$$J = \frac{1}{2} \int_0^{t_f} x^T \mathbf{Q} x + \mathbf{u}^T \mathbf{R} u \, dt \quad (78)$$

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (79)$$

$$u = -\mathbf{K}x \quad (80)$$

In this research, the state dynamics used for the LQR controller are Eq. (72) and Eq. (76) without the wind turbulence effects; the bank angle and roll rate dynamics. Additionally, the weight matrices \mathbf{Q} and \mathbf{R} used in this research are chosen to be,

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{R} = 1 \quad (81)$$

The input vector, \mathbf{x} , into the LQR system is comprised of the bank angle error and roll rate. The bank angle error is simply calculated as the difference in bank angle command, ϕ_{cmd} , and bank angle. The bank angle command is then limited

based on the assumed maximum load factor defined in Table 1 and the corresponding level-turn bank angle defined by Eq. (70)

The output of the LQR command system is a roll-rate command, u_1 , defined by Eq. (82) where \mathbf{K}_P is the 1×2 LQR gain matrix. The roll-rate command is subsequently limited by Eq. (83) or the aircraft roll rate limit, whichever magnitude is less, to prevent the assumed aircraft \dot{N}_Z limit from being exceeded as described in Sec. 3.2. The final output of the system is u_P , the roll-command input to Eq. (76). The bank angle command system is depicted for clarity in Fig. 5

$$u_1 = \mathbf{K}_P [\phi - \phi_{cmd}, P]^T \quad (82)$$

$$P_{N_Z} = \left| \dot{N}_{Z_{lim}} \frac{\cos \phi(t)}{\tan \phi(t)} \right| \quad (83)$$

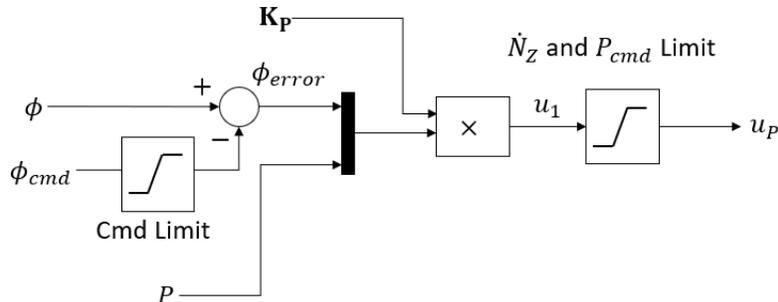


Figure 5. Linear Quadratic Regulator Bank Angle Command System

3.3 Nonlinear Aircraft Model

A nonlinear aircraft model provides a more realistic model to verify the performance of the arrival time algorithm. The model is based on the F-16 and is provided in Appendix A of Ref. [6]. The model calculates the forces and moments due to the aerodynamics and thrust used in the twelve equations of motion, Eqs. (12)-(23), and

calculates the corresponding rates of change. Additionally, the model incorporates a first order lag filter which simulates the engine as it spools up and down.

The states of the model are the twelve states of the aircraft equations of motion and the engine “lag” state. The four control inputs are the aileron deflection, rudder deflection, elevator deflection, and a throttle setting. Deflection and rate limits are applied to the surface deflections based on the limits in Ref. [6]. The throttle setting of 0 corresponds to idle power, 0.77 corresponds to “military power,” and 1.0 corresponds to maximum power (utilizing an afterburner).

Nonlinear Aircraft Model Control Systems.

A control system for the nonlinear model was designed to accept bank angle and speed commands to follow straight and circular path segments while maintaining a desired altitude. It is assumed that the circular paths to be followed may correspond to level turns flown with a load factor of up to 4.5G. The general design process makes a series of loop closures with a proportional, integral, and/or derivative (PID) controller for each loop [6]. Each loop is linearized at the appropriate operating condition and tuned to a desired response as described below. The F-16 model is assumed to be capable of maintaining a speed between 771 (*ft/s*) and 956 (*ft/s*) knots which allows the controller gains to remain constant without affecting stability.

A sliding control system is used to account for the high bank angles which a fighter aircraft may attain; a 4.5G level turn requires a bank angle of 77 *deg* to maintain. This causes the effectiveness of the primary control variables, N_Z and bank angle, to change with regard to tracking altitude or lateral tracking error.

As an example, an aircraft flying with zero bank angle best controls altitude with N_Z along the body pitch-axis. However, a change in N_Z for an aircraft tracking a 2G turn at a 60 *deg* bank angle affect both lateral tracking error and altitude error.

In order to allow for both low and high bank angle path following, a sliding control is used which interconnects the longitudinal and lateral-directional control systems. This allows the most efficient control-axis to be used to track the path at any given time. The design of the low bank angle autopilots is discussed first followed by the sliding control to follow a circular path.

Level Longitudinal Control Systems.

The overall design of the longitudinal control system is as an altitude hold system [6]. The system is a simple cascade of PID controllers which is summarized in Fig. 6. The “ N_Z Steady Reference” block is a part of the sliding control which will be discussed in a later section. However, when wings level the block outputs the level reference N_Z of 1G.

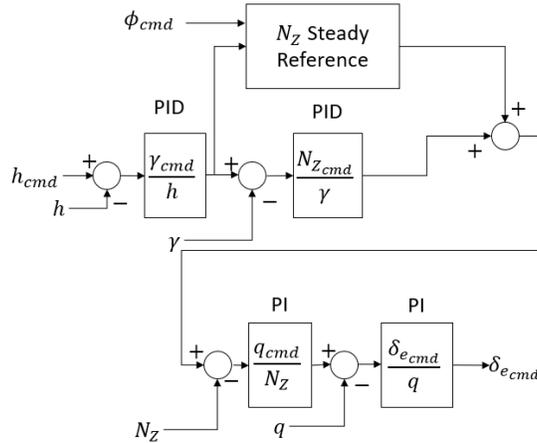


Figure 6. Altitude-Hold Autopilot Overview

The nonlinear model is linearized at an altitude of 10,000 ft and a true airspeed of 864 ft/s. Linear actuator dynamics are included in the linearized model with Eq. (84) where δ is the actuator deflection angle and δ_{cmd} is the commanded deflection. Next, a pitch rate command loop is designed to meet the Level 1 short period handling qualities specifications highlighted in Ref. [6] but derived from Ref. [71]. The tuned

pitch rate command system utilizes a proportional integral (PI) controller, and the resulting closed loop system is critically damped with a natural frequency of 6.7 rad/s.

$$\frac{\delta}{\delta_{cmd}} = \frac{20.2}{s + 20.2} \quad (84)$$

Next, a N_Z command system is looped around the pitch rate command system. Again, a PI controller is used however, the controller was tuned to simply provide a rapid and “deadbeat” N_Z response. The input to this command system is the N_Z differential from zero pitch rate flight. The “Turn Adjustment” is added to the command which serves as a reference for zero pitch rate flight. When the bank angle is zero, this term is simply $\cos \gamma$ which is often seen in altitude hold autopilots. The reference N_Z while turning will be discussed in the circular path tracking section.

A flight path angle command system is used for attitude control instead of pitch attitude because it more directly ties to altitude rate of change than pitch attitude for an aircraft which may not be at a low angle of attack while maneuvering. An equation for the flight path angle is derived by beginning with Eq. (85) which defines the stability axis airspeeds in terms of inertial velocity and earth-fixed wind speed. Multiplying both sides of the equation by the transpose of the rotation matrix simplifies to Eq. (87) due to the definition of the navigation equations of motion and because the rotation matrix is orthogonal.

$$\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} - \mathbf{C}_{frd/ned} \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix} \quad (85)$$

$$\mathbf{C}_{frd/ned}^T \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \mathbf{C}_{frd/ned}^T \begin{bmatrix} U \\ V \\ W \end{bmatrix} - \mathbf{C}_{frd/ned}^T \mathbf{C}_{frd/ned} \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix} \quad (86)$$

$$\begin{bmatrix} \dot{p}_N \\ \dot{p}_E \\ \dot{p}_D \end{bmatrix} = \mathbf{C}_{frd/ned}^T \begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} + \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix} \quad (87)$$

The flight path angle is defined as the angle of the inertial speed vector relative to the horizon which relates the inertial speed to climb rate by Eq. (88). Furthermore, the stability axis airspeeds are defined in terms of true airspeed, angle of attack, and angle of sideslip by Eq. (89) through Eq. (91).

$$\dot{h} = V_T \sin \gamma \quad (88)$$

$$U' = V_T \cos \alpha \cos \beta \quad (89)$$

$$V' = V_T \cos \alpha \sin \beta \quad (90)$$

$$W' = V_T \sin \alpha \quad (91)$$

Setting Eq. (88) equal to $-\dot{p}_D$ from Eq. (87) and substituting in the definition of the stability axis airspeeds results in Eq. (92); note the shorthand notation s and c for sin and cos respectively. Assuming the vertical wind speed is small relative to the true airspeed ($W_D \approx 0$), the flight path angle is estimated solely as function of the

Euler angles, angle of attack and angle of sideslip.

$$\gamma = -\arcsin(W_D/V_T - c\alpha c\beta s\theta + s\beta s\phi s\theta + s\alpha c\beta c\theta c\phi) \quad (92)$$

With an estimate for γ , the process of linearizing the system and determining the appropriate PID gains is repeated. The γ command system utilizes a PID controller and was tuned for a quick, deadbeat response.

Finally, the altitude loop utilizes an output limited PID tuned to have a near deadbeat response with a rise time of approximately 30 seconds. The limiter prevents an unreasonable climb angle from being commanded where the engine thrust would be unable to maintain airspeed. Since it limits γ , the limiter also effectively limits altitude rate which necessitated slower rise time of the response.

The value of the limiter was determined by determining the maximum flight path angle with the power limited to “military” power which could maintain airspeed while in a 2G turn. A limit of ± 10 degrees adhered to these limitations.

Level Lateral-Directional Control Systems.

The lateral directional control system is designed as a bank hold autopilot which incorporates a yaw stability augmentation system [6]. The system is again designed by linearizing the nonlinear model at 10,000 ft and 864 *ft/s* then making successive loop closures. An overview of the control system is given in Fig. 7. The “Climbing Adjustment” block contains the calculations for the higher bank angle sliding control.

The yaw damper utilizes a sideslip and yaw rate stability augmentation system. A yaw rate washout filter accounts for the constant, non-zero, yaw rate which occurs during steady turns. The PID gains are chosen to meet the Level 1 handling qualities requirements for Dutch roll and result in a critically damped Dutch roll with a natural frequency of 4 *rad/s*. The bank hold system consists of a simple proportional roll

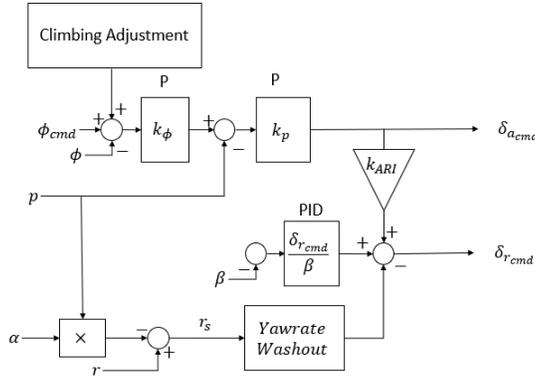


Figure 7. Bank-Hold Autopilot Overview

rate and bank angle gain; k_P and k_ϕ in Fig. 7. Both gains were tuned to achieve a rapid and deadbeat response to any bank angle command.

Finally, an aileron-rudder interconnect (ARI) is included to improve yaw damping during roll maneuvers [6]. The ARI provides a rudder command proportional to the aileron command to proactively address the adverse yaw associated with aileron deflections. In highly maneuverable aircraft, the gain is often a function of angle of attack and Eq. 93 is used to calculate the gain in real time [6].

$$k_{ARI} = 0.3\alpha - 0.7 \quad (93)$$

Circular Path Tracking Control System.

With the nominal lateral and longitudinal autopilots defined, this section discusses the interconnection between the two which allows for higher bank angle circular path tracking. Recall that this autopilot assumes the bank angle command inputs, from an outer loop navigation system, correspond to a constant altitude turn. Therefore, the input bank angle command also corresponds to a load factor for the level turn. This load factor is referred to as the “steady reference” load factor in the N_Z command loop, 1G when at constant altitude and zero bank angle.

Similarly, when at high bank angles the flight path angle commands from the N_Z command loop are converted to a bank angle “climbing adjustment”. Assuming the aircraft is in a steady level turn at constant load factor, changing the bank angle will result in an increase or decrease in altitude. The N_Z “turn adjustment” is discussed first.

The N_Z command is affected by utilizing the flight path angle and bank angle commands to generate an N_Z command resulting in a steady climbing turn; $\dot{\gamma} \approx 0$. This is utilized as the “steady reference” N_Z . The appropriate N_Z may be derived from the three-dimensional simplified equations of motion presented in the previous section, Sec. 3.2.

Recall that two of the assumptions of the simplified dynamics are zero angle of attack and sideslip which also results in an assumption of $\gamma = \theta$. Making this substitution into the $\dot{\theta}$ equations results in Eq. (94) which may be set equal to zero and solved for N_Z , Eq. (95). The command adjustment is calculated with Eq. (96) utilizing the flight path and bank angle commands.

$$\dot{\gamma} = \frac{g}{V_T}(N_Z \cos \phi - \cos \gamma) = 0 \quad (94)$$

$$N_Z = \frac{\cos \gamma}{\cos \phi} \quad (95)$$

$$N_{Z_{turn}} = \frac{\cos \gamma_{cmd}}{\cos \phi_{cmd}} \quad (96)$$

The bank angle command is affected with the control scheme in Fig. 8. The gain, $k_{\gamma\phi}$, is tuned to provide improved altitude tracking performance during turns while not affecting the overall stability of the autopilot. The output of the gain is multiplied by the bank angle command to increase the contribution of flight path into the bank angle command as the bank angle increases. The command is used instead of the

actual bank angle because it represents the state the aircraft should be at in the future.

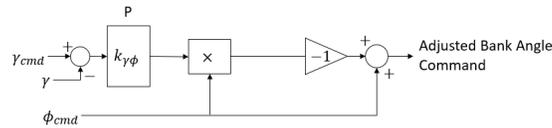


Figure 8. Bank-angle Command Climb Adjustment

Airspeed Control System.

The airspeed control system is a true airspeed-hold autopilot. A simple limited proportional controller was used to control speed error with the throttle command. The integrator is relatively weak to prevent unnecessary lag in the system. However, the system is initialized at the throttle required for steady level flight. A secondary term was added into the throttle command which was proportional to the flight path angle command as method to account for the change in steady state throttle with climbs or dives. The gains were tuned to provide a well damped response for both level and climbing flight.

Summary.

Figure 9 presents a simulation to highlight the performance of the outer-loop autopilots when following commands to follow a series of straight and circular path segments. The circular segments correspond to 2G level turns at 864 *ft/s*. The simulation is initialized at a low altitude to show the path tracking performance while climbing, level, and during the transition between the two. Additionally, moderate turbulence is included the simulation.

Once the desired altitude is achieved, the altitude is maintained within ± 100 *ft* of the set altitude, 10,000 *ft*. The altitude, bank angle, and speed performance are

all well damped while providing adequate path tracking performance.

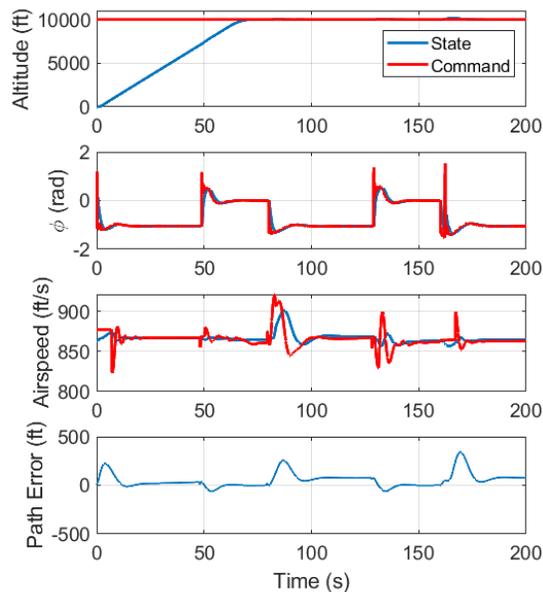


Figure 9. Outer-loop Autopilot Performance

3.4 Wind Turbulence Effects Model

Turbulence is included in this research to increase the realism of the scenarios by incorporating realistic noise into the system. The Dryden Wind Turbulence model is specified as an acceptable turbulence model in Ref. [71]. The model outputs the effect of wind turbulence on the translational rates (U , V , W) and rotational rates (P , Q , R) of the aircraft as noted in Sec. 3.2. The turbulence effects are created by passing white noise through a shaping filter for each of the specified outputs. Refer to Ref. [70, 71] for definitions of the filters and more detailed information on the turbulence model.

The translational effects are primarily defined by the turbulence intensity while the rotational effect also incorporates the size of the aircraft. Thus, the rotational output changes depending on the configuration of the aircraft. First, the global

parameters, independent of aircraft type, used in this research are presented in Tab. 2 and correspond to the recommendations in Ref. [71].

Table 2. Global Wind Turbulence Parameters from Ref. [2]

Altitude	10,000 <i>ft</i>
Scale Length	1750 <i>ft</i>
Light Turbulence Intensity	5.9 <i>ft/s</i>
Moderate Turbulence Intensity	9.4 <i>ft/s</i>

The aircraft dependent parameters are presented in Tab. 3. The airspeed is taken from the previous section on aircraft configuration while the wingspans are chosen to be representative of the aircraft type. The wingspans are based off the C-12, C-130, C-17, and F-16 for the Light Utility, Medium Cargo, Heavy Cargo, and Fighter aircraft categories respectively.

Table 3. Aircraft Dependent Wind Turbulence Parameters

Aircraft Type	Wingspan (<i>ft</i>)	Airspeed (<i>ft/s</i>)
Light Utility	60	293
Medium Cargo	132	390
Heavy Cargo	170	544
Fighter	33	864

A sample output of the turbulence for a Light Utility aircraft is provided in Fig. 10. Two standard deviations of the output are defined by dashed lines. In the lateral axes, the turbulence intensity is constant for each axis. In the rotational axes, each axis has a different intensity which is presented as a dashed line in the same color as the output effect. This behavior of intensity corresponds to the transfer functions of the model itself.

Finally, two standard deviations for each output parameter and aircraft type are given in Tab. 4. Note, the turbulence model effects the rotational axes of larger aircraft less than smaller aircraft.

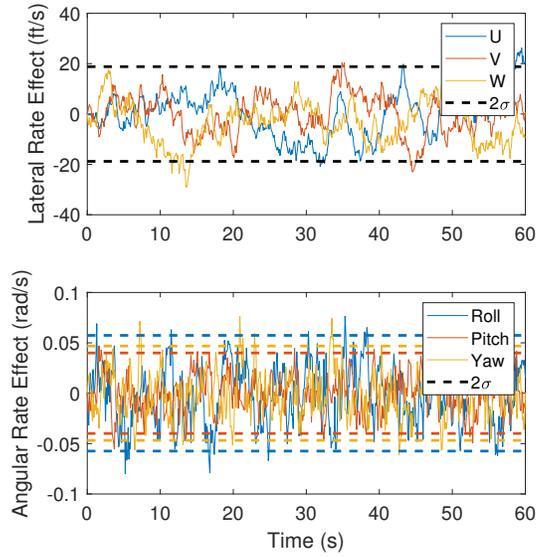


Figure 10. Moderate Turbulence Model Output with Two Standard Deviations (Dashed Lines)

Table 4. Turbulence Model Two Standard Deviations Value (Light/Moderate)

Aircraft Type	Lateral Axis 2σ (ft/s)	Roll Axis 2σ (rad/s)	Pitch Axis 2σ (rad/s)	Yaw Axis 2σ (rad/s)
Light Utility	5.9/9.4	0.030/0.049	0.019/0.031	0.022/0.036
Medium Cargo	5.9/9.4	0.018/0.029	0.013/0.020	0.015/0.023
Heavy Cargo	5.9/9.4	0.015/0.024	0.011/0.017	0.013/0.020
Fighter	5.9/9.4	0.045/0.072	0.026/0.042	0.030/0.048

3.5 Wind Triangle

This section provides a brief explanation of the wind triangle because it is the basis for many calculations in this research. The wind triangle describes the relationship between the airspeed, wind speed, and ground speed vectors. The ground speed vector is simply the vector sum of the wind speed and airspeed. Two important angles result from this definition. The heading, ψ , is the direction of the airspeed vector referenced to some axis; for example, “North” or the y-axis. The course angle, χ , is the direction of the ground speed vector referenced to the same axis.

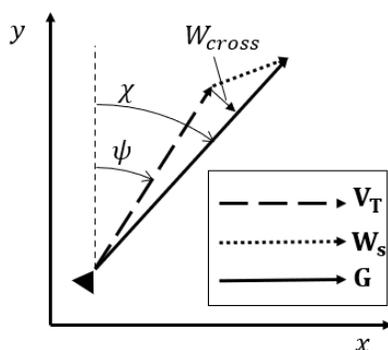


Figure 11. Wind Triangle

The primary use of the wind triangle in this research is the determination of a heading which corresponds to a desired ground course given some wind vector. First, assume the desired course is along the y-axis, or “North.” The required heading angle is one which results in no change in the x-position of the aircraft, $\dot{x} = 0$. The required heading is then derived by solving Eq. (97) for ψ resulting in Eq. (98).

$$\dot{x} = 0 = V_T \sin \psi + W_x \quad (97)$$

$$\psi = \arcsin \left(-\frac{W_x}{V_T} \right) \quad (98)$$

The relationship in Eq. (98) may be made more general by aligning the y-axis of this arbitrary reference frame with the desired ground track. The rotation results in the x-direction being perpendicular to the desired course angle; in the direction of W_{cross} in Fig. 11. The required heading along an arbitrary ground course, χ , is derived by the same method resulting in Eq. (100).

$$\dot{x}_{cross} = 0 = V_T \sin(\psi - \chi) + W_{cross} \quad (99)$$

$$\psi = \arcsin\left(-\frac{W_{cross}}{V_T}\right) + \chi \quad (100)$$

3.6 Wind Measurement Model

To facilitate accurate timing calculations, it is assumed the wind speed is measured by some type of on-board system. In practice, the wind may not be measured directly since the aircraft is moving through the moving wind field [73]. The wind may be calculated by utilizing the definition of the stability axis airspeeds, reproduced in Eq. (101).

$$\begin{bmatrix} U' \\ V' \\ W' \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} - \mathbf{C}_{frd/nd} \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix} \quad (101)$$

A GPS or inertial navigation sensor is used to measure U , V , and W while pitot tubes, angle of attack, and angle of sideslip sensors are used to measure U' , V' , W' . Taking the difference of the inertial and stability axis velocity vectors results in the wind speed vector. To simulate the measurement and calculation steps, a simple low pass filter is applied to the windspeed model which is the sum of the steady-state and

turbulent wind. The filter time constant is chosen to be 10 *s*.

3.7 Aircraft and Environmental Models Summary

This chapter defined and derived the various models to be used in this research. A simplified model was derived from the nonlinear equations of motion to provide a model which may be used analytically and is efficient to simulate. Additionally, a nonlinear model and command control system were derived to provide a realistic model which may verify any results produced by the simplified model. Finally, the modeling of the wind environment is important to this research to ensure any results may hold if flight tested in the open air. The next portion of this research utilizes the simplified model and environmental models to derive and evaluate an analytic solution to an optimal control problem.

IV. Optimal Lead Turn Problem

Fixed-wing aircraft cannot precisely track a Dubins path due to discontinuous changes in the path heading-rate-of-change. The path heading-rate-of-change is simply the velocity of the aircraft divided by turn radius. Thus, a straight segment with an infinite turn radius has a heading rate-of-change of zero. Conversely, a circular segment with a finite, positive turn radius has a finite heading rate-of-change.

The issue with fixed wing aircraft precisely tracking a Dubins path is easily deduced using the simplified constant altitude dynamics from the previous section. Assuming constant altitude, the heading dynamics are given by Eq. (102). The dynamics show that a fixed-wing aircraft would require instantaneous changes in bank angle to generate the discontinuous heading rate required to precisely track a Dubins path.

$$\dot{\psi} = \frac{g}{V_T} \tan \phi \quad (102)$$

The effect of the inability to instantaneously change bank angle is highlighted in Fig. 12. Figure 12 shows the desired straight path segment as a dotted red line and the curved path segment as a dashed red line. If the aircraft begins to change its bank angle at the transition between the straight and circular segment, when the path heading rate changes, the black path results and the aircraft will end in a position off the desired path and not tangent to the path when the desired heading rate is achieved.

Since travel time is distance traveled divided by velocity, a change to the distance traveled, assuming constant velocity, will result in a change to the travel time. If a time constraint is imposed, some form of compensation is required to satisfy the time constraint. Simply adjusting the airspeed will be investigated in later sections



Figure 12. Dubins Path Following Example

however, one other option is to minimize the tracking error so that airspeed compensation is not required. For a minimal tracking-error solution to be useful in a realistic scenario, it must be able to be calculated in near real time.

To correct the tracking error highlighted in Fig. 12, a simple solution is to begin the turn before the transition point resulting in a final position much closer to the path. This is shown by the gray path in Fig. 12 which is the same as the black path, only shifted to begin the turn on the straight segment. Therefore, the design space in minimizing the tracking error is the location of the initial position and the roll control during the maneuver.

The following section develops and solves the problem of finding the tracking-error optimal method to transition between a straight and circular segment, or vice versa. The problem is initially solved numerically to determine the general form of the solution which may not be calculated in real-time. Next, a near-optimal analytic solution is developed by simplifying the original problem. Finally, the near-optimal solution is integrated into existing guidance laws to quantify the benefit of the near-optimal lead turn in realistic scenarios. A portion of the research in this section has been published in [74].

4.1 Optimal Lead Turn Problem Definition

The optimal lead-turn problem is to determine the optimal control and initial position which minimize the tracking error during the segment transitions of a Dubins path. The dynamics model used in the problem is the simplified dynamics model derived in the previous section, reproduced in Eqs. (103)-(107). In addition to the assumption of constant altitude, this problem assumes constant true airspeed throughout the maneuver. As a reminder, the lateral speed effects of wind are modeled through W_x and W_y while the angular rate effects of wind are modeled with ω_P , ω_q , and ω_r for the roll, pitch, and yaw axes, respectively. Finally, the course angle is defined by Eq. (108).

$$\dot{x}(t) = V_T \sin \psi(t) + W_x \quad (103)$$

$$\dot{y}(t) = V_T \cos \psi(t) + W_y \quad (104)$$

$$\dot{\psi}(t) = \frac{g}{V_T} \tan \phi(t) + \omega_q \sin \phi(t) + \omega_r \cos \phi(t) \quad (105)$$

$$\dot{\phi}(t) = p(t) + \omega_P \quad (106)$$

$$\dot{p}(t) = -\frac{1}{\tau_P} (u(t) - (p(t) + \omega_P)) \quad (107)$$

$$\chi(t) = \text{atan2}(\dot{x}(t), \dot{y}(t)) \quad (108)$$

The two versions of the desired path are depicted in Fig. 13. The aircraft flight path is constrained to begin and end tangent to the desired (red-dashed) path. However, the initial position of the aircraft is used as a design variable in the optimization problem. The initial position is defined by y_0 if the initial segment is straight or s_0 if the initial segment is a circular segment. Specifically, s_0 is the arc length prior to the path transition point.

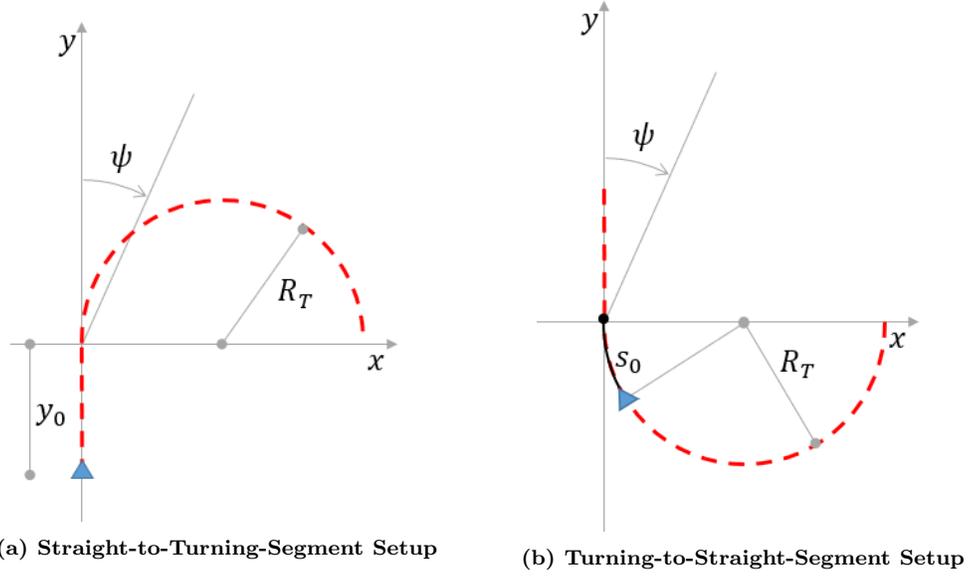


Figure 13. Optimal Lead Turn Problem Setup

Cost Functional.

The cost functional of the problem is chosen to be the integral tracking error as defined by Eqs. (109)-(111). The cost functional is analogous to the \mathcal{L}_2 -norm of the error. The maximum error, or infinity norm, was not used since any tracking error will affect the arrival time and not just the maximum path error.

$$J = \sqrt{\int_0^{t_f} e(t)^2 dt} \quad (109)$$

$$e(t) = x(t), \text{ if on straight segment} \quad (110)$$

$$e(t) = R_T - \sqrt{(x(t) - R_T)^2 + y(t)^2}, \text{ if on circular segment} \quad (111)$$

Terminal Constraints.

The terminal constraints of the problem are for the ground path to be tangent to the desired path with zero tracking error at the beginning and end of the maneuver.

The problem defines the straight path segment to be part of the y-axis, so the x-position must be zero at the appropriate condition. Since wind is included, the course angle, and not the heading angle, must be zero at this condition. Finally, bank angle and roll rate must also be equal to zero.

The turning path constraints are derived from the steady state level turn equations [69]. A constant altitude, constant speed, level turn is defined by Eq. (112). The position constraints are derived by utilizing the parametric definition of a circle, Eqs. (113)-(114). Finally, if the initial path segment is circular, the initial course angle is related to the initial arc-length, s_0 , through Eq. (115).

$$\phi(t) = \arctan\left(\frac{V_T^2}{gR_T}\right) \quad (112)$$

$$x(t) = R_T - R_T \cos \chi(t) \quad (113)$$

$$y(t) = R_T \sin \chi(t) \quad (114)$$

$$\chi(0) = R_T s_0 \quad (115)$$

As the primary focus of this problem is tracking ground-fixed path, the heading constraints are defined in terms of course angle. The relationship between airspeed and groundspeed, the wind triangle, is primarily discussed in Sec. 3.5. Applying the wind triangle to a straight path segment where the direction of travel is along the positive y-axis, the relationship between course angle and heading is given by

$$\psi = \arcsin\left(-\frac{W_x}{V_T}\right) \quad (116)$$

The heading along a circular segment is derived in the same manner but the crosswind must be calculated in the numerator of the arcsine.

$$\psi = \arcsin \left(-\frac{W_x - W_y \sin \chi}{V_T} \right) + \chi \quad (117)$$

Mathematical Definition.

Let the maximum roll-rate command be u_{max} and the final time of the maneuver be t_f . Combining the presented constraints and definitions, the optimal lead turn problem is mathematically defined by Eq. (118)

$$\arg \min_{u(t), y_0} \sqrt{\int_0^{t_f} e(t)^2 dt} \quad (118a)$$

subject to Dynamic Constraint Eqs. (103)-(107),

Control Bound Constraint,

$$|u(t)| \leq u_{max}, \quad (118b)$$

Straight-to-Turn Terminal Constraints,

$$y(0) = y_0, \quad x(0) = 0, \quad \psi(0) = 0, \quad \phi(0) = \phi_0, \quad p(0) = p_0, \quad (118c)$$

$$x(t_f) = R_T - R_T \cos \chi(t_f), \quad y(t_f) = R_T \sin \chi(t_f), \quad (118d)$$

$$\phi(t_f) = \arctan \left(\frac{V^2}{gR_T} \right), \quad p(t_f) = 0, \quad (118e)$$

$$y_0 \leq 0, \quad (118f)$$

Turn-to-Straight Terminal Constraints,

$$x(0) = R_T - R_T \cos(R_T s_0), \quad y(0) = R_T \sin(R_T s_0), \quad (118g)$$

$$\chi(0) = R s_0, \quad \phi(0) = \arctan \left(\frac{V_T^2}{gR_T} \right), \quad p(0) = 0, \quad (118h)$$

$$x(t_f) = 0, \quad \chi(t_f) = 0, \quad \phi(t_f) = 0, \quad p(t_f) = 0, \quad (118i)$$

$$s_0 \leq 0. \quad (118j)$$

Numerical Solutions.

The problem is first solved numerically using GPOPS-II to determine the form of the solution [34]. The GPOPS-II solution for a single straight-to-turning-segment scenario, Fig. 13a, is presented in Fig. 14. Figure. 14 is calculated using an airspeed of 19 ft/s in order to better show planar trajectory behavior. Multiple solutions were obtained for a variety of airspeeds, roll-mode time constants, and desired turn radii representative with all resulting in similar control behavior.

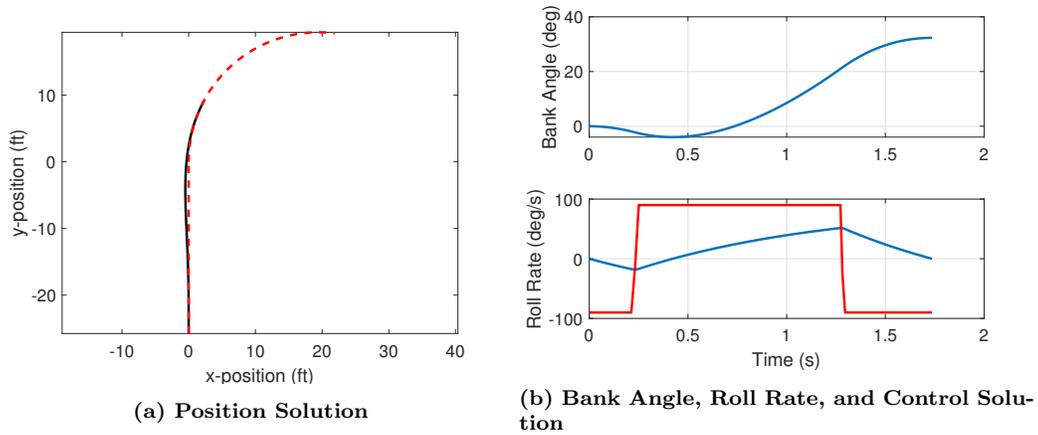


Figure 14. GPOPS-II Numeric Solution to the Optimal Lead Turn Problem

Figure 14b shows the control, in red, switches twice over the course of the solution which was observed in each calculated solution. The result of this behavior is better shown in Fig. 14a where the planar trajectory of the vehicle moves away from the desired path initially before ending precisely on the curved path segment. The bounded control behavior is often seen as the optimal solution to minimum time problems [31]. Subsequently, the problem was solved with GPOPS-II using a minimum time cost functional resulting in identical solutions. This behavior will later be used to simplify the optimization problem.

The multiple control switches in the solution indicate that an analytic optimal solution is likely not achievable. However, an analytic near-optimal solution may be

derived with the addition of several simplifying assumptions.

4.2 Near-Optimal Lead Turn Problem

The primary difficulty of the optimal lead turn problem is the control is unknown. Thus, the first assumption is that the control will simply achieve the final bank angle in minimum time. For now, it is assumed this assumption will allow the bank angle to be analytically defined as a function of time. However, this will be shown to be true in the following section.

The result of this assumption on the control is the maneuver will be completed in minimum time, but the final position of the vehicle cannot be guaranteed to be on the desired path. The cost functional is adjusted such that the terminal distance from the desired path is minimized using the terminal heading and position boundary conditions from Eq. (118). The change to the cost functional effectively minimizes the terminal constraint violation of the original problem.

The modified optimization problem for the turning segment to straight segment problem is defined by Eq. (119). The dynamic constraints are reduced since $\phi^*(t)$ is assumed to be an analytic function of time which incorporates u_{max} , τ_P , $\phi(t_f)$. The initial boundary constraints are defined by Eq. (119e) and Eq. (119f) while the final boundary constraints from Eq. (118) have been incorporated in the new cost functional, Eq. (119a). Finally, it is assumed that the wind, W_x and W_y , remains constant throughout the maneuver.

$$\arg \min_{y_0} \quad (x(t_f) + R_T \cos \chi(t_f) - R_T)^2 + (y(t_f) - R_T \sin \chi(t_f))^2 \quad (119a)$$

$$\text{subject to} \quad \dot{x}(t) = V_T \sin \psi(t) + W_x, \quad (119b)$$

$$\dot{y}(t) = V_T \cos \psi(t) + W_y, \quad (119c)$$

$$\dot{\psi}(t) = \frac{g}{V_T} \tan \phi^*(t), \quad (119d)$$

$$x(0) = \chi(0) = 0, \quad (119e)$$

$$y(0) = y_0 \quad (119f)$$

Similarly, the modified turning path segment to straight path segment problem is defined by Eq. (120) where the dynamics have again been condensed and the terminal boundary constraints incorporated in the modified cost functional. The form of Eq. (120a) is chosen to account for the terminal position and heading terminal constraints of the original problem. Furthermore, the R_T in Eq. (120a) is used as a weighting factor to ensure the relative magnitude of the position and heading terms is similar. The minimum time nature of the problem is preserved since $\phi^*(t)$ is assumed to be a minimum time control.

$$\arg \min_{s_0} \quad x(t_f)^2 + (R_T \chi(t_f))^2 \quad (120a)$$

$$\text{subject to} \quad \dot{x}(t) = V_T \sin \psi(t) + W_x, \quad (120b)$$

$$\dot{y}(t) = V_T \cos \psi(t) + W_y, \quad (120c)$$

$$\dot{\psi}(t) = \frac{g}{V_T} \tan \phi^*(t), \quad (120d)$$

$$x(0) = R_T - R_T \cos(R_T s_0), \quad (120e)$$

$$y(0) = R_T \sin(R_T s_0), \quad (120f)$$

$$\chi(0) = R_T s_0 \quad (120g)$$

4.3 Near-Optimal Lead Turn Solution

Minimum Time-to-Bank Solution.

To determine $\phi^*(t)$, an optimal control problem to achieve a desired bank angle in minimum time will be developed and solved. Without loss of generality, the problem is defined as starting at rest with wings level and commanding a positive desired bank angle. It will later be shown in the derivation that the key parameters of the solution only depend on the total bank angle change.

The resulting optimization problem is a straightforward second order optimal control problem. The solution approach used in this section utilizes Pontryagin's Minimum Principle (PMP) and classical optimal control methods [31]. Equation 121 defines the optimal control problem.

$$\arg \min_u \int_0^{t_f} 1 dt \quad (121a)$$

$$\text{subject to } \dot{\phi}(t) = p(t) \quad , \quad (121b)$$

$$\dot{p}(t) = -\frac{1}{\tau_P} p(t) + \frac{1}{\tau_P} u(t), \quad (121c)$$

$$\phi(0) = p(0) = p(t_f) = 0, \quad (121d)$$

$$\phi(t_f) = \phi_f, \quad (121e)$$

$$|u(t)| \leq u_{max} \quad (121f)$$

As a minimum-time problem, it is hypothesized that the solution will initially command a maximum roll rate and then switch to a minimum roll rate command at a point which achieves the desired bank angle with zero roll rate. Thus, it is expected there will be a single switching time, defined as t_1 , when the control switches from u_{max} to $-u_{max}$.

Hamiltonian & Pontryagin's Minimum Principle.

The Hamiltonian for this problem is defined by Eq. (122). Pontryagin's minimum principle states that an optimal control will minimize the Hamiltonian [31]. The result is Eq. (123) which shows the sign of the control depends on the sign of the second costate, $\lambda_2(t)$.

$$\mathcal{H}(t) = 1 + \lambda_1(t)p(t) + \lambda_2(t) \left(-\frac{1}{\tau_P} p(t) + \frac{1}{\tau_P} u(t) \right). \quad (122)$$

$$u(t) = \left. \begin{array}{ll} u_{max} & , \lambda_2(t) < 0 \\ -u_{max} & , \lambda_2(t) > 0 \\ \text{undetermined} & , \lambda_2(t) = 0 \end{array} \right\} \quad (123)$$

A difficulty of some optimal control problems is the existence of singular intervals. A singular interval is a “time interval of finite duration during which the necessary conditions...provide no information about” the control [31]. In this problem, a singular interval could occur if $\lambda_2(t) = 0$ for a finite amount of time. However, a property of minimum time problems involving linear systems of equations is that a singular interval cannot exist if the system is completely controllable [31]. Recall a system is controllable if the controllability matrix is full rank.

Lemma 4.1. *A singular interval does not exist in the minimum time-to-bank problem defined by Eq. (121)*

Proof. Writing the equations of motion, Eq. (121b) and Eq. (121c), in state-space form results in Eq. (124) which highlights the \mathbf{A} and \mathbf{B} matrix used to determine the controllability matrix. Note, it has been assumed that the states ϕ and p are both observable.

$$\begin{Bmatrix} \dot{\phi}(t) \\ \dot{P}(t) \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau_P} \end{bmatrix} \begin{Bmatrix} \phi(t) \\ P(t) \end{Bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\tau_P} \end{bmatrix} u(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t). \quad (124)$$

The controllability matrix is then formed in Eq. (125). The matrix in Eq. (125) is clearly full rank since the roll mode time constant for an aircraft must be a finite positive value. Thus, the system is completely controllable, and no singular interval

exists for the minimum time-to-bank problem.

$$\mathbf{C} = [\mathbf{B} \ \mathbf{AB}] = \begin{bmatrix} 0 & \frac{1}{\tau_P} \\ \frac{1}{\tau_P} & -\frac{1}{\tau_P^2} \end{bmatrix} \quad (125)$$

■

Costate Response.

A singular interval does not exist due to Lemma 4.1. Therefore, there are only two possible values for the optimal control which depend on the sign of the second costate; shown in Eq. (126).

$$u(t) = \begin{cases} u_{max} & , \lambda_2(t) < 0 \\ -u_{max} & , \lambda_2(t) > 0 \end{cases} \quad (126)$$

The remaining necessary conditions for optimal control are then applied to the problem statement resulting in,

$$\dot{\lambda}_1 = -\frac{\partial \mathcal{H}}{\partial \phi} = 0 \quad (127)$$

$$\dot{\lambda}_2 = -\frac{\partial \mathcal{H}}{\partial p} = -\lambda_1(t) + \frac{1}{\tau_P} \lambda_2(t). \quad (128)$$

Equations (127) and Eq. (128) are in the form of a linear system whose solution is define by

$$\lambda_1(t) = L_1 \quad (129)$$

$$\lambda_2(t) = L_1 \tau_P + e^{\frac{t}{\tau_P}} (\lambda_2(0) - L_1 \tau_P). \quad (130)$$

As $\lambda_1(t)$ is constant, it is simply defined as L_1 . It can be concluded that the control

switching time, t_1 , occurs at the precise time when $\lambda_2(t)$ changes signs and is equal to zero. Next, it is shown that $\lambda_2(t)$ changes sign only once.

Lemma 4.2. *Eq. (130) always has one zero in the context of the minimum time-to-bank problem*

Proof. The derivative of $\lambda_2(t)$ is a simple exponential function defined by Eq. (131). Time is greater than or equal to zero, thus the exponential term will always be positive. Additionally, the terms inside the parentheses are constant. Thus $\dot{\lambda}_2(t)$, Eq. (131), does not change sign indicating that $\lambda_2(t)$ will have at most one zero, as expected. Furthermore, $\lambda_2(t)$ may not be constant and result in an optimal control because a constant $\lambda_2(t)$ would result in a constant control, due to Eq. (126), which would clearly not achieve the terminal boundary constraint of $p(t_f) = 0$. Therefore, $\lambda_2(t)$ will have one zero in the context of the minimum time-to-bank problem.

$$\dot{\lambda}_2(t) = e^{\frac{t}{\tau_P}} \left(\frac{1}{\tau_P} \lambda_2(0) - L_1 \right) \quad (131)$$

■

Due to Lemma 4.2, it is concluded there is only a single control switching which occurs at time t_1 . The value of t_1 is determined by setting Eq. (130) at time $t = t_1$ to zero and solving for t_1 as shown in Eq. (132) and Eq. (133).

$$0 = \lambda_2(t_1) = L_1 \tau_P + e^{\frac{t_1}{\tau_P}} (\lambda_2(0) - L_1 \tau_P) \quad (132)$$

$$t_1 = \tau_P \ln \frac{-L_1 \tau_P}{\lambda_2(0) - L_1 \tau_P} \quad (133)$$

Time Response.

The previous section showed that there is only one control switch, and the control magnitude is either minimized or maximized. Next, analytic definitions for the states will be calculated to apply the boundary conditions. First, take the general form of the solution for a linear system given by,

$$\mathbf{x} = e^{At}\mathbf{x}(0) + e^{At} \int_0^t e^{-A\zeta} B u(\zeta) d\zeta. \quad (134)$$

Next, separate the integral into two consecutive integrals corresponding to the switching time t_1 ,

$$\mathbf{x}(t \leq t_1) = e^{At}\mathbf{x}(0) + e^{At} \int_0^t e^{-A\zeta} B u(\zeta) d\zeta \quad (135)$$

$$\mathbf{x}(t > t_1) = e^{At}\mathbf{x}(0) + e^{At} \int_0^{t_1} e^{-A\zeta} B u(\zeta) d\zeta + e^{At} \int_{t_1}^t e^{-A\zeta} B u(\zeta) d\zeta. \quad (136)$$

The control is constant in each interval but has a different sign due to Eq. (126). The initial control must be the same sign as the final desired bank angle, assuming the terminal bank angle is bounded $(-\pi, \pi)$, due to the equations of motion. Thus $u(t)$ in the first integral is replaced with u_{max} and $u(t)$ in the second integral is replaced with $-u_{max}$ resulting in,

$$\mathbf{x}(t \leq t_1) = e^{At}\mathbf{x}(0) + e^{At} \int_0^t e^{-A\zeta} B u_{max} d\zeta \quad (137)$$

$$\mathbf{x}(t > t_1) = e^{At}\mathbf{x}(0) + e^{At} \int_0^{t_1} e^{-A\zeta} B u_{max} d\zeta + e^{At} \int_{t_1}^t e^{-A\zeta} B (-u_{max}) d\zeta. \quad (138)$$

Expanding the equations using the matrices \mathbf{A} and \mathbf{B} from the equations of motion, Eq. (124), and evaluating the integrals results in Eqs. (139)-(142).

$$\phi(t \leq t_1) = u_{max} \left(t - \tau_P + \tau_P e^{-\frac{t}{\tau_P}} \right) + \phi_0 \quad (139)$$

$$p(t \leq t_1) = u_{max} \left(1 - e^{-\frac{t}{\tau_P}} \right) \quad (140)$$

$$\phi(t > t_1) = u_{max} \left(2t_1 - t + \tau_P - 2\tau_P e^{-\frac{t-t_1}{\tau_P}} + \tau_P e^{-\frac{t}{\tau_P}} \right) + \phi_0 \quad (141)$$

$$p(t > t_1) = u_{max} \left(2e^{-\frac{t-t_1}{\tau_P}} - e^{-\frac{t}{\tau_P}} \right) - u_{max} \quad (142)$$

Returning to the generality of the solution, a negative final bank angle, for an initial bank angle of zero, would simply require a change in the sign of the control due to the equations of motion. The case of a non-zero initial bank angle, $\phi_0 \neq 0$, is less straight forward.

Take Eq. (141) at time $t = t_f$. Simplify subtracting ϕ_0 from both sides yields an equation for the change in bank angle which is identical to Eq. (141) if $\phi_0 = 0$.

$$\phi(t_f) = \phi_f = u_{max} \left(2t_1 - t_f + \tau_P - 2\tau_P e^{-\frac{t_f-t_1}{\tau_P}} + \tau_P e^{-\frac{t_f}{\tau_P}} \right) + \phi_0 \quad (143)$$

$$\phi_f - \phi_0 = u_{max} \left(2t_1 - t_f + \tau_P - 2\tau_P e^{-\frac{t_f-t_1}{\tau_P}} + \tau_P e^{-\frac{t_f}{\tau_P}} \right) \quad (144)$$

Therefore, the solution for an initial bank angle of zero may be used in the case of any initial bank angle by assuming that the desired final bank angle used in the calculations is the desired change in bank angle.

Solving for t_1 and t_f .

