UAV Control and Guidance for Autonomous Cooperative Tracking of a Moving Target

Richard Wise

A research proposal submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2006

Program Authorized to Offer Degree: Aeronautics & Astronautics

University of Washington

Abstract

UAV Control and Guidance for Autonomous Cooperative Tracking of a Moving Target

Richard Wise

Chair of the Supervisory Committee:
Assistant Professor Rolf Rysdyk
Aeronautics and Astronautics

This research compares various different methodologies for tracking a moving target with multiple Unmanned Aerial Vehicles (UAVs). Relative position coordination of UAVs is enforced. The comparison considers minimization of heuristics and robustness of performance and stability when the UAVs are exposed to wind and target motion.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

Nomenclature

a	Helmsman sensitivity parameter
d	Distance
F_b	Body-fixed frame
g	Gravity constant
r	Radius
s	Arclength position along desired path
$\mathbf{V}$	Velocity
V_a	Airspeed
V_g	Inertial speed (ground speed)
V_w	Windspeed (inertial)
x_b, y_b, z_b	Body-fixed axes system
x_N,y_E,z_D	Navigation axes system
x, y, z	Position coordinates
y_s	Cross-track error
$\kappa(s)$	Curvature of desired path at position s
ϕ	Bank angle
ρ	Radius of curvature
χ	Course
χ_w	Wind direction ('from' convention)
ψ	Heading
ψ_w	Wind vector orientation ($\psi_w = \chi_w + \pi$)
Ψ	Bearing angle of aircraft from the target ('Clock angle' relative to target)

Subscripts and Superscripts

b	Body-fixed reference frame
c	Command
e	Earth reference frame
hat $(\hat{\cdot})$	Estimate
icpt	Intercept
s	Serret-Frenet reference frame
t	Target
w	Wind
τ	Tangential to Orbital Path
tilde $(\tilde{\cdot})$	Relative parameter(Vehicle - Target)
minus $(\cdot)^-$	A value delayed by one time step

Chapter 1

INTRODUCTION

This chapter details the cooperative tracking problem and discusses the background motivation that has led to many alternate solution methods.

1.1 Background

There is considerable and mounting interest in using unmanned vehicles to perform a multitude of tasks [1], [13], [21]. UAVs already provide a clear opportunity to reduce the risk of life threatening missions that might otherwise be performed by human-piloted craft. The well-known unmanned drone, the Predator, is remotely piloted over the dangerous skies of Iraq. In recent years, autonomy has mad UAVs significantly more useful. Autonomy is the ability to perform a task (mission) without being directly or remotely controlled by a human. Flights such as the Aerosonde autonomous flight over the Atlantic Ocean have amply demonstrated UAV autonomous feasibility[1].

Achieving complete single-vehicle autonomy requires spanning the three levels of the autonomous hierarchy: strategic, tactical, dynamic. The levels are made distinct by time scale (dynamic rate at which decisions are to be made) and thus by the control problem that is to be solved. The highest or *strategic* level operates on the slowest time scale. Missions such as path planning, task allocation, and searching fall in this category. The time scale is governed by the size of the mission (scope of the problem) relative to the average speed of the vehicles. This is equivalent to the navigation control problem. The middle or *tactical* level operates on a faster time scale and includes such missions as target observation. Here the time scale is associated with the maneuverability of the vehicles. In control terms, this is

[1] "First UAV Across the Atlantic" http://www.aerosonde.com/drawarticle/4, May 2006.

the guidance problem. The lowest or dynamic level operates on the fastest time scale, which is now a function of control actuator dynamics specific to a vehicle. This is the "inner-loop" control problem that translates guidance orientation commands to actuator signals.

Intra-UAV autonomous cooperation is a significant UAV technology that has not been amply demonstrated. Autonomous cooperation implies that the autonomous controls of one vehicle are dependent on the autonomous controls of another vehicle, and vice versa. Nearly all missions can benefit directly by using more than one UAV. Multiple UAVs bring extra sensors, extra communication devices, differing vantage points, and redundancy. If combined and utilized correctly, this can allow for greater efficiency and greater accuracy of determined dynamic parameters than by using a single UAV.

There is an important distinction between *cooperation* and *coordination*. Coordination is used to imply any effort to optimally utilize assets to perform a mission. Cooperation is active decision making among multiple vehicles to accomplish tasks that either cannot be performed by one vehicle alone, or could be performed more efficiently with multiple vehicles. Therefore, coordination is a result or subset of cooperation.

Tracking is a fundamental UAV mission that elucidates the necessity for coordination and the benefits of cooperation. Thus, there are many ongoing efforts to determine optimal methods to track a moving target with multiple autonomous UAVs [8], [10], [16], [18].