In order to determine t_1 and t_f as functions of the problem parameters (τ_P , ϕ_f , etc.), expressions for $\lambda_2(0)$ and L_1 must be determined. The Hamiltonian of the minimum time-to-bank problem is not an explicit function of time and the final time

is free. Therefore, the Hamiltonian must be equal to zero at any time t [31].

$$\mathcal{H}(t) = 1 + \lambda_1(t)p(t) + \lambda_2(t) \left(-\frac{1}{\tau_P}p(t) + \frac{1}{\tau_P}u(t) \right) = 0. \quad (145)$$

Take the Hamiltonian, Eq. (145), at time $t = 0$. Next, substitute the analytic equations for the costates, roll rate, and control (derived in the previous sections) at time $t = 0$ into Eq. (145). Solving for $\lambda_2(0)$ results in Eq. (146).

$$\lambda_2(0) = -\frac{\tau_P}{u_{max}} \quad (146)$$

The final time, t_f , is then determined by expanding the equation for roll rate at $t = t_f$, Eq. (142), and applying the terminal boundary constraint, $p(t_f) = 0$. Taking the resulting Eq. (147), the equation for t_1 , Eq. (133), and $\lambda_2(0)$ are substituted and the entire equation is solved for t_f resulting in Eq. (148).

$$0 = u_{max} \left(2e^{-\frac{t_f-t_1}{\tau_P}} - e^{-\frac{t_f}{\tau_P}} \right) - u_{max} \quad (147)$$

$$t_f = \tau_P \ln \left(\frac{L_1 u_{max} - 1}{L_1 u_{max} + 1} \right) \quad (148)$$

Finally, L_1 is determined by evaluating the roll angle boundary condition at t_f . Equation (141) is set equal to the terminal roll angle condition, ϕ_f assuming an initial bank angle of zero. The resulting equation then rearranged to solve for L_1 resulting in two solutions defined by Eq. (149).

$$L_1 = \pm \frac{1}{u_{max}} e^{\frac{\phi_f}{2\tau_P u_{max}}} \sqrt{\frac{1}{e^{\frac{\phi_f}{\tau_P u_{max}}} - 1}} \quad (149)$$

The sign of the variable L_1 is determined by applying the following assumptions.

First, Eq. (150) is defined in the problem statement. Since u_{max} corresponds to the initial optimal control, it must logically be the same sign as ϕ_f , Eq. (151). Finally, the roll mode time constant (τ_P) is defined to be positive, Eq. (152). Thus, both exponents in Eq. (149) will be positive, the exponentials will both be greater than one, and the sign of L_1 is the same as the plus or minus.

$$\phi_f > 0 \tag{150}$$

$$u_{max} > 0 \tag{151}$$

$$\tau_P > 0 \tag{152}$$

The correct solution is the one which results in a positive t_f , Eq. (148). Inspecting Eq. (148), the inside of the logarithm must be greater than one for t_f to be positive. If L_1 is positive, the numerator will be smaller than the denominator resulting in a negative t_f . A negative L_1 will result in a positive t_f only if the inequality $-1 > L_1 u_{max}$ is true. Taking this inequality and expanding with the negative solution for L_1 results in,

$$-1 > L_1 u_{max} \tag{153}$$

$$-1 > -e^{\frac{\phi_f}{2\tau_P u_{max}}} \sqrt{\frac{1}{e^{\frac{\phi_f}{\tau_P u_{max}}} - 1}} \tag{154}$$

$$-1 > -\sqrt{\frac{e^{\frac{\phi_f}{\tau_P u_{max}}}}{e^{\frac{\phi_f}{\tau_P u_{max}}} - 1}}. \tag{155}$$

Since it has been assumed that $\phi_f > 0$, $\tau_P > 0$, and $u_{max} > 0$, the exponential terms will be greater than 1. Thus, the radicand will always be greater than 1 and the relation is satisfied. Therefore, L_1 must be negative, and Eq. (156) is the correct

solution.

$$L_1 = -\frac{1}{u_{max}} e^{\frac{\phi_f}{2\tau_P u_{max}}} \sqrt{\frac{1}{\frac{\phi_f}{e^{\tau_P u_{max}}} - 1}} \quad (156)$$

Minimum Time-to-Bank Solution Summary.

To summarize the solution to the minimum time-to-bank problem defined by Eq. (121), the optimal control is defined by Eq. (157) through Eq. (161).

$$u^*(t \leq t_1^*) = u_{max} \quad (157)$$

$$u^*(t > t_1^*) = -u_{max} \quad (158)$$

$$t_1^* = \tau_P \ln \frac{L_1 \tau_P}{\lambda_2(0) - L_1 \tau_P} \quad (159)$$

$$L_1 = -\frac{1}{u_{max}} e^{\frac{\phi_f}{2\tau_P u_{max}}} \sqrt{\frac{1}{\frac{\phi_f}{e^{\tau_P u_{max}}} - 1}} \quad (160)$$

$$\lambda_2(0) = -\frac{\tau_P}{u_{max}} \quad (161)$$

The resulting optimal bank angle as a function of time is defined by Eq. (162) and Eq. (163). Finally, the duration of the maneuver, t_f^* , is defined by Eq. (164). However, it is noted that the initial bank angle for this problem is zero, identified as $\phi^\dagger(t)$.

$$\phi^\dagger(t \leq t_1^*) = u_{max} \left(t - \tau_P + \tau_P e^{-\frac{t}{\tau_P}} \right) \quad (162)$$

$$\phi^\dagger(t > t_1^*) = u_{max} \left(2t_1^* - t + \tau_P - 2\tau_P e^{-\frac{t-t_1^*}{\tau_P}} + \tau_P e^{-\frac{t}{\tau_P}} \right) \quad (163)$$

$$t_f^* = \tau_P \ln \left(\frac{L_1 u_{max} - 1}{L_1 u_{max} + 1} \right) \quad (164)$$

To generalize the solution, let ϕ'_0 and ϕ'_f be the actual initial and final bank angle, respectively, while ϕ_f is defined by Eq. (165). The general optimal solution

may then be calculated with Eq. (166) where the change in bank angle, ϕ_f , is used in the optimal control solution defined by Eq. (157) through Eq. (161).

$$\phi_f = |\phi'_f - \phi'_0| \quad (165)$$

$$\phi^*(t) = \text{sign}(\phi'_f - \phi'_0)\phi^\dagger(t) + \phi'_0 \quad (166)$$

The analytically determined results were validated by solving the minimum time-to-bank problem numerically using the GPOPS-II optimal control software [34]. A rough initial guess which follows the desired path and meets the endpoint constraints was used and resulted in a locally minimal solution for each scenario calculated.

Several configurations of bank angle, roll-mode time constant, and maximum control limit were simulated. All simulated configurations achieved the same final time and switching time using both the analytic and numeric methods. A single configuration is shown for comparison. In this case, the desired bank angle is 60 *deg*, the roll mode time constant is 1.4 *s*, and the maximum roll rate limit is 45 *deg/s*.

Figure 15 presents the bank angle and roll rate comparison. It is easily seen the bank angle, roll rate, and control for the GPOPS-II computed solution and the analytically derived solution are well-matched. More precisely, the 2-norm of the difference between the analytic and GPOPS-II solution for bank angle is 3.54×10^{-4} *deg* and for roll rate is 4.64×10^{-2} *deg/s*.

Figure 16 presents the analytic costates and Hamiltonian along with the numerical estimates calculated with GPOPS-II. Again, the parameters from the two solutions are well matched. The analytic control is also plotted over $\lambda_2(t)$ to show that the switching time, t_1 , for the control corresponds to when λ_2 changes sign. Finally, the analytic Hamiltonian is zero, as expected, and the GPOPS-II estimated Hamiltonian is also near zero. The 2-norm of the Hamiltonian estimate is 1.81×10^{-3} where the

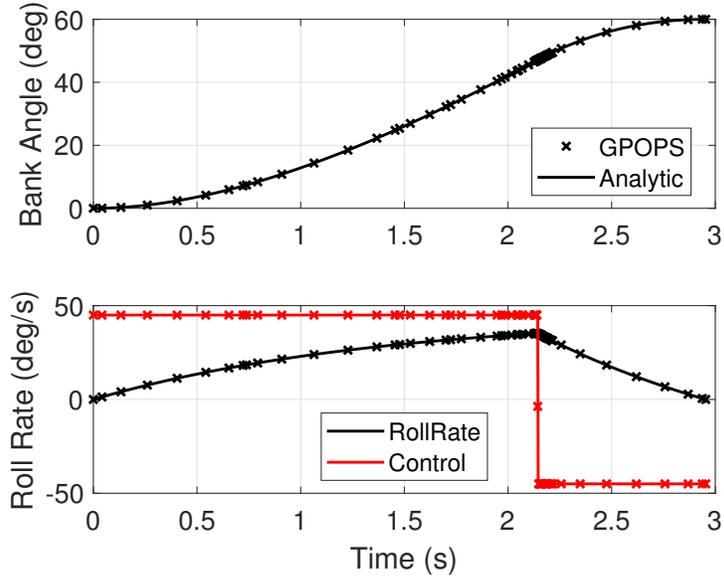


Figure 15. Minimum Time-to-Bank Solution Comparison, States

error is due to numerical calculation errors.

Near-Optimal Lead Turn Solution.

The resulting solution for the bank angle, Eq. (166), is integrated through the equations of motion, Eq. (167) through Eq. (169), to determine the final position and heading of the aircraft after completing the minimum time to bank maneuver. The final position is used to determine the optimal maneuver initiation point with regards to the lead turn problem defined by Eq. (119) and Eq. (120).

$$\dot{x}(t) = V_T \sin \psi^*(t) + W_x \quad (167)$$

$$\dot{y}(t) = V_T \cos \psi^*(t) + W_y \quad (168)$$

$$\dot{\psi}(t) = \frac{g}{V_T} \tan \phi^*(t) \quad (169)$$

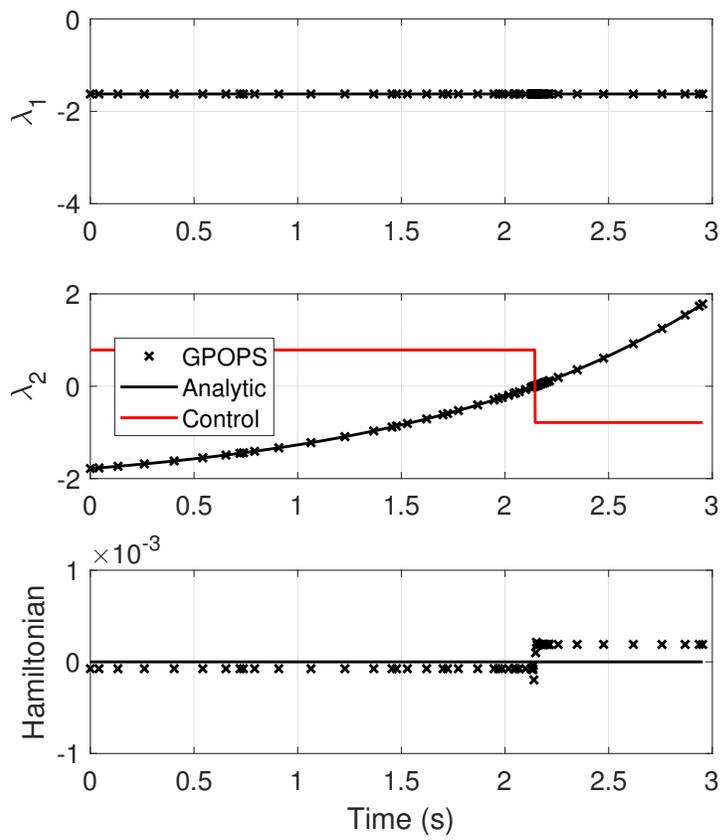


Figure 16. Minimum Time-to-Bank Solution Comparison, Costates

Determining Heading as a Function of Time.

Integrating the heading rate equation, Eq. (169), presents a challenge since the integral contains the term $\int \tan e^x dx$, which does not have an analytic solution. The challenge is addressed in this research by utilizing a Taylor polynomial to approximate the tangent of the bank angle.

Since the bank angle change is not assumed to be small, a different expansion is used for the two intervals in time, $[0, t_1]$ and $[t_1, t_f]$. A second-degree Taylor polynomial approximation is used for both intervals.

The first time-segment approximation is centered at $\frac{1}{4}\phi_f$ and the second is centered at $\frac{3}{4}\phi_f$. Although these points may not correspond to the bank angle at the center of each time segment, it is assumed they are close enough to allow for an accurate Taylor polynomial approximation; the resulting accuracy of this solution will be shown to validate this assumption. The second-degree Taylor polynomial approximation for heading rate, $\dot{\psi}_{ap}(t)$, is given by Eq. (170) where the variable a is the center of the approximation, either $\frac{1}{4}\phi_f$ or $\frac{3}{4}\phi_f$.

$$\dot{\psi}(t) \approx \dot{\psi}_{ap}(t) = \frac{g}{V_T} (\tan a - (\tan^2 a + 1)(a - \phi^*(t)) + (\tan^3 a + \tan a)(a - \phi^*(t))^2) \quad (170)$$

where $a = \frac{1}{4}\phi_f \forall t \leq t_1$, $a = \frac{3}{4}\phi_f \forall t > t_1$

The heading is approximated by integrating the corresponding heading-rate approximation. However, the inclusion of wind into the problem results in the initial heading being a function of the wind and the initial course angle, χ ; discussed in

Sec. 4.1. In the case of a straight segment, the heading is given by

$$\psi = \arcsin\left(-\frac{W_x}{V_T}\right). \quad (171)$$

However, the heading along a circular segment is more complex due to the requirement to determine the component of wind perpendicular to the ground path of the aircraft. Taking the analytic definition of the initial heading while in a turn, Eq. (172), it is assumed that $\chi(0)$ is small. The small angle assumption results in Eq. (173).

$$\psi(0) = \arcsin\left(-\frac{W_x - W_y \sin \chi(0)}{V_T}\right) + \chi(0) \quad (172)$$

$$\psi(0) = \left(-\frac{W_x - W_y \chi(0)}{V_T}\right) + \chi(0) \quad (173)$$

Finally, the heading approximation, $\psi_{ap}(t)$, is defined by Eq. (174) and Eq. (175). The resulting equations are analytic but complex. The equations themselves were calculated utilizing symbolic math functions in MATLAB[®]. The code to derive these equations is provided in Appendix A.

$$\psi_{ap}(t \leq t_1) = \int_0^t \dot{\psi}_{ap}(t) dt \quad (174)$$

$$\psi_{ap}(t > t_1) = \int_0^{t_1} \dot{\psi}_{ap}(t) dt + \int_{t_1}^t \dot{\psi}_{ap}(t) dt \quad (175)$$

Approximating Position as a Function of Time.

Next, the approximated heading is integrated through the position dynamics; Eq. (176) and Eq. (177). The integral again does not have an analytic solution. However, only a small amount of heading change is expected over the course of the

maneuver.

$$\dot{x}(t) = V_T \sin \psi(t) + W_x \quad (176)$$

$$\dot{y}(t) = V_T \cos \psi(t) + W_y \quad (177)$$

To show that the change in heading is small, take an aircraft at an altitude of 5,000 ft above mean sea level traveling at a calibrated airspeed of 180 knots (327 ft/s true airspeed); similar conditions to a landing approach for large aircraft. Assume the aircraft has a roll mode time constant of 1.4 and a maximum roll rate of 30 *deg/s*.

A minimum time-to-bank maneuver, described in Sec. 4.3, from wings level to a bank angle of 30 degrees would take 2.51s. Numerically integrating the bank angle through the equations of motion results in heading change of 3.7 degrees; a small angle by most accounts.

Due to this small observed heading change, a simple small angle approximation will be made to approximate the trigonometric functions in \dot{x} and \dot{y} as shown in Eq. (178) and Eq. (179); the subscript *ap* refers to the approximation. The position rate approximations are subsequently integrated through the two time-segments similarly to the heading rate integration.

$$\dot{x}(t) \approx \dot{x}_{ap}(t) = V_T \psi_{ap}(t) + W_x \quad (178)$$

$$\dot{y}(t) \approx \dot{y}_{ap}(t) = V_T + W_y \quad (179)$$

Straight to Turning Segment Boundary Constraints and Solution.

The initial conditions for the straight to turning problem are applied, resulting in analytic solutions for the final position and heading as summarized by Eq. (180) through Eq. (182). The equations are complex, but analytic which allows for the

determination of an optimal y_0 . The MATLAB[®] code to determine the analytic equations is provided in the digital appendix.

$$x(t_f) \approx x_{ap}(t_f) = V_T \int_0^{t_1} \psi_{ap}(t) dt + V_T \int_{t_1}^{t_f} \psi_{ap}(t) dt + t_f W_x \quad (180)$$

$$y(t_f) \approx y_{ap}(t_f) = V_T t_f + y_0 + t_f W_y \quad (181)$$

$$\psi_{ap}(t_f) = \int_0^{t_1} \dot{\psi}_{ap}(t) dt + \int_{t_1}^{t_f} \dot{\psi}_{ap}(t) dt \quad (182)$$

Taking the cost functional from the straight to turning segment problem definition, Eq. (119), a final small angle approximation is made which simplifies the equation resulting in Eq. (183). Estimating the final heading angle requires the definition of course angle reproduced in Eq. (184). Utilizing the approximations for the position dynamics derived in this section along with a small angle approximation on the arctangent results in an approximation for the final course angle, Eq. (185).

$$J = (x_{ap}(t_f))^2 + (y_{ap}(t_f) - R_T \chi(t_f))^2 \quad (183)$$

$$\chi(t) = \text{atan2}(\dot{x}(t), \dot{y}(t)) \quad (184)$$

$$\chi(t_f) \approx \frac{V_T \sin \psi_{ap}(t_f) + W_x}{V_T \cos \psi_{ap}(t_f) + W_y} \quad (185)$$

The cost functional is then analytically differentiated with respect to y_0 , set equal to zero, and solved for y_0 in accordance with classical optimization theory [75]. The MATLAB[®] code which generates the analytic definition of y_0 is provided in Appendix A.

Turning to Straight Segment Boundary Constraints and Solution.

The process of finding the optimal initial point for the turning to straight segment problem is similar to the straight to turning problem. In this case, different initial conditions are applied to the heading and position variables, based on the initial arc length variable s_0 , before performing a similar integration process as in the previous section. However, a small angle assumption must also be applied to the initial conditions to obtain an analytic solution. The original initial conditions for this problem are described by Eq. (186) and Eq. (187) and the original cost functional is Eq. (188).

$$x(0) = R_T - R_T \cos(R_T s_0) \quad (186)$$

$$y(0) = R_T \sin(R_T s_0) \quad (187)$$

$$J = x(t_f)^2 + R_T \chi(t_f)^2 \quad (188)$$

To obtain an analytic solution, a small angle approximation is applied to the trigonometric functions in the initial conditions for position. This aids the solution by “removing” the design variable, s_0 , from inside trigonometric functions. Additionally, $\chi(t_f)$ is expanded as the terminal heading plus the crab angle, which is defined by Eq. (171). The simplified initial conditions and cost functional for the turning to straight segment problem are now given by Eq. (189) through Eq. (191).

$$x(0) \approx 0 \quad (189)$$

$$y(0) \approx R_T s_0 \quad (190)$$

$$J \approx (x_{ap}(t_f))^2 + \left(\psi_{ap}(t_f) - \arcsin -\frac{W_x}{V_T} \right)^2 \quad (191)$$

The initial conditions are applied to the integrated position dynamics to determine the final positions and heading. The resulting cost functional is again differentiated, but with respect to s_0 , set equal to zero, and the equation solved for s_0 . Again, the equation for the optimal s_0 is complex. The MATLAB[®] code which generates the analytic definition of s_0 is provided in Appendix A.

4.4 Near-Optimality of the Solution

The solution derived in the preceding sections will be referred to as the Taylor Polynomial Approximation Solution (TPAS) in this section. The near optimality of the TPAS across a wide range of scenarios is determined by comparing the tracking error of the TPAS, a numerically calculated optimal solution, and a control solution. In this analysis, the original cost functional of the optimal lead turn problem is used; reproduced in Eqs. (192)-(194).

$$J = \sqrt{\int_0^{t_f} e(t)^2 dt} \quad (192)$$

$$e(t) = x(t), \text{ if on straight segment} \quad (193)$$

$$e(t) = R_T - \sqrt{(x(t) - R_T)^2 + y(t)^2}, \text{ if on circular segment} \quad (194)$$

The control solution is defined as a minimum time-to-bank maneuver performed at the segment transition point, $y_0 = 0$. The optimal solution is optimal with regards to the original lead turn problem which enforces all terminal constraints and is numerically calculated using GPOPS-II [34]. The cost is evaluated over the minimum time-to-bank interval to ensure comparable results among the solutions.

The optimality metric is defined to be the difference between the TPAS cost and

the optimal cost, divided by the control cost for each scenario.

$$\text{Optimality Metric} = \frac{J_{TPAS} - J_{optimal}}{J_{control}} \quad (195)$$

This metric allows for the difference in control scenario cost to be accounted for across a wide range of scenario parameters. The scenarios varied the following parameters:

- Airspeed - 200 *ft/s* to 800 *ft/s*
- Roll mode time constant - 0.5 *s* to 2.0 *s*
- Maximum roll rate - 15 *deg/s* to 180 *deg/s*
- Bank Angle During Turn - 20 *deg* to 60 *deg*
- Wind Speed - 0% to 20% of airspeed
- Wind Direction - Ordinal directions (e.g., North-East, South-East,...)
- Path Segment Order (e.g., Straight-to-Turn or Turn-to-Straight)

Finally, each parameter was evaluated at three separate levels except for wind direction. The result is 1,458 different scenarios which were calculated and evaluated.

The results are summarized in Table 5 with the minimum and maximum metric for each segment order along with the mean and two standard deviation bounds. Although the negative metric values indicate the TPAS had a lower cost than some GPOPS-II solutions, the TPAS does not satisfy all the constraints of the original problem. Overall, the TPAS cost is within 13.3% of the GPOPS-II solution for every scenario and within 7.4% for at least 95% of the simulated scenarios. This suggests that the TPAS is a near-optimal approximation of the optimal GPOPS-II solution for the parameters studied.

Table 5. Taylor Polynomial Approximation Solution (TPAS) Optimality Metric

Segment Order	Min	Mean $\pm 2\sigma$	Max
Straight-to-Turn	-0.6%	3.7% \pm 3.4%	8.0%
Turn-to-Straight	-5.9%	2.7% \pm 7.4%	13.3%

The primary benefit of the TPAS is that, although complicated, it takes significantly less time to compute. The average computation time of all the GPOPS-II solutions was 0.50 *s* while the average computation time of the TPAS was 0.001 *s*. The quick calculation time of the TPAS allows for the potential to update the near-optimal lead turn distance in real time, the original motivation for determining an analytic solution.

4.5 Lead Turn with Existing Guidance Laws

As the TPAS was shown to be near-optimal and efficient to compute, the next step is to determine if the solution may be integrated into existing guidance laws and still result in an improvement to tracking performance. In the following section, the lead turn distance refers to the initial position calculated by the TPAS.

Guidance Law Integration.

The guidance laws selected for evaluation are a nonlinear guidance law from Ref. [3] (referred to as NGL), a vector field follower from Ref. [76] (referred to as VFF), a non-linear dynamic inversion controller from Ref. [12] (referred to as NDI), and a synthetic waypoint guidance controller from Ref. [77] (referred to as SWG). The different controllers are chosen for their varying levels of complexity and performance.

The SWG controller is a type of “carrot-chasing” algorithm which is one of the simplest types of guidance laws to implement because it simply drives the aircraft to point towards an imaginary point moving along the desired path. However, “carrot-chasing” algorithms typically exhibit poor steady-state path tracking performance

[77].

Nonlinear controllers tend to involve complicated calculations derived from an analysis of the path tracking problem. The path tracking performance of nonlinear controllers tends to be better than “carrot-chasing” algorithms with the additional benefit of a single controller mode for any type of path. Finally, vector field controllers derive commands from a vector field around the desired path. Thus, a vector field controller will have zero steady state error around its design path; lines and arcs are used in this work. However, this requires the controller to switch between modes when the path changes from a line to a circle and vice versa [78].

The output of each controller is transformed to a bank angle command which drives an LQR bank-angle hold autopilot providing the roll rate control input, u , into the dynamics model [6]. The VFF and SWG guidance laws also require an intermediate course hold autopilot between the guidance law and bank-angle hold autopilot [79]. The design of this autopilot is discussed in the following section.

The aircraft model used in the testing is the two-dimensional simplified model discussed in Sec. 3.2 and controlled by the LQR bank angle command system described in Sec. 3.2. A condensed summary of the aircraft type parameters defined in Table 1 is reproduced for reference in Table 6. Next, a short description of each guidance law is given, which focuses on the calculation of the bank angle command input into the simplified model.

Table 6. Aircraft Class Parameters; reproduced from Table 1

Aircraft Class	V_{ref} (ft/s)	τ_P (s)
Light Utility	293	1.0
Medium Cargo	390	1.4
Heavy Cargo	544	1.4
Fighter	864	1.0

Nonlinear Guidance Law Integration.

The NGL is a single mode controller which uses a specified look-ahead distance to generate a lateral acceleration command [3]. The look-ahead distance (L_D) variable is the primary design parameter for this control law. Note, the look-ahead distance is defined as L_1 in [3] instead of L_D . The nomenclature was changed in this work due to L_1 being used in another section.

Equation (197) is used to determine the value of L_D based on the results presented in Ref. [3]. The lateral acceleration command (a_s) is then calculated according to Eq. (196) where η is the angle between the aircraft trajectory and the reference point as depicted in Fig. 17. Note that the lateral acceleration command is calculated using ground speed, V_g . The lateral acceleration command is then translated to the bank angle command input to the simplified model using Eq. (198). Equation (198) may be derived from the level turn equations [69].

$$a_s = 2 \frac{G^2}{L_D} \sin \eta \quad (196)$$

$$L_D = \frac{V_{ref} \sqrt{2}}{0.23} \quad (197)$$

$$\phi_{cmd} = \text{sgn}(a_s) \arccos \frac{g}{\sqrt{a_s^2 + g^2}} \quad (198)$$

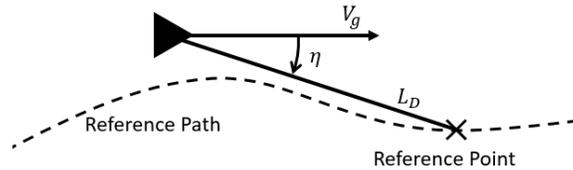


Figure 17. Nonlinear Guidance Law Parameters; Ref. [1]

Vector-Field Follower.

The VFF is a multi-mode controller which uses a vector field to determine course angle (χ) commands which are transformed to bank angle commands via a PI controller [76]. The vector field controller creates a vector field of course angle commands around the desired path as a function of tracking error and course error. The commands are computed with Eq. (199) for a straight path and Eq. (201) for a circular path.

$$\chi_{cmd_{line}} = \Delta\chi - \frac{2\chi_{far}}{\pi} \arctan(k_{line}e) \quad (199)$$

$$k_{line} = \frac{1}{k_{div}V_{ref}} \quad (200)$$

$$\chi_{cmd_{orbit}} = \Delta\chi + D_T \left(\chi_{far} + \arctan\left(\frac{k_{orbit}e}{R_T}\right) \right) \quad (201)$$

In these equations, $\Delta\chi$ is the course angle error, χ_{far} is the desired course command when the aircraft is far from the desired path, e is the path tracking error, k_{line} is the controller gain while following a straight path, and k_{orbit} is the controller gain while following a circular path. In this case, the design variables are k_{div} , χ_{far} , and k_{orbit} . The values of the design variables for the χ -command equations are given in Table 7. Further details of the vector field controller may be found in Ref. [76].

Table 7. Vector Field Follower Parameters

Aircraft Class	k_{div}	k_{orbit}	χ_{far} (rad)
Light Utility	6	4	$\pi/2$
Medium Cargo	4	4	$\pi/2$
Heavy Cargo	9	4	$\pi/2$
Fighter	4	4	$\pi/2$

The course command to bank angle command controller is a simple PI controller as shown in Fig. 18. The proportional and integral gains, k_P and k_I respectively, are

given by Eq. (202) through Eq. (204) with W_χ and ζ_χ as the design parameters; the controller is based on a course-hold autopilot design in Ref. [79].

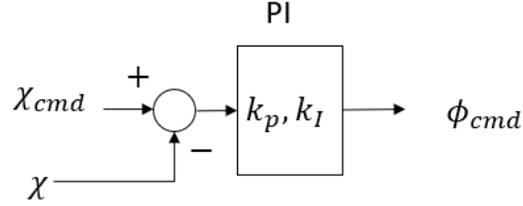


Figure 18. Course Command to Bank Angle Command

$$k_P = \frac{2}{g} V_T \zeta_\chi \omega_{n_\chi} \quad (202)$$

$$k_I = \frac{V_T}{g} \omega_{n_\chi}^2 \quad (203)$$

$$\omega_{n_\chi} = \frac{\tau_P}{W_\chi} \quad (204)$$

The parameter ζ_χ represents the desired damping ratio of the course angle follower while W_χ is a multiple used to separate the natural frequency of the course follower from the natural frequency of the roll mode (τ_P). Finally, the gain k_C adjusts the gain of the controller during a turn. The parameter values used for each aircraft type are presented in Table 8.

Table 8. Vector Field Follower Course Hold Parameters

Aircraft Class	W_χ	ζ_χ	k_T
Light Utility	5	1.0	0.8
Medium Cargo	8	1.0	0.7
Heavy Cargo	10	0.8	0.8
Fighter	6	0.8	1

Nonlinear Dynamic Inversion Controller.

The NDI controller [12] calculates a desired course rate of change based on Eq. (205) where κ is the curvature of the path, ζ_{ndi} is the desired damping ratio, and $\omega_{n_{ndi}}$ is the desired natural frequency. The variables ζ_{ndi} and $\omega_{n_{ndi}}$ are the design parameters for this controller. The design parameter values for all aircraft types is presented in Table 9.

$$\dot{\chi}_{cmd} = \frac{\kappa G^2 \cos(\Delta\chi)^2 / (1 - \kappa e_{path}) - 2\zeta_{ndi}\omega_{n_{ndi}}G \sin(\Delta\chi) - \omega_{n_{ndi}}^2 e_{path}}{\cos(\Delta\chi)G} \quad (205)$$

Table 9. Nonlinear Dynamic Inversion Follower Parameters, all aircraft

Parameter	Value
$\omega_{n_{ndi}}$	0.35 rad/s
ζ_{ndi}	0.8

The values were chosen to provide a balance between stability and path following performance. Since the course angle rate of change is equivalent to the heading rate-of-change, the heading-rate equation of motion may be used to calculate the corresponding bank angle command using Eq. (206); note that ground speed is used when utilizing the course angle rate of change [79].

$$\phi_{cmd} = \arctan \frac{V_g \dot{\chi}_{cmd}}{g} \quad (206)$$

Synthetic Waypoint Guidance.

Finally, the SWG controller guides the aircraft to follow a moving reference point which is a specific distance from the vehicle. The layout of the parameters is the same as the NGL controller and Fig. 17 applies; the variable L_D will again be used

for simplicity. However, the SWG controller is a simple “carrot-chasing” guidance law where the angle η directly relates to the commanded course as described by Eq. (207).

$$\chi_{cmd} = \eta + \chi \quad (207)$$

The χ_{cmd} drives the same course-hold guidance law used by the VFF controller, Fig. 18, but with gains tuned for the SWG controller. The course-command gains used with the SWG controller are given in Table 10 and the SWG controller uses a look-ahead distance (L_D) of $5V_{ref}$ ahead of the aircraft which represents a look-ahead time of 5 seconds.

Table 10. Synthetic Waypoint Guidance Course Hold Parameters

Aircraft Class	W_χ	ζ_χ	k_T
Light Utility	7	1.0	0.8
Medium Cargo	12	1.0	0.7
Heavy Cargo	12	1.2	0.8
Fighter	10	1.0	0.8

Finally, it’s noted that “carrot-chasing” guidance laws inherently have a non-zero steady-state error that will occur when following a circular path with a positive look-ahead distance [78]. The error develops because this type of guidance law drives the line-of-sight angle to the look-ahead point, η , to zero. Analyzing the geometry of the path in Fig. 19 assuming the aircraft has stabilized in an orbit around the center of the desired path, a constant heading rate-of-change. Driving η to zero under these assumptions requires the aircraft to fly an orbit with a turn radius smaller than the desired path resulting in a steady-state error defined by Eq. (208).

$$\text{SWG Steady-State Turning Error} = R_T - \sqrt{R_T^2 - L_D^2} \quad (208)$$

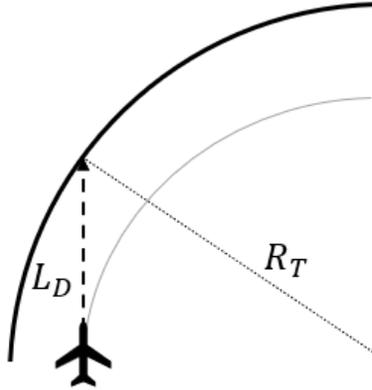


Figure 19. Synthetic Waypoint Guidance Circular Orbit Error

Guidance Law Scenarios.

Three different cases are used to evaluate the lead turn when integrated with existing guidance laws: a control, a lead turn without the optimal roll maneuver, and a lead turn with the optimal roll maneuver. In a control scenario, the guidance laws tracks the desired path, Fig. 13, without change to their original design.

The “Lead” case affects the reference point the guidance law uses to calculate tracking error and generate commands. Prior to the lead turn point, the aircraft tracks the first path segment then switches to the second segment. The “Optimal” scenario behaves similarly to the “Lead” scenario except that a minimum time-to-bank maneuver is performed at the lead turn distance. This method ignores the guidance law commands at the lead turn distance and provides a minimum time-to-bank roll rate command to the dynamics. Following the completion of the maneuver, the controller is reinitialized and tracks the second path segment.

An example of the look-ahead point behavior during “Lead” and “Optimal” scenarios is presented in Fig. 20 for the straight-to-turn scenario. Figure 20 shows the “look-ahead” point, a black star, for the aircraft both before and after the lead turn distance, y_0 .

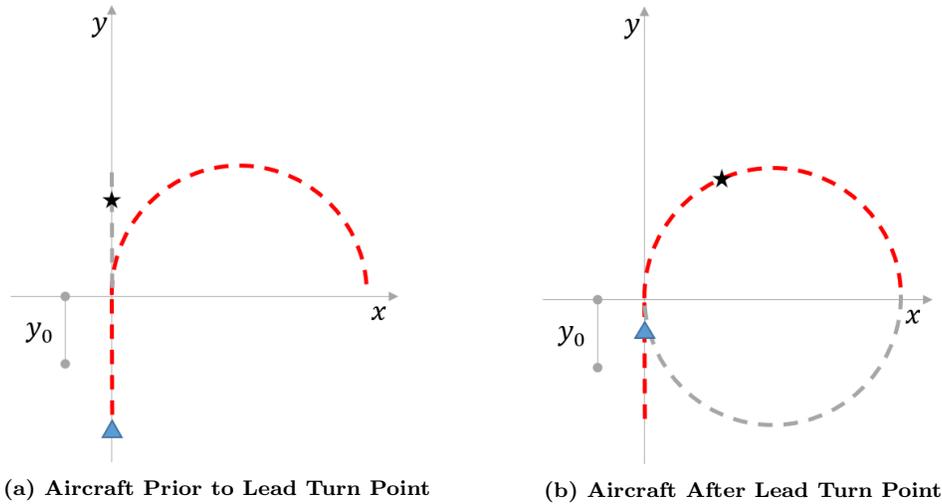


Figure 20. “Lead/Optimal” Scenario Example, Look-ahead Point Identified as Star

Each integration case is then evaluated over a range of parameters: wind, turbulence, aircraft performance, and path segment order. The steady-state wind speed was simulated at 0%, 10%, and 20% of the aircraft airspeed in the four ordinal directions (e.g., northwest, southeast, etc.). Three levels of wind turbulence were tested (none, light, moderate) using the MATLAB[®] Dryden Wind Turbulence Model with 25 simulations per scenario to account for the randomness of turbulence. Next, both path segment orders were tested. Finally, four distinct classes of aircraft were simulated with airspeeds from 293 *ft/s* to 864 *ft/s*; light utility, medium cargo, heavy cargo, and fighter [70]. The simulation setup resulted in a total of 10,800 simulations per guidance law.

The steady state wind speed was simulated at 0%, 10%, and 20% of the aircraft airspeed in the four ordinal directions (e.g., northwest, southeast, etc.). Three levels of wind turbulence were tested (none, light, moderate) using the MATLAB[®] Dryden Wind Turbulence Model. Since the turbulence is generated with random white noise, 25 simulations were performed for each light and moderate scenario. Finally, the four distinct classes of aircraft were simulated with each guidance law simulating airspeeds

from 293 *ft/s* to 864 *ft/s*; refer to Table 1. The simulation setup resulted in a total of 10,800 simulations for each guidance law.

4.6 Guidance Law Integration Performance Analysis

Comparison Metric.

The comparison metric is chosen to be defined by Eq. (209); similar to the original lead turn problem, Eq. (118).

$$ComparisonMetric = \sqrt{\int_{t_0}^{t_f} e(t)^2 dt} \quad (209)$$

The integral is evaluated over the time interval of eight seconds before the calculated lead turn point to twenty seconds after the lead turn point. The interval captures the look-ahead distance for each of the guidance laws as well as the resulting transient behavior. The lead turn point remains the same for each scenario which allows the intervals to remain the same for the three different scenarios for a given controller, aircraft, and wind condition. The metric is then compared as a percent difference among the scenarios to determine the relative effect of the lead turn since each controller has different baseline, or “control,” performance.

Results.

Figure 21 presents a sample scenario with the NGL path follower to highlight the behavior of the different integration cases and the comparison metric. The scenario is a light utility aircraft traveling at 293 *ft/s* with light turbulence, 59 *ft/s* wind speed, and a wind direction of southeast. The metric interval indicates the time interval over which the performance metric, the 2-norm, was evaluated. The “Roll Interval” is the time interval during the “Optimal” case when the optimal roll maneuver is being

performed.

The trajectories are initially identical since the same turbulence is used for all three cases. Since the NGL controller incorporates a “look-ahead” distance, the “Control” case trajectory diverges from the other two trajectories when the “look-ahead” point moves from the straight path to the circular path; this occurs approximately at the “look-ahead” point annotated on the figure. The “Lead” and “Optimal” trajectories remain the same until the lead turn distance, indicated by the start of the “Roll Interval.” The comparison metric for the “Control”, “Lead”, and “Optimal” trajectories in this scenario are 309 *ft*, 287 *ft*, and 110 *ft* respectively.

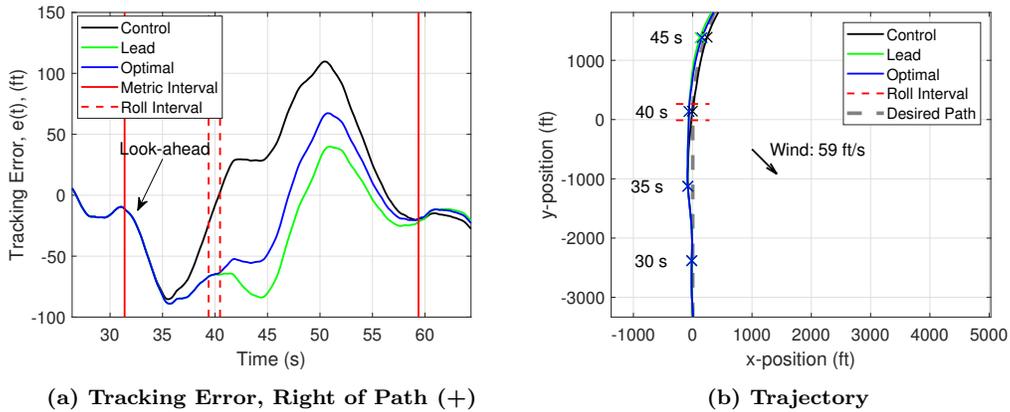


Figure 21. Nonlinear Guidance Law (NGL) Example: Light Utility Aircraft, Light Turbulence, with Wind

Figure 22 presents the percent difference in the comparison metric of every “Lead” and “Optimal” scenario compared to the corresponding “Control” scenario. The data are split up by guidance law and then subdivided into the three turbulence levels. In each subdivision, all data points are colored gray while the average of the “Lead” and “Optimal” cases are presented as a black circle and “x” respectively. Note that the two means are nearly identical for the light and moderate turbulence cases.

The primary result is that the “Lead” trajectories show an average 34% decrease in the comparison metric from the “Control” scenario across all simulations. The

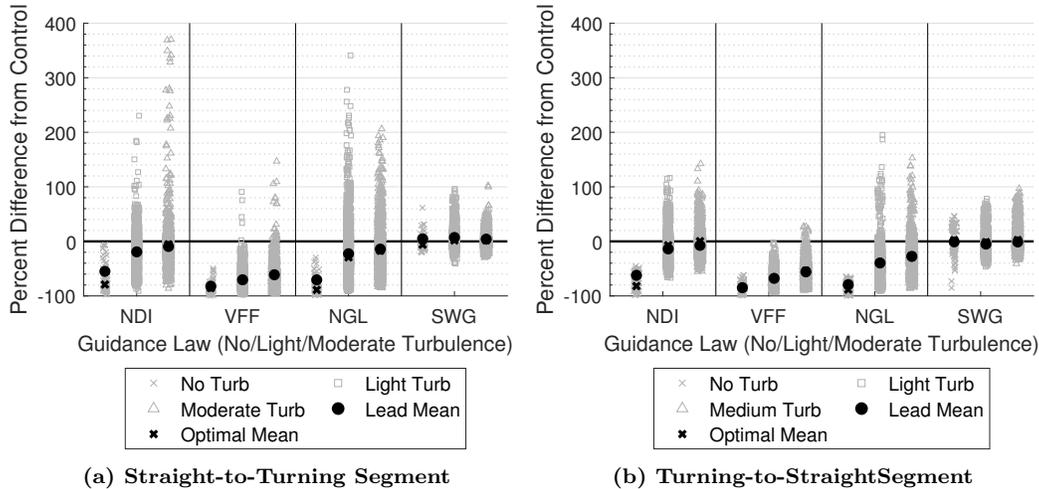


Figure 22. ℓ^2 -norm Percent Difference from Control Scenario

“Optimal” trajectories only improved this performance to a 37% decrease in the comparison metric from the “Control” cases. The results also indicate that the magnitude of the average decrease in cost is heavily dependent on the level of turbulence. This is shown in Fig. 22 by the “Lead Mean” points which are seen to increase as turbulence increases for each controller except the SWG controller. Additionally, the optimal maneuver only shows a significant benefit to the NGL and the NDI controllers in scenarios without turbulence while all other scenarios show a similar 2-norm for the “Lead” and “Optimal” solutions.

It is hypothesized that the SWG controller does not benefit from the lead turn because the design of the SWG controller and the course command controller introduces a steady-state tracking error when following a circular path due to the “pure pursuit” nature of the controller. The resulting steady-state error violates the terminal constraint assumption that the aircraft be on and tangent to the path, which diminishes the benefit of the lead turn.

This behavior is highlighted in Fig. 23. Near the lead turn point, the “Lead” and “Optimal” scenarios do maintain a lower error. However, once the SWG controller reactivates, the error is driven towards the steady-state turn value. While a different

metric interval would likely yield different results, the interval was chosen to capture the time when the trajectories are significantly different to capture the effect of the lead turn.

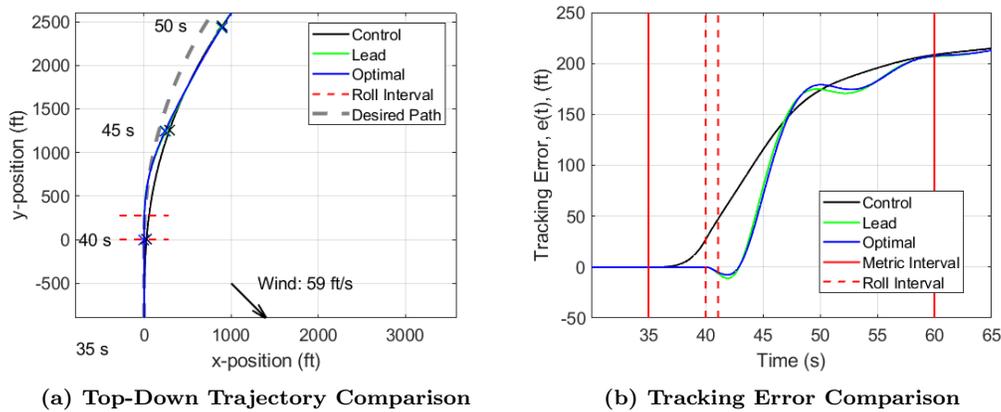


Figure 23. Synthetic Waypoint Guidance Error Comparison

4.7 Near-Optimal Lead Turn Conclusions

This section developed and analyzed a method designed to improve the Dubins path tracking performance of an aircraft by deriving a near-optimal lead turn point and near-optimal maneuver in a windy environment which could be calculated in near real time. The maneuver was then integrated into existing path followers to determine the benefit of the maneuver in a more realistic scenario.

The results show that simply transitioning from one path segment to the next at the computed lead turn point improves tracking performance for three of the four tested controllers on average. However, it was seen that the assumption of zero terminal error in the derivation of the lead turn distance reduced the benefit of the lead turn and reduced the tracking performance in scenarios where the assumption was violated.

The near-optimal roll maneuver further increases performance in simulations with zero turbulence but has a negligible impact, on average, for scenarios with turbulence.

The results demonstrate the use of a simple calculation which may improve the Dubins path tracking performance of an aircraft which is controlled by a wide variety of guidance laws. This maneuver will be utilized in the arrival time algorithm to improve path tracking performance during transitions between straight and circular path segments.

V. Racetrack Algorithm Development

5.1 Introduction

The racetrack algorithm is the primary control algorithm designed to enable a group of aircraft to achieve coordinated arrival time with a real time control algorithm. First, the general principles of how the racetrack algorithm affects arrival time is presented. This is followed by the introduction of the racetrack reference frame which is the foundation for the arrival time estimate calculations. Next, two methods that accurately calculate the time to travel along the racetrack are introduced. The two methods will be compared in the results section and only one method used extensively in simulation.

Then, a short description of the implemented model predictive control path follower is given. Finally, the upper-level racetrack manager is described. The manager enforces keep-out zones and determines which portion of the racetrack the aircraft should be following.

5.2 Primary Controls

To develop an algorithm which can achieve coordinated arrival time in a windy environment which may have keep-out zones, two primary control effectors are chosen: path length and airspeed. First, a path whose definition may be changed by a single variable is assumed, a racetrack. Adjusting the length of the racetrack, as shown in Fig. 24, allows for rapid control over total path length which is able to quickly respond to changes in the environment and scenario constraints. The solution reduces the complexity of the problem by decoupling the path-following controller and the arrival-time controller.

The algorithm is designed so that the airspeed control is always active, but be-

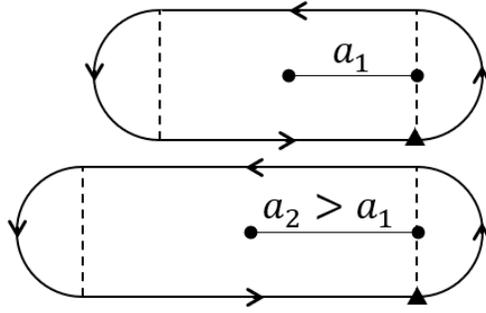


Figure 24. Effect of Adjusting a on the Racetrack Shape

comes the primary arrival time controller when the path length reaches a limit. This limit could be either a minimum limit (a circular orbit) or a maximum limit due to an external keep-out zone. A virtual target-like control is used, which allows for both arrival time and arrival airspeed constraints to be met.

5.3 Racetrack Reference Frame

The path to be followed, a racetrack, is depicted in Fig. 24. To standardize the control algorithm calculations for various turn directions, headings, and final positions, a left-hand racetrack reference frame is introduced in Fig. 25; referred to as the racetrack reference frame.

Figure 25 also presents the variables used in the racetrack reference frame. The bottom right corner of the racetrack is assumed to be fixed to the desired final position for the trajectory; the point $[x_f, y_f]$. Each of the four path segments, two straight lines and two circular arcs, are in four separate areas referred to as quadrants (Q); as an example, quadrant one is labeled as Q_I . The racetrack is oriented such that the final segment in Q_{IV} is in the direction of the final desired ground course in the earth-fixed reference frame, χ_f .

The parameters a and b are used to determine the size of the racetrack. The parameter a defines the distance from the center of the racetrack to the center of

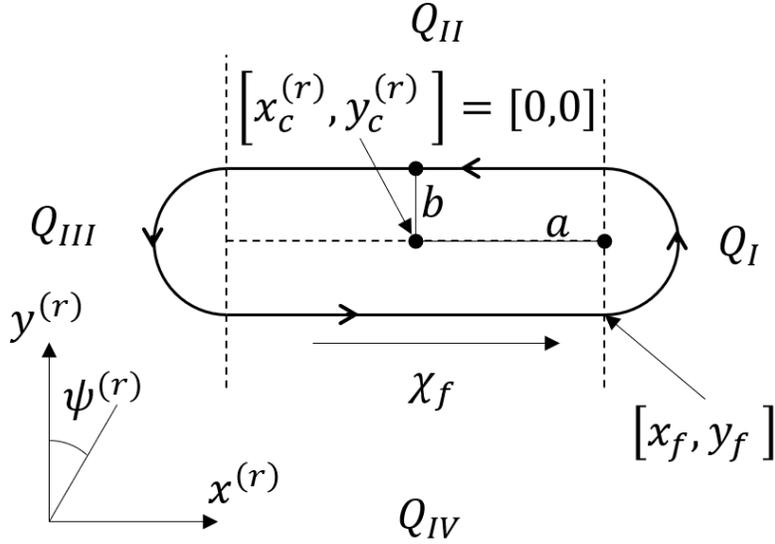


Figure 25. Left-Hand Racetrack Definition

either turn circle such that the length of the straight section in quadrants two and four is $2a$. The parameter b defines the turn radius for the turns on either end of the racetrack. The parameter a will be controlled to meet the coordinated timing requirements while b remains constant.

Figure 26 presents the relationship between the earth-fixed axes, $\{x^{(e)}, y^{(e)}\}$ and the racetrack reference frame axes, $\{x^{(r)}, y^{(r)}\}$, in the case of a right-hand turning racetrack. Note, the racetrack axes appear to flip in the earth-fixed reference frame between Fig. 26 and Fig. 25 because the racetrack reference frame is defined as a counterclockwise racetrack. The origin of the earth-fixed reference frame is arbitrary while the origin of the racetrack reference frame is defined as the center of the racetrack, $[x_c, y_c]$.

To transform variables from the earth-fixed reference frame to the racetrack reference frame, the vehicle and environment state variables must be rotated and flipped into the left-hand racetrack reference frame. A turn direction variable, D_T , is introduced to standardize calculations between counterclockwise and clockwise racetracks. The variable is defined as $D_T = 1$ for counterclockwise racetracks and $D_T = -1$ for

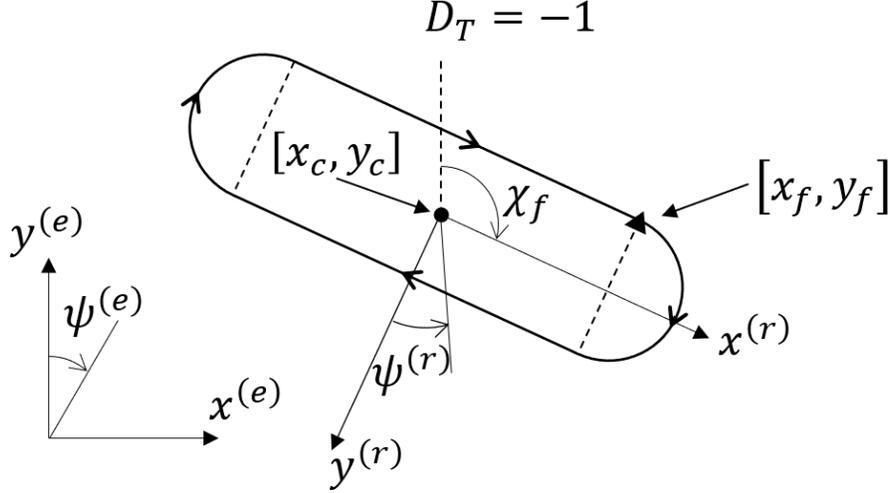


Figure 26. Left-Hand Racetrack Definition

clockwise racetracks. In summary, the entire path for a racetrack is defined by the final position, final course and the variables a , b , and D_T .

The rotation matrix \mathbf{M} , Eq. (210), projects parameters from a reference frame X onto another reference frame Y [6]. If frame Y has been rotated clockwise from frame X , the projection of a point \mathbf{x} on frame X onto the reference frame Y is $\mathbf{x}_Y = \mathbf{M}\mathbf{x}$.

$$\mathbf{M}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (210)$$

When applied to the problem of projecting parameters from the earth-fixed reference frame to the racetrack reference frame, the rotation matrix alone will not suffice because the direction of $y^{(r)}$, with respect to the earth-fixed reference frame, also depends on turn direction. The previously defined turn direction variable, D_T , is used to account for the change in turn direction.

First, the translation of windspeed between the earth-fixed and racetrack reference frame is discussed. The earth-fixed reference frame wind speeds are projected into the racetrack reference frame via Eq. (211), noting the use of the variable D_T to account

for turn direction. The relationship of the earth-fixed and racetrack axes is depicted in Fig. 27. The rotation is through an angle of $\chi_f - \pi/2$ since the earth-fixed heading is measured clockwise positive from the y-axis, “North.”

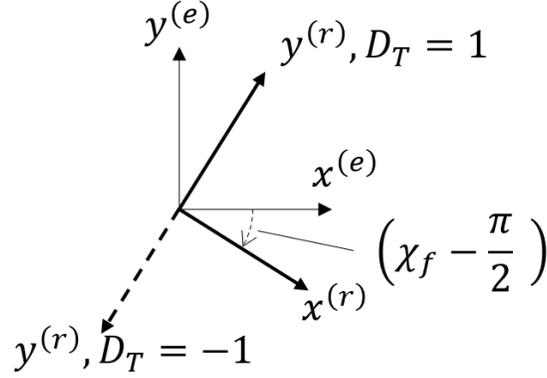


Figure 27. Reference Frame Rotation

$$\begin{bmatrix} W_x^{(r)} \\ W_y^{(r)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & D_T \end{bmatrix} \mathbf{M} \left(\chi_f - \frac{\pi}{2} \right) \begin{bmatrix} W_{x/e} \\ W_{y/e} \end{bmatrix} \quad (211)$$

Next, the center of the racetrack with respect to the earth-fixed reference frame, $[x_c^{(e)}, y_c^{(e)}]$, is determined to compute the aircraft position relative to the center of the racetrack. The calculation is performed with Eq. (212) where the rotation matrix is now applied to parameters in the racetrack reference frame which requires the transpose of the rotation matrix. The calculation of the center of the racetrack accounts for a left or right turn by multiplying b by D_T in Eq. (212).

$$\begin{bmatrix} x_c^{(e)} \\ y_c^{(e)} \end{bmatrix} = \mathbf{M}^T \left(\chi_f - \frac{\pi}{2} \right) \begin{bmatrix} -a \\ D_T b \end{bmatrix} + \begin{bmatrix} x_{f/e} \\ y_{f/e} \end{bmatrix} \quad (212)$$

Finally, the position of the aircraft in the racetrack reference frame is calculated

using Eq. (213).

$$\begin{bmatrix} x^{(r)} \\ y^{(r)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & D_T \end{bmatrix} \mathbf{M} \left(\chi_f - \frac{\pi}{2} \right) \begin{bmatrix} x^{(e)} - x_c^{(e)} \\ y^{(e)} - y_c^{(e)} \end{bmatrix} \quad (213)$$

5.4 Arrival Time Estimation

The most important aspect of the time-coordination problem is to accurately estimate the arrival time of the aircraft. It is assumed that the earth-fixed position of the aircraft is accurately measured along with true airspeed, heading, bank angle, and roll rate. The wind speed measurement is modeled by filtering true wind speed, including turbulence, through a low pass filter to provide an estimate of the current steady-state wind speed; described in Sec. 3.6.

The time-estimate calculations follow a simple formula; distance remaining divided by groundspeed. Two primary assumptions are made in these calculations. First, it is assumed that the current measured wind speed remains constant for the remainder of an orbit. Second, the calculations assume that the aircraft travels at the desired airspeed, V_{ref} , and corresponding groundspeed for the remainder of the orbit. This assumption will be shown to aid in simultaneously achieving a desired arrival time and arrival airspeed; see Sec. 5.5. The racetrack path is comprised of two primary path shapes, a circle and a line, which allows the time estimates to be separated by shape.

Straight Section Time Estimation.

In accordance with the general arrival time formula, the time to travel a straight section is simply the distance remaining in the segment divided by the nominal groundspeed, G_{ref} . The distance remaining is calculated as the distance from the endpoint of the segment to the point defined as the vehicle's position projected onto

the segment as shown in Fig. 28. This requires the assumption that the path is precisely followed.

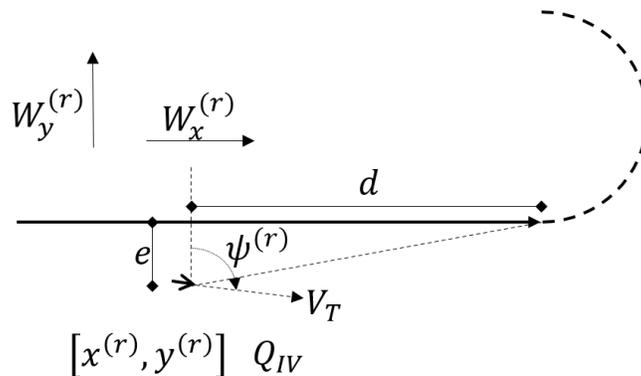


Figure 28. Straight Racetrack Section, Quadrant IV

The time remaining to complete a straight path segment in quadrant IV is approximated using the desired ground velocity according to Eq. (214) and Eq. (215). Eq. (214) is relatively simple because the straight segment is parallel to the race-track x-axis winds, $W_x^{(r)}$. Equation (215) is derived from the wind triangle discussed in Sec. 3.5. In this case, the desired course is $\pi/2$ and the crosswind component is $-W_y^{(r)}$ when calculating the time remaining for quadrant IV. In quadrant II, the desired course is $3\pi/2$ and the crosswind component is $W_y^{(r)}$.

$$T_{remQ_{IV}}(t) \approx \frac{d(t)}{G_{ref}} = \frac{d(t)}{V_{ref} \sin \psi^{(r)} + W_x^{(r)}} \quad (214)$$

$$\psi^{(r)} = \arcsin \left(\frac{W_y^{(r)}}{V_{ref}} \right) + \frac{\pi}{2} \quad (215)$$

Curved Section Time Estimation.

Estimating the time to travel a circular segment is more complicated than a straight segment because the wind direction relative to the path is constantly chang-

ing. As a result, the aircraft groundspeed also changes constantly. Two curved section time estimation techniques are initially investigated and described in this section. However, only a single method is chosen for integration into the algorithm; a comparison is made in Sec. 7.3. The first assumes a ground fixed path with constant turn radius is followed. The second assumes a constant bank angle turn between straight segments.

Gaussian Quadrature Arrival Time Estimation.

The first method assumes the aircraft follows a constant-radius, ground fixed path and utilizes a Gaussian quadrature to accurately estimate the arrival time. Again the general time estimate is calculated as distance divided by groundspeed. However, the groundspeed cannot be assumed to remain constant and requires different assumptions to accurately estimate an arrival time. Figure 29 presents a depiction of the curved section variables for quadrant I. For quadrant III, similar calculations are used by changing the sign for $W_x^{(r)}$ and $W_y^{(r)}$. The remaining calculations will be discussed for quadrant I only.

First, the distance remaining is determined by projecting the aircraft position onto the circular path and determining the angular amount of turn remaining, θ . The distance remaining, d , is calculated as the arc length associated with θ , $d = \theta b$.

Next, the groundspeed is calculated as the magnitude of the vector addition of the airspeed vector and the wind vector. Since the component wind speeds, $W_x^{(r)}$ and $W_y^{(r)}$, are already calculated by the control algorithm, the groundspeed at any angle, θ , may be calculated using Eq. (216).

$$G(\theta) = \left((V_{ref} \sin \psi^{(r)}(\theta) + W_x^{(r)})^2 + (V_{ref} \cos \psi^{(r)}(\theta) + W_y^{(r)})^2 \right)^{\frac{1}{2}} \quad (216)$$

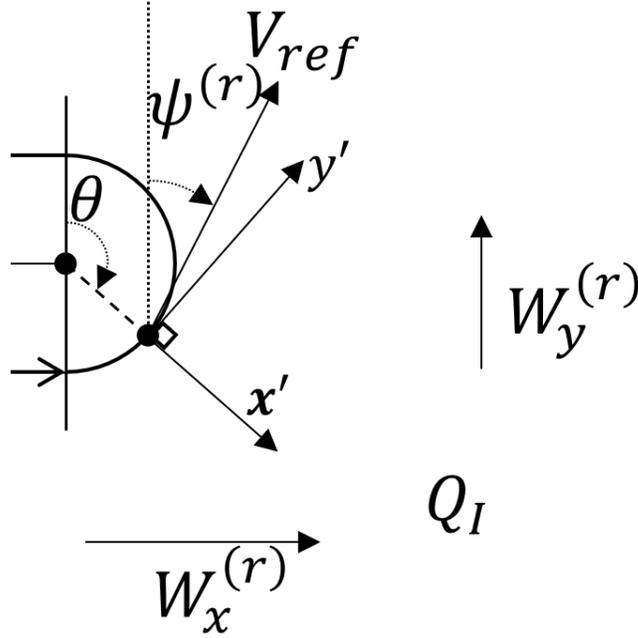


Figure 29. Curved Racetrack Section, Quadrant I

Since the path is assumed to be precisely followed, the heading required to achieve a ground course tangent to the circular path, $\psi^{(r)}(\theta)$, is determined by Eq. (217); derived from Sec. 3.5 assuming a ground course of $\theta - \pi/2$.

$$\psi^{(r)}(\theta) = \arcsin\left(-\frac{W_{cross}}{V_{ref}}\right) + \theta - \frac{\pi}{2} \quad (217)$$

Finally, the crosswind component of wind at any angle θ is calculated using Eq. (218) which projects the racetrack reference frame winds onto x' as shown in Fig. 29. The axes $\{x', y'\}$ are oriented such that y' is tangent to the circular path.

$$W_{cross}(\theta) = W_x^{(r)} \cos\left(\theta - \frac{\pi}{2}\right) - W_y^{(r)} \sin\left(\theta - \frac{\pi}{2}\right) \quad (218)$$

Thus, the magnitude of groundspeed for any angle θ may be calculated using Eq. (216) through Eq. (218).

However, the calculated groundspeed will change throughout the turn when the

wind is non-zero. A potential solution to achieve an arrival time estimate is to compute the time-weighted average ground velocity through the turn defined by Eq. (219).

$$G_{avg_t} = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} G(t) dt \quad (219)$$

However, the time-weighted average requires knowledge of the time remaining and groundspeed as a function of time which are unknown. An alternative is to calculate the angle-weighted integral as defined by Eq. (220).

$$G_{avg_\theta}(\theta) = \frac{1}{\theta} \int_0^\theta G(\theta) d\theta = \frac{1}{\theta} \int_0^\theta G(\theta) d\theta \quad (220)$$

A problem with the angle-weighted formulation is the angle-weighted integral is not equal to the time-weighted integral when winds are non-zero because $d\theta$ is not constant with respect to time. This may be shown by taking time derivative of the definition of arc length, s , and solving for $d\theta$,

$$\theta(t) = \frac{s(t)}{b} \quad (221)$$

$$d\theta(t) = \frac{G(t)}{b} dt. \quad (222)$$

Note that the direction of \dot{s} is tangent to the circle, by definition. Therefore the magnitude of \dot{s} is identical to the groundspeed, G . Since $d\theta$ depends on groundspeed, which changes over time, $d\theta$ must also change over time.

A solution is to approximate Eq. (220) as a weighted-average integral, Eq. (223), and utilize Eq. (222) to determine the integration weight, $h(\theta)$. Examining Eq. (222), the weighting function is chosen to be the inverse of groundspeed as a function of θ . The result is Eq. (224).

$$G_{avg_w}(\theta) = \frac{\int_0^\theta h(\theta)G(\theta) d\theta}{\int_0^\theta h(\theta) d\theta} \quad (223)$$

$$G_{avg_w}(\theta) = \frac{\theta - 0}{\int_0^\theta \frac{1}{G(\theta)} d\theta} \quad (224)$$

The integral in the denominator of Eq. (224) does not have a closed form solution but is continuous. A Legendre-Gauss quadrature is used to approximate Eq. (224) using Eq. (225) [80]. The nodes, θ_i , are obtained by applying an affine transformation to the roots of Legendre polynomials of some fixed degree, transforming them from the interval $[-1, +1]$ to $[0, \theta]$. The quadrature weights, w_i , are defined based on the degree of the polynomials; note the quadrature weights are different from the previously defined integration weight. The change in the numerator from Eq. (224) to Eq. (225) is due to the affine transformation.

$$G_{avg_w} \approx \frac{1 - (-1)}{\sum_{i=1}^N \frac{w_i}{G(\theta_i)}} \quad (225)$$

The accuracy of this estimation is demonstrated in Fig. 30. In this example, the aircraft airspeed is 300 *ft/s* with a 50 *ft/s* constant wind. The wind is oriented such that there is initially a 50 *ft/s* tail wind. Finally, the aircraft completes a 180-degree turn resulting in a 50 *ft/s* head wind at the final time.

First, the true groundspeed was numerically integrated to determine the true average groundspeed over the course of the turn. Next, each polynomial approximation is shown for comparison and the time equivalent location of the nodes, θ_i , are marked with an \times . The degree of the polynomial is n for each approximation p_n . Finally, the percent difference in the calculated average groundspeed is shown in the legend.

All of the investigated approximations were accurate to under 0.05 *ft/s*. However, the largest accuracy improvement appears to be from third to fourth degree

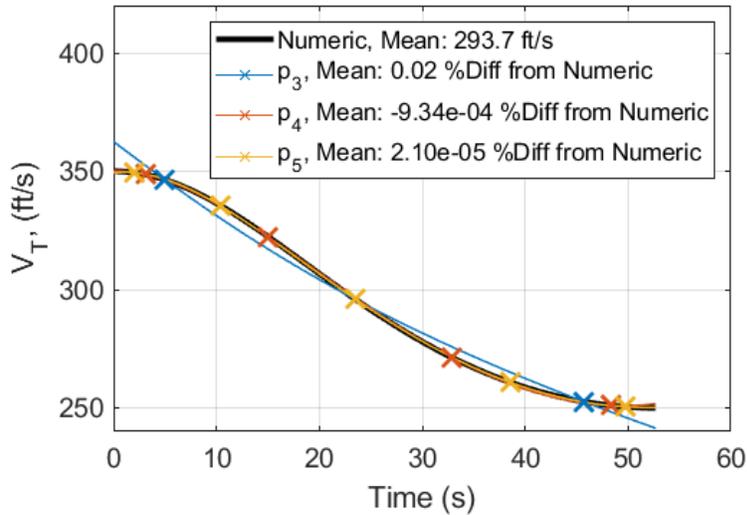


Figure 30. Average Airspeed Quadrature Estimate Comparison

polynomials and it was determined that the accuracy of fifth order polynomials was not needed. Therefore, fourth degree polynomials were chosen for implementation into the algorithm.

Finally, the time remaining in the quadrant is estimated by dividing the estimated distance remaining, $b\theta$, by the estimated average velocity as shown in Eq. (226) where b is the turn radius of the circular path segment.

$$T_{remQ_{1/III}} \approx \frac{b \theta}{G_{avg_w}} \quad (226)$$

Navigation Based Arrival Time Estimation.

The second curved-segment time-estimation method assumes a constant bank angle throughout the turn. In contrast to the Gaussian-quadrature method in the previous section, this method may not result in a circular path when the wind is non-zero. This is because a constant bank angle turn creates a constant radius turn through the moving air mass rather than a constant radius ground-fixed path.

Figure 31 presents an example of a moving turn circle. Assume that an aircraft

starts a turn at time t_0 and ends at time t_1 . At time t_1 the aircraft is required to be traveling in the opposite direction. The aircraft performs a constant bank angle turn resulting in a constant radius turn with turn radius R_T . In this example, assume that the wind is moving from left to right.

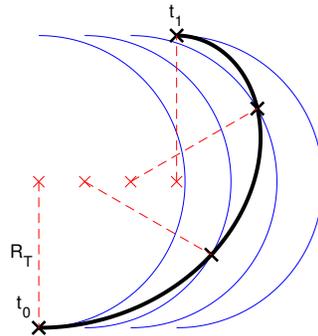


Figure 31. Constant Bank Angle Turn Ground Path

The turn circles at different points in time in the moving air mass are shown in blue with the center in each turn circle identified by a red \times . Each dashed red line is of constant length R_T . The aircraft will follow the turn circle as it moves from left to right resulting in the skewed shape of the aircraft path shown in black. Note, the distance between markers is greater at the beginning of the turn since the airspeed vector and wind-speed vector are in the same direction resulting in a higher groundspeed.

Since a constant bank angle turn follows a circular path through the moving air mass, the ground referenced position at the end of a constant bank angle turn may be calculated by adding the movement of the air mass to the final position of a turn in zero wind. The following calculations will be shown for a turn in Q_I .

Figure 32 presents a depiction of a turn in the moving air mass along with the variables used in the following calculation. Assume that the aircraft begins at an arbitrary heading, $\psi_0^{(r)}$, at an arbitrary position, $[x_0, y_0]$ and is traveling at constant

altitude and airspeed. Let the final position of the turn be defined as the point $[x_1, y_1]$. The final heading of the first turn, $\psi_1^{(r)}$, is the heading which will achieve a ground course of $-\pi/2$ in the presence of some steady-state wind. Finally, it is assumed that the aircraft will complete the turn at a single, negative bank angle, ϕ_1 , since the racetrack reference frame is a counterclockwise racetrack.

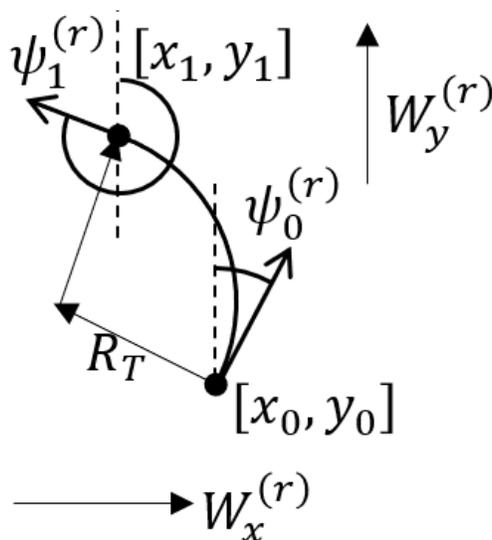


Figure 32. Air Mass Turn Calculations

First, the final position of the aircraft relative to the moving air mass, $[x_1^{(a)}, y_1^{(a)}]$, will be calculated by performing two vector sums defined by Eq. (227). The equation is derived by computing the vectors which relate the initial position of the aircraft to the center of the turn circle and the center of the turn circle to the final desired position. Note that the air mass has not moved at the initial position, $[x_0, y_0]$, and so the value of the initial position is the same relative to the racetrack as to the moving air mass.

$$\begin{bmatrix} x_1^{(a)} \\ y_1^{(a)} \end{bmatrix} = \begin{bmatrix} x_0^{(r)} \\ y_0^{(r)} \end{bmatrix} + R_T \begin{bmatrix} \sin(\psi_0^{(r)} - \frac{\pi}{2}) \\ \cos(\psi_0^{(r)} - \frac{\pi}{2}) \end{bmatrix} + R_T \begin{bmatrix} \sin(\psi_1^{(r)} + \frac{\pi}{2}) \\ \cos(\psi_1^{(r)} + \frac{\pi}{2}) \end{bmatrix} \quad (227)$$

Next, the final heading is defined by Eq. (228) based on the definition of the wind triangle from Eq. (100). In this case, $W_y^{(r)}$ replaces W_{cross} from Eq. (100) since it is perpendicular to the final ground course.

$$\psi_1^{(r)} = \arcsin\left(-\frac{W_y^{(r)}}{V_T}\right) - \frac{\pi}{2} \quad (228)$$

Finally, the turn radius, R_T , for an aircraft traveling at constant altitude, constant airspeed, and in a counterclockwise orbit is defined by Eq. (229) [69]. Note, the signum function is applied to the equation to ensure a positive turn radius value in lieu of a negative bank angle.

$$R_T = \text{sgn}(\phi) \frac{V_T}{\dot{\psi}} = \text{sgn}(\phi) \frac{V_T^2}{g \tan \phi_1} \quad (229)$$

The air mass relative position is then transformed to the racetrack reference frame by accounting for the distance that the air mass moves for the duration of the turn as defined by Eq. (230) where T_1 is the estimated time remaining to complete the turn in the first quadrant.

$$\begin{bmatrix} x_1^{(r)} \\ y_1^{(r)} \end{bmatrix} = \begin{bmatrix} x_1^{(a)} \\ y_1^{(a)} \end{bmatrix} + T_1 \begin{bmatrix} W_x^{(r)} \\ W_y^{(r)} \end{bmatrix} \quad (230)$$

Finally, Eq. (227) through Eq. (229) are substituted into Eq. (230) resulting in Eq. (231), the final position relative to the racetrack reference frame. Additionally, the incremental change in $x^{(r)}$ from the initial to the final position is defined as $x_{turn}^{(r)}$ by Eq. (232). The parameter, $\delta x_{turn}^{(r)}$ will be used in a subsequent section.

$$\begin{bmatrix} x_1^{(r)} \\ y_1^{(r)} \end{bmatrix} = \begin{bmatrix} x_0^{(r)} \\ y_0^{(r)} \end{bmatrix} - \frac{V_T^2}{g \tan \phi_1} \begin{bmatrix} \cos(\psi_0^{(r)}) \\ -\sin(\psi_0^{(r)}) \end{bmatrix} - \frac{V_T^2}{g \tan \phi_1} \begin{bmatrix} \cos(\psi_1^{(r)}) \\ \sin(\psi_1^{(r)}) \end{bmatrix} + T_1 \begin{bmatrix} W_x^{(r)} \\ W_y^{(r)} \end{bmatrix} \quad (231)$$

$$\delta x_{turn}^{(r)} = x_1^{(r)} - x_0^{(r)} \quad (232)$$

Next, the turn time, T_1 , is calculated as the total change in heading divided by the magnitude of heading rate, Eq. (233). Since it was assumed that the aircraft is traveling at constant altitude, airspeed, and bank angle, it follows that the heading rate throughout the turn will also be constant. Again, the signum function is used to ensure a positive turn time. In Eq. (234), $\dot{\psi}^{(r)}$ will have the same sign as ϕ_1 .

$$T_1 = \text{sgn}(\tan \phi_1) \frac{\Delta\psi^{(r)}}{\dot{\psi}^{(r)}} \quad (233)$$

$$\dot{\psi}^{(r)} = \frac{g}{V_T} \tan \phi_1 \quad (234)$$

The required change in heading is given by Eq. (235), assuming that the both heading values, $\psi_0^{(r)}$ and $\psi_1^{(r)}$, are defined $[-\pi, \pi]$. The previously made assumptions result in a positive $\Delta\psi^{(r)}$.

$$\Delta\psi^{(r)} = \psi_0^{(r)} - \psi_1^{(r)} = \psi_0^{(r)} - \left(\arcsin \left(-\frac{W_y^{(r)}}{V_T} \right) - \frac{\pi}{2} \right) \quad (235)$$

The turn time, T_1 , is calculated by substituting Eq. (234) and Eq. (235) into Eq. (233) resulting in Eq. (236).

$$T_1 = \text{sgn}(\phi_1) \frac{V_T}{g} \frac{\psi_0^{(r)} - \psi_1^{(r)}}{\tan \phi_1} \quad (236)$$

Finally, the bank angle required to achieve the desired final $y^{(r)}$ -axis position must be determined to accurately estimate the arrival time and provide a bank angle command. First, the final $y^{(r)}$ -axis position in Eq. (231), $y_1^{(r)}$, is defined as b , the desired turn radius of the racetrack. The equation is then solved for $\text{sgn}(\phi_1) \tan \phi_1$ resulting in Eq. (237), where ϕ_1 is the bank angle which will achieve the correct y-axis position at the end of the turn.

$$\text{sgn}(\phi_1) \tan \phi_1 = \frac{V_T \left(W_y^{(r)} \psi_1 - W_y^{(r)} \psi_0^{(r)} + V_T \sin \psi_1^{(r)} - V_T \sin \psi_0^{(r)} \right)}{g \left(b - y_0^{(r)} \right)} \quad (237)$$

Both sides of Eq. (237) are positive allowing for the magnitude of the bank angle to be easily determined by taking the arctangent of the right-hand side. Since a counterclockwise turn is assumed, the value of ϕ_1 will be negative. Subsequently, the time remaining in the turn may be estimated with Eq. (236), assuming that the aircraft maintains a constant bank angle, ϕ_1 , throughout the turn.

The previous calculations specifically apply to a turn in quadrant I. However, quadrant III turns are calculated using the same equations by rotating the reference frame by π such that quadrant III is identical to quadrant I. Once the initial position, heading, and winds have been rotated, the calculations are identical.

Path Acquisition.

A key assumption of the previous section is that the path is precisely followed. However, it is desired for the aircraft to be able to begin off path and still maintain accurate timing. In fact, it will be shown that this occurs whenever an orbit is completed against a moving target, see Sec. 5.8. The approach chosen is to guide the aircraft to the path quickly where the path tracking assumptions are valid.

This operation is performed by navigating the aircraft at a right angle to the path. However, the result is that the projection of the aircraft onto the path, used in estimating the time remaining, will not change position. Subsequently the estimated time remaining will not change, and the estimated arrival time will increase as time increases. In the case where the aircraft is not on the racetrack, an estimate for the time to return to the racetrack is made to provide a more accurate overall arrival time estimate.

The estimate is calculated in the same manner as the straight path segments, Sec. 5.4. The estimate assumes that the aircraft flies directly towards the path and is calculated as the path error (the distance remaining) divided by the current reference groundspeed. The groundspeed is calculated by rotating the winds to be tangent to the closest point on the path and adding the appropriate component to airspeed, a similar rotation to Eq. (211).

Since some path tracking error is expected, a dead zone is utilized to prevent excessive additions to the time estimate once the path is acquired. Based on initial simulations, a dead zone equivalent to the desired turn radius is used. Although the path follower is able to stay well within this value, see Sec. 5.7, it was observed that this value provided minimal deviation in arrival time estimate when acquiring the path. Once the aircraft is within the dead-zone, it is assumed that the aircraft is once again successfully following the path and that the time estimates of the previous sections are valid.

5.5 Arrival Time Control

Taking the arrival time estimates of each segment from the previous section, the arrival time at the end of the racetrack is estimated as the sum of the estimated time to travel along each remaining segment added to the current time. The arrival time

error, T_e , is defined by Eq. (238) where T_{rem} is the estimated time remaining until the endpoint, T_{ref} is the reference or desired arrival time, and t is the current time. A positive arrival error indicates that the aircraft will arrive “late.” Two controls are available to affect $T_{rem}(t)$: airspeed and length of path.

$$T_e(t) = T_{rem}(t) + t - T_{ref} \quad (238)$$

Arrival Time Control Via Airspeed.

First, assume that the path length is fixed and airspeed is the only control which may affect the arrival time estimate. Recall, the arrival time estimate assumes that the aircraft travels at the desired airspeed for the remainder of the trajectory, as opposed to the current airspeed or some non-constant airspeed profile. The desired airspeed is used because it aids the control algorithm in achieving a desired final airspeed in addition to a desired arrival time; this is best shown with an example.

At some time t_1 , let there be an aircraft a distance d from the endpoint; represented by a triangle in Fig. 33. The aircraft is traveling from left to right at a constant airspeed of V_T such that the dynamics of the aircraft position relative to the endpoint are $\dot{d}(t) = -V_T$; assuming that the wind speed is zero. The arrival time estimate is defined to be the distance remaining, d , divided by the desired airspeed, V_{ref} as defined in Eq. (239). The arrival time error for the aircraft is then defined by Eq. (240)

$$T_{rem}(t) = \frac{d(t)}{V_{ref}} \quad (239)$$

$$T_e(t) = \frac{d(t)}{V_{ref}} + t - T_{ref} \quad (240)$$

Setting the error equation, Eq. (240), to zero at time, t_1 , allows a distance d^* to

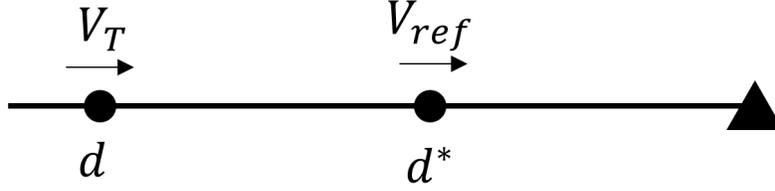


Figure 33. Velocity used in Time Estimation Example

be computed by Eq. (241). The variable d^* describes a position at the current time, t_1 , where the desired arrival time will be achieved if the aircraft travels at the desired airspeed, V_{ref} .

$$0 = \frac{d^*(t_1)}{V_{ref}} + t_1 - T_{ref}, \quad (241)$$

Now, assume that the current arrival time error at time t_1 is greater than zero indicating the aircraft is estimated to arrive “late.”

$$0 < T_e(t) = \frac{d(t_1)}{V_{ref}} + t_1 - T_{ref}, \quad (242)$$

Combining Eq. (242) and Eq. (241) results in the inequality,

$$\frac{d^*(t_1)}{V_{ref}} + t_1 - T_{ref} < \frac{d(t_1)}{V_{ref}} + t_1 - T_{ref} \quad (243)$$

$$d^*(t_1) < d(t_1) \quad (244)$$

as shown in Fig. 33.

Therefore, a controller which drives the arrival time error to zero will also drive the position of the aircraft to a moving position d^* . Since d^* is moving at V_{ref} , the arrival time and arrival airspeed constraints will be simultaneously satisfied if $d = d^*$. The result is similar to a virtual target approach without explicitly tracking a virtual target [45]. In Chapter VI, the arrival time error dynamics will be derived and analyzed for this airspeed control scheme.

Arrival Time Control Via Path Length.

Now assume that the path-length is not fixed. The path length may only be adjusted in quadrants I and II because adjusting the path in quadrant III would result in large tracking errors while adjusting the path length in quadrant IV would not have any affect since the endpoint is fixed. This can be seen in Fig. 34. The racetrack size could simply be controlled with a PI controller and drive the arrival time error to zero. However, this section will derive an analytic expression for the desired racetrack size as a function of the estimated time remaining and the desired arrival time.

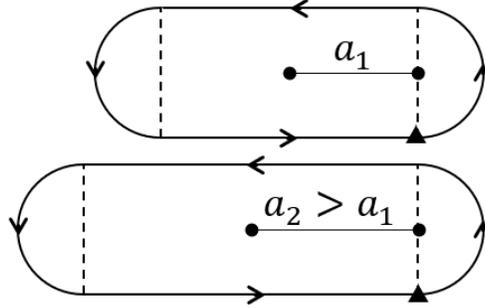


Figure 34. Effect of Adjusting a on the Racetrack Shape

Equations (245)-(246) expand the definition of the estimated time remaining as the sum of each remaining path segment if the aircraft is in quadrant I or II. The time remaining for the current segment will decrease as the aircraft moves along the path. However, the time remaining for each straight segment is a function of the racetrack size since the length of each segment is defined to be $2a$. Finally, with the aircraft in quadrant I or II, the estimated time remaining in quadrant III is constant.

$$T_{rem}(t) = T_{rem_{Q_I}}(t) + T_{rem_{Q_{II}}}(a) + T_{rem_{Q_{III}}} + T_{rem_{Q_{IV}}}(a), \text{ in } Q_I \quad (245)$$

$$T_{rem}(t) = T_{rem_{Q_{II}}}(t, a) + T_{rem_{Q_{III}}} + T_{rem_{Q_{IV}}}(a), \text{ in } Q_{II} \quad (246)$$

Recalling that the estimated time remaining for each of the straight segments is simply distance divided by desired groundspeed, the time remaining equations are expanded to Eqs. (247)-(248). In this expression, V_{sref} is the crosswind corrected reference speed defined by Eq. (249) which was derived using the wind triangle from Sec. 3.5. Adding the current headwind/tailwind component of windspeed to V_{sref} results in the desired groundspeed, the denominator component.

$$T_{rem}(t) = T_{remQ_I}(t) + \frac{2a}{V_{sref} + W_x^{(r)}} + T_{remQ_{III}} + \frac{2a}{V_{sref} - W_x^{(r)}}, \text{ in } Q_I \quad (247)$$

$$T_{rem}(t) = \frac{x^{(r)}(t) + a}{V_{sref} + W_x^{(r)}} + T_{remQ_{III}} + \frac{2a}{V_{sref} - W_x^{(r)}}, \text{ in } Q_{II} \quad (248)$$

$$V_{sref} = V_{ref} \sin \arccos \left(-\frac{W_y^{(r)}}{V_{ref}} \right) \quad (249)$$

In the racetrack algorithm, all of the parameters in Eqs. (247)-(249) have already been calculated as part of the time estimate calculations. The definition of arrival time error, Eq. (250), is then set equal to zero and solved for the racetrack size a . After simplifying the expression, the result is Eqs. (251)-(252).

$$T_e(t) = T_{rem}(t, a) + t - T_{ref} \quad (250)$$

In Q_I ,

$$a = -\frac{\left(V_{s_{ref}}^2 - W_x^{(r)2}\right) \left(T_{rem_{Q_I}} + T_{rem_{Q_{III}}} + t - T_{ref}\right)}{4V_{s_{ref}}} \quad (251)$$

In Q_{II} ,

$$a = -\frac{\left(V_{s_{ref}} + W_x^{(r)}\right) \left(x^{(r)}(t) + V_{s_{ref}} \left(T_{rem_{Q_{III}}} + t - T_{ref}\right)\right)}{3V_{s_{ref}} - W_x^{(r)}} + \frac{\left(V_{s_{ref}} + W_x^{(r)}\right) \left(W_x^{(r)} \left(T_{rem_{Q_{III}}} + t - T_{ref}\right)\right)}{3V_{s_{ref}} - W_x^{(r)}} \quad (252)$$

When this expression is active in the algorithm, the arrival time error is zero, by definition. Small perturbations to the error are expected in simulation due to a discrete time step. If a is calculated at each time step, then there will be a small amount of time between calculations when the information used to calculate a is not exactly the true value. These perturbations are minimized with a small time step.

5.6 Path Tracking Effect on Arrival Time

If the size of the racetrack is fixed, the aircraft may intentionally fly off the desired path to affect arrival time. Additionally, it is expected there will be some path tracking error in simulation due to the dynamics of the path follower interacting with the effects of wind and turbulence. This section evaluates the effect path tracking error may have on the arrival time error. The main cause of this effect is the difference between true groundspeed of the aircraft and the groundspeed of the reference point moving along the path.

Straight Segments.

This effect of path tracking error on arrival time error is shown in Fig. 35 for the case of a straight segment. Note, true groundspeed and ground course angles are used. In Sec. 5.4, the time to travel a straight segment is calculated by projecting the aircraft position onto the segment. Similarly, the rate of change of that projection is based on the projected groundspeed, G_s . When the ground course error ($\Delta\chi$) is small, the difference between G and G_s is negligible. It is reasonably assumed that the ground course error will be small while the path follower is successfully tracking the path. Therefore, in the case of a straight path segment while the aircraft is near the path, the effects of path error are negligible.

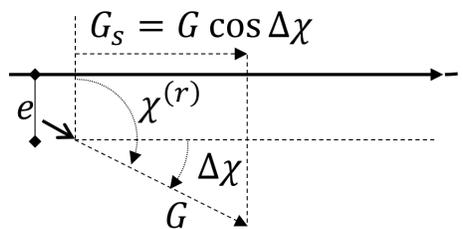


Figure 35. Straight Racetrack Section, Quadrant IV

However, in the case of an aircraft that is reacquiring the path, the heading may not be small and the speed of the projected point could be zero when perpendicular to the path. Therefore, a path error correction term, discussed in Sec. 5.4, is added to approximate the time to get to the path and alleviates any large changes in the arrival time estimate when acquiring a path segment.

Curved Segments.

In the case of a curved segment, the effect of path track error on arrival time is only discussed for the Gaussian Quadrature estimation method, Sec. 5.4 because the Navigation method does not track a path during a turn. Take the scenario in Fig. 36

where the aircraft is tracking a curved path, turn radius R_T , but has a tracking error of e . Based on simple geometry, the arc length, s , of some arc angle θ will be larger for the desired path than the aircraft's current circular trajectory, shown as a bold line.

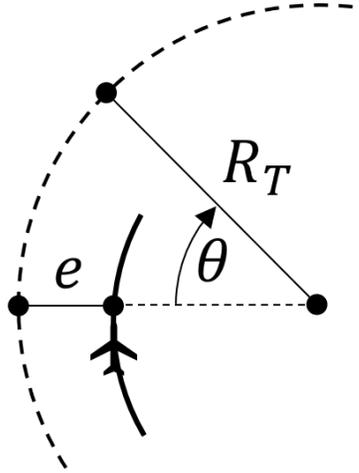


Figure 36. Circular Segment Error

Recall, the quadrature method projects the aircraft position to the circular path in order to estimate the time remaining to complete the segment. The arc length is related to arc angle by some turn radius, R , as defined by Eq. (253). Taking the derivative of this equation, assuming constant R , results in Eq. (254). Equation (254) may be used to relate the measured groundspeed of the aircraft, G_{m_t} to the measured groundspeed of the projected position on the reference path, G_m .

$$s = \theta R \quad (253)$$

$$\dot{s} = \dot{\theta} R \quad (254)$$

$$\dot{\theta}_t = \frac{G_{m_t}}{R_T - e} \quad (255)$$

$$G_m = \frac{G_{m_t} R_T}{R_T - e} \quad (256)$$

The angular rate of change is a function of the aircraft position rate of change, equivalent to groundspeed. Therefore, the true angular rate of change, $\dot{\theta}_t$, is defined by Eq. (255). Finally, the rate of change of a point projected on the path, G_m , is determined utilizing the true angular rate of change as defined by Eq. (256). Note, the variable G_m is used as the rate of change of the projected point because it is equivalent to G_m which will be used Sec. 6.2 and assumes the aircraft remains on the path. The actual of effect of flying off the desired path is analyzed in the results, Sec. 7.4

5.7 Model Predictive Control Path Follower

As previously mentioned, the racetrack algorithm effectively decouples arrival time control from path following control. This requires a navigation controller which can accurately track the path and provide quick path acquisition when the aircraft is away from the desired path. Any controller which can meet these requirements could be integrated into the racetrack algorithm.

However, this research utilized a model predictive control (MPC) navigation controller primarily because of its flexibility with regards to providing fine path tracking in addition to gross acquisition. In general, a MPC controller determines a control output by finding the control which minimizes some cost function over a finite time horizon.

Choosing the cost function to be tracking error, a single MPC controller will direct the aircraft towards the path when far away and provide fine tracking when near the path. In this research, the MPC controller determines a single bank angle, ϕ_m which minimizes Eq. (257) at every time step t ; t_h is the time horizon of the controller.

$$J(\phi) = \int_t^{t+t_h} e^2(\phi_m, t) + k_\delta \delta^2(\phi_m, t) \quad (257)$$

The position error (e) and heading error (δ), depicted in Fig. 37, are determined by integrating the single bank angle through the simplified dynamics and LQR bank angle command system presented in Chapter III then comparing the output to the desired path. The dynamics are reproduced in Eqs. (258)-(264). Recall, u_{max} is the aircraft defined roll rate limit and $\dot{N}_{Z_{max}}$ is the assumed maximum rate of change of N_Z the aircraft can generate; see Sec. 3.2.

$$\dot{x}^{(r)}(t) = V_T \cos \psi^{(r)}(t) + W_x^{(r)} \quad (258)$$

$$\dot{y}^{(r)}(t) = V_T \sin \psi^{(r)}(t) + W_y^{(r)} \quad (259)$$

$$\dot{\psi}^{(r)}(t) = \frac{g}{V_T} \tan \phi(t) \quad (260)$$

$$\dot{\phi}(t) = p(t) \quad (261)$$

$$\dot{P}(t) = \frac{1}{\tau_p} (P_{cmd}(t) - P(t)) \quad (262)$$

$$P_{cmd}(t) = \int_{P_{min}}^{P_{max}} -\mathbf{K}_p [\phi(t) - \phi_m, P(t)]^T \quad (263)$$

$$P_{max} = -P_{min} = \min \left(u_{max}, \left| \dot{N}_{Z_{lim}} \frac{\cos \phi(t)}{\tan \phi(t)} \right| \right) \quad (264)$$

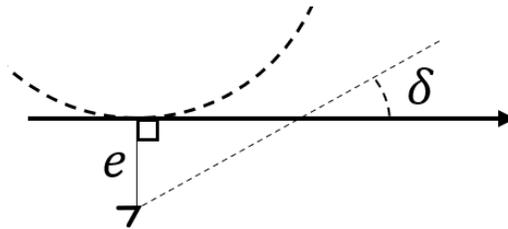


Figure 37. Path Follower Variables

The cost is calculated assuming constant airspeed based on the airspeed at each time step (i.e., the airspeed may change each time a new bank angle is calculated). Finally, the path error is calculated as the tangential distance from the circular or

straight path segment. The error is calculated differently depending on the quadrant mode of the controller; the quadrant mode logic is discussed in detail in Sec. 5.8. Nominally, the distance from each evaluated point to the full racetrack is used to determine the tracking error. However, this method guides the aircraft to the closest portion of the racetrack. When acquiring the path or executing a re-orbit, it is desired for the follower to ignore portions of the racetrack and fly to the desired segment.

As an example, a re-orbit may be commanded in quadrant IV to send the aircraft back to quadrant II. The aircraft is closest to the quadrant IV segment and needs to ignore that portion of the path. Therefore, a re-orbit is commanded, Sec. 5.8, or when commanded to fly to quadrant II while acquiring the path, the path segments in quadrants I and IV are ignored. A loop is prevented because the quadrant mode will return to a nominal condition once the quadrant II path is reacquired.

The course error gain, k_δ , is a variable that is a function of both heading error and path error. The gain is zero when the aircraft is near the path, a positive value when acquiring the path, and zero again when far from the path.

Combined with the path error component of Eq. (257), this approach ensures that the aircraft begins to orbit in the correct direction when acquiring the path and allows the aircraft to turn around if initialized near the path in the wrong direction.

The optimization problem is numerically solved with a finite number of iterations to ensure a bank angle is computed before the next control-algorithm time step. The output of the MPC controller is a bank angle command which is limited based on the assumed turn performance for the aircraft; defined in Chap. III.

Arrival Time Control Features.

As previously mentioned, arrival time control via speed is limited by the capability of the aircraft. To improve the ability of the controller to negate large arrival time

errors to zero, two path adjustment features were added to the MPC algorithm: s-turns and turn radius adjustment. Both features are implemented by adding or subtracting a bounded parameter from the calculated tracking error. This effectively “commands” a non-zero path error which the MPC algorithm attempts to achieve. The features are only enabled if the absolute arrival time error is greater than five seconds.

The purpose of the s-turn is to increase the arrival time in the event of racetrack size limitation due to a keep-out zone. The s-turn causes the aircraft to weave across a section of path which increases the length of the actual path compared to the path segment. This feature is only enabled in Quadrant II since flying inside of a turn would cause the arrival time to decrease. The s-turn is implemented by adding a sine signal to the actual path error. The magnitude of the signal is 700 *ft* for fighter aircraft and 300 *ft* for all others.

The turn radius adjustment feature directs an aircraft in a turn to fly inside or outside of a turn to affect its arrival time. Clearly, a smaller turn radius will result in a shorter path and an aircraft traveling at the same speed will get to the end of the turn faster with a smaller turn radius. The magnitude of this adjustment is also limited to 80% of the reference airspeed value which was chosen based on observed performance. A gain of 40 is applied to any time error outside of ± 5 seconds to calculate the non-limited adjustment. Overall, these features enable the algorithm to quickly diminish large arrival time errors in the event the desired endpoint or path abruptly changes.

Model Predictive Control Parameters and Performance.

The MPC time horizons were selected for each aircraft to achieve a maximum path following error of 500 *ft* in an environment with a windspeed of half the aircraft

airspeed and moderate turbulence. The time horizons are listed in Table 11.

Table 11. Model Predictive Control Aircraft Parameters

Aircraft Type	Horizon (s)
Light Utility	6
Medium Cargo	8
Heavy Cargo	7
Fighter	6

An example of the MPC tracking performance is also presented in Fig. 38; the bank angle command and desired quadrant are given in the first graphs. The simulation is a heavy cargo aircraft, $V_{ref} = 544 \text{ ft/s}$, following a racetrack in moderate turbulence and a steady state wind of 190 ft/s (112 knots). The error is within 200 ft for the duration of the orbit and the control does not exhibit overly active behavior.