In this research, we consider active tracking, or maintaining continuous knowledge of the parameters of the target to be tracked. By coordinating UAV positions, the combined sensor information produces more accurate target position determination than a single UAV can provide. A single sensor provides an estimate of the target position as being within an uncertainty ellipsoid, which is nonsymmetric and wider in the direction perpendicular to the line of site. By using multiple UAVs, the uncertainty is bounded to the intersection of the corresponding multiple uncertainty ellipsoids. See Figure [1.1]. For example, when there are two UAVs, the bound is minimized by maintaining a 90 degree clock angle separation. Through coordination, this lower uncertainty bound is advantageous even when considering the noise of multiple sensor signals [3].

Through cooperation, we will provide the answers to decisions such as:

- Is the target position information good enough, or is another UAV needed to improve it?

- Is it worth the effort of moving another UAV from another task?

- With a communication or sensor failure causing a degradation in target position information, can the target following task be performed reliably?

- If the target is hostile, is is worth the risk to continue tracking?

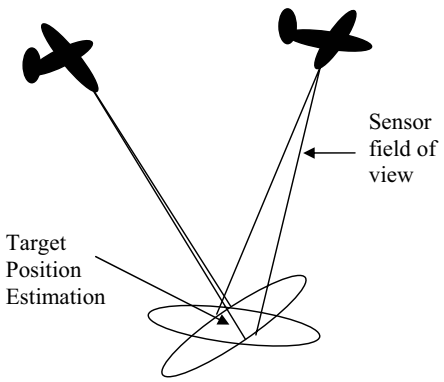


Figure 1.1: Reducing Target Estimation Error By Using Multiple Vehicles

1.2 Problem Statement

This research will investigate multiple, minimal heuristic methods of coordinated tracking of a moving target in the presence of wind. The first goal is to provide a rigorous comparison of the performance, robustness, and stability of each method. The second goal is to determine the impact of each method's level of cooperation on multi-vehicle autonomy.

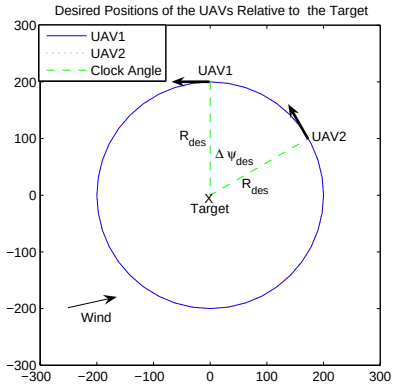


Figure 1.2: The Desired Relative Picture

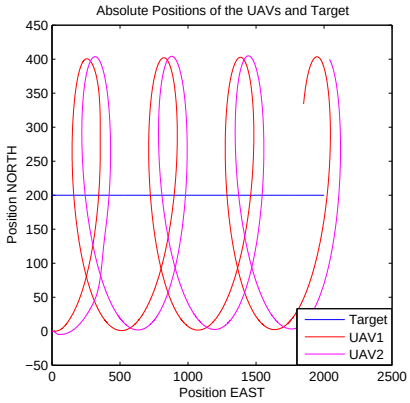


Figure 1.3: The Desired Absolute Picture

1.2.1 The Problem of Coordinated Tracking of a Moving Target in Wind

Tracking coordination implies that, relative to the target, the UAVs maintain a specified, constant angular separation (constant relative clock angle). The relative picture is represented in Figure [1.2]. Figure [1.3] shows the corresponding absolute motion of the UAVs when tracking a target that moves Easterly at constant speed. From Figure [1.2], we see that clock angle is a function of time-varying, relative positions:

$$\Psi_{UAV}(t) = \tan^{-1}\left(\frac{y(t) - y_t(t)}{x(t) - x_t(t)}\right) = \tan^{-1}\frac{\tilde{y}(t)}{\tilde{x}(t)} \tag{1.1}$$

The clock angle rate is therefore related to the relative, tangential velocity:

$$\dot{\Psi}(t) = \frac{|\tilde{\mathbf{V}}_\tau|(t)}{\tilde{r}(t)} \tag{1.2}$$

where the relative velocity depends on the components of the UAV, target, and wind velocities in the orbit tangential direction:

$$\dot{\Psi}(t) = \frac{(V_\tau(t) - V_{t\tau}(t) - V_{w\tau}(t))}{\tilde{r}(t)} \tag{1.3}$$