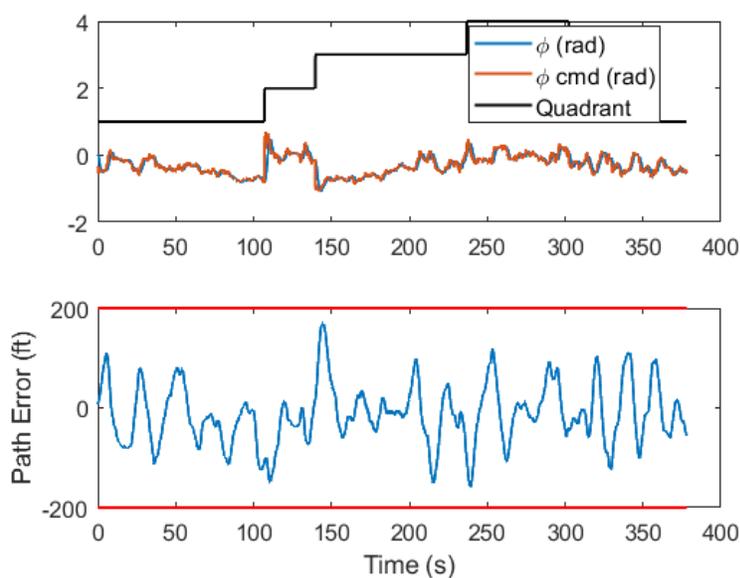


Figure 38. Heavy Cargo Aircraft Path Following Performance Example

Finally, Fig. 39 presents a simulation which highlights the path acquisition, s-turn, and turn radius adjustment logic. The reference airspeed is 864 ft/s and the windspeed is 259 ft/s with a heading of 97 deg . The path acquisition is nominal and followed by s-turns at approximately 50-100 seconds. After traveling the remainder of

the orbit nominally, the turn radius adjustment is activated after passing the endpoint (red triangle) at approximately 250 seconds. The adjustment commands the follower to fly outside the desired turn radius to increase the estimated arrival time. The command slowly decreases as the arrival time error decreases below a set threshold value.

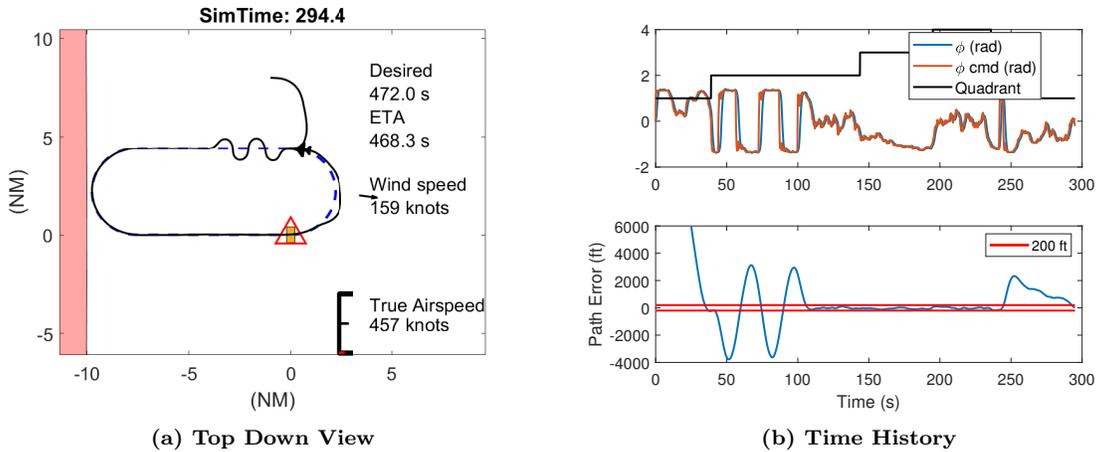


Figure 39. Fighter Aircraft Arrival Time Control Feature Example

An identical simulation run with only the turn radius adjustment active reached the s-turn deactivation threshold upon entering the final quadrant; coordinates (0, 8). A subsequent simulation without s-turns or turn radius adjustment results in reaching the s-turn deactivation threshold when crossing the endpoint. This highlights the benefit of both the s-turn and turn radius adjustment logic.

5.8 Racetrack Manager

The racetrack manager determines a number of high-level parameters which affect the calculations and behavior of the racetrack. The manager calculates the desired arrival time, the maximum size of the racetrack, the orbit direction, the location of the endpoint, whether airspeed control is enabled or not, and the current quadrant the aircraft is commanded to follow.

Arrival Time Communication.

The arrival time for a single aircraft is straight forward; this is equivalent to the lead aircraft case. It is simply the desired orbit duration added to the start time of the current orbit. The start time is either the initialization of the simulation or the time when the aircraft last overflowed the endpoint.

In the case of multiple aircraft, the desired time spacing is added to the estimated arrival time of the preceding aircraft. Finally, to simulate data communication, the estimated arrival time of the preceding aircraft is delayed by 0.7 seconds and updated once per second. This provides an approximate simulation of data calculation and transmission in a realistic environment. The effect of a delay on stability of the system is of interest but beyond the scope of this dissertation.

Enabling Airspeed Control.

Airspeed control is allowed to be turned on and off to prevent undesirable behavior during racetrack initialization. To achieve this, the airspeed command is set to the desired airspeed for the first 7 seconds of each orbit to allow any large perturbations in arrival time error due to endpoint changes to stabilize.

Endpoint Estimation.

When evaluating the arrival time and airspeed performance of the algorithm, it is with reference to the endpoint of the racetrack. In this research, it is assumed that the desired outcome is overflight of some ground target at a desired time, heading, and airspeed. The case of a static target is trivial, and the endpoint will overlay the target itself.

There are two primary approaches to overfly a moving target; predict the endpoint or track the target directly. In either case, it is first assumed that the aircraft has

some method to accurately measure the position and velocity vector of the target.

The first approach is to extrapolate the movement of a target to the estimated arrival time, depicted in Fig. 40. This method assumes that the target continues at a constant speed and direction through the estimated arrival time. The estimated arrival time is used, as opposed to the desired arrival time, because if the arrival time is not met, the target is still overflown. If the desired arrival time were used, both arrival time and overflight of the target may not be met.

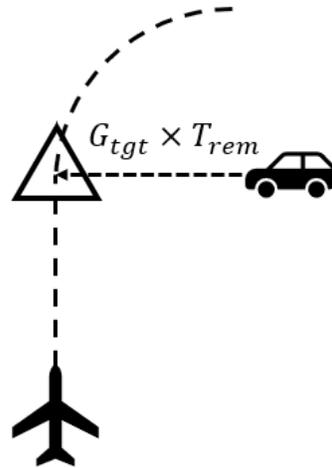


Figure 40. Method 1: Constant Target Speed, Fixed End Point

Calculating the estimated endpoint is done by multiplying the velocity vector by the estimated time remaining. If the target velocity vector is constant, the estimated endpoint in Fig. 40, the triangle, will remain constant. The disadvantages of this method occur if the target changes its motion before the arrival time. The result is a large step change in the endpoint which moves the entire racetrack being followed. The aircraft must reposition to the new racetrack and adjust its speed accordingly.

The second approach is to let the endpoint move with the target, depicted in Fig. 41. In this situation, the racetrack pattern will always be moving which isn't

inherently a disadvantage. In fact, this method is more robust to changes in target speed. However, this movement will cause the ground course that the aircraft overflies the target to be affected by the motion of the target, shown as the vector \mathbf{G} in Fig. 41. This is because the motion of the racetrack must be added to the motion of the aircraft following the racetrack.

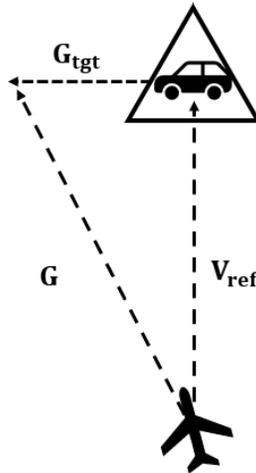


Figure 41. Method 2: Constant Target Speed, Moving End Point

In this research, priority is given to overflying the target at the desired heading and it is assumed that the target will maintain a relatively constant speed and heading. Therefore, the endpoint extrapolation method, Fig. 40 is implemented in the algorithm.

Racetrack Reset Trigger.

The racetrack manager triggers several separate functions if a large change to the endpoint is detected. This is defined as a “Racetrack Reset”. Two different types of resets are used.

The first type of reset triggers the keep-out zone calculations and is termed an “endpoint reset.” The initial location of the endpoint is saved as a reference point. If

the current endpoint moves greater than 500 ft from the reference point, an “endpoint reset” is triggered.

The second type of reset is triggered when the filtered velocity of the endpoint is greater than seven times the reference airspeed; the filter is a low pass filter with a one second time constant. This may occur when large changes to the estimated time remaining are experienced such as stopping targets or at the beginning of an orbit. This reset, referred to as a “Quadrant Reset” causes the quadrant logic to execute a hold and reset; described in Sec. 5.8.

The reason for two different sets of resets is that recomputing the keep-out zone limits is computationally intensive and a rapid succession of resets resulted in simulation errors. On the other hand, the quadrant control is able to handle a quick succession of resets. Since the quadrant command is part of the arrival time calculations, it is necessary to ensure it is up to date, discussed in Sec. 5.8.

Keep-Out Zone Determination.

In this research, keep-out zones are enforced by determining the maximum race-track size which does not overlap a keep-out zone. This method limits ensures the desired path does not violate the keep-out zone but does not directly limit the aircraft trajectory. As an example, the turn radius adjustment in Sec. 5.7 could cause the aircraft to violate the keep-out zone when the desired path is touching the keep-out zone. To accommodate the expected variation in path tracking, a “pad” is included in the calculations which ensures the desired path remains a specified distance from the keep-out zone, identified as the variable *pad* in calculations.

Additionally, determining the maximum size of the racetrack could be performed by a separate navigation algorithm which is specifically designed to manipulate shapes around exclusion zones. However, the racetrack algorithm is set up as a single model

which includes its own keep-out zone enforcement.

The keep-out zone enforcement is performed as an optimization problem to maximize the size of the racetrack while honoring the keep-out zone constraints. The problem is also formulated to allow for an arrival heading tolerance. This is implemented to allow the racetrack to rotate to avoid a keep-out zone or provide a larger maximum arrival time.

To formulate a feasible problem, it is assumed that any keep-out zones do not enclose the endpoint and do not overlap a racetrack of size $a = 0$, a circle. This represents an infeasible scenario which is outside the scope of this algorithm.

It is assumed that any keep-out zones are ground-fixed and time invariant. Subsequently, it is assumed that any keep-out zone may be modeled by a series of ground-fixed elliptical or rectangular areas. These assumptions allow the keep-out zones to be modeled as superellipsoids [81].

Superellipsoids are a special set of closed curves where the curve is defined by Eq. (265). The variables a_1 and a_2 determine the size of the semi-major and semi-minor axes while ε_2 determines how “rectangular” the curve is; note ε_1 is used for three-dimensional shapes. Figure 42 displays multiple superellipsoids with varying ε_2 ; $a_1 = a_2 = 1$ in each graph.

$$\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}} = 1 \quad (265)$$

The strength of superellipsoids is that the definition of the curve is related to an “in-out” function, Eq. (266). This function may be used to determine if a point is inside or outside the shape. If $F = 1$ the “query” point is on the curve. If $F > 1$ the “query” point is outside the curve and if $F < 1$ the “query” point is inside the curve. This function is referenced to the center of the superellipsoid in a specific orientation.

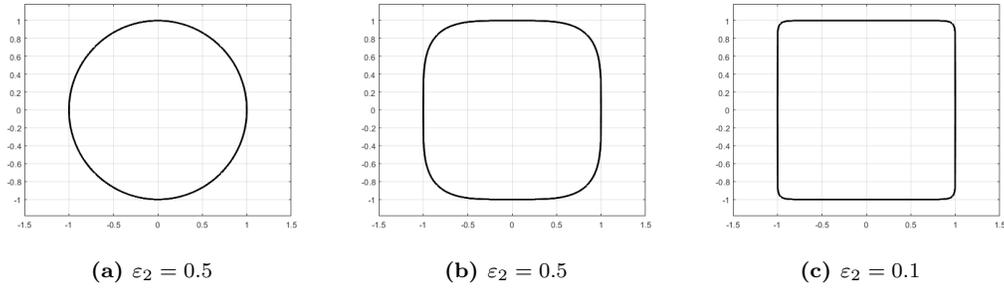


Figure 42. Change in Superellipsoids with ε_2

To allow for rotated keep-out zones which may be placed somewhere around the racetrack, any “query” point in the earth-fixed reference frame must first be referenced to the center of the superellipse and then rotated to the orientation shown in Fig. 42.

$$F(x, y) = \left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}} \quad (266)$$

To develop the optimization problem, the “query” points must be defined. The “query” points are selected to be 10 evenly spaced points on each circular segment of the racetrack, to include the endpoints of each circle, for 20 points total. The linear segments of the racetrack are not queried because the initial guess of the optimization problem is a circle. As the racetrack lengthens, the linear segment remains between the endpoints of the circular segment. Therefore, the circular segment will encounter a constraint before the linear segment if the optimization is performed correctly. The racetrack reference frame is reproduced in Fig. 43 for reference.

Utilizing the racetrack reference frame, the “query” points in the earth-fixed reference frame may be defined as a linear function of the racetrack size. Let $\mathbf{E}_I^{(r)}$ be a 2×1 vector of the $x^{(r)}, y^{(r)}$ coordinates of a “query” point in the first quadrant. The keep-out zone “pad” is added to the desired racetrack turn radius when computing the “query” points.

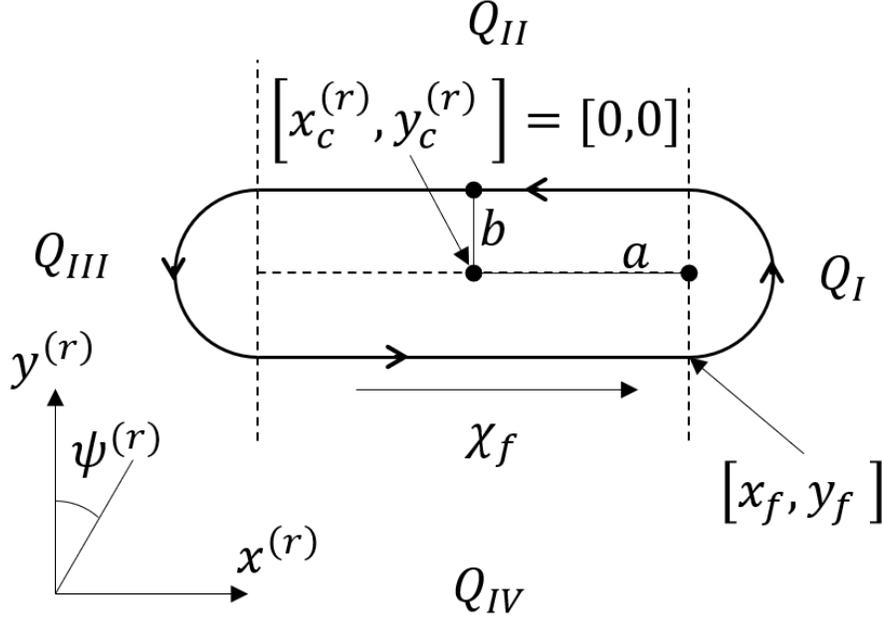


Figure 43. Left-Hand Racetrack Definition

Recall that \mathbf{M} is simply a clockwise axis rotation matrix, therefore \mathbf{M}^T rotates the racetrack axes counterclockwise and the racetrack is rotated clockwise. Additionally, recall that D_T is the turn direction variable which is defined to be 1 for a counterclockwise orbit and -1 for a clockwise orbit.

Then the earth-fixed coordinates of a quadrant I point may be defined by Eq. (267). Note that quadrant I is independent of the racetrack size variable a , as expected. Similarly, let $\mathbf{E}_{III}^{(r)}$ be a 2×1 vector of $x^{(r)}, y^{(r)}$ coordinates of a quadrant III “query” point with racetrack size $a = 0$. Again, an adjusted turn radius to include the keep-out zone “pad” is used to calculate $\mathbf{E}_{III}^{(r)}$.

In the racetrack reference frame, quadrant III is simply translated along the negative x-axis as the racetrack size increases. Therefore, the earth-fixed coordinates of a quadrant III point may be defined by Eq. (268); note that this is linear with respect to a .

$$\mathbf{E}_I^{(e)} = \mathbf{M}^T \left(\chi_f - \frac{\pi}{2} \right) \begin{bmatrix} 1 & 0 \\ 0 & D_T \end{bmatrix} \mathbf{E}_I^{(r)} + \begin{bmatrix} x_f^{(e)} \\ y_f^{(e)} \end{bmatrix} \quad (267)$$

$$\mathbf{E}_{III}^{(e)}(a) = \mathbf{M}^T \left(\chi_f - \frac{\pi}{2} \right) \begin{bmatrix} 1 & 0 \\ 0 & D_T \end{bmatrix} \left(\mathbf{E}_{III}^{(r)} + \begin{bmatrix} -2a \\ 0 \end{bmatrix} \right) + \begin{bmatrix} x_f^{(e)} \\ y_f^{(e)} \end{bmatrix} \quad (268)$$

Each “query” point is evaluated with Eqs. (267)-(268) to result in a 2×20 set of coordinates, $\mathbf{E}^{(e)}$. Let the rotation of a keep-out zone be defined as $\psi_k^{(e)}$ and the earth-fixed coordinates of the center of the keep-out zone be $(x_k^{(e)}, y_k^{(e)})$. The “query” points are shifted and rotated into the keep-out zone reference frame, see Fig. 42, using Eq. (269). Finally, the inequality constraint, Eq. (270), is evaluated using Eq. (266) for each “query” point and each keep-out zone in the scenario.

$$\mathbf{E}^{(k)} = \mathbf{M}(\psi^{(e)}(k)) \mathbf{E}^{(e)} - \begin{bmatrix} x_k^{(e)} \\ y_k^{(e)} \end{bmatrix} \quad (269)$$

$$F(\mathbf{E}^{(k)}) \geq 1 \quad (270)$$

The resulting optimization problem may be summarized by Eq. (271). This optimization problem is evaluated whenever an “Endpoint Reset” is commanded as described in Sec. 5.8

$$\text{minimize} \quad -a \quad (271a)$$

$$\text{subject to} \quad -F(\mathbf{E}^{(k)}(a)) \leq -1 \quad (271b)$$

Quadrant Control.

To provide flexibility in arrival time control, the quadrant the aircraft is following is controlled by a state machine modeled in the StateFlow language of MATLAB[®]. The output of this state machine is the quadrant “mode” of the algorithm (i.e., the portion of the racetrack the aircraft should be following). The state machine is used because it allows for a wide range of edge cases to be addressed. The primary benefit of the quadrant control is commanding the correct quadrant in the event the endpoint moves, such as after orbit completion with a moving target. This requires a robust state machine since the endpoint may move to any racetrack quadrant depending on the target heading, target speed, wind direction and windspeed.

In the state machine, the physical quadrant that the aircraft is in is labeled as *ActualQuad*; this is an input into the state machine. The output of the state machine is the quadrant mode which is labeled as *SetQuad*. This parameter directs which part of the racetrack the aircraft should be following. When the simulation begins, the quadrant mode is initialized to the quadrant the aircraft is physically in unless the simulation is a multi-pass simulation. In the latter case, the quadrant initializes as if an orbit was just successfully completed.

The quadrant control varies depending on how the curved section arrival time is estimated. Either a ground-fixed racetrack is assumed, the Gaussian Quadrature method, or a constant bank angle turn is assumed, see Sec. 5.4. The latter results in a racetrack whose size changes with the magnitude of the wind. This necessitates a slightly different method to control the quadrant mode. Three overall control loops are used: nominal, reset, and re-orbit.

Ground-fixed Path Nominal Flow.

In the nominal ground-fixed case, it is assumed that the aircraft precisely follows the desired racetrack, Fig. 44. When the aircraft physically moves into the next quadrant, defined by Fig. 44, then the quadrant mode is incremented to the quadrant.

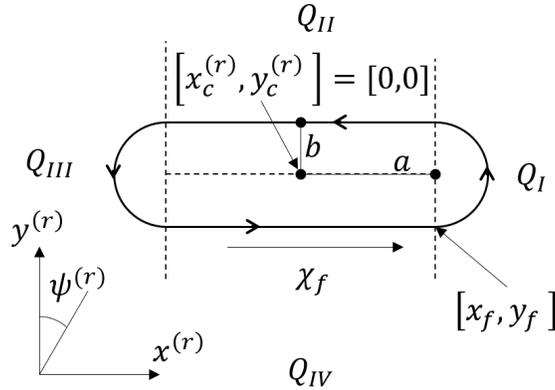


Figure 44. Left-Hand Racetrack Definition

The lead turn time calculated in Chapter IV is also used as a condition to increment to the next quadrant. In the case of a repeat pass completed nominally, the quadrant mode is set to either one or two depending on the relative location of the aircraft to the new racetrack. This forces the aircraft to fly towards the desired quadrant and ensures the racetrack size may adjust. Once the aircraft is within a turn radius of the desired path, the quadrant logic returns to its nominal flow. This loop is depicted in Fig. 45.

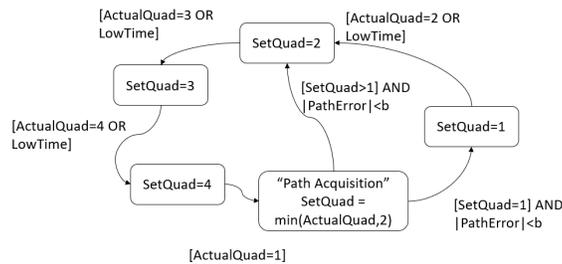


Figure 45. Ground-fixed Path Nominal State Machine

Constant Bank Angle Nominal Flow.

In the case of a constant bank angle turn, the endpoint of each turning path segment is variable and dependent on Eq. (272); from Sec. 5.4. The variability of these endpoints requires a modification to the previous method of quadrant determination based on how the wind affects the ground track.

$$\begin{bmatrix} x_1^{(r)} \\ y_1^{(r)} \end{bmatrix} = \begin{bmatrix} x_0^{(r)} \\ y_0^{(r)} \end{bmatrix} - \frac{V_T^2}{g \tan \phi_1} \begin{bmatrix} \cos(\psi_0^{(r)}) \\ -\sin(\psi_0^{(r)}) \end{bmatrix} - \frac{V_T^2}{g \tan \phi_1} \begin{bmatrix} \cos(\psi_1^{(r)}) \\ \sin(\psi_1^{(r)}) \end{bmatrix} + T_1 \begin{bmatrix} W_x^{(r)} \\ W_y^{(r)} \end{bmatrix} \quad (272)$$

$$\delta x_{turn}^{(r)} = x_1^{(r)} - x_0^{(r)} \quad (273)$$

The variability of the turning segment endpoints is highlighted in Fig. 46 where the endpoint of each turn is shifted in the direction of the $x^{(r)}$ -axis wind. Dotted lines are used to delineate the different quadrants are to either straight or curved path segments. The wind in the upper racetrack image is moving from right to left and left to right in the lower image. To correctly determine the quadrant, two assumptions and a racetrack size limit are required.

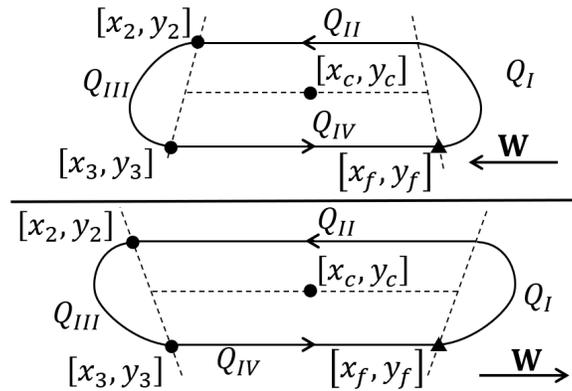


Figure 46. Wind Direction Effect on Ground Track

First, the end of a turning quadrant is defined as some fixed $y^{(r)}$ -axis position. This is implemented because the required bank angle determined by Eq. (237) approaches zero as the required change in heading approaches zero. This could result in a trajectory which asymptotically approaches the next straight path segment without ever reaching the segment itself. Therefore, the transition from turning to straight segment must occur before the remaining change in heading approaches zero. This results in a transition point which is not precisely on the straight path segment but is close enough for the straight-segment path follower to quickly eliminate any error. A value of 95 percent of the turn radius was determined to provide reasonable timing performance and resulted in smooth quadrant transitions.

The second assumption fixes key points relative to the center of the racetrack. It is desired that the transition point $[x_3, y_3]$ be at a racetrack relative position of $[-a, -b]$. However, this point is the endpoint of a turn which is variable. Since the relative change in $x^{(r)}$ between x_2 and x_3 is known from the turn calculations, $\delta x_{turn}^{(r)}$ from Eq. (273), the beginning of the turn, point $[x_2, y_2]$, may selected to achieve the desired end position assuming a nominal turn. Thus, the start of the turn in quadrant III is defined as Eq. (274) and Eq. (275).

$$[x_2^{(r)}, y_2^{(r)}] = [-a', b] \quad (274)$$

$$a' = a + \delta x_{turn}^{(r)} \quad (275)$$

Finally, it can be seen that if $a = 0$ in the case of the upper image in Fig. 46, then the point $[x_2, y_2]$ would need to occur in quadrant I, before the end of the first turn, to achieve the correct end of turn position, x_3 . This issue is remedied by increasing the minimum bound of a to account for $\delta x_{turn}^{(r)}$ in the case where $W_x^{(r)} < 0$, the upper image

of Fig. 46. Combining these assumptions, the aircraft’s current quadrant is governed by Fig. 47. Repeat passes are treated identically to the Gaussian quadrature method. In the case of a repeat pass completed nominally, a “Quadrant Reset” is commanded.

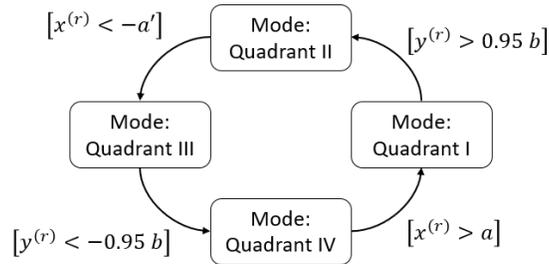


Figure 47. Constant Bank Angle Quadrant Determination

Quadrant Reset.

The quadrant logic is reset to one whenever a significant change in the endpoint is detected. The endpoint will move if the estimated end time changes rapidly due to the preceding aircraft or upon orbit completion. The reset occurs when a “Quadrant Reset,” Sec. 5.8, is triggered by the Racetrack Manager while the aircraft is in quadrants I, II, and III. When a reset is triggered, the quadrant is simply re-initialized to quadrant I. Initializing to quadrant I allows the racetrack size to re-adjust while also allowing the quadrant mode to increment to an appropriate number. A similar “reset” occurs upon successful orbit completion however, this is considered nominal behavior and is discussed in Sec. 5.8

The primary purpose of specifying the quadrant is to prevent the aircraft from following the “wrong” portion of the racetrack after a reset. Take Fig. 48 for example and let the aircraft currently be on time with the current racetrack shown as a solid path. If the target stops at some time t_1 , the racetrack will move to the dashed

path and the aircraft will move from quadrant II to IV. If the mode were switched to quadrant IV, the aircraft would likely arrive early due to its proximity to the endpoint.

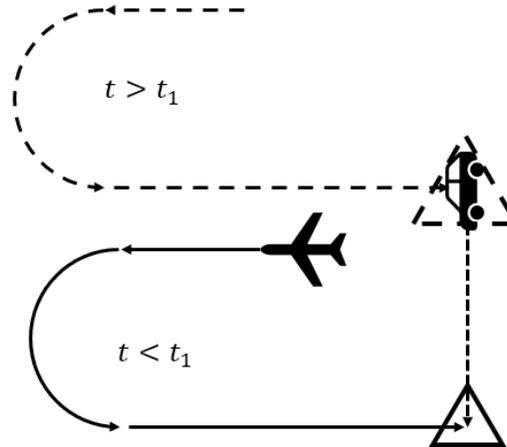


Figure 48. Target Stop at time $t = t_1$

Re-Orbit.

A “re-orbit” logic is used to improve the arrival time performance in cases where the aircraft is significantly early, and the size of the racetrack may not change. This could occur due to an endpoint change which results in the aircraft ending up in the final two quadrants or due to a keep-out zone preventing the racetrack from extending to the desired size. Recalling that one of the primary arrival time controls is path length, if the racetrack size is fixed, the aircraft may simply return to an early portion of the racetrack to increase the path length.

To prevent a re-orbit from resulting in a late arrival, the logic is only allowed when the arrival time error is more negative than an estimate of a time to complete a re-orbit. The re-orbit time estimate is based on the calculation of the time to complete half an orbit at the desired turn radius and airspeed. In quadrant IV, the arrival time

error must be negative and a greater magnitude than the re-orbit time estimate. In quadrant III, half the re-orbit time estimate is used. The re-orbit is performed by commanding the aircraft to fly towards the quadrant II segment until within a turn radius of the path. At that point, the quadrant control returns to its nominal flow.

Full State Machine.

Combining the nominal, reset, and re-orbit logic of the previous sections results the entire state machine which determines the quadrant mode for a ground-fixed path is depicted in Fig. 49. A similar quadrant mode state machine may be derived using the mode changes of Fig. 47 for the constant bank angle time estimate.

The three primary loops are highlighted in Fig. 49. The nominal loop sets the quadrants in numerical order; highlighted in blue. The reset loop allows transitions from quadrant II or III to quadrant I if a reset is triggered due to a change in the position of the endpoint; highlighted in red. Finally, the re-orbit loop allows the quadrant to move from III or IV back to II to increase the estimated arrival time; highlighted in orange.

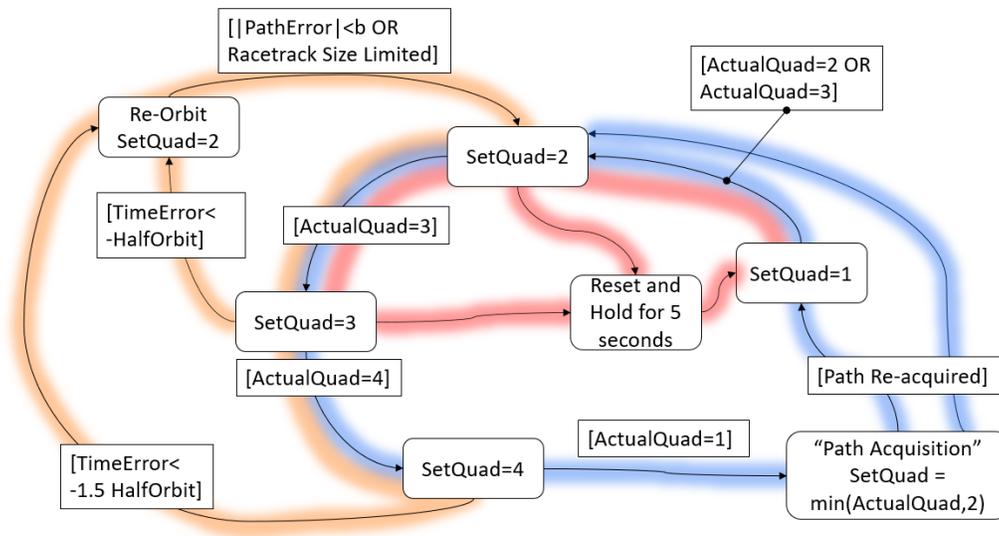


Figure 49. Full Quadrant State Machine

VI. Racetrack Algorithm Stability Analysis

String stability was introduced in Sec. 2.6 as a method to characterize the stability of an interconnected group of vehicles. This chapter applies the theory of string stability to the racetrack algorithm to determine a set of airspeed control gains and limits which result in a stable system under a wide set of environmental conditions. This theory is not applied to the racetrack size control since the size of the racetrack is calculated to result in zero arrival time error, see Sec. 5.5

6.1 Definitions

First, the string stability definitions used in this chapter are defined following the definition given in Ref. [53]. Consider a group of N non-autonomous interconnected systems, $i \in \mathcal{I}_N := \{1, 2, \dots, N\}$, where the leader is referenced as vehicle $i = 1$ and whose dynamics may be described by the system

$$\begin{aligned}\dot{\mathbf{x}}_1 &= f(t, \mathbf{x}_1, 0, \mathbf{w}_1) \\ \dot{\mathbf{x}}_i &= f(t, \mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{w}_i)\end{aligned}\tag{276}$$

where $f : [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is locally Lipschitz in \mathbf{x}_i and \mathbf{w}_i . Note that each vehicle i may have multiple states or disturbances such that $n \geq N$.

Definition 6.1. *The system, Eq. (276), is string stable if for any $\epsilon > 0$, there exists a $\delta > 0$ such that, for all $N \in \mathbb{Z}$,*

$$\sup_{i \in \mathcal{I}_N} \|\mathbf{x}_i(t_0)\|_\infty < \delta \rightarrow \sup_{i \in \mathcal{I}_N} \|\mathbf{x}_i(t)\|_\infty < \epsilon, \quad \forall t \geq t_0.\tag{277}$$

Definition 6.2. *The system, Eq. (276), is asymptotically string stable if the system*

is string stable and

$$\sup_{i \in \mathcal{I}_N} \|\mathbf{x}_i(t_0)\|_\infty < \delta \rightarrow \lim_{t \rightarrow \infty} \sup_{i \in \mathcal{I}_N} \|\mathbf{x}_i(t)\|_\infty = 0. \quad (278)$$

Next, a stability definition based on input-to-state stability is presented [82]. First, class \mathcal{K} and \mathcal{KL} functions are defined. A function $\alpha : [0, a) \rightarrow [0, \infty)$ belongs to class \mathcal{K} if it is strictly increasing and $\alpha(0) = 0$. A function $\beta : [0, a) \times [0, \infty) \rightarrow [0, \infty)$ belongs to class \mathcal{KL} if for each fixed s , the mapping $\beta(r, s)$ belongs to class \mathcal{K} with respect to r and, for each fixed r , the mapping $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \rightarrow 0$ as $s \rightarrow \infty$ (Def. 4.2 and Def. 4.3 from Ref. [82]).

Definition 6.3. *A system, $\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{u})$, is input-to-state stable if there exists a class \mathcal{KL} function β and class \mathcal{K} function γ such that for any initial state and bounded input $\mathbf{u}(t)$, the solution $\mathbf{x}(t)$ exists for all $t \geq t_0$ and satisfies*

$$\|\mathbf{x}(t)\|_\infty \leq \beta(\|\mathbf{x}(t_0)\|_\infty, t - t_0) + \gamma\left(\sup_{t_0 \leq \tau \leq t} \|\mathbf{u}(\tau)\|_\infty\right). \quad (279)$$

Finally, the definition of exponential stability is included from Ref. [82] for completeness. Consider a single member of the interconnected system, Eq. (276),

$$\dot{\mathbf{x}} = f(t, \mathbf{x}). \quad (280)$$

Definition 6.4. *The equilibrium point $\mathbf{x} = \mathbf{0}$ of Eq. (280) is exponentially stable if there exist positive constants c , k , and λ such that*

$$\|\mathbf{x}(t)\| \leq k\|\mathbf{x}(t_0)\|e^{-\lambda(t-t_0)}, \quad \forall \|\mathbf{x}(t_0)\| < c. \quad (281)$$

6.2 Arrival Time Problem Dynamics

Building on the general dynamics of Eq. (276) the arrival time problem is introduced based on the algorithm described in Chapter V and the simplified aircraft model described in Chapter III. Consider a group of airspeed heterogeneous aircraft, $i \in \mathcal{I}_N := \{1, 2, \dots, N\}$, which follow individual ground-fixed paths and desire to overfly a specified endpoint at some reference time and airspeed. The reference, or desired, arrival time for each aircraft is defined to be the arrival time of the preceding aircraft plus some spacing in time. If the spacing is set to zero, all aircraft would arrive at their endpoints simultaneously.

It is assumed that each aircraft has airspeed limitations such that the airspeed state is bounded by a minimum and maximum airspeed, V_{min/max_i} along with a minimum and maximum airspeed rate of change, A_i and B_i respectively; the rate of change limits may be asymmetric. The aircraft are assumed to be flying in a non-zero wind field which is modeled as the sum of a measured component and an unmeasured component similar to turbulence. These two components simulate an aircraft which can measure a steady state windspeed but is not sensitive enough to pick up small changes in windspeed due to turbulence.

The state variables, as they apply to Eq. (276), are the arrival time error and the arrival airspeed error. In the sense of string stability, the variable of interest is only the arrival time error since limits have been assumed on the airspeed which subsequently bound the airspeed error.

Finally, this section also assumes that the desired ground track is precisely followed, similar to the racetrack algorithm time estimates. There may be some tracking error due to external disturbances. However, these errors will be shown to have a similar effect on stability as turbulence, see Sec. 5.6.

Wind and Airspeed Relationship.

The challenge with wind is that many aircraft parameters, including the primary speed control, rely on airspeed while navigation and timing rely on groundspeed. The overarching relationship is that groundspeed is the vector sum of windspeed and airspeed; often referred to as the wind triangle which was discussed in Sec. 3.5. However, this relatively simple relationship becomes challenging if the true windspeed is not known.

In this research, it has been assumed that the wind vector is measured by some on-board system which has the effect of filtering out noise due to turbulence, see Sec. 3.6. Thus, the true wind speed vector, \mathbf{W}_s , is the sum of the measured (filtered) wind vector, \mathbf{W}_f , and the unmeasured (turbulent) effects on the wind speed vector, \mathbf{W}_t .

Since it is assumed that the path is precisely followed, the airspeed vector can be projected onto the desired path using only the measured crosswind and true airspeed; the variable V_{s_i} represents this projection. It is defined by Eq. (282) where W_{cross_i} is the component of measured wind perpendicular to the ground path at the aircraft's current position; shown in Fig. 50.

$$V_{s_i} = V_{T_i} \sqrt{1 - \frac{W_{cross_i}^2}{V_{T_i}^2}} \quad (282)$$

Additionally, the true groundspeed, G_i , and the measured groundspeed, G_{m_i} , need to be separated. This difference will be clarified in the next section, but the overarching reason for the difference is because the control algorithm estimates arrival time with the measured windspeed/groundspeed while the dynamics are affected by the true groundspeed.

The difference between G_i and G_{m_i} is W_{u_i} , the unmeasured windspeed vector projected along the direction of the ground path. Figure 50 depicts the variables used in this work and their geometric relationship. The aircraft identification subscript i has been omitted from the figure for clarity but is used in the remainder of this work.

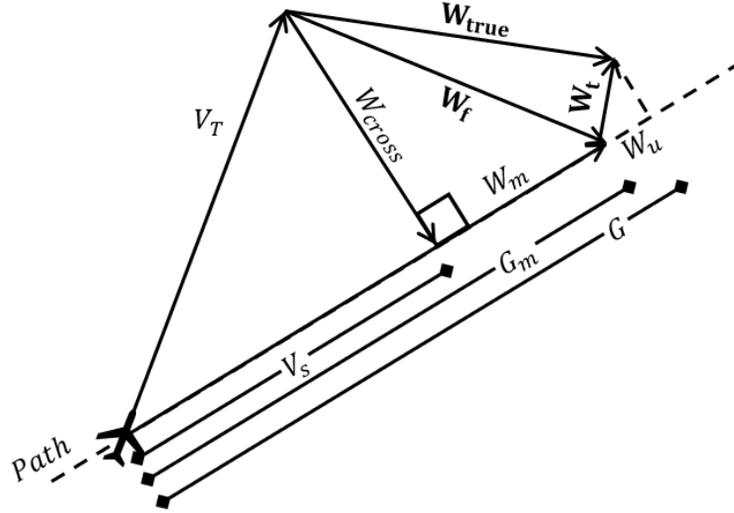


Figure 50. Wind Triangle Variables

Since it is assumed that the path is followed with negligible error, the directions of the variables G_i , G_{m_i} , W_{m_i} , W_{u_i} , and V_{s_i} are also assumed to be along the desired path. This assumption allows the variables to be mathematically related by Eq. (283) and Eq. (284).

$$G_{m_i}(t) = V_{s_i}(t) + W_{m_i}(t) \quad (283)$$

$$G_i(t) = G_{m_i}(t) + W_{u_i}(t) \quad (284)$$

Wind Assumptions.

One of the two primary constraints of the coordinated arrival time problem is that each aircraft should arrive at its designated endpoint at some desired airspeed. From

Chapter V, this airspeed is defined to be V_{ref_i} for each aircraft. A scale factor, F_{V_i} , is also introduced which relates the minimum airspeed to the reference airspeed by Eq. (285).

$$V_{T_{min_i}} = F_{V_i} V_{ref_i}, \quad 0 \leq F_{V_i} \leq 1 \quad (285)$$

The coordinated arrival time problem becomes infeasible if the windspeed is greater than airspeed capability of the aircraft since the aircraft would not be able to decrease its distance to the endpoint. To practically bound the problem space, let the true windspeed be less than the lowest minimum airspeed of all aircraft multiplied by some a factor $F_W < 1$, Eq. (286). This assumption ensures that the distance to the endpoint may always decrease for any aircraft. Since the true windspeed magnitude is bounded, it follows that the unmeasured windspeed, $W_{u_i}(t)$, is also bounded. This bound would be based on the level of turbulence and properties of the wind filter. This bound is defined by Eq. (287).

$$\sup_{i \in \mathcal{I}_N} \|\mathbf{W}_{true_i}(t)\| \leq F_W \min_{i \in \mathcal{I}_N} F_{V_i} V_{ref_i}, \quad 0 \leq F_W < 1, \quad \forall t \geq t_0 \quad (286)$$

$$0 \leq |W_{u_i}(t)| \leq W_{u_{max_i}}, \quad \forall t \geq t_0 \quad (287)$$

Applying the reference airspeed to variables from the previous section, $V_{s_{ref_i}}$ and $G_{ref_i}(t)$ are defined by Eq. (288) and Eq. (289) which are derived from Eq. (282) and Eq. (283) with $V_{T_i}(t) = V_{ref_i}$.

$$V_{s_{ref_i}}(t) = V_{ref_i} \sqrt{1 - \frac{W_{cross_i}(t)^2}{V_{ref_i}^2}} \quad (288)$$

$$G_{ref_i}(t) = V_{s_{ref_i}}(t) + W_{m_i}(t) \quad (289)$$

Error Definitions.

With bounds on windspeed and defined reference airspeeds, a relationship between airspeed and groundspeed error may be derived. Let the airspeed error be defined by Eq. (290) and the groundspeed error be defined by Eq. (291) for each aircraft i in the string. Similarly, the minimum and maximum airspeed limits are converted to airspeed error limits by Eq. (292).

$$V_{e_i}(t) = V_i(t) - V_{ref_i} \quad (290)$$

$$G_{e_i}(t) = G_{m_i}(t) - G_{ref_i}(t) \quad (291)$$

$$V_{e_{min/max_i}} = V_{T_{min/max_i}} - V_{ref_i} \quad (292)$$

Next, the variable $\sigma_i(t)$ is introduced and defined by Eq. (293) such that Eq. (282) may be simplified to Eq. (294). The wind magnitude assumptions allow $\sigma_i(t)$ to be bounded by Eq. (295). Similarly, the variable $\sigma_{ref_i}(t)$ is introduced defined by Eq. (296) which simplifies Eq. (288) in the same manner as Eq. (294).

$$\sigma_i(t) = \sqrt{1 - \frac{W_{cross_i}(t)^2}{V_{T_i}(t)^2}} \quad (293)$$

$$V_{s_i} = \sigma_i(t)V_{T_i}(t) \quad (294)$$

$$\sqrt{1 - F_W^2} \leq \sigma_i(t) \leq 1 \quad (295)$$

$$\sigma_{ref_i}(t) = \sqrt{1 - \frac{W_{cross_i}(t)^2}{V_{ref_i}^2}} \quad (296)$$

Next, the measured groundspeed error is simplified to Eq. (299) by utilizing the

definition of V_{s_i} in Eq. (283) and Eq. (294). However, $V_{e_i}(t)$ is the variable of interest.

$$G_{e_i}(t) = G_{m_i}(t) - G_{ref_i}(t) \quad (297)$$

$$G_{e_i}(t) = V_{s_i}(t) + W_{m_i}(t) - (V_{s_{ref_i}}(t) + W_{m_i}(t)) \quad (298)$$

$$G_{e_i}(t) = \sigma_i(t)V_i(t) - \sigma_{ref_i}(t)V_{ref_i} \quad (299)$$

In order to isolate $V_{e_i}(t)$, a variable, $s_i(t)$, is introduced which is the ratio of groundspeed to airspeed according to Eq. (300). This equation is expanded to Eq. (301) by substituting Eq. (299) and Eq. (293). Since, $s_i(t)$ is a function of wind and airspeed, its magnitude is bounded. Solving for $s_i(t)$ results in Eq. (302) which is mathematically undefined when $V_{T_i}(t) = V_{ref_i}$. However, since the groundspeed error is defined to be zero when the airspeed error is zero, this point will be ignored for now.

$$G_{e_i}(t) = s_i(t)V_{e_i}(t) \quad (300)$$

$$s_i(t)V_{e_i}(t) = \left(\sqrt{V_{T_i}^2(t) - W_{cross_i}(t)^2} - \sqrt{V_{ref_i}^2 - W_{cross_i}(t)^2} \right) \quad (301)$$

$$s_i(t) = \frac{\left(\sqrt{V_{T_i}^2(t) - W_{cross_i}(t)^2} - \sqrt{V_{ref_i}^2 - W_{cross_i}(t)^2} \right)}{V_{T_i}(t) - V_{ref_i}} \quad (302)$$

The lower bound of $s_i(t)$ is 1 which occurs when the magnitude of the crosswind is zero. This case results in the groundspeed error being equal to the airspeed error. The upper bound of $s_i(t)$ occurs with maximum windspeed and minimum airspeed. The derivation of the conditions associated with the limits (e.g., zero airspeed for minimum $s_i(t)$) is presented in Appendix B.1. The derivation also addresses the point where $s_i(t)$ is undefined, zero airspeed error. Based on the assumptions of the problem, $s_i(t)$ is bounded by Eq. (303) which is only dependent on the wind and

airspeed limitations.

$$1 \leq s_i(t) \leq \frac{\sqrt{1 - F_{V_i}^2 F_W^2} - \sqrt{F_{V_i}^2 (1 - F_W^2)}}{1 - F_{V_i}} \quad (303)$$

Virtual Target Control.

Arrival Time Dynamics.

To control the arrival time error and airspeed error, the controller in Ref. [83] implemented a virtual target airspeed control system. This section derives the error dynamics of a similar control system.

Let there be a virtual target for each aircraft which is moving at the reference airspeed/groundspeed along the corresponding path. Next, let the virtual target be positioned such that it will achieve the arrival time constraints if it traverses the remainder of the path at the desired airspeed. If the error between aircraft i and the corresponding virtual target is zero, it will achieve the desired arrival time and desired arrival airspeed simultaneously [45, 83]. Figure 51 presents the virtual target setup if the paths of vehicle i and $i - 1$ are the same.