If the orbit radius is maintained constant, then the clock angle rate is also related to the absolute velocities by:

$$\dot{\Psi}^2(t) \quad = \quad \frac{(V_a(t)\cos(\psi(t)) - V_t(t)\cos(\psi_t(t)) - V_w(t)\cos(\psi_w(t)))^2}{\tilde{r}^2}$$

$$
\begin{aligned}
+ \quad & \frac{\left(V_a(t)\sin(\psi(t)) - V_t(t)\sin(\psi_t(t)) - V_w(t)\sin(\psi_w(t))\right)^2}{\tilde{r}^2} \\
= \quad & \frac{V_g(t)^2 + V_t^2 - 2V_g(t)V_t(t)\cos(\tilde{\psi}(t) + \psi_t(t))}{\tilde{r}^2} \\
= \quad & \frac{V_g(t)^2 + V_t^2 + 2V_g(t)V_t(t)\sin(\Psi(t) - \psi_t(t))}{\tilde{r}^2}
\end{aligned} \tag{1.4}
$$

where $\tilde{\psi}(t)$ is the relative velocity heading. Note that maintaining radius implies that UAV course is always such that the relative velocity is tangential so that $\hat{\chi}(t) = \pi/2 + \Psi(t)$. If radius is not maintained constant, we can take the tangential component of the relative velocity:

$$
\dot{\Psi}^2(t) = \frac{\left(V_g(t)^2 + V_t^2 + 2V_g(t)V_t(t)\sin(\Psi(t) - \psi_t(t))\right)_\tau}{\tilde{r}(t)^2} \tag{1.5}
$$

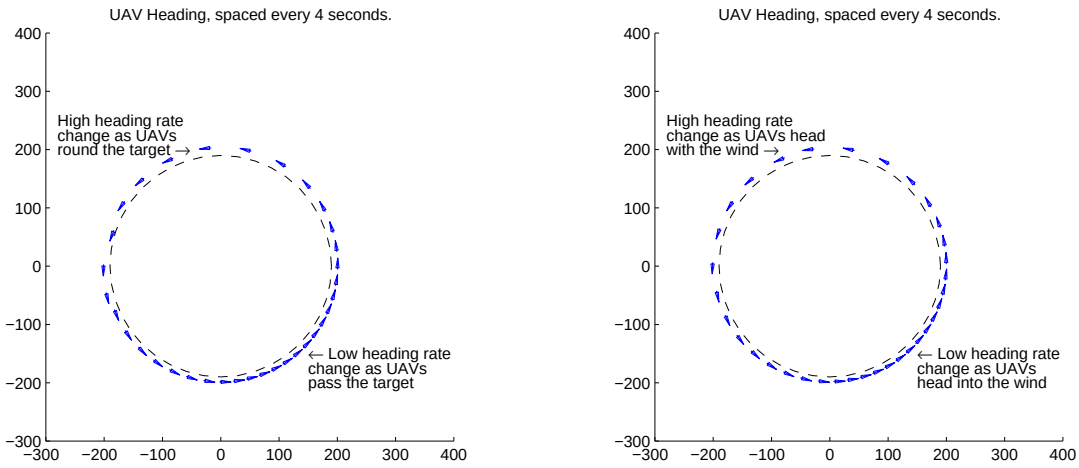
Where we see that clock angle rate depends on UAV velocity, distance from the target, and current clock angle as well as target velocity and wind velocity.

Tracking a Moving Target Without Wind

First consider Figure [1.4(a)] which shows the local velocity vector of a UAV relative to a moving target. The vectors are evenly spaced in time. The UAV rotates counter-clockwise about the target. As the UAV can only pass the target at a relative speed lessened by the target forward motion, the target bearing rate, as 'seen' by the UAV, is also lessened and so is the UAV clock angle rate about the target ($\dot{\Psi} = (|\mathbf{V} - \mathbf{V}_t|)/r$). This is seen in Figure [1.4(a)]: the vectors seem bunched up. When the UAV has rotated past the front of the target, its relative speed increases to include the target motion in the opposite direction. This higher speed results in high target bearing rate and thus heightened clock angle rate. This is visible in Figure [1.4(a)] as the vectors spreading apart.

Tracking a Stationary Target With Wind

Now consider the problem of target tracking in the presence of a wind field, as shown in Figure [1.4(b)]. Here a UAV orbits a stationary target, but with a constant wind from right to left. As in Figure [1.4(a)], the velocity vectors are evenly spaced in time, but are bunched up when the vehicle travels against the wind and spread out when the vehicle is with the wind. Unlike the target, however, the wind affects the vehicle dynamics. When the UAV



(a) Effect of Target Motion On Course Rate of Change

(b) Effect of Wind On Course Rate of Change

Figure 1.4: Relative Motion Effects on Course Rate of Change

flies into the wind, the aircraft is more responsive to bank angle increasing its ability to change course. However, the UAV's lowered ground speed has decreased its clock angle rate around the target. Conversely, when the UAV flies with the wind, its ground speed increases reducing its ability to change course but increasing its clock angle rate around the target.

Tracking a Moving Target With Wind

Combining the effects of a moving target with wind may render a real UAV, with actuator limits and response dynamics, incapable of maintaining or even converging to the desired circular orbit in Figure [1.2].