Let the estimated time remaining of aircraft i , $T_{rem_i}(t)$, be a function of the actual distance remaining, $d_i(t)$, and the average reference groundspeed along the remainder of the path, $G_{ref_{avg_i}}(t)$. The average reference groundspeed is calculated as an average value integral as defined by Eq. (304).

$$G_{ref_{avg_i}}(t) = \frac{1}{(T_{rem_i}(t) + t) - t} \int_t^{T_{rem_i}(t)+t} G_{ref_i}(\tau) d\tau \quad (304)$$

Since a constant desired airspeed is specified, the desired groundspeed varies as the direction of the path relative to the wind changes. The average remaining ground-

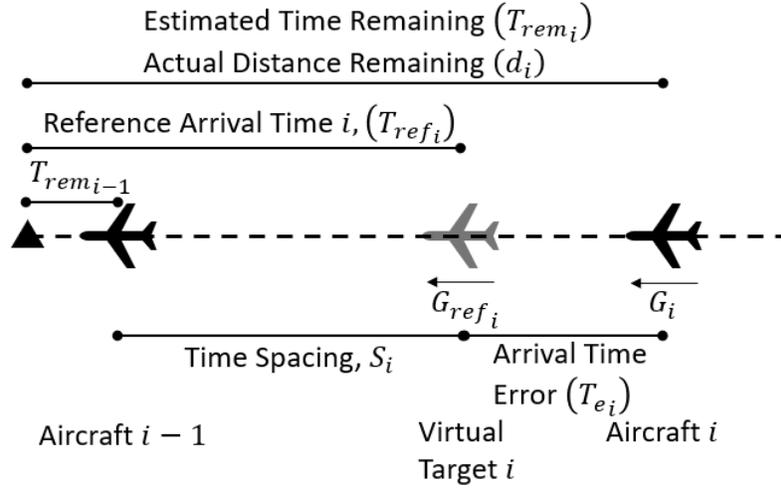


Figure 51. Virtual Target Variables

speed was used in Ref. [83] because it allows the effect of these changes to be captured. The calculation of the average groundspeed assumes the measured wind vector with respect to the ground remains constant for the remainder of the path. Substituting Eq. (304) into Eq. (305) results in Eq. (306).

$$d_i(t) = T_{rem_i}(t)G_{ref_{avg_i}}(t) \quad (305)$$

$$d_i(t) = \int_t^{T_{rem_i}(t)+t} G_{ref_i}(\tau) d\tau \quad (306)$$

The arrival time error is defined as the difference between desired and estimated arrival time, Eq. (307). The variable $T_i(t)$ is the estimated arrival time of aircraft i and T_{ref_i} is the reference arrival time for the aircraft. In the case of the first aircraft, the target arrival time T_{ref_i} is constant. However, in the case of following aircraft, this term is a function of the preceding aircraft arrival time and the desired time spacing, S_i .

$$T_{e_i}(t) = T_i(t) - T_{ref_i}(t) = T_{rem_i}(t) + t - T_{ref_i}(t) \quad (307)$$

$$T_{ref_i}(t) = Constant, \quad i = 1 \quad (308)$$

$$T_{ref_i}(t) = T_{rem_{i-1}}(t) + t + S_i, \quad i > 1 \quad (309)$$

Taking the time derivative of Eq. (307) results in Eq. (310) as a general form of the arrival time error dynamics. Next, the dynamics of the estimated time remaining, $R_i(t)$, are derived.

$$\dot{T}_{e_i}(t) = \dot{T}_{rem_i}(t) + 1 - \dot{T}_{ref_i}(t) \quad (310)$$

Equation (305) may be rearranged to solve for the estimated time remaining as a function of $d_i(t)$ and $G_{ref_i}(t)$, Eq. (311). Note that $G_{ref_i}(t)$ must be positive due to the wind assumptions. Taking the derivative of Eq. (311) results in Eq. (312) which has been simplified using the Leibniz integral rule. This equation is further simplified with a few substitutions.

$$T_{rem_i}(t) = \frac{d_i(t)}{G_{ref_{avg_i}}(t)} = \frac{d_i(t) ((T_{rem_i}(t) + t) - t)}{\int_t^{T_{rem_i}(t)+t} G_{ref_i}(\tau) d\tau} \quad (311)$$

$$\begin{aligned} \dot{T}_{rem_i}(t) = & \frac{T_{rem_i}(t)\dot{d}_i(t) + d_i(t)\dot{T}_{rem_i}(t)}{\int_t^{T_{rem_i}(t)+t} G_{ref_i}(\tau) d\tau} \\ & + \frac{T_{rem_i}(t)d_i(t) \left(G_{ref_i}(t) - G_{ref_i}(t + T_{rem_i}(t)) \left(\dot{T}_{rem_i}(t) + 1 \right) \right)}{\left(\int_t^{T_{rem_i}(t)+t} G_{ref_i}(\tau) d\tau \right)^2} \end{aligned} \quad (312)$$

First, the derivative of distance remaining, \dot{d}_i , is defined as negative groundspeed

since the path is assumed to be perfectly followed, Eq. (313). The reference ground-speed integral term is also equivalent to distance remaining; Eq. (306). Next, the term, $G_{ref_i}(t + T_{rem_i}(t))$, is the reference groundspeed at the estimated arrival time, which occurs at the end of the path by definition. This term is redefined as $G_{ref_{f_i}}(t)$ and is constant if V_{ref_i} and the wind vector are constant. Making these substitutions and collecting $\dot{T}_{rem_i}(t)$ to one side of the equation results in Eq. (314).

$$\dot{d}_i(t) = -G_i(t) \quad (313)$$

$$\dot{T}_{rem_i}(t) = -\frac{G_i(t) - G_{ref_{f_i}}(t)}{G_{ref_{f_i}}(t)} - 1 \quad (314)$$

Returning to the arrival time error, Eq. (314) is substituted into Eq. (310) resulting in Eq. (315). The groundspeed term is separated into its measured and unmeasured components, from Eq. (284), resulting in Eq. (316). Finally, the definition of groundspeed error, Eq. (291), is substituted and the dynamics are separated into three primary terms, Eq. (317): a groundspeed term which includes the measured effects of wind, an unmeasured wind term, and the reference time dynamics which are either constant or a function of the preceding vehicle.

$$\dot{T}_{e_i}(t) = -\frac{G_i(t) - G_{ref_{f_i}}(t)}{G_{ref_{f_i}}(t)} - \dot{T}_{ref_i}(t) \quad (315)$$

$$\dot{T}_{e_i}(t) = -\frac{G_{m_i}(t) + W_{u_i}(t) - G_{ref_{f_i}}(t)}{G_{ref_{f_i}}(t)} - \dot{T}_{ref_i}(t) \quad (316)$$

$$\dot{T}_{e_i}(t) = -\frac{G_{e_i}(t)}{G_{ref_{f_i}}(t)} + \frac{W_{u_i}(t)}{G_{ref_{f_i}}(t)} - \dot{T}_{ref_i}(t) \quad (317)$$

The previously introduced variables, $s_i(t)$ and $\sigma_i(t)$, are then used to obtain dynamics which are a function of airspeed error. First, the final reference groundspeed

is defined by Eq. (318). The variable, $\sigma_{ref_{f_i}}$, is calculated using the reference airspeed and the crosswind at the end of the path while $W_{m_{f_i}}$ is defined as the measured wind-speed projected along a line tangent to the end of the path. Making the appropriate substitutions, the result is Eq. (319).

$$G_{ref_{f_i}}(t) = \sigma_{ref_{f_i}}(t)V_{ref_i} + W_{m_{f_i}} \quad (318)$$

$$\begin{aligned} \dot{T}_{e_i}(t) = & -\frac{s_i(t)}{\sigma_{ref_{f_i}}(t)V_{ref_i} + W_{m_{f_i}}}V_{e_i}(t) \\ & -\frac{W_{u_i}(t)}{\sigma_{ref_{f_i}}(t)V_{ref_i} + W_{m_{f_i}}} - \dot{T}_{ref_i}(t) \end{aligned} \quad (319)$$

Next, V_{e_i} is non-dimensionalized by V_{ref_i} resulting in the non-dimensionalized airspeed error \mathcal{V}_{e_i} being defined by Eq. (320). The non-dimensional airspeed error allows a unified stability analysis to apply to aircraft flying at a wide range of airspeeds. Introducing $\mathcal{V}_{e_i}(t)$ into Eq. (319) results in Eq. (321)

$$\mathcal{V}_{e_i}(t) = \frac{V_{e_i}(t)}{V_{ref_i}} \quad (320)$$

$$\begin{aligned} \dot{T}_{e_i}(t) = & -\frac{s_i(t)V_{ref_i}}{\sigma_{ref_{f_i}}(t)V_{ref_i} + W_{m_{f_i}}}\mathcal{V}_{e_i}(t) \dots \\ & -\frac{W_{u_i}(t)}{\sigma_{ref_{f_i}}(t)V_{ref_i} + W_{m_{f_i}}} - \dot{T}_{ref_i}(t) \end{aligned} \quad (321)$$

Although Eq. (321) is complicated, it is noted that the coefficient to $\mathcal{V}_{e_i}(t)$ and the unmeasured wind term are comprised entirely of bounded variables; therefore, each term is also bounded. The bounded terms are grouped into the multiplicative airspeed term, $m_i(t)$, and the additive windspeed term $w_i(t)$ resulting in Eq. (322). The terms $m_i(t)$ and $w_i(t)$ are defined by Eq. (323) and Eq. (324) respectively. Utilizing the assumptions of the previous section, $m_i(t)$ is bounded by Eq. (325) while $u_i(t)$ is

bounded by Eq. (326).

$$\dot{T}_{e_i}(t) = -m_i(t)\mathcal{V}_{e_i}(t) - w_i(t) - \dot{T}_{ref_i}(t) \quad (322)$$

$$m_i(t) = \frac{s_i(t)V_{ref_i}}{\sigma_{ref_{f_i}}(t)V_{ref_i} + W_{m_{f_i}}} \quad (323)$$

$$w_i(t) = \frac{W_{u_i}(t)}{\sigma_{ref_{f_i}}(t)V_{ref_i} + W_{m_{f_i}}} \quad (324)$$

$$0 < \frac{1}{F_{V_i}F_{W_i} + 1} \leq m_i(t) \leq \frac{F_{V_i}\sqrt{1 - F_{W_i}^2} - \sqrt{1 - F_{V_i}^2F_{W_i}^2}}{(F_{V_i} - 1)\left(-F_{V_i}F_{W_i} + \sqrt{1 - F_{W_i}^2}\right)} \quad (325)$$

$$0 \leq |w_i(t)| \leq \frac{W_{u_{max_i}}}{-V_{ref_i}F_{V_i}F_{W_i} + V_{ref_i}\sqrt{1 - F_{W_i}^2}} \quad (326)$$

Although this may appear to be an oversimplification, the terms $m_i(t)$ and $w_i(t)$ are exogenous with respect to the dynamics because they primarily depend on the magnitude of the wind. To bring some physical meaning to these terms, a positive $w_i(t)$ translates to an unmeasured tailwind component. In the sense of arrival time, a positive $w_i(t)$ would cause the true groundspeed to be greater than the measured groundspeed therefore, the aircraft would arrive early if corrections were not made.

However, the estimated arrival time is also dependent on distance to the endpoint. Thus, in the dynamics, a positive $w_i(t)$ causes the estimated arrival time to decrease, $\dot{T}_{e_i} < 0$. In terms of magnitude, moderate turbulence generated by the MATLAB[®]Dryden Turbulence model varies by approximately ± 30 ft/s. Assuming an aircraft traveling at 293 ft/s and with zero steady state wind, $|w_i(t)| \leq 0.1$ (s/s).

The variable $m_i(t)$ is more convoluted since it is multiplicative and the result of multiple simplifications. However, the entire term $m_i(t)\mathcal{V}_{e_i}(t)$ represents the effect measured groundspeed error has on the arrival time and so $m_i(t)$ is the scale factor which effectively translates airspeed into groundspeed with respect to their effect on arrival time error. Additionally, the numerator of $m_i(t)$ contains the ‘‘local’’ wind

effects which are a function of the current aircraft heading while the denominator terms are based on the windspeed projected along the final direction of the path.

Finally, the derivative of the definitions for reference arrival time of the 1st or i 'th vehicle, the derivative of Eqs. (308)-(309), are substituted into Eq. (322) resulting in the arrival time error dynamics. Equation (327) defines the arrival time error dynamics for the first aircraft while Eq. (328) defines the arrival time error dynamics for the following aircraft.

$$\dot{T}_{e_i}(t) = -m_i(t)\mathcal{V}_{e_i}(t) - w_i(t), \quad i = 1 \quad (327)$$

$$\dot{T}_{e_i}(t) = -(m_i(t)\mathcal{V}_{e_i}(t) + w_i(t)) + (m_{i-1}(t)\mathcal{V}_{e_{i-1}}(t) + w_{i-1}(t)), \quad i > 1 \quad (328)$$

Airspeed Error Dynamics.

The arrival time error is affected by changing the current aircraft's airspeed which is modeled as a rate-limited first order airspeed command system. In order to enforce the rate limit and airspeed limits for each aircraft, let the saturation function be defined by Eq. (329).

$$\int_a^b(x) = \begin{cases} b, & x > b \\ x, & b \geq x \geq a \\ a, & x < a \end{cases} \quad (329)$$

Since the non-dimensional airspeed was defined in the previous section, the upper and lower airspeed rate limits, A_i and B_i respectively, are also scaled by the desired airspeed to conform to the non-dimensionalized airspeed model.

$$\mathcal{A}_i = \frac{A_i}{V_{ref_i}} \quad (330)$$

$$\mathcal{B}_i = \frac{B_i}{V_{ref_i}} \quad (331)$$

Applying the rate-limits and saturation function, the airspeed error dynamics are modeled by Eq. (332) where $u(t)$ is the non-dimensional airspeed error command. The model assumes that the aircraft can achieve a critically damped airspeed response.

$$\dot{\mathcal{V}}_{e_i}(t) = \int_{\mathcal{A}_i}^{\mathcal{B}_i} (u_i(t) - \mathcal{V}_{e_i}(t)) \quad (332)$$

Lemma 6.1. *Take the dynamics defined by Eq. (332) and let \mathcal{M}_i be a compact subset of \mathbb{R} , $\mathcal{A}_i < 0$, and $\mathcal{B}_i > 0$. If $u(t) \in \mathcal{M}_i$ and $\mathcal{V}_{e_i}(t_0) \in \mathcal{M}_i$ then $\mathcal{V}_{e_i}(t) \in \mathcal{M}_i$ for $t \geq t_0$.*

Proof. The system defined by Eq. (332) is stable with an equilibrium point of $u(t) \in \mathcal{M}_i$ when the saturation function is inactive. Without saturation, the system is a single variable linear system with \mathbf{A} matrix, and eigenvalue, equal to -1 . Thus, the unsaturated system is stable and critically damped towards the equilibrium point resulting in no overshoot. Since it is assumed that $\mathcal{V}_{e_i}(t_0) \in \mathcal{M}_i$, therefore $\mathcal{V}_{e_i}(t) \in \mathcal{M}_i$ for $t \geq t_0$ when the system is unsaturated.

Since $0 \in [\mathcal{A}_i, \mathcal{B}_i]$, it follows that

$$\text{sgn} \left(\int_{\mathcal{A}_i}^{\mathcal{B}_i} (u_i(t) - \mathcal{V}_{e_i}(t)) \right) = \text{sgn} (u_i(t) - \mathcal{V}_{e_i}(t)) \quad (333)$$

and $\mathcal{V}_{e_i}(t)$ is driven towards the equilibrium point when the saturation function is active. Furthermore, since $\mathcal{A}_i \neq 0$ and $\mathcal{B}_i \neq 0$, the saturation function must become inactive at some time $t = t_1$ prior to $\mathcal{V}_{e_i}(t)$ reaching the equilibrium point.

Since $\mathcal{V}_{e_i}(t_0) \in \mathcal{M}_i$, then $\mathcal{V}_{e_i}(t_1)$ must also be within \mathcal{M}_i . Therefore, t_1 may be viewed as a new initial condition where the saturation function will not activate again prior to reaching the equilibrium point. ■

Control Bounds.

In the previous section, it was shown that the airspeed error dynamics are bounded given the bounds of the control $u(t)$. This section will determine the appropriate bounds on the control so that an equilibrium arrival time error is reachable for an interconnected system. However, it is assumed that $w_i(t)$ is unknown and therefore may not be used to determine the bounds on the control. Subsequently, the steady state analysis is performed assuming $w_i(t) = 0$. The interconnected arrival time error dynamics with $w_i(t) = 0$ are reproduced in Eq. (334).

$$\dot{T}_{e_i}(t) = -m_i(t)\mathcal{V}_{e_i}(t) + m_{i-1}(t)\mathcal{V}_{e_{i-1}}(t) \quad (334)$$

Let the airspeed bounds of aircraft i at a given point in time be defined as $\mathcal{M}_i(t)$. The equilibrium point of the arrival time error occurs when Eq. (335) is satisfied. The equilibrium point is reachable if there exists a $\mathcal{V}_{e_i}(t) \in \mathcal{M}_i(t)$ for any $\mathcal{V}_{e_{i-1}}(t) \in \mathcal{M}_{i-1}(t)$.

$$m_i(t)\mathcal{V}_{e_i}(t) = m_{i-1}(t)\mathcal{V}_{e_{i-1}}(t) \quad (335)$$

One solution to ensure a reachable equilibrium is to define $\mathcal{M}_i(t)$ in terms of the windspeed and a common airspeed limit. Let $\mathcal{M}_i(t)$ be defined by Eqs. (336)-(338). Recall, that $V_{e_{max_i}}$, $V_{e_{min_i}}$, V_{ref_i} , and m_{min} are constants defined by the assumed aircraft and environmental limits.

$$\mathcal{M}_i(t) = \left[\frac{m_{min}}{m_i(t)} \mathcal{V}_{e_{min}}, \frac{m_{min}}{m_i(t)} \mathcal{V}_{e_{max}} \right] = [\mathcal{V}_{e_i}^-(t), \mathcal{V}_{e_i}^+(t)] \quad (336)$$

$$\mathcal{V}_{e_{max}} = \min_i \left\{ \frac{V_{e_{max_i}}}{V_{ref_i}} \right\}, \quad \mathcal{V}_{e_{min}} = \max_i \left\{ \frac{V_{e_{min_i}}}{V_{ref_i}} \right\} \quad (337)$$

$$m_{min} \leq m_i(t) \quad (338)$$

This formulation creates symmetric limits, $\mathcal{V}_{e_{max}}$ and $\mathcal{V}_{e_{min}}$, which are scaled by the measured windspeed variable, $m_i(t)$. First, $\mathcal{V}_{e_i}^+ \leq \mathcal{V}_{e_{max}}$ and vice versa for the minimum limit, due to Eq. (338). Next, if $\mathcal{V}_{e_{i-1}}(t)$ is assumed to equal the upper bound of $\mathcal{M}_{i-1}(t)$ then the equilibrium point is the upper bound of $\mathcal{M}_i(t)$.

$$m_i(t) \mathcal{V}_{e_i}(t) = m_{i-1}(t) \mathcal{V}_{e_{i-1}}^+(t) \quad (339)$$

$$m_i(t) \mathcal{V}_{e_i}(t) = m_{i-1}(t) \frac{m_{min}}{m_{i-1}(t)} \mathcal{V}_{e_{max}} \quad (340)$$

$$\mathcal{V}_{e_i}(t) = \frac{m_{min}}{m_i(t)} \mathcal{V}_{e_{max}} \quad (341)$$

Therefore, the arrival time error equilibrium point is always reachable if $\mathcal{V}_{e_i}(t) \in \mathcal{M}_i(t)$ where $\mathcal{M}_i(t)$ is defined by Eqs. (336)-(338).

Control Selection.

This research investigates a nonlinear control formulation which utilizes information assumed to be available to the racetrack algorithm, arrival time error and airspeed error. This control minimized the amount of information required to be transmitted between aircraft; estimated arrival time of the previous vehicle and whether the current vehicle is the leader or follower.

First, the airspeed command, $u_i(t)$, is scaled by the variable $m_i(t)$ which is hypothesized to improve performance among aircraft experiencing different wind conditions.

This results in airspeed dynamics now being defined by Eq. (342).

$$\dot{\mathcal{V}}_{e_i}(t) = \int_{\mathcal{A}_i}^{\mathcal{B}_i} \left(\frac{u_i(t)}{m_i(t)} - \mathcal{V}_{e_i}(t) \right) \quad (342)$$

Next, two proportional gains are applied to the arrival time error and non-dimensional airspeed error, k_T and k_V respectively. Combining the assumed control bounds and airspeed control into the dynamics, the non-dimensional airspeed error dynamics are defined by Eq. (343) where $\mathcal{V}_{e_i}^+(t)$ and $\mathcal{V}_{e_i}^-(t)$ are defined in the previous section by Eq. (336).

$$\dot{\mathcal{V}}_{e_i}(t) = \int_{\mathcal{A}_i}^{\mathcal{B}_i} \left[\int_{\mathcal{V}_{e_i}^-(t)}^{\mathcal{V}_{e_i}^+(t)} \left(\frac{k_T}{m_i(t)} T_{e_i}(t) + \frac{k_V}{m_i(t)} \mathcal{V}_{e_i}(t) \right) - \mathcal{V}_{e_i}(t) \right] \quad (343)$$

6.3 String Stability Analysis

The combined non-dimensionalized arrival time and airspeed error dynamics are summarized by Eq. (344) through Eq. (348) where m_{min} may be calculated as the lowest lower bound of Eq. (325) for all vehicles in the system. This section analyzes the stability of the system with and without unmeasured airspeed disturbances $w_i(t)$.

$$\dot{T}_{e_i}(t) = \begin{cases} -m_i(t)\mathcal{V}_{e_i}(t) - w_i(t) & , i = 1 \\ -(m_i(t)\mathcal{V}_{e_i}(t) + w_i) + (m_{i-1}(t)\mathcal{V}_{e_{i-1}}(t) + w_{i-1}(t)) & , i > 1 \end{cases} \quad (344)$$

$$\dot{\mathcal{V}}_{e_i}(t) = \int_{\mathcal{A}_i}^{\mathcal{B}_i} \left[\int_{\mathcal{V}_{e_i}^-(t)}^{\mathcal{V}_{e_i}^+(t)} \left(\frac{k_T}{m_i(t)} T_{e_i}(t) + \frac{k_V}{m_i(t)} \mathcal{V}_{e_i}(t) \right) - \mathcal{V}_{e_i}(t) \right] \quad (345)$$

$$\mathcal{V}_{e_i}^-(t) = \frac{m_{min}}{m_i(t)} \mathcal{V}_{e_{min}}, \quad \mathcal{V}_{e_i}^+(t) = \frac{m_{min}}{m_i(t)} \mathcal{V}_{e_{max}} \quad (346)$$

$$\mathcal{V}_{e_{min}} = \min_i \left\{ \frac{V_{e_{min_i}}}{V_{ref_i}} \right\}, \quad \mathcal{V}_{e_{max}} = \min_i \left\{ \frac{V_{e_{max_i}}}{V_{ref_i}} \right\} \quad (347)$$

$$m_{min} \leq m_i(t) \quad (348)$$

First, the areas where the rate limit or saturation are active are characterized by analyzing the airspeed error dynamics, Eq. (345). The positively limited areas are defined by Eq. (349)-(356) where \mathcal{M} is the set of reachable airspeeds, \mathcal{R}^+ is the set where the positive rate limit is active, \mathcal{W}^+ is the set where the positive command saturation is active but not the positive rate-limit, and \mathcal{U} is the set where no limits are active (the unlimited dynamics). Note that the “-” in Eq. (355)-Eq. (356) refers to a set difference.

$$\mathcal{M}_i(t) = \left\{ \mathcal{V}_{e_i}(t) : \frac{m_{min}}{m_i(t)} \mathcal{V}_{e_{min}} \leq \mathcal{V}_{e_i}(t) \leq \frac{m_{min}}{m_i(t)} \mathcal{V}_{e_{max}} \right\} \quad (349)$$

$$\mathcal{S}_i^+(t) = \left\{ T_{e_i}(t), \mathcal{V}_{e_i}(t) \in \mathcal{M}_i(t) : T_{e_i}(t) \geq \frac{m_{min}}{k_T} \mathcal{V}_{e_{max}} - \frac{k_V}{k_T} \mathcal{V}_{e_i} \right\} \quad (350)$$

$$\mathcal{Q}_i^+(t) = \{ T_{e_i}(t), \mathcal{V}_{e_i}(t) \in \mathcal{M}_i(t) : T_{e_i}(t) \geq \Omega_i(t) \} \quad (351)$$

$$\Omega_i(t) = \frac{m_i(t)}{k_T} \left(\left(1 - \frac{k_V}{m_i(t)} \right) \mathcal{V}_{e_i}(t) + \mathcal{B}_i \right) \quad (352)$$

$$\mathcal{P}_i^+(t) = \left\{ \mathcal{V}_{e_i}(t) : \mathcal{V}_{e_i}(t) \leq \frac{1}{m_i(t) - k_V} (m_{min} \mathcal{V}_{e_{max}} - \mathcal{B}_i m_i(t)) \right\} \quad (353)$$

$$\mathcal{R}_i^+(t) = \mathcal{Q}_i^+(t) \cap \mathcal{P}_i^+(t) \quad (354)$$

$$\mathcal{W}_i^+(t) = \mathcal{S}_i^+(t) - \mathcal{P}_i^+(t) \quad (355)$$

$$\mathcal{U}_i(t) = \mathcal{M}_i(t) - \mathcal{R}_i^+(t) - \mathcal{W}_i^+(t) - \mathcal{R}_i^-(t) - \mathcal{W}_i^-(t) \quad (356)$$

The areas are presented graphically in Fig. 52 for clarity. Similar definitions may be derived for the negative limits utilizing \mathcal{V}_e^- , \mathcal{A}_i , and changing the sign of all the inequalities except for \mathcal{M} .

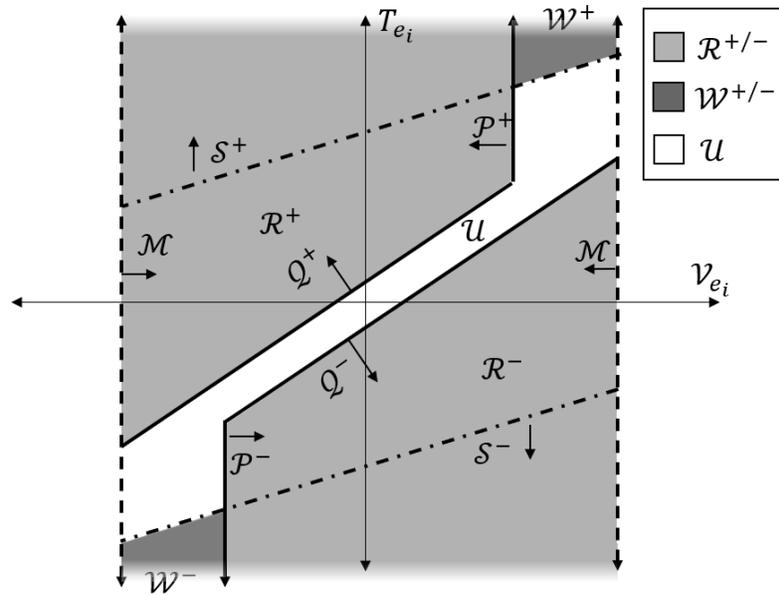


Figure 52. Rate-Limiting and Saturated Sets

Equilibrium Point.

The airspeed equilibrium point of Eqs. (344)-(345) is important to this problem because the range of permissible airspeeds is limited by the command saturation representing the limits of the aircraft. With regards to the stability analysis of the system, the airspeed equilibrium point is reachable where the equilibrium point defined by Eq. (357) must be reachable, $\mathcal{V}_{e_{ieq}}(t) \in \mathcal{M}_i(t)$.

$$\mathcal{V}_{e_{ieq}}(t) = \frac{m_{i-1}(t)\mathcal{V}_{e_{i-1}}(t) + w_{i-1}(t) - w_i(t)}{m_i(t)} \quad (357)$$

Since \mathcal{M}_i is defined by the measured windspeed, it is possible to violate the assumption for certain combinations of $w_i(t)$ and w_{i-1} . However, it is hypothesized that the random nature of turbulence will only result in small intervals where $\mathcal{V}_{e_{ieq}}(t) \notin \mathcal{M}_i(t)$. Although the response of these intervals is not bounded by this paper, the response is expected to remain relatively small. Simulations of this condition are discussed in the results. Simulations of this condition are discussed in Sec. 7.4.

Simplifying the Time Variant System.

To address the command saturation and rate-limit non-linearities, variables are introduced to transform the nonlinear system with command saturation and rate-limits, Eqs. (344)-(345), into a which may be viewed as a polytropic linear differential inclusion (LDI). The benefit is that polytropic LDIs may be used to determine gains resulting in an exponentially stable system.

When saturated, the following inequality must be true, by definition,

$$\mathcal{V}_{e_i}^+ \leq \left| \frac{k_T}{m_i(t)} T_{e_i}(t) + \frac{k_Y}{m_i(t)} \mathcal{V}_{e_i}(t) \right|. \quad (358)$$

The variable $\delta(t)$ is introduced to replace the saturation function. If the system is unsaturated, $\delta(t)$ is defined to equal one. When saturated, $\delta(t)$ will vary between $0 < \delta(t) \leq 1$ to satisfy Eq. (359).

$$\mathcal{V}_{e_i}^{+/-} = \delta(t) \left(\frac{k_T}{m_i(t)} T_{e_i}(t) + \frac{k_Y}{m_i(t)} \mathcal{V}_{e_i}(t) \right) \quad (359)$$

The process is repeated with the rate-limit saturation function. The variable $\rho(t)$ is introduced to model the rate limit resulting in Eq. (360). When the rate-limit is active, Eq. (361) will be satisfied, and when the rate-limit is inactive, $\rho(t)$ is defined to be one. It follows that $\rho(t)$ is bounded identically to $\delta(t)$; $0 < \rho(t) \leq 1$.

The system, Eq. (360) models the effects of the unlimited, saturated, and rate-limited system. The variables $\delta(t)$ and $\rho(t)$ are dependent on the state variables and the degree to which Eq. (360) is linear may be questioned. However, the variables are also bounded and in the next section, it will be shown if a linear system has a bounded \mathbf{A} matrix, then the stability of the system may be determined for any \mathbf{A} within the defined bounds.

$$\begin{aligned} \begin{Bmatrix} \dot{T}_{e_i}(t) \\ \dot{\mathcal{V}}_{e_i}(t) \end{Bmatrix} &= \begin{bmatrix} 0 & -m_i(t) \\ \rho(t)\delta(t)\frac{k_T}{m_i(t)} & \rho(t)\left(\delta(t)\frac{k_Y}{m_i(t)} - 1\right) \end{bmatrix} \begin{Bmatrix} T_{e_i}(t) \\ \mathcal{V}_{e_i}(t) \end{Bmatrix} \\ &+ \begin{bmatrix} 0 & m_{i-1}(t) \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} T_{e_{i-1}}(t) \\ \mathcal{V}_{e_{i-1}}(t) \end{Bmatrix} + \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} w_i(t) \\ w_{i-1}(t) \end{Bmatrix} \end{aligned} \quad (360)$$

$$\mathcal{A}_i \text{ or } \mathcal{B}_i = \rho(t) \left(\delta(t) \left(\frac{k_T}{m_i(t)} T_{e_i}(t) + \frac{k_Y}{m_i(t)} \mathcal{V}_{e_i}(t) \right) - \mathcal{V}_{e_i}(t) \right) \quad (361)$$

Single Vehicle Stability.

The stability of the system, Eq. (360), is first analyzed as a polytopic LDI allowing constraints to be derived which guarantee exponential stability, defined by Def. 6.4. The problem is then formulated and solved numerically as an optimization problem. First, an overview of polytopic LDI stability is given. From Ref. [84], for a system defined by

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}, \quad \mathbf{A}(t) \in \{\mathbf{A}_1, \dots, \mathbf{A}_C\} \quad (362)$$

where the set $\{\mathbf{A}_1, \dots, \mathbf{A}_C\}$ is convex, the necessary and sufficient conditions for quadratic stability are

$$\mathbf{P} > 0, \quad \mathbf{A}_j^T \mathbf{P} + \mathbf{P} \mathbf{A}_j < 0, \quad j = 1, \dots, C. \quad (363)$$

Simply put, if Eq. (363) is satisfied and $\mathbf{A}(t) \in \{\mathbf{A}_1, \dots, \mathbf{A}_C\}$, then the system is exponentially stable for any $\mathbf{A}(t) \in \{\mathbf{A}_1, \dots, \mathbf{A}_C\}$.

In the case of Eq. (360), the vertices of the convex set which contains $\mathbf{A}(t)$, \mathbf{A}_j , correspond to the eight possible combinations of the upper and lower bounds of $m_i(t)$, $\delta(t)$, and $\rho(t)$. Although $\delta(t)$ and $\rho(t)$ may be dependent on the state, they are bounded allowing for Eq. (360) to be represented in the form Eq. (362).

To determine the matrix P , three optimization design (i.e., independent) variables are defined. The first two are the two elements of a diagonal matrix Q which are part of the Lyapunov equation, Eq. (364). The third variable is the identifier for a vertex, $n \in \{1, 2, \dots, C\}$, of the set $\{\mathbf{A}_1, \dots, \mathbf{A}_C\}$ to be used to solve Eq. (364).

The three variables allow the calculation of a symmetric, positive definite matrix, \mathbf{P} , which satisfies Eq. (364). The matrix \mathbf{P} is calculated for vertex n and then used to evaluate the stability constraints at the remaining vertices, Eq. (363).

$$\mathbf{A}_n^T \mathbf{P} + \mathbf{P} \mathbf{A}_n = -\mathbf{Q} \quad (364)$$

The exponential stability constraints and P matrix definition are then formulated into an optimization problem defined by Eq. (365). The design variables of this problem are the system gains, elements of the diagonal matrix \mathbf{Q} , and the guessed vertex, n , used to calculate the matrix \mathbf{P} . The cost functional to be minimized, J , may be chosen based on the desired performance of the system. As an example, maximizing the value of k_γ tends to decrease the settling time of the system. However, the stability of the system does not depend on J , only on the constraints.

$$\begin{aligned} & \underset{q_1, q_2, k_T, k_\gamma, n}{\text{minimize}} && J \\ & \text{subject to} && \mathbf{A}_n^T \mathbf{P} + \mathbf{P} \mathbf{A}_n = - \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}, \\ & && 1 \leq n \leq 8, \\ & && \mathbf{A}_j^T \mathbf{P} + \mathbf{P} \mathbf{A}_j < 0, \quad j = 1, \dots, 8, \\ & && q_1, q_2, k_T, -k_\gamma > 0 \end{aligned} \quad (365)$$

To numerically solve the optimization problem, estimates on reasonable lower bounds of $\delta(t)$ and $\rho(t)$ are made using Eq. (366) and Eq. (367) respectively. These estimates must be made since the lower bound for any state, $\{T_{e_i}(t), \mathcal{V}_{e_i}(t)\}$, approaches but does not reach zero. This is a limitation of numerical solvers which are designed to address \leq types of inequalities rather than $<$ inequalities.

$$\delta_{min} = \frac{\mathcal{V}_{e_{max}} m_{min}^2}{m_{max} (k_T T_{e_{max}} + k_V \mathcal{V}_{e_{min}})} \quad (366)$$

$$\rho_{min} = \frac{\min_i (\min\{|\mathcal{A}_i|, |\mathcal{B}_i|\})}{2\mathcal{V}_{e_{max}}} \quad (367)$$

Equation (366) assumes some worst case $T_{e_{max}}$ and solves Eq. (359) for the smallest value while Eq. (367) assumes a saturated system with \mathcal{V}_{e_i} the opposite sign of the command. The numerator determines the lowest magnitude rate-limit of each aircraft then selects the smallest. Since a successful solution results in an exponentially stable system, this formulation effectively assumes the initial arrival time error may be bounded by $\pm T_{e_{max}}$.

This assumption is valid in the context of the racetrack algorithm since the dynamics in this section become active when the racetrack size becomes fixed. Since a free racetrack size is continually adjusting to set the arrival time error to zero, it is expected that the arrival time error will also be near zero when the racetrack transitions to a fixed state.

The problem was successfully formulated and numerically solved utilizing the MATLAB[®] routine `fmincon`. To summarize, if $|T_{e_i}(t)| \leq T_{e_{max}}$, Eq. (357) is satisfied, and a solution to Eq. (365) is found, then the system is exponentially stable.

String Stability Analysis.

Finally, the string stability of Eqs. (344)-(345) is addressed. Let there be a finite number of aircraft in the string, $i \in \mathcal{I}_N = \{1, \dots, N\}$. Furthermore, let $\mathbf{A}(t)$ be a member of the convex set defined by $\{\mathbf{A}_1, \dots, \mathbf{A}_C\}$ whose vertices correspond to the bounds of $m_i(t)$, $\delta_i(t)$, $\omega_i(t)$; $[m_{min}, m_{max}]$, $[\delta_{min}, 1]$, $[\omega_{min}, 1]$ respectively.

Lemma 6.2. *Assume that $|w_i(t)| < r$ for some positive constant r , $T_{e_{max}} \geq \Delta$, and*

$\mathcal{V}_{e_{ieq}}(t) \in \mathcal{M}_i(t)$ is satisfied for all t . If a feasible solution to Eq. (365) exists, then the system defined by Eqs. (344)-(345) is string stable according to Definition 6.1.

Proof. Equations (344)-(345) are equivalent to the LTV system defined by Eq. (360) with state variables T_{e_i} and \mathcal{V}_{e_i} and input variables $T_{e_{i-1}}$, $\mathcal{V}_{e_{i-1}}$, w_i , and w_{i-1} . The system is globally Lipschitz in \mathcal{M} since $m_{i-1}(t)$ is bounded and continuously differentiable. Additionally, the unforced system is exponentially stable, Def. 6.4 since a solution to Eq. (365) exists. Therefore, the Eq. (360) is also input-to-state stable which yields Eq. (368) where $\mathbf{x}_i(t)$ is the state vector of aircraft i while γ , ξ , and v are class \mathcal{K} functions (see Lemma 4.6 in Ref. [82]). Equivalently, the nonlinear system defined by Eq. (360) is also input-to-state stable; from Definition 6.3.

$$\begin{aligned} \|\mathbf{x}_i(t)\|_\infty &\leq \beta(\|\mathbf{x}_i(t_0)\|_\infty, t - t_0) + \gamma \left(\sup_{\tau \geq t_0} |\mathcal{V}_{e_{i-1}}(\tau)| \right) \\ &\quad + \xi \left(\sup_{\tau \geq t_0} |w_i(\tau)| \right) + v \left(\sup_{\tau \geq t_0} |w_{i-1}(\tau)| \right) \end{aligned} \quad (368)$$

Next, $\mathcal{V}_{e_i}(t)$ is uniformly bounded for all aircraft i due to the identical limits of the saturation function defined by Eq. (347). Additionally, it has been assumed that $w_i(t)$ is uniformly bounded. Then Eq. (368) may be rewritten as Eq. (369) where Γ , Ξ , and Υ are positive constants associated with the largest values of γ , ξ , and v respectively.

$$\|\mathbf{x}_i(t)\|_\infty \leq \beta(\|\mathbf{x}_i(t_0)\|_\infty, t - t_0) + \Gamma + \Xi + \Upsilon \quad (369)$$

The expression is then evaluated over all aircraft i resulting in Eq. (370). Since β is a class \mathcal{KL} function, it is bounded as a function of $\|x_i(t_0)\|$. Therefore, the entire

system is bounded by some positive Δ independent of time which is equivalent to Definition 6.1. Therefore, the system under the stated assumptions is string stable in the case of non-zero disturbances.

$$\sup_{i \in \mathcal{I}_N} \|\mathbf{x}_i(t)\|_\infty \leq \beta \left(\sup_{i \in \mathcal{I}_N} \|\mathbf{x}_i(t_0)\|_\infty, t - t_0 \right) + \Gamma + \Xi + \Upsilon \leq \Delta \quad (370)$$

Note that the condition, $T_{e_{max}} \geq \Delta$, may be verified via simulation after determining the control gains which satisfy Eq. (365). ■

Corollary 6.2.1. *The system defined by Eqs. (344)-(345) is asymptotically string stable with no external disturbances, $w_i(t) = 0$.*

Proof. From Theorem 6.2, each individual aircraft is input-to-state stable, Eq. (371). In the case of the first aircraft, $i = 1$, the input is zero ($\mathcal{V}_{e_{i-1}} = 0$). Subsequently, the input to the second aircraft, $\mathcal{V}_{e_{i-1}}(t)$, will be bounded by Eq. (371) as shown in Eq. (372).

$$\|\mathbf{x}_1(t)\|_\infty \leq \beta(\|\mathbf{x}_1(t_0)\|_\infty, t - t_0) \quad (371)$$

Subsequently, the input of the second aircraft will also approach zero as time approaches infinity. This logic may be cascaded to all vehicles such that the input to aircraft N approaches zero as time approaches infinity, Eq. (372). Since the system is string stable and Eq. (372) holds, the system with no external disturbances meets the definition of asymptotic string stability, Definition 6.2.

$$\sup_{i \in \mathcal{I}_N} \|\mathbf{x}_i(t)\|_\infty \rightarrow 0, t \rightarrow \infty \quad (372)$$



6.4 Stability Analysis Summary

This chapter performed a string stability analysis on the controller used by the racetrack algorithm described in Chapter V to control aircraft airspeed. The analysis derived a method to select control gains which result in a string stable system for a variety of wind conditions to include unmeasured turbulence. The analysis guarantees that each individual aircraft is input-to-state stable and that the error state of an entire string of aircraft is bounded.

Applying this guarantee to the racetrack algorithm, the lead aircraft is exponentially stable towards zero arrival time error with a small tolerance which is dependent on the magnitude of turbulence. This tolerance represents the expected arrival time error range when an aircraft completes a nominal orbit.

The arrival time error of each of the following aircraft is also bounded and independent of the number of vehicles. This ensures that the arrival time error of the second vehicle will be near the origin when the lead aircraft arrives and the second aircraft becomes the lead aircraft. Therefore, the aircraft which is about to arrive will always be exponentially stable to the origin. The results of this method are analyzed in the next chapter.

VII. Simulation Results and Performance Characterization

The performance characterization of the arrival time algorithm discussed in this research is split in four parts. First, two early versions of the algorithm are compared and a single version selected for further evaluation. Next, the arrival time stability of is analyzed based on the gains selected using the methods described in Chapter VI. This analysis evaluates the string stability assuming zero path error and presents an analysis of the effect of tracking error on the arrival time.

Then, the single vehicle performance of the selected controller is characterized across a wide range of environment and target scenarios. Finally, the performance of the controller and its various features are demonstrated in a set of challenging multi-vehicle scenarios utilizing both the simplified and full nonlinear aircraft models discussed in Chapter III.

7.1 Performance Metrics and Variables

The primary performance metric is the arrival time error calculated as the difference between the actual and desired arrival times. First, the actual arrival point is determined as the point in space when the aircraft was closest to the target during the terminal portion of an orbit. To prevent the initial point from being identified as the arrival point, the evaluation algorithm only considers points which are at a time greater than half the desired arrival time.

The actual arrival time is the time associated with the actual arrival point. The secondary performance metric is arrival airspeed error calculated as the difference between actual and desired airspeed at the arrival point.

Next, several variables are non-dimensionalized to evaluate the algorithm across aircraft traveling at different airspeeds. First, the desired arrival time is non-dimensionalized

by a “minimum orbit time,” and the resulting parameter is referred to as the track-time scale-factor ($TTSF$), Eq. (373). The minimum orbit time, $T_{minorbit}$ is calculated as the time to complete a circular orbit with zero wind at the desired airspeed. Equation (374) defines $T_{minorbit}$. The nominal or reference bank angle, ϕ_{ref} , is the bank angle which corresponds to ground turn radius of b with zero wind. As an example, a $TTSF$ of 1.0 sets the desired arrival time equal to the predicted time to complete a circular orbit without wind. This formulation was chosen because the reference orbit size is the minimum size of the racetrack.

$$TTSF = \frac{T_{ref}}{T_{minorbit}} \quad (373)$$

$$T_{minorbit} = \frac{2\pi}{\dot{\psi}_{ref}} = \frac{2\pi V_{ref}}{g \tan \phi_{ref}} \quad (374)$$

Next, the wind speed and target ground speed are non-dimensionalized by the aircraft reference speed, V_{ref} , and are referred to as the wind speed ratio and target speed ratio respectively. In the non-cooperative simulations, the target stop-time ratio is used to define when the target stops. It is calculated as the actual target stop time divided by the desired arrival time with respect to the beginning of an orbit.

Finally, a notable derived variable is the direction of the wind or target relative to the $y^{(r)}$ axis. This variable is calculated via Eq. (375) and combines the effects of turn direction and wind or target direction.

$$\psi_{tgt/wind}^{(r)} = \psi_{tgt/wind}^{(e)} - \left(\psi_f^{(e)} + \frac{\pi}{2} (-D_T) \right) \quad (375)$$

7.2 Solver Configurations

The racetrack algorithm and dynamics model are modeled and evaluated using MATLAB[®] and the associated modeling software Simulink; MATLAB[®] version 2019b was used to produce all simulations. Since Simulink is designed to handle a variety of dynamics models, multiple types of numerical solvers are available to integrate the dynamics. The primary considerations in choosing a solver are the number and types of states. The racetrack algorithm is simply a discrete command system with a small number of states only relating to signal filters. The arrival time control has no states since the primary command is proportional to airspeed and arrival time, which are explicitly calculated based on the current aircraft state. The simplified model is an entirely continuous model while the nonlinear model is a mix of continuous states and discrete states utilized in the various autopilots. The specific solvers are discussed below but were chosen for accuracy in evaluating the different types of models based on the recommendations in the MATLAB[®] documentation [85].

The simplified dynamics model is solved utilizing the MATLAB[®] solver *ode45*, with default settings, since the model is primarily made up of continuous states. The solver *ode45* is a fifth-order variable-step Runge-Kutta numerical integration method based on the formulae developed by Dormand and Prince [86]. Variable step methods change the integration step size at each integration based on an estimate of the solution error at the current time step. In general, the error of variable-step Runge-Kutta methods is determined by comparing the solutions of two different orders. In the case of *ode45*, the fifth order solution is compared to a fourth order solution to determine the local error. When the error increases above some specified tolerance, the time step is decreased until the error is below some specified tolerance. This method allows the numerical integration to remove time steps when the dynamics are slowly varying and add time steps with rapidly changing dynamics, increasing the

overall efficiency of the algorithm.

The accuracy of the simplified model solution calculated with *ode45* was qualitatively evaluated by recomputing the solution with a lower relative tolerance as recommended by the MATLAB® documentation [87]. The accuracy was evaluated using the non-cooperative scenario in Sec. 7.7 which results in large changes to the endpoint. The accuracy evaluation was performed by lowering the relative tolerance from the default value of 1×10^{-3} to 1×10^{-4} . The arrival time error of any vehicle changed by less than 0.01 s and the arrival airspeed error changed by less than 0.1 ft/s. The arrival tracking error of any vehicle changed by less than 1 ft. Therefore, it is concluded the solver *ode45* with default settings provides accurate solutions to simplified model scenarios.

The increased complexity of the nonlinear dynamics model necessitates a different approach to evaluating the Simulink model. Initial simulations showed the *ode45* algorithm utilized for the simplified model is inefficient in solving multi-aircraft simulations using the nonlinear model. This behavior is expected due to the mix of continuous and discrete states in the nonlinear model which may cause a variable-step solver to try and accurately model sharp step changes in the dynamics [85]. A fixed-step solver was used due to the greater number of discrete states in the model [85]. The solver *ode8* was used with a fixed time-step of 0.05 s which corresponds to the smallest discrete step in the model, the aircraft control system.

The solver *ode8* is an eighth-order fixed-step Runge-Kutta method also developed by Dormand and Prince [88]. Similar to the simplified model, the accuracy was evaluated using the multi-aircraft nonlinear non-cooperative scenario in Sec. 7.8 to evaluate the solution with large changes to the endpoint. The accuracy evaluation was performed by lowering the time-step from the default value of 0.05 s to 0.01 s. The arrival time error of any vehicle changed by less than 0.02 s and the arrival airspeed

error changed by less than 0.1 ft/s . The arrival tracking error of vehicles with low arrival time error changed by less than 2 ft . However, the scenario resulted in an initial vehicle with a higher arrival time error and the path tracking error changed by 57 ft . Although high, this error is acceptable since it is associated with an aircraft which did not arrive on time and is still well within the demonstrated tracking error performance. Therefore, it is concluded that the solver *ode8* with a fixed time step of 0.05 s provides accurate solutions to nonlinear model scenarios.

7.3 Initial Controller Performance Comparison

In Chapter V, two different arrival time estimation methods were developed as potential candidates for the final racetrack algorithm. The two methods are referred to as the “Quad” controller and the “Nav” controller. The Quad controller follows a ground-fixed path, Sec. 5.4, while the Nav controller assumes a constant bank angle during turns, Sec. 5.4. This approach was chosen as it allowed the initial development to focus on two timing methods, chose the better performing method, and focus the remaining development effort on a single method.

In the initial performance comparison, the arrival time error is evaluated as the average absolute value of the arrival time error for a given set of data. This provides a single variable to compare the two time-estimation methods. Finally, the initial performance comparison only evaluated the fighter aircraft type since initial testing showed similar performance among the aircraft types.

Controller Configurations.

The initial controllers tested do not have the full functionality discussed in Chapter V and were not analyzed for string stability. The controllers do not have the re-orbit, Sec. 5.8 or keep-out-zone avoidance Sec. 5.8 enabled. This allows the initial

comparison to focus on performance in a more benign environment.

The following airspeed command gains were used in the tested initial controllers:

$$k_T = 0.07 \tag{376}$$

$$k_V = 0 \tag{377}$$

The gains were selected to provide well damped single vehicle performance.

Static Target Simulations.

The static target simulations evaluated the performance of the controllers while varying the following parameters:

- *TTSF*: 1 to 1.7
- Wind Speed Ratio: 0 to 0.5 (half the reference airspeed)
- Wind Direction: 0 deg to 360 deg in increments of 45 degrees
- Orbit Direction: Counterclockwise

High wind speeds were studied in this comparison to determine the point where the controllers' arrival time performance degraded. The contours effectively define a "feasible set" of arrival times and wind speeds which achieve some average performance defined by the contour lines. As an example, the one-second feasible set for a controller would be all combinations of wind speed and *TTSF* below the one-second contour for the controller.

The "Quad" controller clearly has a larger feasible set than the "Nav" controller. The "Nav" controller shows a slight increase in performance with an increase in *TTSF* but it appears that there is a limit to the wind speed the "Nav" controller can compensate for regardless of *TTSF* in this configuration; approximately a 0.3

wind speed ratio to achieve an average arrival time error of one second. However, arrival time performance between the two controllers was similar within the one-second feasible set of the “Nav” controller. This indicates that both controllers achieve similar arrival time errors in “easier” scenarios with lower wind speeds and higher desired arrival times. It was also observed that the racetrack size was saturated on the minimum bound in scenarios with an absolute value of error greater than one. This indicates that airspeed control alone cannot compensate for high wind speeds in every situation.

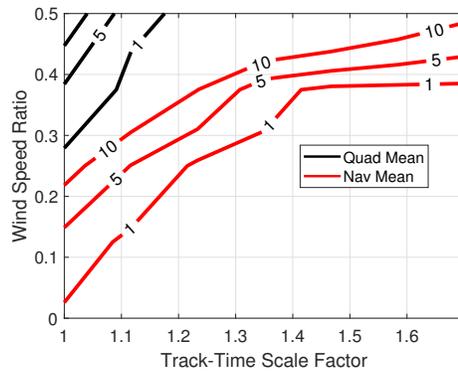


Figure 53. Initial Comparison Static Endpoint Arrival Time Error (s)

The arrival airspeed performance is not presented because it is closely related to arrival time performance and results in a similar relationship as shown in Fig. 53. This is because arrival airspeed is not being directly controlled; the commanded airspeed is simply a function of current arrival time error. Arrival airspeed errors of 67 ft/s tend to correspond to arrival time errors of one second for both controllers, indicating a saturated airspeed condition.

The smaller feasible set of the “Nav” controller is due to the assumption of a constant bank angle turn. This effect is best shown by Fig. 54 with two scenarios where the position of each aircraft at equal points in time is noted. In both scenarios, the *TTSF* is 1.4 and the wind direction is 0 deg (North). In Fig. 54a, the wind speed ratio was 0.2 and both controllers achieved an arrival time error less than 0.1

seconds. However, the semi-major axis of the “Nav” racetrack is smaller due to the longer length of the turns.

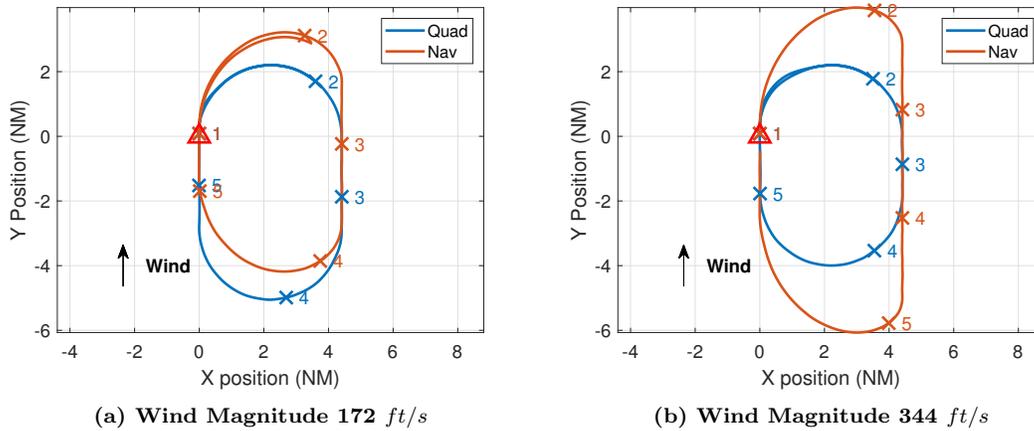


Figure 54. Ground Track Comparison, ‘×’ Markers are Equivalent Time

The same scenario with a wind magnitude of 0.4 knots is shown in Fig. 54b. In this case, the “Nav” controller had an arrival time error of 12 seconds while the “Quad” controller arrival time error was -0.3 seconds. This presents a case where the scenario is outside the one-second “feasible set” of the “Nav” controller but within the one-second “feasible set” of the “Quad” controller. The ground track of the “Nav” controller is clearly longer than the “Quad” controller and equivalent time marks, the ‘×’, show that the “Nav” controller was always behind the “Quad” controller. Although, the “Nav” controller did not achieve the desired time, it accurately predicted it would not achieve the time and commanded maximum airspeed during the entire simulation.

Finally, Fig. 55 presents a comparison of the average time-of-arrival error as it varies with wind direction. The results highlight the skewing effect of the wind on the “Nav” controller racetrack which results in higher time-of-arrival errors for wind directions parallel to the $x^{(r)}$ -axis of the racetrack; 0 and 180 degrees. Conversely, crosswinds, wind directions of 90 and 270 degrees, relative to the $x^{(r)}$ -axis result in

much lower time-of-arrival error which are similar to the “Quad” controller.

Crosswinds have a lower effect on the time-of-arrival error because they do not substantially increase the path length. As an example, holding the bank angle constant and increasing the crosswind would change the resulting $y^{(r)}$ -axis position of the aircraft at the end of a turn which would not line up with the desired straight segment. However, the controller calculations solve for the bank angle which achieves the correct $y^{(r)}$ -axis position compensating for the effects of wind in this direction. The “Quad” controller performance is not affected by wind direction since the path of the “Quad” controller does not change with the wind.

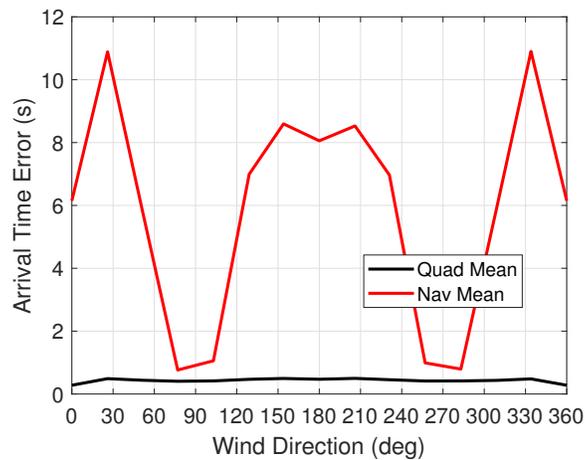


Figure 55. Static Endpoint Arrival Time Errors (s) v. Wind Direction (deg)

Moving Target Simulations.

Multi-pass Simulations.

The multi-pass simulations create scenarios where the aircraft accomplishes multiple passes on a target moving with constant speed and heading. This simulates a continuous monitoring scenario with a moving target. In this initial comparison, the scenarios are setup such that the aircraft begins over the point where the target will be at the desired arrival time for the first pass. After the first pass the aircraft must

fly to the path associated with the next target endpoint.

The arrival time performance for the last pass in each simulation is used as the data point for each simulation. The simulations result in a step change in the endpoint, similar to the non-cooperative scenarios except that the change occurs at the beginning of each pass. In this initial comparison, the wind direction and magnitude were not varied to only compare the effects of a moving target. Finally, the parameters varied in this comparison are:

- *TTSF* 1.4
- Clockwise and count-clockwise orbits
- Target Speed Ratio: 0 to 0.5 (half the desired airspeed)
- Target Heading: 0 deg to 360 deg in increments of 30 deg
- Number of Repeat passes: 1 to 4

Figure 56 presents the contours of average absolute arrival time error as it changes with number of passes and target speed. The performance of both controllers does not significantly change for more than two passes. This is expected since additional passes repeat the same re-positioning maneuver of the previous pass. However, the “Quad” controller demonstrated an ability to achieve one second arrival time performance for faster vehicles than the “Nav” controller. This observation is used to design the final configuration multi-pass characterization in Sec. 7.6.

Finally, it was noted that both controllers achieved arrival time performance under 0.5 seconds for any target speed simulated, up to a target speed ratio of 0.5, when the target heading relative to the y_r -axis was 180 degrees. Since the final desired course is “North,” a target relative heading of 180 degrees corresponds to a target traveling “East” for a counterclockwise orbit and “West” for a clockwise orbit. This observation may inform more advanced logic of future iterations of the controller.

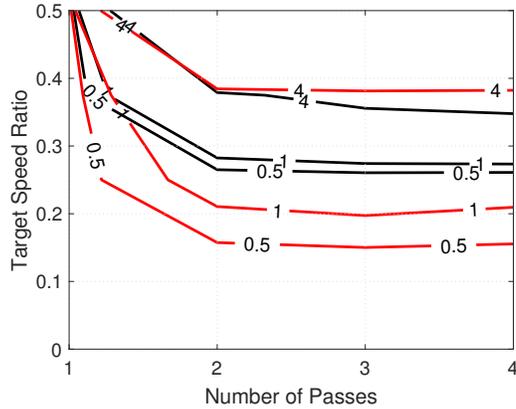


Figure 56. Initial Comparison Multi-pass Average Arrival Time Error (s)

Non-Cooperative Target Simulations.

The non-cooperative target simulations create a scenario where the target is initially traveling at constant heading and speed, then stops at some point during the simulation. The result is a step change in the final desired position of the path when the target stops due to the calculation of the estimated arrival point, see Sec. 5.8. The scenario challenges the ability of the controller to correct for a new final position while still attempting to achieve the desired arrival time.

Again, the wind direction and magnitude were set to zero to isolate the effects of the target stopping. The primary variables changed in the non-cooperative target sensitivity were:

- *TTSF* 1.4
- Clockwise and count-clockwise orbits
- Target Speed Ratio: 0 to 0.5 (half the desired airspeed)
- Target Heading: 0 deg to 360 deg in increments of 30 deg
- Target Stop-Time Ratio: 0.2 to 0.8

Figure 57 presents the time-of-arrival error contours as they vary with target speed and target stop-time scale factor. In the case of the “Quad” controller, the relatively constant error below a target stop-time scale factor of 0.4 corresponds to targets which stop in quadrant I or II allowing the size of the racetrack to change and compensate for the error induced by the stopping target. The performance decreases when the target stops in quadrant III, the final turning quadrant, because the size of the racetrack is fixed and cannot compensate for any arrival time error. Finally, the performance increases above a stop-time scale factors of 0.6 since the target stops closer to the original endpoint.

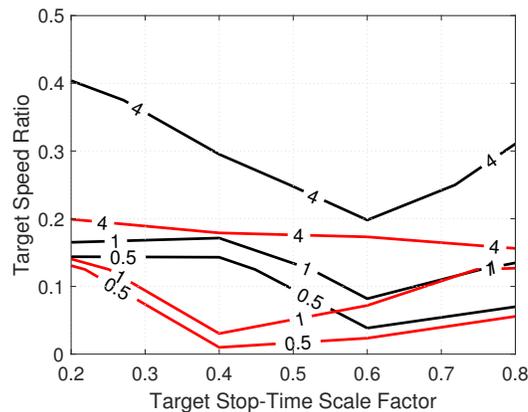


Figure 57. Initial Comparison Non-cooperative Target Average Magnitude of Arrival Time Error (s)

The “Nav” controller non-cooperative target performance is similar to the “Quad” controller for higher target stop-time scale factors because the two controllers are identical in quadrant IV. However, the “Nav” controller is unable to correct for earlier target stop-times as well as the “Quad” controller. This is likely due to the constant bank angle turns calculated by the controller which may not maneuver the aircraft to the new path in a timely manner. This may result in an unnecessarily long path to the new endpoint which prevents a lower arrival time error.

Controller Selection.

The initial comparison results show the “Quad” controller has lower time-of-arrival errors on average in scenarios where the target endpoint are static, moving with multiple passes, and a non-cooperative stopping scenario. Additionally, the assumptions of the “Nav” controller result in larger trajectories for certain wind conditions when compared to the “Quad” controller. Finally, the ground-fixed path of the “Quad” controller is expected to be able to integrate with the keep-out-zone avoidance algorithm.

Thus, it is concluded that the quadrature-based controller performs better over a wider range of scenarios and will be used to further develop the arrival time algorithm. However, it should be noted that both controllers achieve arrival time errors near zero in scenarios with low wind speed and slower targets. Therefore, the “Nav” controller may be a better controller for some future work where a steady aircraft bank angle is more desirable to keep-out-zone avoidance and high-wind performance.

7.4 Arrival Time Stability Analysis

The following section analyzes the string stability performance of the airspeed controller. In this section, the dynamics presented in Sec. 6.3 are assumed. Recall, the primary assumptions of these dynamics are that path length is fixed, the path is perfectly followed, and the arrival time estimates are accurate. This limits any change in arrival time to the airspeed controller, steady state wind, and turbulence.

The airspeed command gains used in the final controller were selected to be string stable, Chapter VI, and result in a multi-aircraft steady-state error of approximately

0 ± 0.5 s, discussed in this section,

$$k_T = 0.4922, \quad (378)$$

$$k_V = -1.9635. \quad (379)$$

As discussed in Section 6.3, a maximum arrival time error was assumed to calculate the gains utilizing the numeric optimization routine. The chosen gains satisfied the constraints of the optimization problem with an assumed maximum error, $T_{e_{max}}$ in Eq. (366), of 1000 s indicating the gains result in a string stable system for a large range of initial conditions. Arrival time errors near 1000 s were not observed during any simulations discussed in this Chapter.

String Stability.

Figure 58 presents the results for a twenty aircraft simulation. In this scenario the upper and lower aircraft rate-limits are randomized between a magnitude of 5 ft/s^2 and 15 ft/s^2 and the reference airspeed is randomized across a range of 200 ft/s to 800 ft/s . The lead aircraft, $i = 1$, is initialized with an arrival time error of 5 seconds, $T_{e_1}(0) = 5$.

Moderate turbulence was generated separately for each aircraft and the measured wind was generated as a bounded random sine wave with varying mean, frequency, amplitude, and phase. The parameters of the random sine wave assume the magnitude of the measured wind remains less than half the reference airspeed for each aircraft and the rate of change is less than an aircraft performing a 2G turn; the parameters were randomly selected for each individual aircraft.

The tailwind component of wind (measured plus unmeasured) is shown in the fourth plot assuming the wind direction is aligned with the aircraft heading. Although the magnitude change may seem extreme, keep in mind an aircraft experiencing a

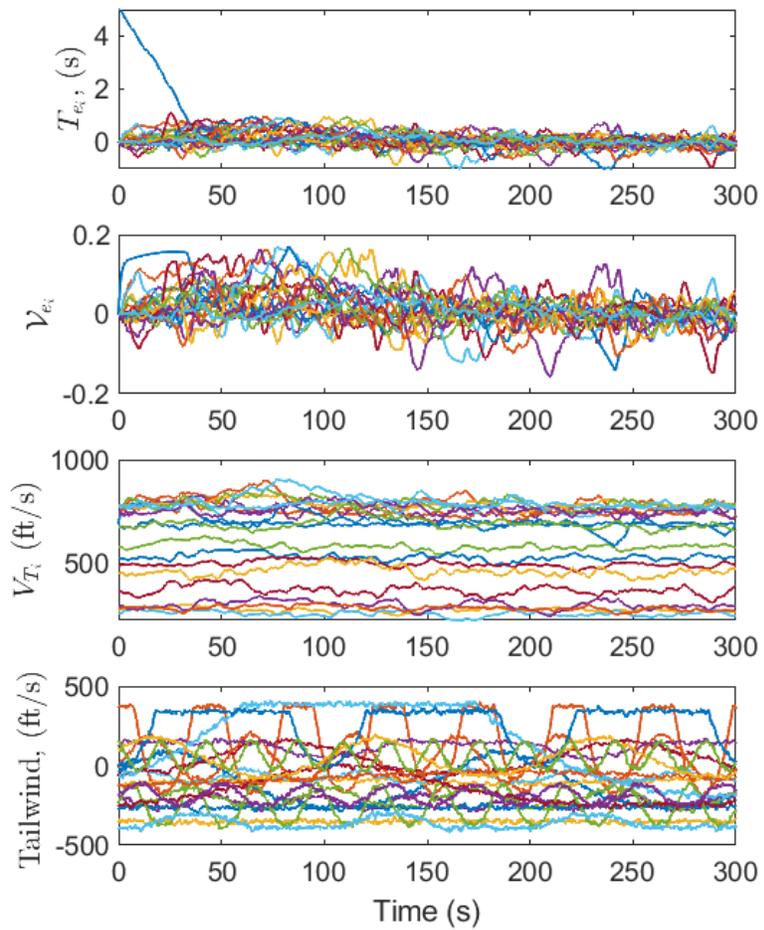


Figure 58. Twenty Aircraft, String Stable with Wind and Turbulence

100 *ft/s* tailwind will experience a -100 *ft/s* tailwind (a headwind) if it turns around and flies in the opposite direction.

Figure 58 highlights that the control system and gains discussed in this paper result in a string stable system in an environment with both measured and unmeasured wind effects. Qualitatively, the system exhibits a string stable response in arrival time error as desired. Once the initial error diminishes at approximately 175 seconds, the arrival time error of all the following vehicles, $i \geq 2$, remained between ± 0.5 seconds for the duration of the simulation. Additionally, the third subplot shows that the control system may handle a wide range of true aircraft speeds and remaining stable due to the non-dimensionalization of the airspeed.

Single Aircraft Nominal Performance.

The previous section highlighted that the arrival time error for a string of aircraft decreases to a small range once any initial error has been diminished. In terms of the racetrack algorithm, the actual arrival time is always determined by the first vehicle in the string. This is because the first vehicle becomes the last once it overflies the endpoint, and the second becomes the first. Mathematically this causes the w_{i-1} term and $\mathcal{V}_{e_{i-1}}$ term to be zero for the arriving aircraft, see Sec. 6.3.

Figure 59 presents a simulation of the lead aircraft only with changing wind and turbulence; the aircraft is a light utility aircraft. In this case, the arrival time error is bounded by $[-0.32, 0.79]$ seconds the entire simulation. This performance is demonstrated with a large change in the steady state wind and turbulence, shown in the fourth plot.

Mathematically, the equilibrium point of the first vehicle is determined by Eq. (380) which gives an expectation of actual arrival time performance since the system is exponentially stable. First, it is clear the minimum value of $m_i(t)$ maximizes the value

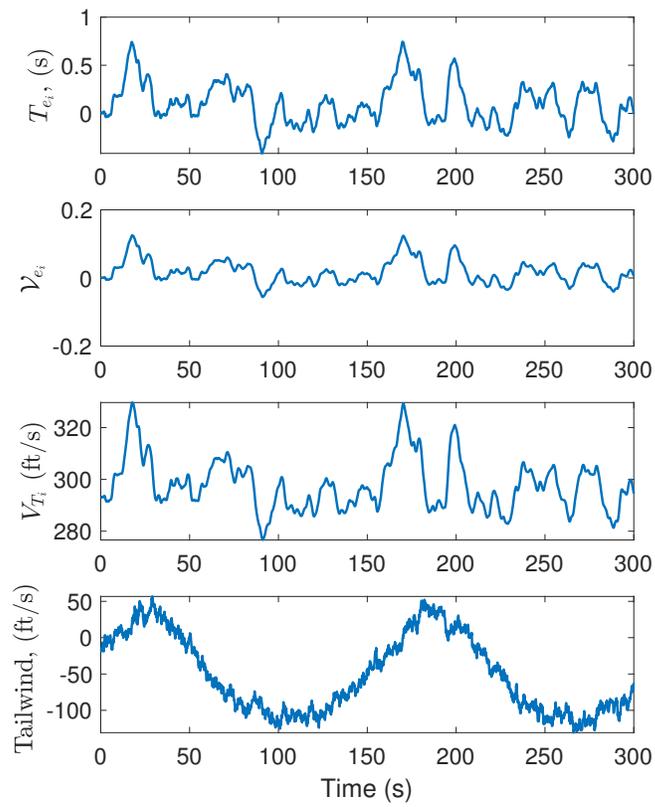


Figure 59. Single Aircraft with Turbulence and Wind

of the equilibrium point holding the other terms constant.

$$T_{e_{ieq}}(t) = w_i(t) \left(\frac{k\gamma}{k_T m_i(t)} - 1 \right) \quad (380)$$

Table 12 presents the maximum and nominal value of the equilibrium point along with the maximum observed value of $w_i(t)$ for each aircraft class. A wind term of $m_i(t) = 0.67$ is used to calculate $T_{e_{ieq}}$ which corresponds to the minimum bound of $m_i(t)$, Eq. (325) in Sec. 6.2. The maximum value of $w_i(t)$ is determined through a 300 second simulation of the Dryden Wind Turbulence model in MATLAB[®] with the output turbulence being converted to $w_i(t)$ using Eq. 324.

Table 12. Nominal Arrival Error and Turbulence Effects

Aircraft Class	$T_{e_{eqnom}}$ (s)	w_σ (1/s)	$T_{e_{eqmax}}$ (s)	w_{max} (1/s)
Light Utility	1.32	0.16	3.95	0.28
Medium Cargo	0.50	0.06	2.88	0.21
Heavy Cargo	0.37	0.05	2.00	0.14
Fighter	0.23	0.03	1.28	0.09

The maximum value of the equilibrium point defines the steady state arrival time error if a constant unmeasured wind with magnitude w_{max} is encountered. This bound is significantly larger than the observed variation in arrival time error in Fig. 59. Since the turbulence is randomly generated, it is unlikely the maximum magnitude will be sustained for any length of time. However, one standard deviation of $w_i(t)$, w_σ , better estimates the observed variation in arrival time error.

Therefore, the term nominal arrival time performance is used in the remainder of this section to define arrival time error performance which is bounded by $\pm T_{e_{eqnom}}$ which is calculated utilizing w_σ in Eq. (380). Since the control is similar to a virtual target, the nominal performance level may be viewed as variation around the desired virtual target. This value corresponds to the expected level of performance when the

lead aircraft is not airspeed saturated.

Saturated Airspeed and Turbulence.

As discussed in Chapter VI, the stability guarantees do not apply to certain saturated conditions. An example of such a condition is when the airspeed is at its upper limit to reduce time error, but the headwind is strong enough to cause the arrival time error to increase.

Figure 60 presents a simulation where this condition occurs multiple times. In the scenario, the lead ($i = 1$) aircraft is initialized with a high arrival time error and saturated airspeed to force the following aircraft to saturate its airspeed on an upper limit. The baseline result, in dashed line, has zero unmeasured wind effects, $w_i(t) = 0$.

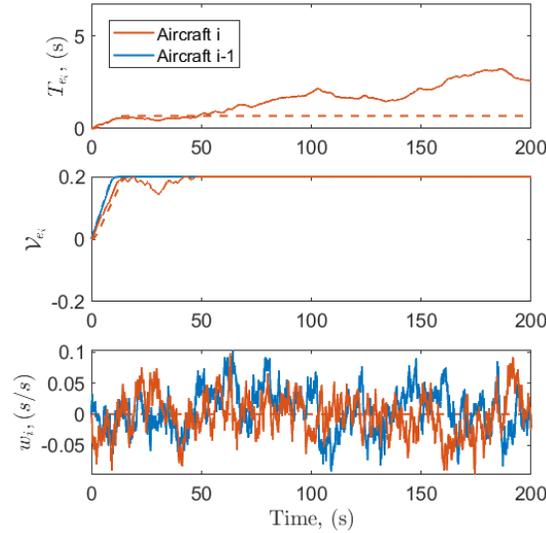


Figure 60. Airspeed Saturated Aircraft with Turbulence; $T_{e_{i-1}}(t) > 5$

The simulation with turbulence, the solid line, shows that the airspeed may desaturate depending on the magnitude of the unmeasured wind but remains saturated for most of the simulation. While saturated, the arrival time error is dependent on the unmeasured turbulence of the current and preceding aircraft which is not controllable.

Although this means the response may not be mathematically bounded, this simulation highlights that the magnitude of this effect due to realistic turbulence is small enough to not cause unstable error behavior. Furthermore, Fig. 58 showed that any initial arrival error is diminished rapidly enough that it is unlikely for two sequential aircraft to both have a saturated airspeed for more than a few seconds.

Effect of Path Tracking Error on Arrival Time.

Section 5.6 identified that tracking error while following, not acquiring, the path significantly affects arrival time error during curved racetrack segments with the selected time estimate method, the Gaussian Quadrature method. This error may be due to typical path track performance or the turn radius adjustment feature of the MPC follower which intentionally flies inside or outside the path to affect arrival time error.

Equation (382) reproduces Eq. (316) which defines the rate of change of arrival time error as a function of measured ground speed. Next, assume that G_m is equal to Eq. (381), from Sec. 5.6. Taking the result, Eq. (383), $0 = G_m(t) - G_m(t)$ is added to the numerator to obtain the original groundspeed error term, $G_m(t) - G_{ref}$.

$$G_m = \frac{G_{m_t} R_T}{R_T - e} \quad (381)$$

$$\dot{T}_e(t) = -\frac{G_m(t) + W_u(t) - G_{ref}(t)}{G_{ref_f}(t)} - \dot{T}_{ref_i}(t) \quad (382)$$

$$\dot{T}_e(t) = -\frac{\frac{G_{m_t}(t) R_T}{R_T - e(t)} - G_{ref}(t) + W_u(t)}{G_{ref_f}(t)} - \dot{T}_{ref}(t) \quad (383)$$

$$\dot{T}_e(t) = -\frac{G_m(t) - G_{ref}(t) + W_u(t) + \frac{G_{m_t}(t) R_T}{R_T - e(t)} - G_m(t)}{G_{ref_f}(t)} - \dot{T}_{ref}(t) \quad (384)$$

$$G_{e_{path}} = \frac{G_{m_t}(t) R_T}{R_T - e(t)} - G_m(t) \quad (385)$$

Equation (385) defines the ground speed error as a function of path tracking error. With regards to the string stability dynamics, this term may be treated as an additional component of $W_u(t)$ with the assumption that the path tracking error is bounded while following a path. This does not address path acquisition which affects the \dot{T}_{ref} term.

Table 13 compares the magnitude of the change in groundspeed error, $G_{e_{path}}$, to the magnitude of the effect of moderate turbulence, $|W_u|$. The comparison assumes zero steady-state wind, the aircraft traveling at the desired ground speed, and a tracking error magnitude equivalent to 80% the ft of tracking error to the inside of a circular turn; based on the maximum turn radius adjustment in Sec. 5.7. The maximum values of $W_u(t)$ were determined via a 200 second simulation for each aircraft. The data show that the effect of path error is the same order of magnitude as the effect of turbulence for all aircraft.

Table 13. Zero steady wind, Moderate Turbulence, 500 ft path error Comparison

Aircraft	$ W_u(t) , (ft/s)$	$ G_{e_{path}} , (ft/s)$	$\mathcal{V}_{e_{max}}, (s/s)$	$w_{max}, (s/s)$
Light Utility	32.5	15.4	0.20	0.24
Medium Cargo	29.8	15.2	0.20	0.16
Heavy Cargo	34.2	15	0.14	0.13
Fighter	41.0	47.0	0.11	0.10

The maximum non-dimensional airspeed is shown in addition to the bound on w_i which corresponds to the observed $|W_u(t)|$ for each aircraft. When the wind magnitude is zero, these two terms are simply added together to determine the rate of change of the arrival time error, see Sec. 6.2. Since the track error term may be added to the turbulence term, it is concluded that the effect of tracking error on arrival time error during a turn is similar to the effect of turbulence.

In a non-zero wind condition, flying inside a turn has approximately half the effect on the rate of change of arrival time error as flying at the maximum airspeed

as indicated by the similar values of $\mathcal{V}_{e_{max}}$ and w_{max} for all aircraft except the fighter. For fighter aircraft, the effect is approximately the same.

A more physical example for a light utility aircraft is that flying 234 ft inside a turn (a magnitude of 80% of V_{ref}) is similar to flying 1.5 times as fast on the desired path with zero wind. For a fighter aircraft, it is equivalent to flying on the desired path twice as fast with zero wind. This quantifies the effect of the turn radius adjustment function of the MPC controller may have on arrival time error. Similar performance may be shown when flying outside the path to increase the estimated arrival time.

The stability guarantees remain unchanged with the assumption that the path error is also bounded. Based on this conclusion, it is expected that the arrival time error will vary more in turns than in straight segments during simulation due to turbulence. However, this is not expected to affect the non-saturated arrival time performance since the endpoint follows a straight segment.

7.5 Final Controller Configurations

The final version of the controller consists of all features discussed in Chapter V to include the re-orbit, keep-out-zone, and the turn radius adjustment features which were not enabled in the initial configurations. As discussed in the previous section, the final version of the controller utilizes the quadrature-based arrival time estimate, see Sec. 5.4. The remainder of this section documents the values of pertinent parameters

As discussed in the Sec. 7.4, the airspeed command gains used in the final controller and were selected to be string stable and result in a multi-aircraft steady-state error of approximately 0 ± 0.5 s:

$$k_T = 0.4922, \tag{386}$$

$$k_V = -1.9635. \tag{387}$$

Additionally, the previous section identified the effect of flying inside or out a turn to provide additional control over arrival time error. The turn radius adjustment is an offset applied to the desired path which causes the aircraft to shift the tracking reference point left or right. The adjustment is directly applied to the error calculation in the MPC controller, Sec. 5.7, which is formulated in the racetrack reference frame. Therefore, a positive adjustment causes the aircraft to track outside the turn while a negative adjustment tracks inside the turn. The adjustment is enabled when the arrival time error is outside of ± 5 s and is calculated as

$$e_{adj}(t) = \int_{0.8V_{ref}}^{-0.8V_{ref}} -40 (T_e(t) - 5\text{sgn}(T_e(t))). \quad (388)$$

The saturation limits and gain on the error were chosen based on the results in Sec. 7.4.

Next, the keep-out zone avoidance logic includes a “pad” to account for the typical path following performance of the MPC path follower. As mentioned in Sec. 5.8, the keep-out zone avoidance method determines the maximum of the racetrack which does not cross into a keep-out zone. If zero pad were used, the desired path would “touch” the edge of the keep-out zone. However, some tracking error is expected since the MPC path follower was tuned to ensure nominal path tracking performance within 500 ft. Therefore, a pad of $0.8V_{ref} = 500$ is used to account for the typical path tracking performance combined with the turn adjustment limits.

Finally, Table 14 provides a summary list of pertinent aircraft specific parameters.

Table 14. Aircraft Dependent Parameters

Aircraft Class	V_{ref} (ft/s)	Max N_Z (g)	MPC Horizon, t_h (s)
Light Utility	293	2	7
Medium Cargo	390	2	7
Heavy Cargo	544	2	6
Fighter	864	4.5	6

7.6 Final Controller Performance Characterization

This section identifies the overall level of performance of the racetrack algorithm and its sensitivity to changes in the scenario. This section is not intended to fully model the performance of the algorithm in every situation. Similar to the initial controller comparison, multiple variables are changed however, a finer grid of points is used to obtain a high-fidelity characterization.

In this section, the performance data are presented as a mean and standard deviation plot rather than just the average absolute error used in Sec. 7.3. This format is chosen because whether the aircraft arrives early or late is meaningful to the characterization.

The data from all simulations for a given subset of data are used to generate each plot. In the case of Fig. 61a, each data point of the contour plot represents the average of all simulations with a given windspeed factor and $TTSF$. Similarly in Fig. 61b, each data point is the standard deviation of all simulations for a given windspeed factor and $TTSF$ combination.

These metrics characterize the average performance of a variable while also providing a sense of the range in performance at the windspeed and $TTSF$ condition. For example, a low standard deviation indicates all simulations for a given condition produced similar results while a higher standard deviation indicates more variation in the data. However, the data are not assumed to be normally distributed, therefore the standard deviation metric is treated as qualitative metric.

Static Target Simulations.

The first set of data evaluates the algorithms performance against a static target. Each simulation is initialized with the aircraft over the target in the desired heading. The following parameters were varied in the characterization resulting in 12,544

simulations:

- *TTSF*: 1 to 1.5
- Orbit Direction: Clockwise and Counterclockwise
- Wind Speed Ratio: 0 to 0.5 (half the reference airspeed)
- Wind Direction: 0 deg to 360 deg in increments of 22.5 degrees
- All aircraft types
- Turbulence: None and Moderate

A simple linear regression was initially performed to determine which variables significantly affected the arrival time error in the static scenarios. The primary significant variables ($p\text{-value} \leq 0.05$) are the wind speed ratio and *TTSF*. Additionally, the aircraft type was treated as a categorical variable, and it was shown that the performance of the fighter type aircraft was significantly different ($p\text{-value} \leq 0.05$) from the other three aircraft types. However, the other three aircraft types were not significantly different ($p\text{-value} > 0.05$) from each other. Notably, the linear regression did not show that turbulence had a statistically significant on the average arrival time error. Since the turbulence is random, it is not unexpected that the mean does not change.

Figure 61 presents the average arrival time error and two standard deviations for different combinations of wind speed ratio and *TTSF*. Moderate turbulence is shown in solid lines while a single contour of the no turbulence data, dashed line, is shown as a comparison. The minimum average error of the turbulent data set is -0.02 s while the minimum of the no turbulence data is -0.1 s. Additionally, the red lines bound the region for each data set where the nominal level of performance was achieved for all scenarios at that data point, corrected for aircraft type.

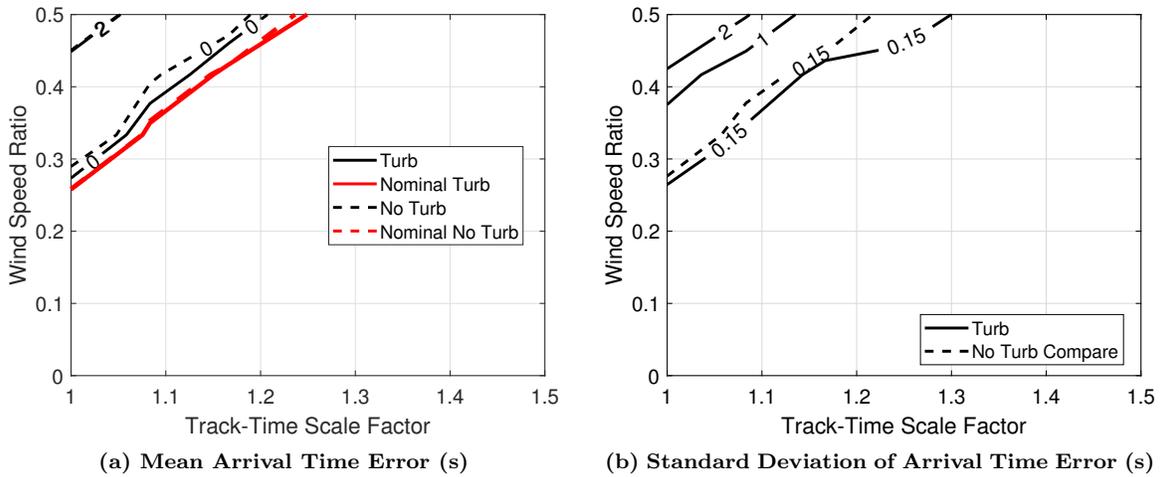


Figure 61. Final Algorithm Static Arrival Time Error

Overall, higher wind speeds, shorter arrival times, and turbulence tend to increase the error dramatically once a certain limit is reached. Higher wind speeds and shorter arrival times correspond to configuration when the racetrack size is minimized, a circular orbit, and the aircraft commands maximum speed the entire orbit. This effectively represents a saturated system and any decrease in arrival time or increase in wind will increase the arrival time error. The standard deviation highlights that this also increases the variability in the data indicating that different scenarios saturate at different times others. Conversely, the simulations show no scenarios where the aircraft arrives earlier than the nominal performance estimates. This is expected since the wind magnitude tends to increase the arrival time error but there is not a variable in this data set which would reduce the arrival time.

Comparing the turbulent and non-turbulent data shows there are conditions where the turbulent and non-turbulent data behave differently. However, the difference is not statistically significant. Generally, a nominal level of performance may be achieved for a slightly wider set of conditions with no turbulence.

Figure 62 presents a similar set of data for the arrival airspeed error. The general relationship between airspeed error, wind speed ratio, and $TTSF$ is the same,

but with larger error magnitudes. This is expected since a portion of the airspeed command is proportional to arrival time error. However, turbulence tends to increase the average error and standard deviation of the error more so than the arrival time error. In fact, a performing a linear regression on the arrival airspeed error shows that turbulence is a significant variable ($p \leq 0.05$).

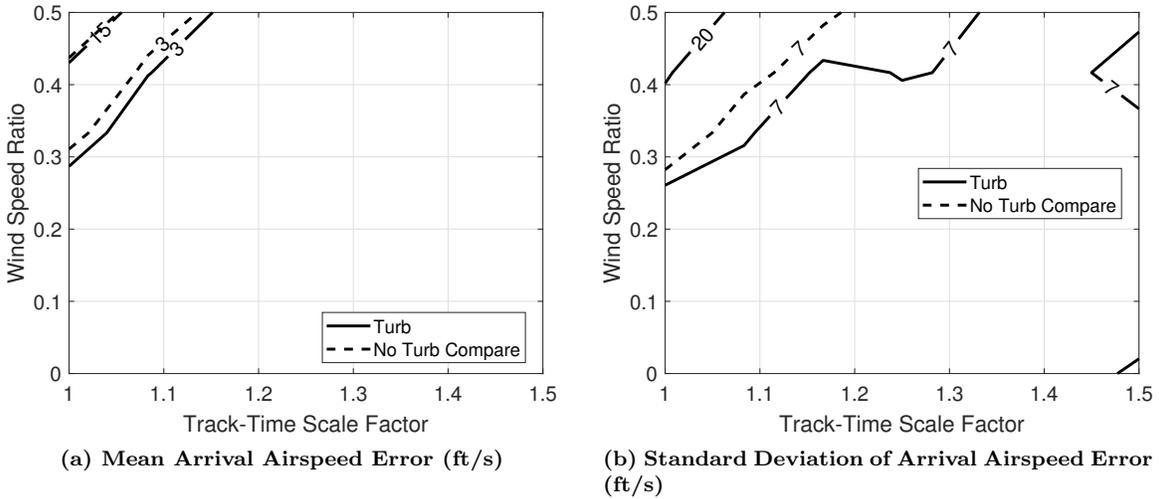


Figure 62. Final Algorithm Static Arrival Airspeed Error

Finally, although the aircraft type was statistically significant, it was observed that an analysis of the fighter type aircraft alone exhibited similar trends and error magnitudes as the average of all aircraft together. Analyzing the remaining aircraft individually showed that the arrival error was nearer to zero on average compared to the average of all aircraft together. Overall, the algorithm still demonstrates a nominal level of performance for a wide range of wind conditions and arrival time requirements regardless of aircraft types.

Moving Target Simulations.

Multi-pass Simulations.

The multi-pass simulations are calculated for a single pass but are initialized as if the aircraft had just completed a pass. Figure 63 depicts this setup at the beginning

of a repeat simulation. The target is represented by an orange box with a speed vector arrow attached to it. A dashed line connects the target to the predicted endpoint which is used as the final point of the racetrack. The aircraft began directly over the target heading “East” and must now reacquire the racetrack for the next pass.

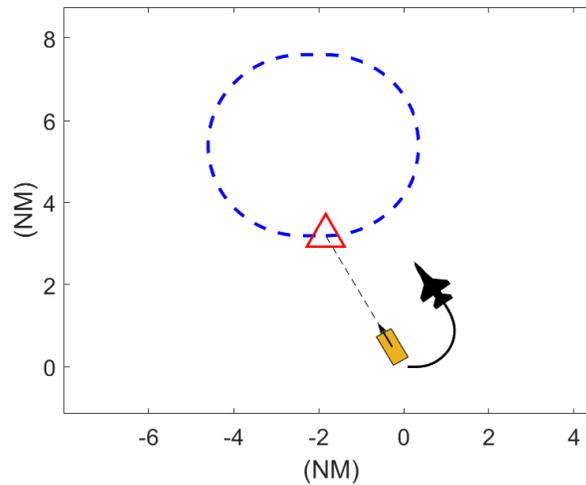


Figure 63. Multi-Pass Setup Example

A similar set of parameters were varied in these simulations, as compared to the static simulations, with the exception of turbulence. Due to the trends in performance between the turbulent and non-turbulent data sets of the static scenarios, only a moderate turbulence level was evaluated. The following parameters were varied in the characterization resulting in 38,400 simulations:

- *TTSF*: 1 to 1.5
- Orbit Direction: Clockwise and Counterclockwise
- Wind Speed Ratio: 0 to 0.5
- Wind Direction: 0 deg to 360 deg in increments of 60 degrees
- Target Speed Ratio: 0 to half the reference airspeed

- Target Direction: 0 deg to 360 deg in increments of 36 degrees
- All aircraft types
- Turbulence: Moderate

Initially, a simple linear regression was performed on the data set to determine which variables significantly affected the arrival time performance and warrant investigation. The regressions identified wind speed ratio, $TTSF$, wind direction relative to $y^{(r)}$, target direction relative to $y^{(r)}$, target speed ratio, and aircraft type as significant variables.

Figure 64 presents the same type of plots used in the static scenario analysis, average error and standard deviation of error contours. Additionally, a single contour from the static simulation data set is included for comparison on each plot.

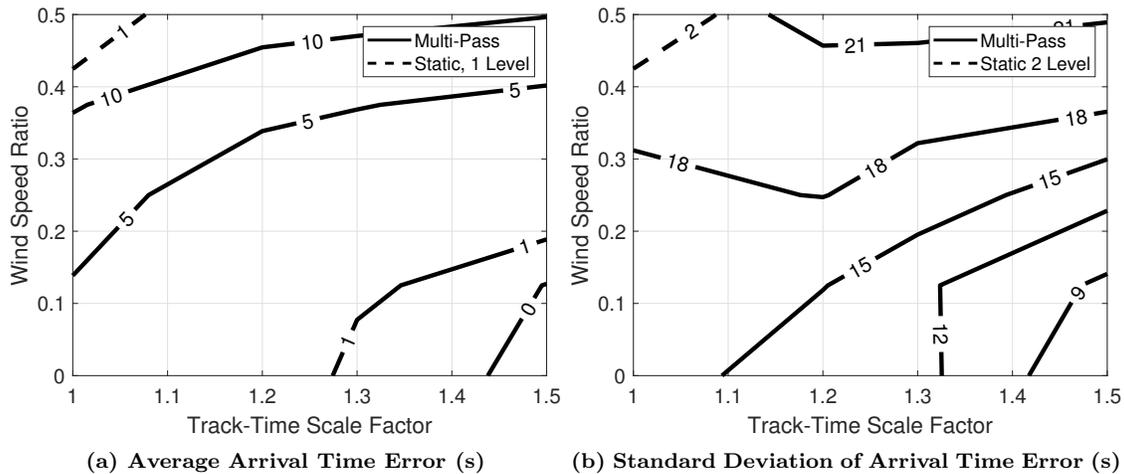


Figure 64. Repeat Pass Comparison to Static Target

On average, the arrival time error increases substantially compared to the errors achieved in static scenarios. Note, the no single data point had nominal performance for all scenarios in the point (indicated by a red line in the static section). The one-second average error contour occurs at higher arrival times ($TTSF$) and lower wind speeds than with the static scenarios. This is expected since each data point

is calculated with every target speed and heading combination. The repeat scenarios may require the aircraft to travel farther than a corresponding static scenario. Additionally, the standard deviation shows that there is more variation in all the corresponding sets of scenarios than the static target. The high standard deviation indicates that, as shown by the regression, additional variables likely impact the arrival time error. There is not a single set of scenarios where all scenarios achieved nominal performance.

Figure 65 presents the arrival time error data set averaged for different combinations of target speed ratio and wind speed ratio. Intuitively, the arrival time error decreases with lower wind and target speeds. This is because higher target speeds result in a larger change in racetrack position from one pass to the next requiring a longer distance to be traveled. This causes the racetrack size to be minimized for lower wind speeds when compared to the static case. Additionally, there is a small portion of scenarios, which achieved nominal performance for all tested scenarios; bounded by the red line. The next set of data will show that the arrival time performance is also heavily dependent on target heading.

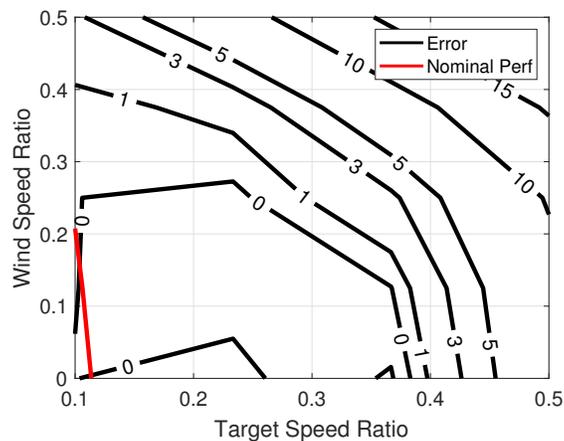


Figure 65. Multi-pass Average Arrival Time Error (s), Varying Wind Speed and Target Speed

Figure 66 presents the arrival time error data set averaged for different combi-

nations of target speed ratio and heading relative to the $y^{(r)}$ axis. A clear desired heading is seen in around a relative heading of 180 deg . Viewed from an aircraft on the final path segment, this is a target moving from left-to-right for a counterclockwise orbit. This is a “desirable” heading because it causes Quadrant *II* of the racetrack to move towards Quadrant *IV* once the orbit is complete. However, no set of scenarios achieved nominal performance for all tested scenarios.

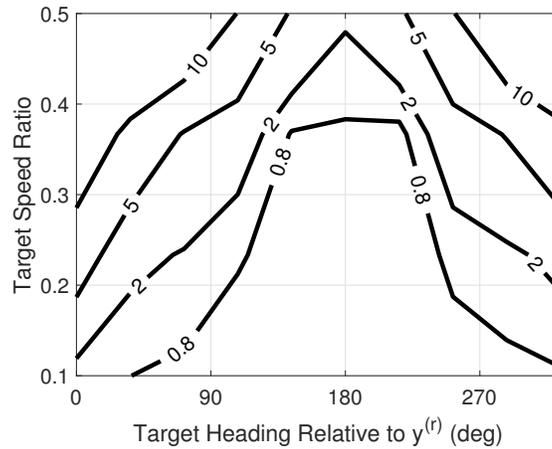


Figure 66. Multi-pass Average Arrival Time Error (s), Varying Target Speed and Heading

The behavior is highlighted in Fig. 67. The aircraft is positioned such that it just completed an orbit and the solid target and trajectory represent a 180 deg relative heading while the empty target and dashed trajectory are a 0 deg relative heading. The desired path of the target moving at a 180 deg relative heading is much closer than the target moving in the opposite direction. This information may be used to choose an orbit direction if it does not matter to a specific scenario.

Figure 68 compares the relative wind and target heading for two data sets. Figure 68a utilizes all scenarios in the multi-pass data while Fig. 68b is limited to $TTSF = 0.5$, Wind Speed Ratio ≤ 0.4 , and Target Speed Ratio ≤ 0.4 .

The dashed lines are values of relative wind heading as a function of the target heading. The main observation in both data sets is that the best arrival time errors

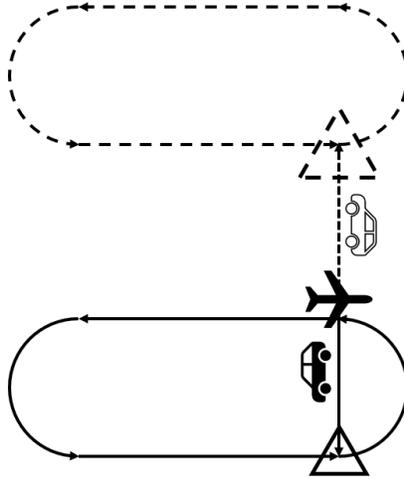


Figure 67. Changing Target Heading for a Counterclockwise Orbit

occur in an oblong area with a semi-major axis which roughly corresponds to an equal wind and target direction. The area is again centered at 180 deg which matches the observations in Fig. 66. However, no set of scenarios achieved nominal performance for all tested configurations in Fig. 68a.

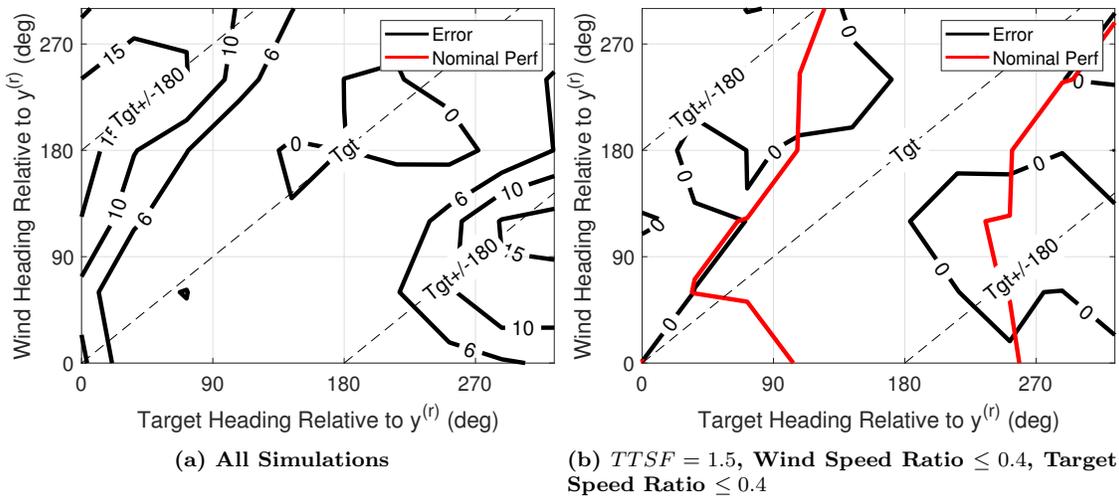


Figure 68. Multi-pass Average Arrival Time Error (s), Varying Wind Direction and Target Direction

Figure 68a also shows that the worst performance occurs with a target heading near 0 deg relative to $y^{(r)}$ with a windspeed in an opposite direction. However, the performance in the same area is significantly better in Fig. 68b. The average perfor-

mance suffers when both target and windspeed are near half the reference airspeed but in opposite directions. The poor performance in this area is due to a combination of variables. The target heading near zero requires the aircraft to fly farther to acquire the path and travel in the same direction as the target; the light car and dashed path in Fig. 67. The wind direction results in a full headwind. When the windspeed and target speed are both half the reference speed, then the reference groundspeed is the same as the target speed, preventing the aircraft from quickly catching up to the target. The result is a very late arrival.

However, Fig. 68b highlights that this degradation in performance is primarily due to a small number of extreme cases. Limiting the data to higher *TTSF* and marginally lower wind and target speeds results in a large portion of the data which achieves nominal performance.

Again, the relative change in parameters did not change with aircraft type. In repeat scenarios, the relative performance of the aircraft corresponded with the turning capability (minimum turn radius) of the respective aircraft. Thus, aircraft types from best to worst performing are light utility, medium cargo, fighter, and heavy cargo. This result is attributed to longer distance a higher turn radius aircraft must fly to reacquire the desired path. Aircraft speed is likely not a factor since the target speeds are scaled off the desired aircraft speed.

Overall, the algorithm performed well in multi-pass scenarios and a clear region of headings was shown to result in significantly better performance. This characterization may be used to choose a preferred arrival heading or turn direction when not constrained by the scenario.

Non-Cooperative Target Simulations.

The non-cooperative scenarios are setup such that the aircraft begins in a position to overfly the moving target and then the target will stop at some point prior to the aircraft completing the orbit. Figure 69a shows the position of the aircraft and target prior to the stop. The target is off the racetrack but moving towards the predicted endpoint (red triangle). Note the aircraft began at the current predicted endpoint. Figure 69b shows the position of the vehicles after the target stopped and the endpoint has shifted to the now static target. The shift is most notable by comparing the initial path of the aircraft (black) and the new desired path (dashed blue).

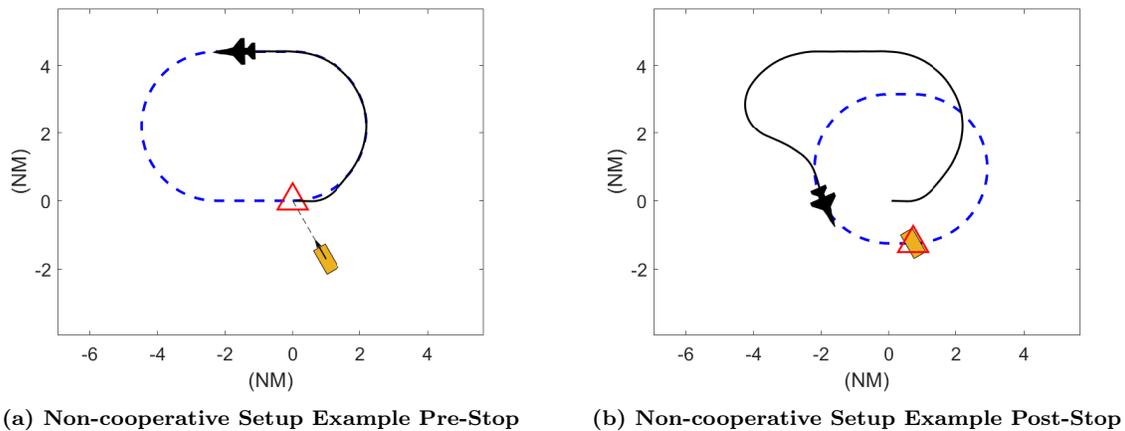


Figure 69. Final Algorithm Non-cooperative Arrival Airspeed Error

The following parameters were varied in the characterization resulting in 19,200 simulations:

- *TTSF*: 1.3
- Orbit Direction: Clockwise and Counterclockwise
- Wind Speed Ratio: 0 to 0.3
- Wind Direction: 0 deg to 360 deg in increments of 60 degrees

- Target Speed Ratio: 0 to half the reference airspeed
- Target Direction: 0 deg to 360 deg in increments of 36 degrees
- Target Stop-Time Ratio: 0.1 to 0.9
- All aircraft types
- Turbulence: Moderate

The initial simple linear regression showed that windspeed ratio, target speed ratio, target stop-time ratio, relative target heading, and aircraft type were all significant variables ($p \leq 0.05$) affecting the arrival time error. First, Fig. 70 presents the average and standard deviation of arrival time error as it varies with target speed and target stop time.

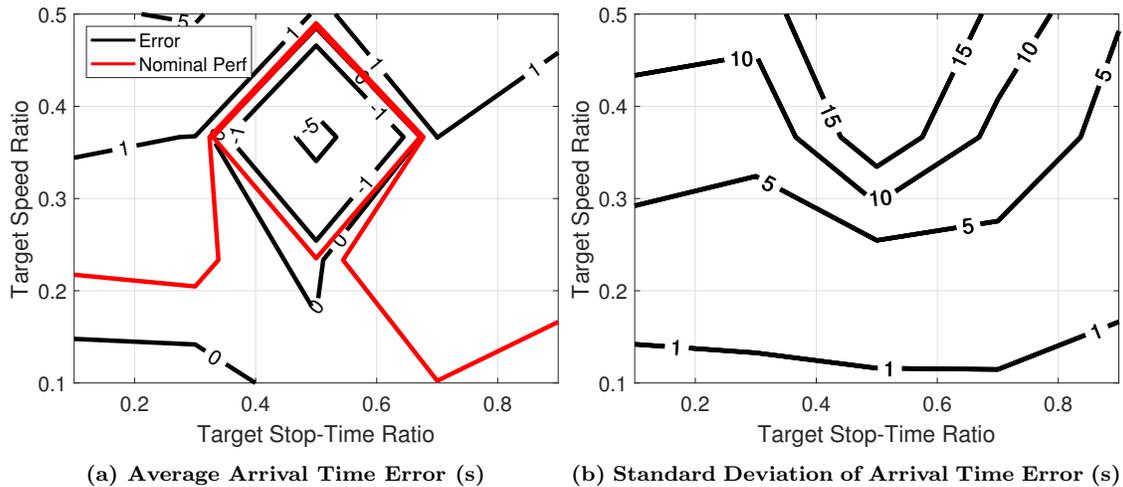


Figure 70. Non-Cooperative Target Speed Ratio v. Target Stop Time Ratio

The average arrival error plot shows interesting behavior due to the highly nonlinear nature of the scenario. Of note, the red contour demonstrates if a slow target stops, nominal performance is achievable regardless of when the target stops. However, interesting behavior occurs at a target stop-time ratio of 0.5. This is a particularly

difficult time because halfway through the orbit is approximately when the aircraft will enter the final turn and the racetrack size becomes fixed.

The average arrival error differences are primarily due to the simulation length. The slowest (maximum) arrival time errors are limited by the length of the simulation while the earliest are not. The standard deviation plot highlights that there is a high variation in arrival time errors at this point.

Figure 71 presents the average and standard deviation arrival time error as it varies with target stop time and relative target heading. Similar behavior is seen as with Fig. 70. In general, the average arrival time error magnitude and standard deviation are lower when the target heading is near 0 *deg* relative to the racetrack $y^{(r)}$ axis.

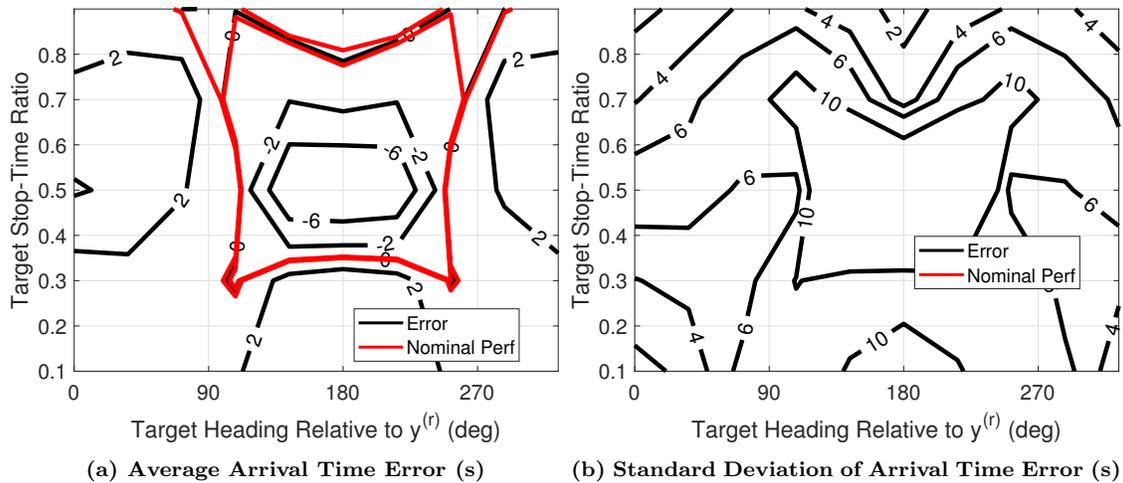


Figure 71. Non-Cooperative Target Stop Time Ratio v. Target Relative Heading

The highest magnitude errors are observed near 180 *deg* target heading and 0.5 stop time scale factor. These scenarios correspond to an aircraft at the beginning of the final turn and the target moving away from the aircraft. After the target stops, the final leg of the desired path moves towards the aircraft resulting in an early arrival time. The standard deviation plot highlights that there is an increased variation in arrival time errors at this condition.

Overall, the racetrack algorithm is not as successful at achieving the desired arrival time with a non-cooperative target compared to a steady moving target or a static target. This is not unexpected since the algorithm was designed to achieve good performance with multiple passes on a constantly moving target. Furthermore, this data set was only simulated for a single *TTSF* and it is expected that errors would improve when as *TTSF* increases based on the static and multi-pass performance. An increased desired arrival time allows the aircraft more time to correct for any error which is a result of the target stopping.

Single Vehicle Analysis.

A single vehicle scenario is analyzed to highlight the detailed behavior of different parts of the racetrack algorithm. The scenario is a non-cooperative moving target scenario with a keep-out zone. A light utility aircraft is flying with a steady windspeed of 88 *ft/s* (52 knots) which is 30 % of the reference airspeed. The desired arrival time is set at 175 *s* which corresponds to a *TTSF* of 1.7 to activate the keep-out zone and target is initially traveling at 58 *ft/s* (20% of the reference airspeed) at a heading of 300 degrees. The target stops at 52 seconds (30% the desired arrival time). Finally, the initial positions of the aircraft and target were set such that the initial estimated endpoint was at the coordinates (0,0) and the aircraft began over that point as well.

Figure 72 presents the results of the simulation at two different points in time to highlight the movement of the desired racetrack. The arrival time error at the end of the orbit is -0.08 *s* with -0.7 *ft/s* arrival airspeed error, within the nominal performance range. The vehicle trajectory abruptly turns just after the snapshot in Fig. 72a to correct to the now stopped target. The effect is that the estimated endpoint, represented by the red triangle, moved from (0,0) to its final location in Fig. 72b, shifting the desired racetrack.

The keep-out zone avoidance logic is also active in this sample scenario. The racetrack size is initially free but near the point of becoming saturated. Once the target stops, the racetrack size saturates. The space between the keep-out zone and the final desired path is equal to the specified “pad” distance. Note that, as expected, this “pad” space is approximately equal to the distance outside the path the flown by the aircraft due to the turn radius adjustment.

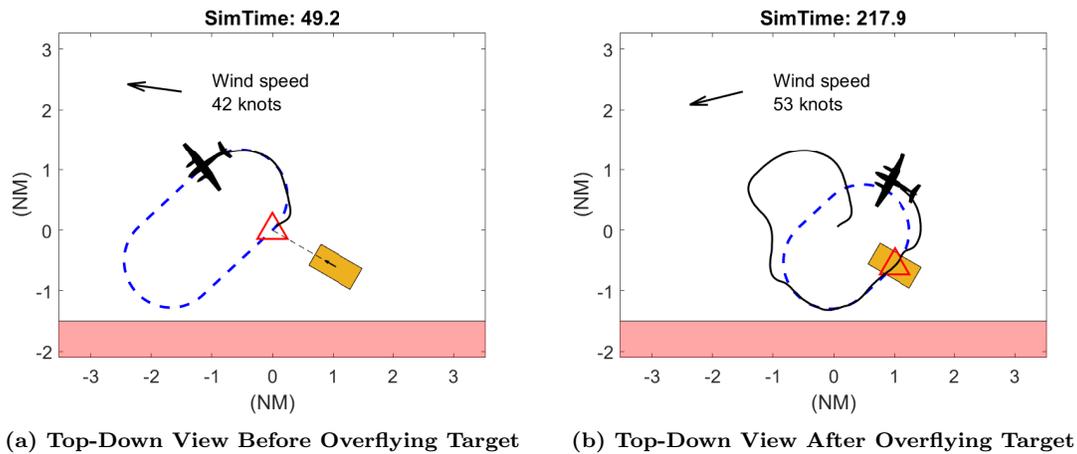


Figure 72. 58 ft/s Target stopping with aircraft in first turn

Inspecting the parameter time histories in Fig. 73 expands on the behavior shown in Fig. 72. First, the rapid changes in arrival time error and racetrack size at the beginning of the simulation are due an artifact of the algorithm initialization. Next, a sharp jump in the racetrack size, the parameter a , occurs when the target stops at 52 seconds. Before this point, the racetrack size changes slightly to keep the arrival time error near zero. Once the target stops, the racetrack size reduces to the keep-out zone constrained value resulting in an arrival time error which shows an early arrival. This commands the airspeed to a minimum value and triggers the turn radius adjustment logic resulting in the tracking error to plateau near 1500 ft just prior to 100 s. The tracking error decreases with arrival time error as expected. Once the arrival time error is near zero at approximately 150 s, the algorithm behaves nominally until the

end of the orbit at 175 s.

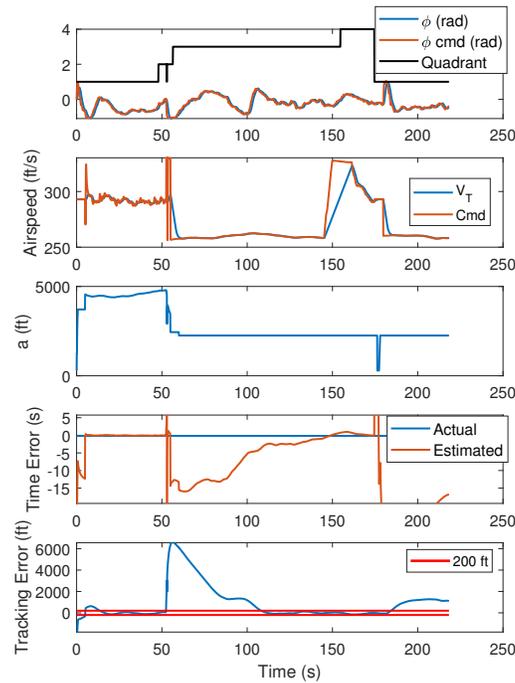


Figure 73. Sample Scenario Time History

7.7 Cooperative Control Simulations

This section analyzes the performance of a coordinated group of four aircraft for in three constrained scenarios. The scenarios are chosen to demonstrate the string stability performance of the algorithm while activating the advanced logic features of the algorithm such as keep-out zone avoidance, s-turns, turn radius adjustment, etc.

In these simulations, the aircraft are color-coded by arrival order: blue, red, yellow, purple (respectively). The heterogeneous aircraft simulations all utilize the same order of aircraft class: light utility, medium cargo, heavy cargo, fighter. Although the order remains constant, the stability analysis in Chapter VI showed that the algorithm performance is independent of the aircraft order based on the selection of

the command gains and limits. Finally, lead aircraft becomes the last aircraft upon orbit completion.

In the top-down views, aircraft trajectory is shown as a solid line in the corresponding color, the desired racetrack is a dashed line, the predicted endpoint a color-coded triangle, and keep-out zones are a shaded red area. All aircraft are initialized at a common point indicated by the black circle. The target is an orange rectangle with a speed vector arrow. The arrival time plots are also color coordinated and the end of each orbit is marked by a black vertical line.

Videos of both the simplified dynamics model and nonlinear dynamics model scenarios may be found at the following YouTube link, <https://www.youtube.com/watch?v=PV1fj08ocFY>. The location of each specific scenario is provided in the video description and a short description of each scenario is given prior to each simulation.

Static Target with Keep-Out Zones.

The static scenario commands a heterogeneous group of aircraft to overfly a static target at equally spaced intervals while avoiding keep-out zones. The keep-out zones, both rectangles and ovals, are positioned so that the size of each aircraft's racetrack is constrained. This ensures the keep-out-zone logic works as intended and forces any arrival time error to be corrected by airspeed only. The main parameters of the scenario are:

- Initial Orbit Arrival Time: 260 *s*
- Time Spacing: 65 *s*
- Steady Wind Speed: 59 *ft/s* (20% of slowest reference airspeed)
- Steady Wind Heading: 315 *deg*

Table 15 presents the arrival error results for each aircraft. All errors are within the nominal range for each aircraft as expected due to the wind conditions, desired arrival times, and the static target.

Table 15. Simplified Model Static Results at Orbit Completion

Aircraft	Arrival Time Error (s)	Arrival Airspeed Error (ft/s)	Distance to Target (ft)
1	0.08	4.0	27
2	-0.02	-1.0	124
3	0.02	1.9	58
4	0.00	0.5	126

Figure 74 presents the resulting trajectories of the static scenario. First, note the seemingly errant turns in the purple and yellow trajectory which occur at the approximate coordinates of $(-5, 19)$ and $(-2, 5)$ respectively. The turns are short activation of the keep-out zone logic at approximately 100 s when the arrival time error decreases below the threshold to activate the s-turn logic, 5 seconds.

In Fig. 74a it is clear the keep-out zone logic works as intended since the aircraft approach but do not cross into the keep-out zones during the first orbit. Since the aircraft begin at the same location, the size of the subsequent racetracks for aircraft 2-4 are shorter for the subsequent pass. This is because the first arrival time included the time for the first vehicle to complete a full orbit plus the time for the other aircraft's time spacing. After the first orbit, the aircraft are correctly spaced apart and each orbit will be the desired orbit time. Finally, the small amount of space between the trajectories and the keep-out zones is due to the "pad" variable used in the keep-out zone constraint; see Sec. 5.8.

In Fig. 74b it is clear the system is string stable in arrival time error since the errors remain bounded. Additionally, the variation in error of a subsequent aircraft noticeably decreases when the preceding aircraft arrives. This highlights how the lead aircraft is able to maintain a lower bounds on arrival time error because its

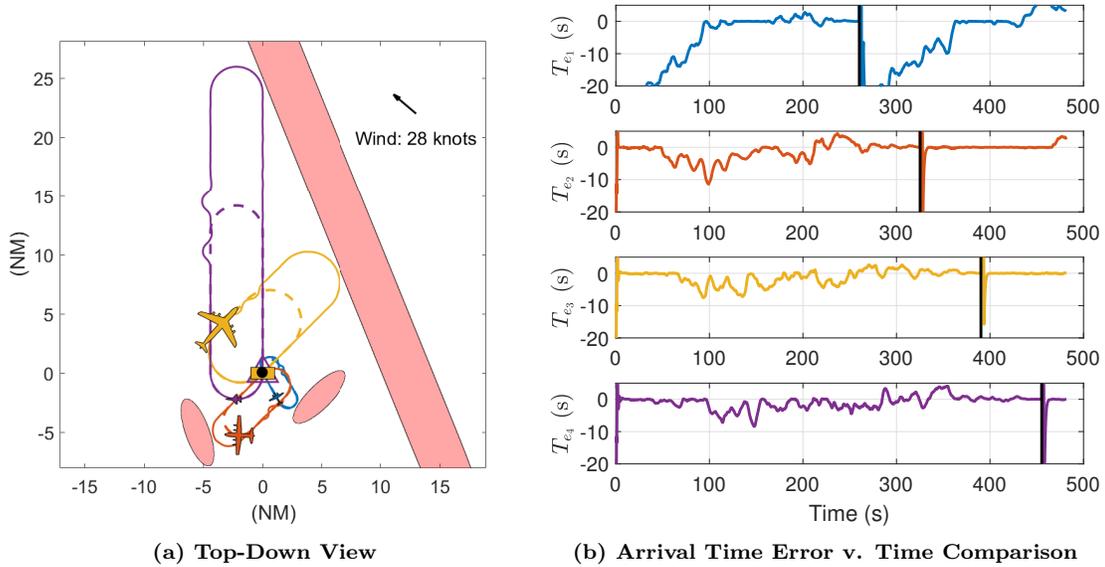


Figure 74. Scenario 1: Static with Keep Out Zone

equilibrium point is zero, see Sec. 6.3.

The equilibrium point for the remaining aircraft depends on the preceding aircraft which causes the error to drift towards the error of the preceding aircraft. This effect can be seen as the time error of aircraft two through four decreases towards the value of the preceding aircraft for the first half of the simulation after which the error returns to zero. Overall, this scenario demonstrates the arrival time performance and stability of the racetrack algorithm while managing keep-out zones against a static target.

Moving Target with Keep-Out Zones.

The next scenario is representative of a continuous monitoring scenario where each aircraft must overfly a moving target at equally spaced intervals. In this scenario, the arrival heading for each aircraft remains constant for each pass. Non-constant headings are demonstrated in Sec. 7.8. A keep-out zone is included which limits the racetrack size of one of the aircraft on the first pass; the fighter aircraft (purple). Finally, the aircraft are all initialized at the same (x, y) coordinates and must acquire

the initial desired racetrack.

- Initial Orbit Arrival Time: 260 *s*
- Time Spacing: 65 *s*
- Steady Wind Speed: 59 *ft/s* (20% of slowest reference airspeed)
- Steady Wind Heading: 315 *deg*
- Target Speed: 147 *ft/s* (50% of slowest reference airspeed)
- Target Heading: 270 *deg*

The arrival error results for each orbit are presented in Table 16. All aircraft demonstrated nominal arrival time error performance and low tracking error from the target at orbit completion due to the predictive calculation of the endpoint; see Sec. 5.8.

Table 16. Simplified Model Multi-Pass Results at Orbit Completion

Orbit 1			
Aircraft	Arrival Time Error (<i>s</i>)	Arrival Airspeed Error (<i>ft/s</i>)	Distance to Target (<i>ft</i>)
1	0.08	4.2	71
2	-0.01	-0.0	47
3	0.02	1.8	46
4	0.01	1.4	16
Orbit 2			
Aircraft	Arrival Time Error (<i>s</i>)	Arrival Airspeed Error (<i>ft/s</i>)	Distance to Target (<i>ft</i>)
1	0.21	11.9	85
2	0.19	13.1	61
3	0.26	26.9	64
4	0.00	1.1	105

Of note is the high target speed relative to the slowest (blue) aircraft; 50% of the blue aircraft reference airspeed. Although the speed is associated with higher errors,

on average, the heading and turn direction of the blue aircraft results in a near zero target heading relative to the blue aircraft's $y^{(r)}$ -axis, a heading associated with lower error on average. Additionally, it is interesting that most of the aircraft resulted in a slightly higher arrival time error on the second orbit however, this is likely due to wind turbulence since the error is within the range of nominal performance.

The top-down trajectory is shown in Fig. 75a. The purple trajectory is clearly limited by the keep-out zone as designed which also activates the s-turn logic. The purple trajectory approaches but does not violate the keep-out zone as desired. In this case, the turn radius adjustment is active pushing the trajectory to the edge of the keep-out zone “pad,” see Sec. 5.8.

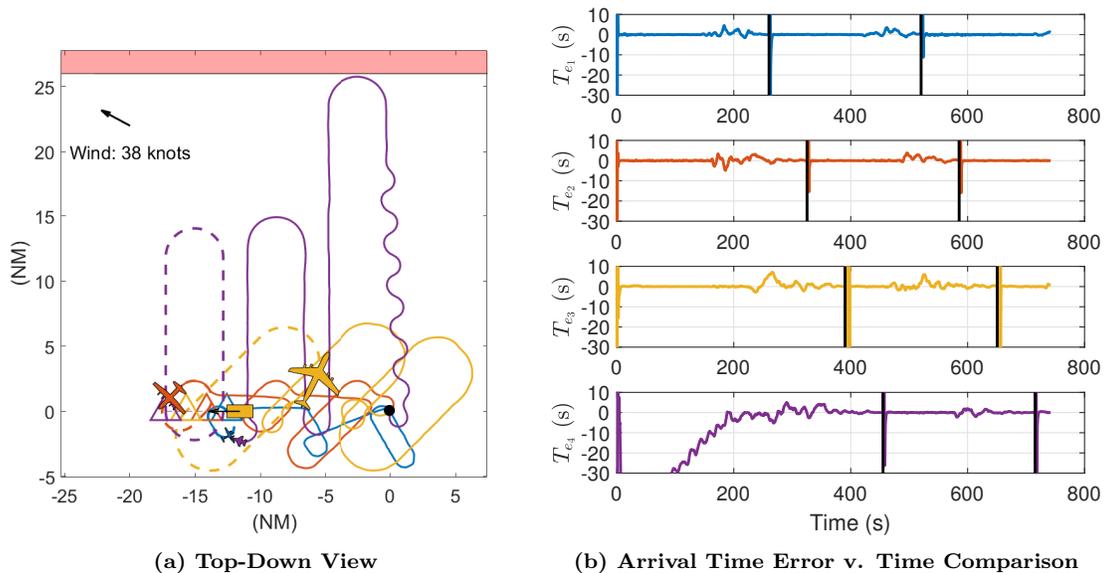


Figure 75. Scenario 2: Moving Target with Keep Out Zone

Figure 75b presents the arrival time errors. The error of the first three aircraft is small for the initial portions of the orbits. These sections with error near zero correspond to times when the racetrack size is not limited and is able to efficiently correct any arrival time error. As expected, the error of the fourth aircraft is initially negative (early) prior to slowly increasing towards zero due to the racetrack size limitation caused by the keep-out zone.

This simulation also highlights the behavior where the error variation increases as an aircraft enters the final turn. This occurs just before 200 *s* for the first aircraft. The arrival time error variation increases due to a combination of the racetrack size becoming fixed and the increased variation in arrival time error expected during turns, see Sec. 5.6. The increased variation in arrival time error of the first vehicle then propagates to the following vehicles once their racetrack sizes become saturated. The errors remain bounded due to the string stable design of the algorithm. Overall, this scenario demonstrates the algorithm's ability to repeat this pattern indefinitely in lieu of a higher target speed and keep-out zone.

Non-Cooperative Target with Keep-Out Zones.

The final scenario is a non-cooperative target with keep-out zones. The keep-out zone is positioned to affect the first aircraft and require the maximum racetrack size to decrease when the target stops. Although this presents a challenging scenario, the nominal arrival time error performance is achieved for all aircraft. Similar to the moving target scenario, the aircraft are initialized at a common point and must first acquire their respective desired paths. The parameters of the scenario are:

- Initial Orbit Arrival Time: 279 *s*
- Time Spacing: 70 *s*
- Steady Wind Speed: 88 *ft/s* (30% of slowest reference airspeed)
- Steady Wind Heading: 330 *deg*
- Target Speed: 57 *ft/s* (20% of slowest reference airspeed)
- Target Heading: 270 *deg*
- Target Stop Time: 139 *s* (50% of first aircraft desired arrival time)

The results at the completion of each aircraft’s orbit are presented in Table 17. It is not unexpected that the first aircraft arrives early due to the stopping of the target however, the error is within nominal performance tolerances for a light utility aircraft. The stopping target forces the aircraft to re-position to a new orbit while achieving the same desired arrival time.

Table 17. Simplified Model Non-Cooperative Results at Orbit Completion

Aircraft	Arrival Time Error (s)	Arrival Airspeed Error (ft/s)	Distance to Target (ft)
1	-1.00	-31.4	67
2	0.01	1.7	9
3	0.03	2.9	56
4	-0.04	-6.2	96

Figure 76a presents a close-up view of the trajectories; the upper portion of the purple trajectory is not shown because it behaves nominally. The most notable feature of the trajectory are the sharp turns which correspond to the time the target stops. The effect is seen in the figure at the (x,y) coordinates of approximately $(-6, 12)$ where the purple trajectory turns sharply. The parallel portions before this turn indicate the initial and final racetrack position.

In the arrival time plot, Fig. 76b, the target stops at 139 seconds where the arrival time error of all aircraft exhibits a large change due to the racetrack algorithm resting to the new endpoint. The blue aircraft becomes keep-out zone limited immediately after this stop as indicated by the slower decrease in arrival time error after the target stops. The blue aircraft increases its turn radius and executes a number of s-turns to drive the arrival time error to zero.

Each following aircraft’s arrival time error decreases once the aircraft the racetrack size becomes fixed due to reaching the final turn (purple and red aircraft) or due to the keep-out zone (yellow aircraft). The decrease in error occurs because the blue aircraft is expected to arrive “early” after the target stops and the following aircraft

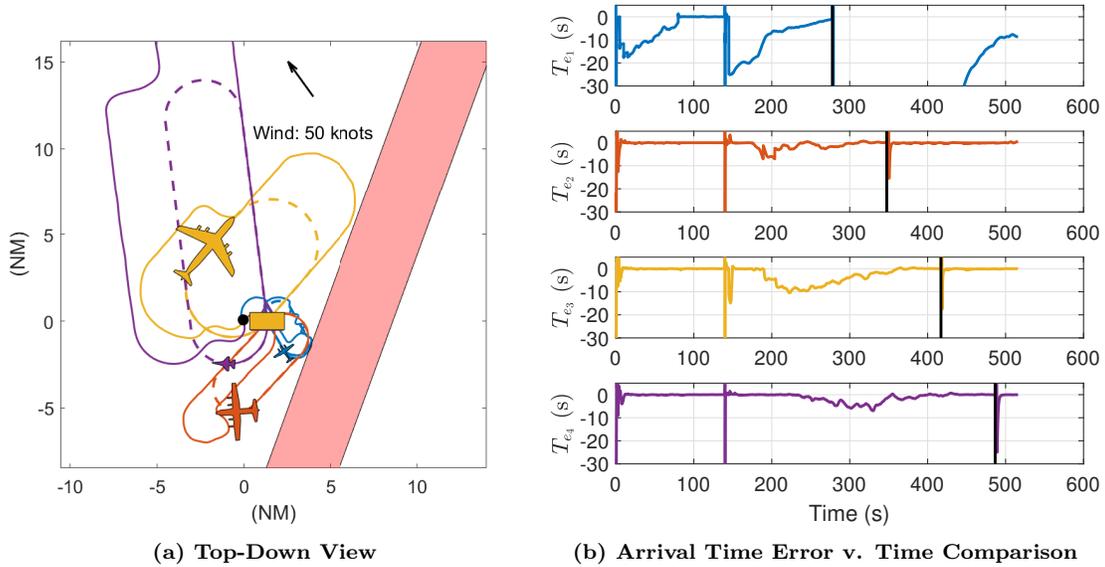


Figure 76. Scenario 3: Non-Cooperative Target with Keep Out Zone

adjust their racetrack sizes to maintain spacing off the estimated “early” time. The errors decrease towards zero once the error of the preceding aircraft also nears zero.

The string stable design of the algorithm ensures the errors remain bounded in lieu of the error variation of the lead vehicle. Overall, this scenario demonstrates the ability of the algorithm to adapt to a moving endpoint while adjusting for previously inactive keep-out zone constraints.

7.8 Nonlinear Aircraft Model Simulations

This section is the primary evaluation of the nonlinear aircraft model developed in Chapter III. The model is first compared to the simplified model to verify the assumption that the simplified model accurately represents the nonlinear model. Then, the nonlinear model and racetrack algorithm are evaluated in graduation exercises similar to those in the previous section.

Comparison to Simplified Model.

Chapter III simplified the full nonlinear equations of motion with several assumptions. In addition to allowing the stability analysis in Chapter VI, the time to calculate the trajectory utilizing the simplified model is significantly lower than when using the nonlinear model. First, the simplified and nonlinear model are compared in an identical scenario to determine if the simplified model characterization is likely to apply to the nonlinear model.

A multi-pass single-aircraft scenario was chosen to stress the algorithm to observe the differences between the two models. The scenario begins with the aircraft over the target as if a pass was just completed. The target continues at a steady speed while the aircraft reacquires the racetrack to overfly the target a second time. The wind and target speeds are chosen to represent a difficult scenario for any aircraft based on their scale-factors. Note that the speeds are less realistic when applied to the desired speed of the fighter aircraft. Finally, turbulence is included in the simulations. The set of parameters for this scenario is listed below:

- *TTSF*: 1.6
- Target-Speed Scale-Factor: 0.3 (259 *ft/s* or 153 knots),
- Target Heading: 133 *deg*,
- Wind speed Scale-Factor: 0.4 (346 *ft/s* or 205 knots),
- Wind Heading: 257 *deg*,
- Final Racetrack Ground Course: 10 *deg*,
- Counterclockwise Orbit.

Figure 77 presents the trajectory and parameter comparison of the two models; the aircraft are initialized at the coordinates (0, 0). Equal points in time are identified

by \times marks in both plots. The simplified model resulted in an arrival time error of 0.03 s and an airspeed error of 3.5 ft/s while the nonlinear model resulted in an arrival time error of 0.03 s and an airspeed error of 6.3 ft/s. The similarity in arrival errors supports the assumptions of the simplified model remain valid and allow the simplified model to be an accurate approximation of the nonlinear model when a nominal level of performance is achieved. However, it is also of interest to evaluate the difference between the two models over the course of the entire simulation.

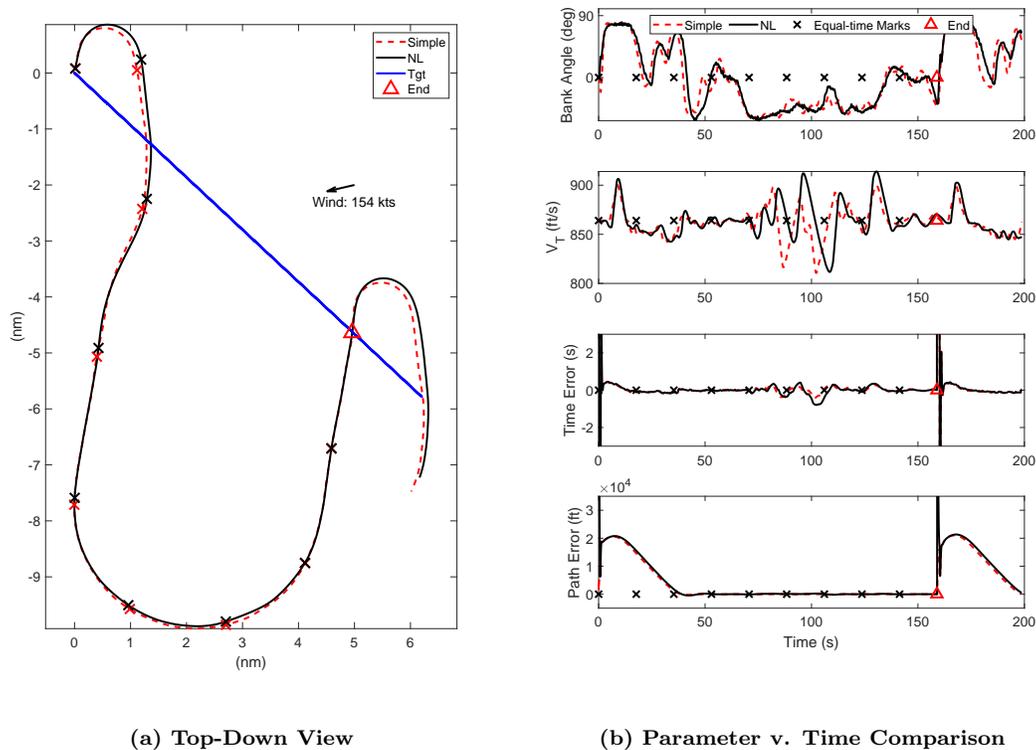


Figure 77. Simplified Model and Nonlinear Model Repeat Pass Comparison

In Fig. 77a the simplified model performs a slightly tighter initial turn resulting in a position further south of the nonlinear model trajectory once both are tracking the southward leg the racetrack. This is due to the simplified model reaching the maximum load factor more rapidly than the nonlinear model.

The difference in turns results in an average distance between the two trajectories

of 1254 *ft*. However, the racetrack algorithm determines a racetrack size based on the current position and time. The result is that the two simulations have a slightly different racetrack size as indicated by the difference in final turn location near the bottom of the plot. The effect of the different racetrack sizes is that the trajectories converge during the final quadrant of the racetrack.

Moreover, the average difference between the simulations during the final straight segment is only 35 *ft*. A similar behavior is apparent in the average arrival time error. The average arrival time error difference prior to orbit completion is 0.54 *s* while the average arrival time error difference in the final quadrant is only 0.02 *s*. The behavior during this final segment indicates the simplified model dynamics are an acceptable approximation of the nonlinear model dynamics.

The difference in track size will affect the overall performance characterization for scenarios near racetrack saturation with large path acquisition turns. Therefore, it is concluded that a nonlinear model characterization would not be significantly different from the simple model in static scenarios. Moving and non-cooperative scenarios would likely see a small increase in average arrival time error for scenarios which require more path acquisition (fast target and/or early stop times). However, the racetrack algorithm performs as expected for both models and the trends in arrival time error with respect to scenario parameters are not expected to change between the models.

Nonlinear Cooperative Control Scenarios.

The nonlinear cooperative control scenarios are similar to the three scenarios presented Sec. 7.7. However, only a fighter-like nonlinear model is analyzed and the scenarios were modified to demonstrate similar algorithm functions as demonstrated in Sec. 7.7. The nonlinear dynamics model itself is described in Chapter III.

As in the simplified model scenarios, aircraft trajectory is shown as a solid line in the corresponding color, the desired racetrack is a dashed line, the predicted endpoint a color coded triangle, and keep-out zones are a shaded red area. All aircraft are initialized at a common point indicated by the black circle. The target is an orange rectangle with a speed vector arrow. The arrival time plots are also color coordinated and the end of each orbit is marked by a black vertical line.

Videos of both the simplified dynamics model and nonlinear dynamics model scenarios may be found at the following YouTube link, <https://www.youtube.com/watch?v=PV1fj08ocFY>. The location of each specific scenario is provided in the video description and a short description of each scenario is given prior to each simulation.

Static Target with Keep-Out Zones.

The static case is designed to cause each aircraft to be keep-out zone limited and force the algorithm to correct some initial arrival time error. The pertinent scenario parameters are:

- Initial Orbit Arrival Time: 139s
- Time Spacing: 34.7 s
- Wind Speed: 172 *ft/s* (20 % of reference airspeed)
- Steady Wind Heading: 315 *deg*

The scenario and resulting trajectories are presented in Fig. 78 and the actual arrival errors are shown in Table 18. All aircraft achieved nominal performance.

In this scenario, the first and fourth aircraft are immediately racetrack size limited while the second and third are not. The estimated arrival time of the first aircraft slowly increases corresponding to the slow increase in arrival time error. The race-track size of the second and third aircraft increases to match the desired spacing

Table 18. Nonlinear Model Static Results at Orbit Completion

Aircraft	Arrival Time Error (s)	Arrival Airspeed Error (ft/s)	Distance to Target (ft)
1	-0.02	-4.2	88
2	0.05	6.7	144
3	0.02	5.5	109
4	0.00	-0.6	110

until limited by the keep-out zone at approximately 50 s when the arrival time error decreases slightly.

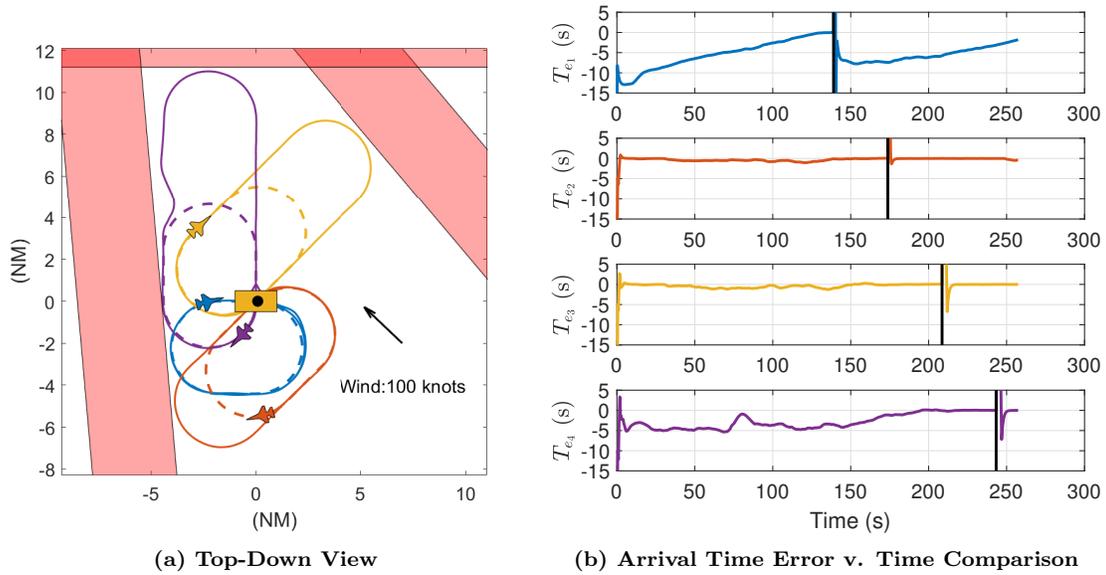


Figure 78. Nonlinear Model: Static Target with Keep Out Zone

Also of note is the behavior of the fourth aircraft where the arrival time error remains near -5 s for the majority of the simulation. This is because the keep-out zones resulted in early estimated arrival times for both the first and fourth aircraft.

In the case of the first aircraft, the desired arrival time is constant while the estimated arrival time increases. In the case of the fourth aircraft, the desired arrival time is also increasing since it is a function of the preceding aircraft. Therefore, it is expected that the fourth aircraft's arrival time error beings decreasing at approximately the same time the first aircraft's arrival time error approaches zero. Finally,

a short segment of s-turns is initiated in the purple trajectory at approximately 75 *s* when the arrival time error passes the threshold to activate the logic, an arrival time error of -5 *s*. Overall, this scenario demonstrated nominal performance and avoided keep-out zones as desired.

Multi-pass.

The multi-pass scenario is designed to activate keep-out zone logic while the algorithm guides the aircraft to overfly a moving target at specified intervals. Unlike the simplified demonstration, this scenario changes the arrival heading between the first and second orbit. The pertinent scenario parameters are:

- Initial Orbit Arrival Time: 149 *s*
- Time Spacing: 37.2 *s*
- Target Speed: 173 *ft/s* (20 % of reference airspeed)
- Target Heading: 90 *deg*
- Wind Speed: 259 *ft/s* (30 % of reference airspeed)
- Steady Wind Heading: 330 *deg*

The actual arrival time errors for each pass are shown in Table 19. All aircraft achieved nominal performance on each orbit.

Figure 79a presents the trajectory for the nonlinear multi-pass scenario. Note, the arrival heading changes from first to second pass. The red aircraft trajectory is constrained and avoids the keep-out zone as designed.

Figure 79b shows the arrival time error behaves as expected due to the effect keep-out zones in the case of the red aircraft. Additionally, the blue aircraft has an increase in arrival time error after completing the first orbit due to a combination

Table 19. Nonlinear Model Multi-Pass Results at Orbit Completion

Orbit 1			
Aircraft	Arrival Time Error (s)	Arrival Airspeed Error (ft/s)	Distance to Target (ft)
1	-0.00	0.4	83
2	-0.03	-2.5	84
3	-0.06	-7.1	144
4	-0.00	2.9	75

Orbit 2			
Aircraft	Arrival Time Error (s)	Arrival Airspeed Error (ft/s)	Distance to Target (ft)
1	0.03	2.0	89
2	-0.10	-9.5	83
3	-0.01	-1.8	91
4	0.01	-0.8	61

of becoming a following aircraft and a large change in the endpoint conditions. The blue aircraft's first arrival point was at the coordinates (0, 0) with a heading approximately East while the second point is approximately at (-5, 0) in a South-southeast direction. Overall, this demonstration highlights the ability of the algorithm to adapt to changing final headings with a moving target and keep-out zones.

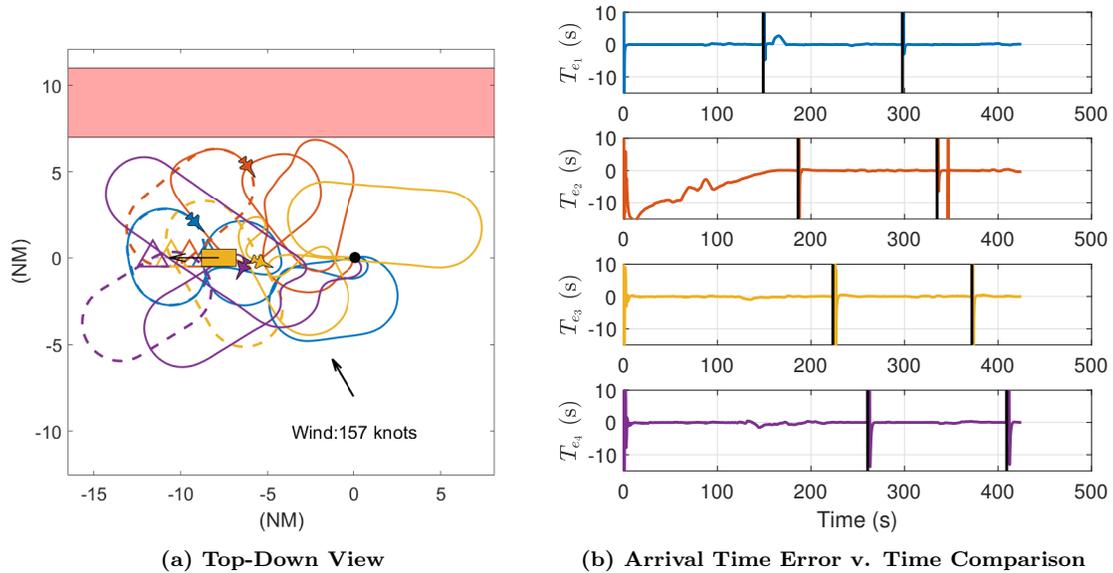


Figure 79. Nonlinear Model: Moving Target with Keep Out Zone

Non-Cooperative Target with Keep-Out Zones.

The final nonlinear cooperative scenario is a non-cooperative target scenario. Similar to the simplified case, the scenario was set up such that the keep-out zone logic for at least one aircraft would activate after the stopped moving. The pertinent parameters of the scenario are:

- Initial Orbit Arrival Time: 149 *s*
- Time Spacing: 37.2 *s*
- Target Stop Time: 89 *s* (60% of first aircraft desired arrival time)
- Target Speed: 259 *ft/s* (30 % of reference airspeed)
- Target Heading: 221 *deg*
- Wind Speed: 259 *ft/s* (30 % of reference airspeed)
- Steady Wind Heading: 330 *deg*

The actual arrival time errors of the non-cooperative scenario are presented in Table 20. This scenario highlights a case where the first aircraft was unable to achieve the desired arrival time due to the time the target stopped. The target stop time ratio corresponds to conditions which resulted in large arrival time errors in the performance characterization. However, the remaining aircraft all adapted and demonstrated nominal performance.

Figure 80 presents the trajectories of the scenario. First note the trajectory of the lead aircraft, in blue. The sharp turn in the blue trajectory at the coordinates $(-2, -5)$ correspond to the point when the target stopped. The relative position of the aircraft, target position, and target airspeed caused the endpoint to shift to the Northeast. The result is the blue aircraft was approximately at the start of the final

Table 20. Nonlinear Model Non-Cooperative Results at Orbit Completion

Aircraft	Arrival Time Error (s)	Arrival Airspeed Error (ft/s)	Distance to Target (ft)
1	-12.38	-83.3	134
2	0.07	9.6	150
3	-0.02	1.5	116
4	-0.03	-0.4	80

turn prior to the target stop and just over halfway through the final turn prior to the target stop. This results in the large decrease in arrival time error indicating an early arrival. The re-orbit logic did not occur in this scenario because a full orbit would result in a late arrival.

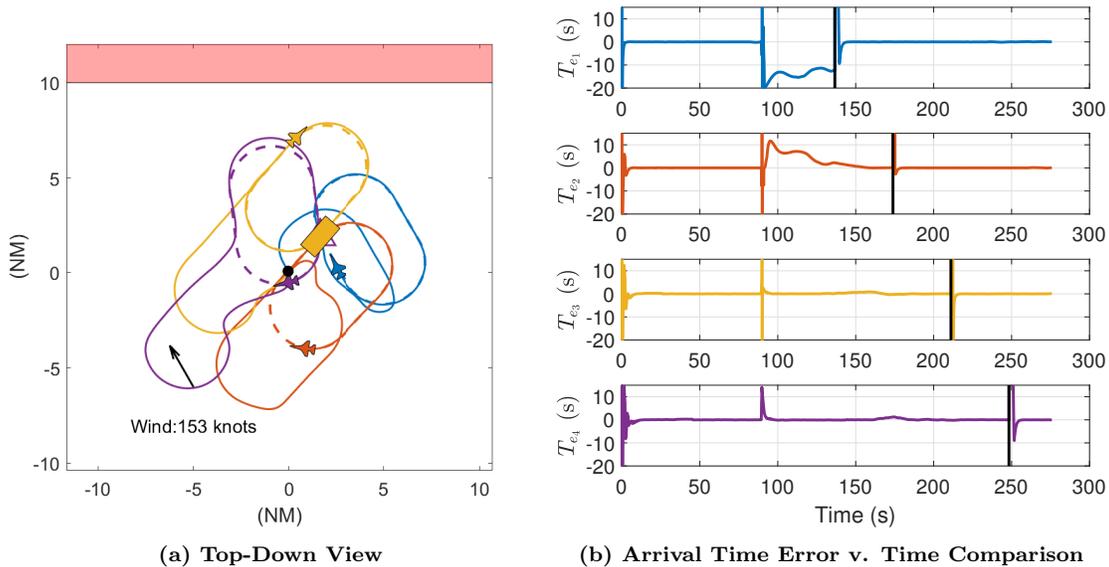


Figure 80. Nonlinear Model: Stopping Target

Although the first aircraft did not meet its initial arrival time, the remaining aircraft were able to adapt and achieve nominal performance. Figure 80b shows that the red aircraft exhibits large arrival time error changes since the target stopped while the red aircraft was in its final turn, increasing the distance to the endpoint. However, aircraft apparently had sufficient time to correct the error while utilizing the turn radius adjustment to fly outside the desired path. Overall, the scenario

demonstrated the ability of the racetrack algorithm to adapt to a non-cooperative target utilizing the nonlinear dynamics model.

7.9 Results Summary

Overall, the results demonstrated that the stability and performance of the racetrack algorithm utilizing the Gaussian Quadrature arrival time estimation method. The performance was characterized for a single aircraft across a wide range of scenarios which were similar to multi-aircraft scenarios. Next, the string stability analysis from Chapter VI was verified through simulation. Finally, three graduation exercises were used to demonstrate the algorithm's performance in realistic scenarios utilizing both the simplified model with a heterogeneous group of aircraft and the nonlinear model with a group of fighter aircraft.

VIII. Conclusion

8.1 Research Questions and Contributions

The goal of this research is to develop a practical control algorithm which solves the coordinated arrival time problem in real time while considering environmental effects, airspace, and maintaining a predictable flight path. The goal of this research is to allow scenarios similar to the continuous monitoring problem where the arrival time of an aircraft may either be fixed or dependent on a preceding vehicle. This resulted in the hypothesis:

A control algorithm exists which may predictably guide an aircraft through a constrained environment to achieve a desired final position, heading, and velocity at a specified arrival time by adjusting both the speed and flight path of the aircraft

Overall, this research supported the hypothesis with the development and analysis of the racetrack algorithm. Four research questions were identified and answered which support this hypothesis:

1. What is the time optimal lead turn time for an aircraft, controlled by roll rate, to closely follow a Dubins path?
2. What technique is best suited to control the time of arrival and velocity of an aircraft in near real time?
3. Is a keep-out zone avoidance feature able to be integrated into the arrival time control algorithm?
4. How does a arrival time control algorithm perform in the presence of environmental anomalies and time dependent constraints?

To support these questions, five contributions to the control of autonomous vehicles were made:

- A near-optimal solution to the optimal lead turn time problem,
- A framework for a real-time arrival time controller which compensates for steady-state winds, is robust to random turbulence, allows for repeat visits, and operates within keep-out zone constraints,
- A method to guarantee the string stability of a group of airspeed heterogeneous, nonlinear aircraft with a cascaded interconnection utilizing the developed arrival time algorithm,
- A performance characterization of the real-time arrival time controller across a range of wind conditions and arrival requirements,
- A demonstration of the arrival time controller in single and multi-vehicle scenarios with cooperative and non-cooperative targets.

In order to develop an algorithm which can achieve coordinated arrival time in a windy environment which may have keep-out zones, two primary control effectors have been identified: path length and airspeed. The desired path is defined as a racetrack because the overall length of the path may be changed by a single variable, the width of the racetrack. This scheme allows for rapid control over the remaining path length, allowing the estimated arrival time to rapidly be controlled. Furthermore, the racetrack shape is well defined as a set of straight and circular path segments allowing for keep-out zone constraints to be enforced. Finally, utilizing this approach reduces the complexity of the coordinated arrival time problem by decoupling the path-following controller and the arrival-time controller.

The arrival time algorithm is built around a framework which allows parts of algorithm analyzed in this paper to be interchanged. The path follower, keep-out zone control, and upper-level path manager may be changed out or handled by a separate algorithm or computer. This may be desired when implementing the algorithm in

hardware based on the constraints of the hardware system. Finally, the racetrack algorithm and subsequent analysis are performed in MATLAB[®] and Simulink[®]. Two different dynamics models were used to develop and analyze the algorithm.

The dynamics models used in this research are derived in Chapter III. A simplified model identified key assumptions which allowed the number of dynamic states to be reduced from the nonlinear equations of motion. With fewer state variables, the model allowed the formation of the optimal lead turn problem, a robust string stability analysis, and an extensive characterization of the racetrack algorithm. To further the realism of the simulation analysis, a control system which accepted airspeed, altitude, and bank angle commands was developed for an existing nonlinear dynamics model of a fighter aircraft [6]. Finally, two key assumptions of this research are that the windspeed and target speed are limited to half the desired airspeed of the slowest aircraft in a group. This assumption prevents the infeasible scenario where the aircraft and target relative speed is zero.

Optimal Lead Turn.

The first research question addressed in this research is, “What is the time optimal lead turn time for an aircraft, controlled by roll rate, to closely follow a Dubins path?” This question is addressed in Chapter III and Chapter IV. The question is derived from the chosen path, a racetrack, which is a type of Dubins path where the initial and final pose are connected. The simplified dynamics model developed in Chapter III allows for the development of the optimal lead turn problem, Chapter IV, which seeks to determine the optimal control that maneuvers a fixed wing aircraft from one section of a Dubins path to the next with minimal path error.

Several assumptions have been applied to the optimal lead turn problem to allow a near-optimal, but analytic, solution to be developed, referred to as the TPAS solution

in this research. The three primary assumptions of the TPAS solution are that the maneuver is completed in minimum time, that the aircraft begins tangent to the initial path segment, and that the position and heading dynamics may be approximated Taylor polynomials. The resulting solution consists of a lead turn point to begin the segment transition and a definition of the optimal roll maneuver.

The near optimality of the TPAS solution has been confirmed in this research by comparing the TPAS solution to a numerically calculated optimal solution. The chosen performance metric is the tracking-error cost relative to the tracking-error cost of a control scenario where no lead turn is performed. The results show that the TPAS solution is within 3.2% of the numerically calculated optimal solution on average but varies between -5.9% to 13.3% over 1,458 different scenarios. This level of similarity supports the claim that the TPAS solution is near-optimal.

This research further evaluated the TPAS solution by integrating the TPAS solution into existing guidance laws to determine if the maneuver improved tracking performance in more realistic scenarios with non-zero wind and turbulence. Four types of existing guidance laws have been studied: carrot-chasing algorithm, nonlinear dynamic inversion, nonlinear guidance law, and a vector field follower. The results show that simply transitioning to the next path segment at the lead turn point improved tracking performance by 34% on average for three of the four controllers when compared to the baseline controller performance. However, the results also show that the benefit of the lead turn decreases as the level of turbulence increases because turbulence the terminal state constraints used to calculate the TPAS solution to become invalid (e.g. the TPAS assumed zero initial tracking error which could not be guaranteed in a realistic environment with turbulence).

Overall, the contribution of the optimal lead turn research is the derivation of the optimal lead turn problem and the subsequent development and analysis of a

near-optimal solution to the optimal lead turn problem which is shown to provide a tracking benefit in realistic scenarios with non-zero wind and turbulence.

Arrival Time and Airspeed Control.

The second research question addressed by this dissertation is, “What technique is best suited to control the time of arrival and velocity of an aircraft in near real time?” This question is addressed in Chapter V-VII. This question is derived from the requirement of the coordinated arrival time problem to ensure that an aircraft arrives at some desired time which may be fixed or a function of the arrival time of a preceding aircraft. First, it has been shown that assuming a ground-fixed racetrack allows the time to fly along the path to be accurately estimated even with non-zero wind conditions. This estimate utilizes a Gaussian quadrature to accurately incorporate the non-constant effect wind has on the aircraft ground speed which, in turn, determines the estimated arrival time.

With an accurate time estimate, it is then shown that estimating the time remaining assuming the aircraft traveled at some desired airspeed allows for both a desired arrival time and desired arrival airspeed to be achieved. This formulation is similar to a virtual target controller in that the arrival time error is defined to be zero at some virtual point on the path. The point is determined such that an aircraft at the point which travels along the path at the desired airspeed will arrive at the endpoint at the desired time. Therefore, a controller which drives the arrival time error to zero will also drive the airspeed error to zero. In order to show the controller does drive the arrival time error to zero, string stability theory has been applied.

The stability implications of a group of interconnected vehicles is analyzed in Chapter VI. The literature review in Chapter II discusses the concept of string stability as the stability of a group of interconnected dynamic systems. This research

expands upon the string stability body-of-knowledge by deriving a method to guarantee the string stability of a group of airspeed heterogeneous aircraft with respect to arrival time error.

The airspeed control includes two proportional terms, arrival time error and arrival airspeed error, which minimizes the amount of information transmitted between aircraft. Utilizing this control, only the estimated arrival time and the relative order of each aircraft is required to be transmitted between aircraft. The arrival time error and airspeed dynamics of the chosen control scheme, utilizing a desired airspeed to estimate the arrival time, are derived and non-dimensionalized in airspeed error. The dynamics are then transformed into nonlinear time-variant system to formulate the method to determine appropriate control gains.

An exponentially stable set of control gains is derived by re-formulating the dynamics as a polytropic linear differential inclusion. The benefit of this formulation is that it defines the constraints which guarantee that a specific set of gains result in an exponentially stable system across all wind conditions, assuming the wind magnitude is bounded. The aircraft defined airspeed command limits are adjusted to account for the speed of the other aircraft and the windspeed. The combination of the exponentially stable gains and airspeed command limits proves that the entire interconnected system of aircraft, regardless of number of aircraft, is string stable.

Regarding the racetrack algorithm, this guarantees that any arrival time disturbances experienced by a leading aircraft would not lead to unbounded disturbances of the following aircraft. Once the lead aircraft arrives, the arrival time estimate of the next aircraft is guaranteed to be bounded and then exponentially decrease towards the desired arrival time. Utilizing the calculated control gains in simulation, the simulation results confirmed string stable behavior for as many as 20 aircraft assuming all are accurately following their desired paths. The results also confirm string stable

behavior for smaller groups of four aircraft in realistic scenarios utilizing the racetrack algorithm.

Two contributions resulted from this research question. The first is a method to guarantee the string stability of a group of airspeed heterogeneous, nonlinear aircraft with a cascaded interconnection. The second contribution is the development of a real-time arrival controller which is able to compensate for steady-state winds, is robust to random turbulence, allows for repeat visits.

Keep-Out Zone Constraints.

The third research question addressed by this research is, "Is a keep-out zone avoidance feature able to be integrated into the arrival time control algorithm?" This is addressed in Chapter V and Chapter VII. The question is derived from the imposed constraint that the coordinated arrival time problem should include airspace-like constraints. These types of keep-out, or keep-in constraints, are constructed as large, non-moving keep-out zones. Additionally, a secondary constraint of this question is that the keep-out zone avoidance method must be implementable in MATLAB[®] and Simulink[®] since these programs are chosen to develop the racetrack algorithm.

The formulation of the racetrack as the desired path allows keep-out zones to be enforced by simply ensuring the desired path, the racetrack, does not violate a keep-out zone. While this method does not actively prevent an aircraft from flying into a keep-out zone, a distance "pad" is used to keep the racetrack a certain distance from the keep-out zone. This distance corresponds to the expected path tracking performance of the algorithm. If more control over keep-out zones is desired, a path follower with active keep-out zone avoidance would ensure the aircraft does not overly the keep-out zone.

The keep-out zone constraints are enforced by determining the maximum race-

track size which does not violate the keep-out zone constraints. The optimization is formulated using superquadrics to define the fixed borders of the keep-out zones. Superquadrics allow rectangular and circular shapes to be evaluated with a simple “in-out” function which returns a value greater than one outside the shape and less than one inside the shape. The optimization is performed using FMINCON[®] in MATLAB[®] whenever a large change to the desired endpoint is detected by the race-track algorithm. A limitation of this method is that FMINCON[®] is computationally inefficient when integrated into a Simulink[®]. This limitation results in a recommendation to improve the efficiency of the algorithm if implemented.

The framework of the algorithm does not require the keep-out zone enforcement to occur within the actual racetrack algorithm. The enforcement occurs within the algorithm due to the requirements of Simulink and the chosen simulation architecture. For implementation, it is recommended the maximum racetrack size be determined outside the racetrack algorithm and simply provide the maximum track size as a variable input to the algorithm.

Overall, the simulation results show that this keep-out zone avoidance method ensures the racetrack size was limited to avoid keep-out zones as desired for both rectangular and circular static keep-out zones. This supports the contribution of the development of an arrival time controller which adheres to keep-out zone constraints.

Arrival Time Control Algorithm Performance.

The final research question addressed by this dissertation is, “How does an arrival time control algorithm perform in the presence of environmental anomalies and time dependent constraints?” This question is addressed in Chapter VII. Given the variety of wind, target, aircraft, and keep-out zone conditions, an exhaustive analysis of the racetrack algorithm is impractical. With the string stability guarantees made

in Chapter VI, an extensive single-vehicle characterization has been performed and a limited number of multi-vehicle “graduation exercises” ensure the nominal string stable performance is attained in realistic scenarios.

As discussed in Sec. 8.1, a “nominal” level of performance is determined based on the string stability analysis and the aircraft type. This level of performance determines the expected variation of arrival time error due to wind and turbulence assuming the path is accurately followed. Since the control is similar to a virtual target, this variation may be viewed as variation around the desired virtual target. The highest nominal performance value is 1.32 *s* associated with the light utility aircraft and the lowest nominal performance value was 0.23 *s* associated with the fighter aircraft.

The single vehicle characterization is performed utilizing the simplified dynamics model for three primary scenario types: static, multi-pass, and non-cooperative moving target. The multi-pass results most closely mimic the continuous monitoring problem which is a primary motivation of this research. The non-cooperative scenario study the effect of a moving target which stops and requires the aircraft to adjust to a new racetrack to overfly the target at the desired time.

In all scenarios, the aircraft successfully overflew the target. The characterization showed that a nominal level of performance was achievable for a wide range of conditions. An increase in windspeed and target speed were generally associated with an increase in arrival time error while increasing the desired arrival time was shown to improve arrival time error on average. Additionally, the multi-pass and non-cooperative moving target scenarios showed that certain combinations of wind and target heading resulted in lower arrival time error. However, these variables do not affect arrival time error linearly.

Returning to the idea of “nominal” performance, if an aircraft achieves nominal

performance at some time during a scenario, the error will remain at that level due to the stability guarantees if the scenario parameters do not change. Therefore, scenarios with error outside of the nominal performance range represent cases where the aircraft did not have enough time to acquire the “nominal” level of performance, which is analogous to acquiring the discussed virtual target. Given a large enough desired arrival time, the algorithm could achieve a nominal level of performance for the tested environmental configurations.

Following the single-vehicle characterization, three “graduation exercise” scenarios have been chosen to evaluate the algorithm in a multi-vehicle, constrained environment with a group of four airspeed heterogeneous aircraft is simulated with desired speeds ranging from 173 knots to 511 knots. The static scenario is designed such that the desired racetrack of each aircraft is size limited by the keep-out zones. The multi-pass scenario is setup to ensure a keep-out zone is always active for one aircraft. Finally, the non-cooperative scenario is setup so that one aircraft is initially keep-out zone limited and the stopping of the target moves estimated endpoint towards the keep-out zone. This requires the racetrack size to shrink to stay out of the, now, closer keep-out zone. The results of the graduation exercises utilizing the simplified dynamics model show the algorithm attained the nominal level of performance based on the single-vehicle performance characterization.

Next, the performance of the racetrack algorithm is evaluated utilizing the nonlinear dynamics model. The nonlinear model is not used during the extensive simulation characterization due to its complexity and the large computational cost required. In the comparison, the only simulation component to change is the dynamics model.

The comparison shows that the racetrack algorithm is able to achieve the same nominal levels of performance regardless of the dynamics model. However, the behavior of the nonlinear model during turns resulted in the conclusion that moving

target scenarios would likely have a higher average arrival time error due to the observation that the nonlinear model racetrack size would saturate, and increase arrival time error, for slightly “easier” scenarios than the simplified model. Although the non-nominal arrival time errors may be increased, there is no indication that the relative changes in error due to the scenario parameters would change. Therefore, it is concluded the error trends in the simplified model characterization would hold for the nonlinear model.

Finally, three additional graduation scenarios are developed to evaluate the race-track algorithm integrated with the nonlinear model. Since a single nonlinear fighter aircraft model with control system was developed, the original graduation scenarios are tailored for four fighter aircraft. The static scenario results show a nominal performance level for all aircraft in the group. The multi-pass and non-cooperative scenario simulations each result in one aircraft which does not achieve nominal performance due to the impact of a keep-out zone and the stopping of the target respectively. This result is not unexpected based on the single-vehicle performance characterization.

Overall, this research question results in the development and evaluation of a framework for a real-time arrival time controller which is able to compensate for steady-state winds, is robust to random turbulence, allows for repeat visits, and operates within keep-out zone constraints. The performance of this controller has been characterized for wind and target speeds up to half the desired airspeed with varying arrival requirements. Finally, the performance of the controller has been demonstrated with cooperative and non-cooperative targets to include challenging multi-vehicle graduation exercises utilizing both the simplified and nonlinear dynamics model.

Summary.

To conclude, this research develops and analyzes a practical control algorithm which solves the coordinated time of arrival problem in real time while considering environmental effects, airspace, and maintaining a predictable flight path. The results characterize the performance of the algorithm across a wide range of conditions and demonstrate its performance in realistic scenarios. The next step in this research is integration of the algorithm into a hardware system and testing onboard an aircraft.

8.2 Future Work and Recommendations

Three primary areas are identified to expand and refine the research of this dissertation. First, flight testing the racetrack algorithm will determine the validity of the assumptions in this research and provide a real-world demonstration of the racetrack algorithm performance. Next, specific portions of the racetrack algorithm are recommended for re-design because they may prohibit successful integration into a hardware system. Finally, an area of the optimal lead turn research is identified to expand the evaluation of the results to another type of guidance law.

Flight Test.

The primary focus of the-near term future work is the integration and flight testing of the racetrack algorithm on board an aircraft. A potential flight test platform for consideration is the Variable Stability In-Flight Simulator (VSS) Learjet operated by Calspan [89]. The aircraft is predominantly used by the United States Air Force Test Pilot School to allow students to experience flying a wide variety of good and bad flying aircraft. The aircraft's control system allows the operator to "match the open-loop flight dynamics" of the desired aircraft while providing a safety envelope which disables the poor characteristics should the limits be exceeded.

The VSS Learjet also allows the integration and operation of developmental control laws and algorithms within a “safety envelope” which will disconnect the developmental guidance law when certain flight limits are exceeded. The configuration of the aircraft allows the developmental software to easily be disabled in the event of unexpected behavior. While it is not the recommendation of the author to specifically use the VSS Learjet, it is recommended to use a system with a similar ability to safely integrate and test a developmental control law.

To integrate the racetrack algorithm into any flight hardware, it is recommended to integrate the algorithm into an appropriate ground test bench first. Integrating into the ground test bench first would ensure the system is compatible with the aircraft hardware and software systems prior to testing the system during flight test. This ground integration provides a confidence to the researcher that the flight test will provide useful data.

Since the algorithm is designed to simply accept a desired endpoint, the flight testing may utilize a virtual target to not limit the testing to a physical vehicle on the ground. A build-up approach is recommended for this type of flight test beginning with simple static endpoints and increasing to multiple passes on a moving and non-cooperative target. The limits and test conditions should also be used to define the limits for another simplified aircraft model which simulates the flight characteristics of the chosen flight test platform. This model should then be used to simulate the desired test conditions for comparison to the flight test data.

Racetrack Algorithm Research.

The graduation exercises highlighted that the keep-out zone avoidance method can be computationally expensive since it is integrated into the racetrack algorithm itself. When integrating the racetrack algorithm into a hardware system, this could

prevent the successful implementation of the algorithm.

It is recommended to implement the keep-out zone avoidance method in a separate loop outside of the control algorithm since keep-out zones only need to be evaluated when the endpoint substantially moves. This would remove the computationally expensive portion of the algorithm from the primary control loop.

Additionally, the comparison between the simplified and nonlinear fighter model showed that small differences in capability affect the overall limitations of the race-track algorithm. In the results, it was observed that the simplified model could initiate turns quicker than the nonlinear model resulting in the simplified model acquiring the desired path faster. Therefore, it is recommended to develop a specific simplified model which matches the turning and acceleration capabilities of an aircraft prior to implementation on a physical aircraft. This would allow an accurate and complete performance characterization.

Optimal Lead Turn Research.

The near-optimal lead turn solution was evaluated on a variety of guidance laws with the notable exception of the MPC guidance law used by the racetrack algorithm. The MPC guidance law was chosen for its ability to acquire the racetrack when far away from the path and its ability to follow the simple racetrack well. It is of interest to this research to determine if the lead turn would improve the path tracking performance of an MPC guidance law. Both the lead turn and the MPC guidance law minimize a path tracking error although the time window of the optimization and control assumptions are different. The lead turn minimizes the path tracking error during the optimal time-to-bank maneuver while the MPC guidance law chooses a single bank angle which minimizes the path tracking error over some time horizon; note the time horizons of all aircraft tested in this research are longer

than the associated lead turn times.

Appendix A. Near-Optimal Lead Turn Solution Code

This appendix contains a summary of the MATLAB[®] symbolic commands which will output the near-optimal Taylor solution.

Straight-to-Turning Segment.

```
syms t t1 tf tau umax V R g phif Wx Wy

%% Optimal Time-to-Bank Results
l10 = -exp(phif/(2*tau*umax))/(umax*(exp(phif/(tau*
    umax)) - 1)^(1/2));
l20 = -tau/umax;
t1 = tau*log(-(l10*tau)/(l20 - l10*tau));
tf = -tau*log((l10*umax + 1)/(l10*umax - 1));
phit1(t) = t*umax - tau*umax + tau*umax*exp(-t/tau);
phitf(t) = (2*t1 - t + tau - 2*tau*exp(-(t - t1)/tau)
    + tau*exp(-t/tau))*umax;

%% Integrate Phi with two 3rd order Taylor Series
    Expansions
a = phif/4;x=phit1;
psidott1 = g/V*(tan(a) - (tan(a)^2 + 1)*(a - x) + (
    tan(a)^3 + tan(a))*(a - x)^2);
a = phif*3/4;x = phitf;
psidottf = g/V*(tan(a) - (tan(a)^2 + 1)*(a - x) + (
    tan(a)^3 + tan(a))*(a - x)^2);
psit1(t) = int(psidott1 , t , 0 , t)+asin(-Wx/V);
psitf(t) = int(psidottf , t , t1 , t)+psit1(t1);
```

```

%% Integrate Psi with Small Angle Assumptions
xt1(t) = int((V*(psit1)),t,0,t);
yt1(t) = int((V),t,0,t);
xtf(t) = int((V*(psitf)),t,t1,t)+xt1(t1);
ytf(t) = int((V),t,t1,t)+yt1(t1);
%Final state after maneuver
xf = xtf(tf)+Wx*tf;
yf = ytf(tf)+Wy*tf;
psif=psitf(tf);
%% Solve for Optimal Lead Turn (Small Angle
    Assumptions in Cost)
syms y0
xfc = xf+R;
yfc = yf-R*(abs((V*sin(psif)+Wx)/(V*cos(psif)+Wy)))+
    y0;
J = (((xfc-R)^2+(yfc)^2));
dJ =diff(J,y0);
outy1 = solve(dJ==0,y0);

```

Turning-to-Straight Segment.

```

syms t t1 tf tau umax V R g phif s0 Q Wx Wy
%% Optimal Time-to-Bank Results
l10 = -exp(phif/(2*tau*umax))/(umax*(exp(phif/(tau*
    umax)) - 1)^(1/2));
l20 = -tau/umax;
t1 = tau*log(-(l10*tau)/(l20 - l10*tau));

```

```

tf = -tau*log((l10*umax + 1)/(l10*umax - 1));
phit1(t) = -(t*umax - tau*umax + tau*umax*exp(-t/tau)
    )+phif;
phitf(t) = -((2*t1 - t + tau - 2*tau*exp(-(t - t1)/
    tau) + tau*exp(-t/tau))*umax)+phif;
%% Integrate Phi with two 3rd order Taylor Series
    Expansions
a = 3*phif/4;x=phit1;
psidott1 = g/V*(tan(a) - (tan(a)^2 + 1)*(a - x) + (
    tan(a)^3 + tan(a))*(a - x)^2);
a = 1*phif/4;x=phitf;
psidottf = g/V*(tan(a) - (tan(a)^2 + 1)*(a - x) + (
    tan(a)^3 + tan(a))*(a - x)^2);
%% Define Initial Condition based on s0
angle0 = s0/R;
%Small Angle Assumption
x0 = R-R*1;
y0 = R*angle0;
Wcross = Wx-Wy*angle0;%small angle
psi0 = -Wcross/V+angle0;%another small angle

%Integrate Psidot
psit1(t) = int(psidott1 , t , 0 , t)+psi0;
psitf(t) = int(psidottf , t , t1 , t)+psit1(t1);
%Integrate Position
xt1(t) = int((V*(psit1)) , t , 0 , t)+x0;

```

```

yt1(t) = int((V),t,0,t)+y0;
xtf(t) = int((V*(psitf)),t,t1,t)+xt1(t1);
ytf(t) = int((V),t,t1,t)+yt1(t1);
%Final state after maneuver
yf = ytf(tf)+Wy*tf;
xf = xtf(tf)+Wx*tf;
psif=psitf(tf);
%% Solve for Optimal Lead Turn (Small Angle
    Assumptions in Cost)
J = xf^2+(Q*(psif-asin(-Wx/V)))^2;
dJ =diff(J,s0);
outy2 = solve(dJ==0,s0);

```

Appendix B. Proofs

B.1 Bounds on $s_i(t)$

Let $s_i(t)$ be defined by Eq. (389) assuming $0 < V_{min} \leq V_i \leq V_{max}$, $0 < V_{min} \leq V_{n_i} \leq V_{max}$, and $0 \leq W_{cross_i} < V_{min}$.

$$s = \frac{\left(\sqrt{V_i^2 - W_{cross_i}^2} - \sqrt{V_{n_i}^2 - W_{cross_i}^2} \right)}{V_i - V_{n_i}} \quad (389)$$

Next, let V_i and W_{cross_i} be defined by the scalars K_V and K_W as defined by Eq. (390) and Eq. (391). The previous assumptions translate to $0 < K_{V_{min}} \leq K_V \leq K_{V_{max}}$, $0 < K_{V_{min}} \leq 1 \leq K_{V_{max}}$, and $0 \leq K_W < K_{V_{min}}$.

$$V_i = K_V V_{n_i} \quad (390)$$

$$W_{cross_i} = K_W V_{n_i} \quad (391)$$

Substituting these definitions for V_i and W_{cross_i} into Eq. (389) results in Eq. (392).

$$s = \frac{\sqrt{K_V^2 - K_W^2} - \sqrt{1 - K_W^2}}{K_V - 1} \quad (392)$$

Lemma B.1. *Assume $K_V \neq 1$ then $\frac{\partial s}{\partial K_V} \leq 0$*

Proof.

$$\begin{aligned}
\frac{\partial s}{\partial K_V} &= \frac{K_W^2 - K_V + \sqrt{1 - K_W^2} \sqrt{K_V^2 - K_W^2}}{(F_V - 1)^2 \sqrt{K_V^2 - K_W^2}} \leq 0 \\
K_W^2 + \sqrt{1 - K_W^2} \sqrt{K_V^2 - K_W^2} &\leq K_V \\
\sqrt{1 - K_W^2} \sqrt{K_V^2 - K_W^2} &\leq K_V - K_W^2 \\
-K_W^2 K_V^2 + K_V^2 + K_W^4 - K_W^2 &\leq K_V^2 - 2K_V K_W^2 + K_W^4 \\
-K_W^2 K_V^2 - K_W^2 &\leq -2K_V K_W^2 \\
-K_V^2 - 1 &\leq -2K_V \\
-(K_V - 1)^2 &\leq 0
\end{aligned}$$

■

Lemma B.2. *Assume $K_V \neq 1$ then $\frac{\partial s}{\partial K_W} > 0$*

Proof.

$$\begin{aligned}
\frac{\partial s}{\partial K_W} &= \frac{K_W}{(K_V - 1) \sqrt{1 - K_W^2}} - \frac{K_W}{(K_V - 1) \sqrt{K_V^2 - K_W^2}} > 0 \\
&\frac{1}{(K_V - 1) \sqrt{1 - K_W^2}} - \frac{1}{(K_V - 1) \sqrt{K_V^2 - K_W^2}} > 0
\end{aligned}$$

If $K_V > 1$,

$$\begin{aligned}
\frac{1}{\sqrt{1 - K_W^2}} - \frac{1}{\sqrt{K_V^2 - K_W^2}} &> 0 \\
\frac{1}{\sqrt{1 - K_W^2}} &> \frac{1}{\sqrt{K_V^2 - K_W^2}} \\
K_V^2 - K_W^2 &> 1 - K_W^2 \\
K_V^2 &> 1
\end{aligned}$$

which holds with $K_V > 1$.

If $K_V < 1$,

$$\begin{aligned} \frac{1}{\sqrt{1 - K_W^2}} - \frac{1}{\sqrt{K_V^2 - K_W^2}} &< 0 \\ \frac{1}{\sqrt{1 - K_W^2}} &< \frac{1}{\sqrt{K_V^2 - K_W^2}} \\ K_V^2 - K_W^2 &\leq 1 - K_W^2 \\ K_V^2 &< 1 \end{aligned}$$

which holds with $K_V < 1$. ■

Lemma B.3. *The point $K_V = 1$ is a point discontinuity of the function s and the derivative $\frac{\partial s}{\partial K_V}$ for all $0 \leq K_W < K_{V_{min}}$*

Proof. Utilizing the rationalization technique, it can be shown that

$$\begin{aligned} \lim_{K_V \rightarrow 1} s &= \frac{1}{\sqrt{1 - K_W^2}} \\ \lim_{K_V \rightarrow 1} \frac{\partial s}{\partial K_V} &= \frac{K_W^2}{2(1 - K_W^2)^{3/2}} \end{aligned}$$

■

Theorem B.1. *Given the function defined by Eq. (389) whose variables are bounded such that $0 < K_{V_{min}} \leq K_V \leq K_{V_{max}}$, $0 \leq K_W < K_V$, $K_W < 1$, s is maximized when $K_W = \min\{K_{V_{min}}, 1\}$ and $K_V = K_{V_{min}}$*

Proof. By Lemma B.1 and Lemma B.2 the maximum of s will occur at the largest bounded value of K_W and the lowest bounded value of K_V . Although a discontinuity exists at $K_V = 1$, Lemma B.3 shows that it is a point discontinuity where the derivative approaches the same value from both sides of the limit. ■

Bibliography

1. S. Park, “Autonomous Aerobatic Flight by Three-Dimensional Path-Following with Relaxed Roll Constraint,” in *AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, August 2011.
2. MathWorks, “Dryden Wind Turbulence Model.” <https://www.mathworks.com/help/aeroblks/drydenwindturbulencemodelcontinuous.html>, 2020. Accessed 15 July 2020.
3. S. Park, J. Deyst, and J. P. How, “Performance and lyapunov stability of a nonlinear path-following guidance method,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.
4. Merriam-Webster Inc., “Automation.” <https://www.merriam-webster.com/dictionary/automation>. Accessed 16 December 2019.
5. W. Scheck, “Lawrence Sperry: Autopilot Inventor and Aviation Innovator.” <https://www.historynet.com/lawrence-sperry-autopilot-inventor-and-aviation-innovator.htm>, November 2004. Accessed 6 December 2019.
6. B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Hoboken, NJ, USA: John Wiley & Sons, Inc, November 2015.
7. National Aeronautics and Space Administration, “NASA Armstrong Fact Sheets: F-8 Digital Fly-By-Wire Aircraft — NASA.” <https://www.nasa.gov/centers/armstrong/news/FactSheets/FS-024-DFRC.html>, 2014. Accessed 6 December 2019.
8. Federal Aviation Administration, “Advanced Avionics Handbook (FAA-H-8083-6),” 2009.
9. United States Department of Defense, “Unmanned Systems Integrated Roadmap: 2017-2042,” 2017.
10. LeMay Center for Doctrine, “AFDP 1, The Air Force,” 2021.
11. Federal Aviation Administration, “PBN NAS Navigation Strategy,” 2016.
12. A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*. Chichester, UK: John Wiley & Sons, Ltd, November 2010.

13. L. E. Dubins, "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, pp. 497–516, July 1957.
14. A. M. Shkel and V. Lumelsky, "Classification of the Dubins set," *Robotics and Autonomous Systems*, vol. 34, pp. 179–202, 2001.
15. S. G. Manyam, D. Casbeer, A. L. Von Moll, and Z. Fuchs, "Shortest Dubins path to a circle," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, January 2019.
16. T. J. Arnett, K. Cohen, B. Cook, and D. Casbeer, "A Supervisory Neural Network and Fuzzy Inference System Approach to an Unmanned Aerial Vehicle Tail-Chase Scenario," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, January 2019.
17. S. S. Stephens, R. V. Grandhi, and D. L. Kunz, "Modified Nonlinear Traveling Salesman Problem with Delivery Time Windows and Item Constraints," in *AIAA Scitech 2020 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, January 2020.
18. T. McGee, S. Spry, and K. Hedrick, "Optimal path planning in a constant wind with a bounded turning rate," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, vol. 5, (Reston, Virginia), pp. 3456–3466, American Institute of Aeronautics and Astronautics, August 2005.
19. T. G. McGee and J. K. Hedrick, "Optimal Path Planning with a Kinematic Airplane Model," *Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 629–633, March 2007.
20. L. Techy and C. A. Woolsey, "Minimum-Time Path Planning for Unmanned Aerial Vehicles in Steady Uniform Winds," *Journal of Guidance, Control, and Dynamics*, vol. 32, pp. 1736–1746, November 2009.
21. B. Ayhan and C. Kwan, "Time-Constrained Extremal Trajectory Design for Fixed-Wing Unmanned Aerial Vehicles in Steady Wind," *Journal of Guidance, Control, and Dynamics*, vol. 41, pp. 1569–1576, July 2018.
22. M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. A. White, "3D dubins sets based coordinated path planning for swarm of UAVs," in *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2006*, vol. 3, (Reston, Virginia), pp. 1418–1437, American Institute of Aeronautics and Astronautics, August 2006.
23. R. T. Farouki and T. Sakkalis, "Pythagorean hodographs," *IBM Journal of Research and Development*, vol. 34, no. 5, pp. 736–752, 1990.

24. R. T. Farouki, “The Bernstein polynomial basis: A centennial retrospective,” *Computer Aided Geometric Design*, vol. 29, pp. 379–419, August 2012.
25. R. Choe, V. Cichella, E. Xargay, N. Hovakimyan, A. C. Trujillo, and I. Kaminer, “A Trajectory-Generation Framework for Time-Critical Cooperative Missions,” in *AIAA Infotech@Aerospace (I@A) Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, August 2013.
26. R. Choe, J. Puig-Navarro, V. Cichella, E. Xargay, and N. Hovakimyan, “Cooperative Trajectory Generation Using Pythagorean Hodograph Bézier Curves,” *Journal of Guidance, Control, and Dynamics*, vol. 39, pp. 1744–1763, August 2016.
27. R. Choe, *Distributed Cooperative Trajectory Generation for Multiple Autonomous Vehicles Using Pythagorean Hodograph Bezier Curves*. Dissertation, University of Illinois at Urbana-Champaign, 2017.
28. M. Fliess, J. Lévine, P. Martin, and P. Rouchon, “Flatness and defect of non-linear systems: introductory theory and examples,” *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
29. M. J. Van Nieuwstadt and R. M. Murray, “Real-time trajectory generation for differentially flat systems,” *International Journal of Robust and Nonlinear Control*, vol. 8, pp. 995–1020, September 1998.
30. M. Van Nieuwstadt, M. Rathinam, and R. M. Murray, “Differential flatness and absolute equivalence,” in *Proceedings of the IEEE Conference on Decision and Control*, vol. 1, pp. 326–332, IEEE, 1994.
31. D. Kirk, *Optimal Control Theory*. Mineola, New York: Dover Publications, Inc., 1998.
32. J. D. Boissonnat, A. Cerezo, and J. Leblond, “Shortest paths of bounded curvature in the plane,” in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2315–2320, IEEE Comput. Soc. Press, 1992.
33. S. G. Park and J.-P. Clarke, “Optimal Control Based Vertical Trajectory Determination for Continuous Descent Arrival Procedures,” *Journal of Aircraft*, vol. 52, pp. 1469–1480, September 2015.
34. A. V. Rao, D. A. Benson, C. Darby, M. A. Patterson, C. Francolin, I. Sanders, and G. T. Huntington, “GPOPS: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method,” *ACM Transactions on Mathematical Software*, vol. 38, pp. 1–2, November 2011.
35. S. G. Park, *Optimal Control Based Method for Design and Analysis of Continuous Descent Arrivals*. Dissertation, Georgia Institute of Technology, 2014.

36. S. S. Stephens, J. Hess, and D. L. Kunz, "Optimal Control Approach to Terrain Following Trajectory Generation," in *AIAA Scitech 2020 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, January 2020.
37. I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre, "Trajectory Tracking for Autonomous Vehicles: An Integrated Approach to Guidance and Control," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1.
38. P. Bares, S. Lazarus, A. Tsourdos, and A. Savvaris, "Adaptive guidance for UAV based on Dubins path," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, August 2013.
39. A. Wolek and C. Woolsey, "Feasible Dubins Paths in Presence of Unknown, Unsteady Velocity Disturbances," *Journal of Guidance, Control, and Dynamics*, vol. 38, pp. 782–787, April 2015.
40. D. R. Nelson, T. W. McLain, and R. W. Beard, "Experiments in Cooperative Timing for Miniature Air Vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 956–967, August 2007.
41. R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 911–922, December 2002.
42. P. Chandler, S. Rasmussen, and M. Pachter, "UAV cooperative path planning," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, August 2000.
43. D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, pp. 519–529, June 2007.
44. A. Saleem and A. Ratnoo, "MAV waypoint guidance with arrival angle and time scheduling constraints," in *2016 AIAA Guidance, Navigation, and Control Conference*, 2016.
45. J. Bélanger, A. Desbiens, and E. Gagnon, "UAV guidance with respect of arrival specifications," *2008 Mediterranean Conference on Control and Automation - Conference Proceedings, MED'08*, no. 4, pp. 689–694, 2008.
46. G. A. Harrison, "Hybrid guidance law for approach angle and time-of-arrival control," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1104–1114, 2012.
47. I. E. Weintraub, R. G. Cobb, W. Baker, and M. Pachter, "Direct Methods Comparison for the Active Target Defense Scenario," in *AIAA Scitech 2020 Forum*,

(Reston, Virginia), American Institute of Aeronautics and Astronautics, January 2020.

48. C. Canuto and A. Quarteroni, "Spectral Methods," in *Encyclopedia of Computational Mechanics*, Chichester, UK: John Wiley & Sons, Ltd, November 2004.
49. A. Wachter and L.T. Biegler, "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
50. P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, vol. 47, pp. 99–131, January 2005.
51. K. Chu, "Decentralized Control of High-Speed Vehicular Strings," *Transportation Science*, vol. 8, pp. 361–384, November 1974.
52. D. Swaroop and J. Hedrick, "String stability of interconnected systems," in *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, pp. 1806–1810, American Automatic Control Council, 1995.
53. D. Swaroop and J. Hedrick, "String stability of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 41, pp. 349–357, March 1996.
54. D. Swaroop and K. Rajagopal, "A review of constant time headway policy for automatic vehicle following," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pp. 65–69, IEEE, 2001.
55. B. Besselink and K. H. Johansson, "String Stability and a Delay-Based Spacing Policy for Vehicle Platoons Subject to Disturbances," *IEEE Transactions on Automatic Control*, vol. 62, pp. 4376–4391, September 2017.
56. Jiann-Shiou Yang, "A simulation study for the control of a platoon of vehicles," in *Proceedings of 1994 American Control Conference - ACC '94*, vol. 1, pp. 423–424, IEEE, 1983.
57. S. Sheikholeslam and C. A. Desoer, "Longitudinal Control of a Platoon of Vehicles," in *1990 American Control Conference*, pp. 291–296, IEEE, may 1990.
58. S. Sheikholeslam and C. A. Desoer, "A system level study of the longitudinal control of a platoon of vehicles," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 114, no. 2, pp. 286–292, 1992.
59. E. Shaw and J. K. Hedrick, "Controller design for string stable heterogeneous vehicle strings," in *2007 46th IEEE Conference on Decision and Control*, pp. 2868–2875, IEEE, 2007.

60. L. Weitz and J. Hurtado, "A Time-to-Go Control Law for Spacing Vehicles to a Point," in *AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, August 2011.
61. Federal Aviation Administration, "ADS-B – Interval Management (IM) Applications." <https://www.faa.gov/nextgen/programs/adsb/pilot/ima/>, 2021. Accessed 23 July 2021.
62. D. Ivanescu, C. Shaw, E. Hoffman, and K. Zeghal, "Towards Performance Requirements for Airborne Spacing- A Sensitivity Analysis of Spacing Accuracy," in *6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, (Reston, Virginia), pp. 25–27, American Institute of Aeronautics and Astronautics, September 2006.
63. K. Zeghal, E. Hoffman, D. Ivanescu, and C. Shaw, "Analysis of constant time delay airborne spacing between aircraft of mixed types in varying wind conditions," in *5th USA/Europe Air Traffic Management Seminar*, (Budapest, Hungary), 2003.
64. L. A. Weitz, "Investigating string stability of a time-history control law for Interval Management," *Transportation Research Part C: Emerging Technologies*, vol. 33, pp. 257–271, August 2013.
65. L. A. Weitz and K. A. Swieringa, "Comparing Interval Management Control Laws for Steady-State Errors and String Stability," in *2018 AIAA Guidance, Navigation, and Control Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, January 2018.
66. M. Chen, Q. Hu, J. F. Fisac, K. Akametalu, C. Mackin, and C. J. Tomlin, "Reachability-based safety and goal satisfaction of unmanned aerial platoons on air highways," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 1360–1373, January 2017.
67. K. A. Swieringa, "The String Stability of a Trajectory-Based Interval Management Algorithm in the Midterm Airspace," in *15th AIAA Aviation Technology, Integration, and Operations Conference*, no. June, (Reston, Virginia), American Institute of Aeronautics and Astronautics, June 2015.
68. K. A. Swieringa, S. R. Wilson, B. T. Baxley, R. D. Roper, T. S. Abbott, I. Levitt, and J. Scharl, "Flight test evaluation of the ATD-1 interval management application," in *17th AIAA Aviation Technology, Integration, and Operations Conference, 2017*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, June 2017.
69. D. E. Bossert, S. L. Morris, W. F. Hallgren, and T. R. Yechout, *Introduction to Aircraft Flight Mechanics*. Reston, VA: American Institute of Aeronautics and Astronautics, January 2003.

70. United States Department of Defense, “MIL-F-8785C: Flying Qualities of Piloted Aircraft,” 1980.
71. United States Department of Defense, “MIL-HDBK-1797: Flying Qualities of Piloted Aircraft,” 1997.
72. J. B. Burl, *Linear optimal control : H2 and H[infinity] methods*. Addison Wesley Longman, 1999.
73. National Aeronautics and Space Administration, “Relative Velocity - Aircraft Reference.” <https://www.grc.nasa.gov/www/k-12/airplane/move2.html>. Accessed 27 May 2021.
74. S. S. Stephens, D. Casbeer, D. L. Kunz, W. Baker, and J. A. Hess, “Controlling Arrival Time and Airspeed with Constant Bank Angle Turns,” in *AIAA Scitech 2021 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, January 2021.
75. J. S. Arora, *Introduction to Optimum Design*. Elsevier, 2004.
76. D. Nelson, D. Barber, T. McLain, and R. Beard, “Vector Field Path Following for Miniature Air Vehicles,” *IEEE Transactions on Robotics*, vol. 23, pp. 519–529, June 2007.
77. E. D. B. Medagoda and P. W. Gibbens, “Synthetic-Waypoint Guidance Algorithm for Following a Desired Flight Trajectory,” *Journal of Guidance, Control, and Dynamics*, vol. 33, pp. 601–606, March 2010.
78. P. B. Sujit, S. Saripalli, and J. B. Sousa, “Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles,” *IEEE Control Systems*, vol. 34, pp. 42–59, February 2014.
79. R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton: Princeton University Press, December 2012.
80. J. Shen, T. Tang, and L.-L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, vol. 41 of *Springer Series in Computational Mathematics*. Berlin, Heidelberg: Springer, 2011.
81. A. Jaklič, A. Leonardis, and F. Solina, *Segmentation and Recovery of Superquadrics*, vol. 20. 2000.
82. H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 3rd ed., 2002.
83. S. S. Stephens, D. L. Kunz, W. P. Baker, J. A. Hess, and D. W. Casbeer, “A Control Algorithm Framework for Time-of-Arrival and Arrival Airspeed Control,” in *Proceedings of the 2020 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, October 2020.

84. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, January 1994.
85. MathWork, “Choose a Solver - MATLAB & Simulink.” <https://www.mathworks.com/help/simulink/ug/choose-a-solver.html>, 2021. Accessed 12 July 2021.
86. J. Dormand and P. Prince, “A family of embedded Runge-Kutta formulae,” *Journal of Computational and Applied Mathematics*, vol. 6, pp. 19–26, March 1980.
87. MathWorks, “Check and Improve Simulation Accuracy - MATLAB & Simulink.” <https://www.mathworks.com/help/simulink/ug/improving-simulation-accuracy.html>, 2021. Accessed 12 July 2021.
88. P. Prince and J. Dormand, “High order embedded Runge-Kutta formulae,” *Journal of Computational and Applied Mathematics*, vol. 7, pp. 67–75, March 1981.
89. Calspan, “Operation Of Variable Stability In-Flight Simulators.” <https://www.calspan.com/services/aircraft-operation/flight-testing/inflight-simulation/>, 2020. Accessed 1 January 2021.

Vita

Major Shawn Stephens is a PhD student studying optimal control in the Department of Aeronautics and Astronautics at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. Captain Stephens earned his Bachelor of Science in Aerospace Engineering from Embry-Riddle Aeronautical University, Prescott, Arizona. Captain Stephens subsequently earned two Master of Science degrees. The first in Engineering from California Polytechnic State University, Pomona, Calif. The second is in Flight Test Engineering from the United States Air Force Test Pilot School.

Major Stephens was first assigned to the 773rd Test Squadron, Edwards AFB, Calif. as a performance and flying qualities engineer supporting the developmental testing of the F-35A. He was then assigned to the 445th Flight Test Squadron, Edwards AFB, Calif. as the lead operations engineer where he planned and executed test support missions for the F-35A, Harpoon missile, P-8, and the Navy's AEGIS Weapon System. Major Stephens was then selected to attend the United States Air Force Test Pilot School, Edwards AFB, Calif as a Flight Test Engineer. Upon completion of Test Pilot School, Major Stephens was assigned to the 780th Test Squadron, Eglin Air Force Base, Fla where he was responsible for the test and evaluation of the Air Force's newest air-to-air and air-to-ground weapons.

Major Stephens was then accepted to the in-residence Ph.D. program at AFIT studying Autonomous Aircraft and Optimal Control. Upon earning his Ph.D. from AFIT, Maj Stephens will be assigned as an instructor at the United States Air Force Test Pilot School, Edwards Air Force Base, Calif.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) 09-16-2021		2. REPORT TYPE Doctoral Dissertation		3. DATES COVERED (From — To) Aug 2018 — Sept 2021	
4. TITLE AND SUBTITLE A Real-Time Algorithm to Achieve Precise Coordinated Arrival Times in a Time-Variant Environment				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Maj. Shawn S. Stephens				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENY-DS-21-S-109	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Control Science Center 2210 8th St WPAFB OH 45433 DSN 785-8495, COMM 937-255-8495 Email: david.casbeer@us.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RQQA	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The persistent monitoring problem is a type of coordinated arrival problem where a stationary or moving ground target must be constantly observed by a group of aircraft. This research simplifies the problem to overfly the target at specific time intervals relative to the other aircraft in the group while increasing the difficulty of the problem by considering aircraft traveling at vastly different airspeeds. The result of this research is a real time guidance algorithm which accurately guides a group of airspeed heterogeneous aircraft to achieve a desired relative arrival time, predictable flight path, and accounts for varying wind conditions, keep-out zones, and non-cooperative targets. A wide variety of scenarios were evaluated with aircraft ranging in speed from 174 knots to 511 knots. However, a notable scenario utilizing a nonlinear fighter aircraft model demonstrated the algorithm's ability to guide a group of four aircraft to achieve the desired time spacing to within 0.1 seconds in a scenario with 160 knot winds, a target moving at 102 knots, keep-out zones, and moderate wind turbulence.					
15. SUBJECT TERMS Unmanned Aircraft, Coordinated Arrival Time, Autonomy, Optimal Control					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. D. R. Kunz, AFIT/ENY
U	U	U	U	306	19b. TELEPHONE NUMBER (include area code) (937) 255-3636, x4548; donald.kunz@afit.edu