For example, assume that the wind is in a direction opposite the motion of the target. As the UAV passes the target, its ground speed will be reduced by the wind velocity resulting in a lesser clock angle rate, but an increased bank angle responsiveness. Assuming that the ground speed of the UAV remains higher than the target, the UAV should be able to

continue its orbital path. As the UAV rounds the target, its ground speed will increase by the wind speed resulting in a higher clock angle rate necessary to continue to orbit. The bank angle of the UAV must increase to accommodate, possibly to saturation, but will have reduced responsiveness. This may cause the UAV to lag behind the target, diverging from the orbit, until the UAV turns back into the wind to catch back up with the target. In other words, the control authority of the UAV is increased when we don't need it and degraded just when we do. Of course, if the wind were to go in the opposite direction, everything would be easier, but we can't expect that.

Coordinated Tracking of a Moving Target With Wind

For coordinated tracking, the clock angle rate of change of the two UAVs must be matched to maintain a constant relative clock angle. In order to optimize the benefit of coordinated tracking, a clock angle separation between the UAVs is commanded. The control must maneuver the UAVs to achieve this separation.

$$\Delta(\Psi_{des}) = \Delta\Psi_c \rightarrow \Psi_2 = \Psi_1 - \Psi_c$$

Once steady on the separation command, the clock angle rate for two UAVs must be matched. Starting with Equation[1.5], we find:

$$\begin{aligned} & \frac{1}{\tilde{r}_1^2(t)} \left(V_{g1}(t)^2 + V_t^2 + 2V_{g1}(t)V_t(t)\sin(\Psi_1(t) - \psi_t(t)) \right)_\tau \\ = \quad & \frac{1}{\tilde{r}_2^2(t)} \left(V_{g2}(t)^2 + V_t^2 + 2V_{g2}(t)V_t(t)\sin(\Psi_1(t) - \Psi_c - \psi_t(t)) \right)_\tau \end{aligned} \tag{1.6}$$

To affect clock angle rate, the UAV can either change speed or heading. Changing speed directly affects clock angle rate, but also changes the relative speed and thus the relative distance and thus the clock angle rate. Changing heading affects clock angle rate by changing the tangential component of the relative speed but also affects the future relative position affecting the clock angle rate. Target motion and wind velocity affect clock angle similarly. Therefore, the UAV control system must compensate for these effects, within actuator limits and behavioral (dynamic response) bounds, while simultaneously tracking the target.

1.3 Kinematic Coordinated Turn UAV Model

1.3.1 Kinematics

As with most conventional aircraft, the UAV control problem would normally involve the following degrees of freedom : airspeed, aerodynamic side-slip angle, turn rate, and flight path angle. The available aerodynamic control surfaces for rotation about the three body axes are the ailerons, elevator, and rudder. The propulsion (throttle) remains and will be assumed to work in conjunction with the elevator to control altitude and airspeed. As this control problem addresses guidance at constant altitude, it will be assumed that airspeed and altitude can be held (or controlled) independent of lateral-directional commands.

Aileron and rudder signals are normally combined to zero sideslip providing a "truly banked turn"or a 'turn coordination'. 'Turn coordination' relates the bank angle and turn rate, kinematically, as follows:

$$\tan\phi = \frac{V_g}{g}\dot{\psi} \tag{1.7}$$

Thus, *turn rate is the single degree of freedom available for the tracking coordination problem.* Recalling Equation [1.2], turn rate does not explicitly affect clock angle rate. However, by turning, heading angle changes resulting in a change in clock angle rate, as in Section [1.2.1].

We will simply relate navigation over the ground via a flat earth model. As we will allow for wind, we distinguish between heading (horizontal orientation of the body frame) and course (horizontal orientation of the UAV over ground, i.e. in the inertial frame).

1.3.2 Dynamics and Limits

To account for aircraft dynamics, a command in airspeed will be assumed to have a first order response by the UAV as follows:

$$V_a = \frac{1}{s+1}V_c \tag{1.8}$$

A command saturation of $\pm 20\%$ of nominal speed is used.

Bank angle response shows a first order behavior:

$$\phi = \frac{1}{0.3s + 1}\phi_c \tag{1.9}$$

A bank angle rate limit of $\pm 45 deg/s$ is used as well as a command saturation of $\pm 45 deg$.

1.4 Overview of This Paper

This chapter has outlined the cooperative target tracking problem.

Chapter 2 outlines the proposed research, including that which has been completed and that which is forthcoming.

Chapter 3 details the first three methods for solving the tracking problem, generally employing feedback.

Chapter 4 details the last two methods for solving the tracking problem, generally employing predictive methods.

Chapter 5 summarizes the proposed research.

Chapter 2

PROPOSED RESEARCH

To address the problem stated in Section 1.2, we propose to implement five different methods for solving the cooperative tracking problem for two UAVs. The methods, detailed in the following chapters, are summarized as follows:

Reactive Methods

1: Helmsman Behavior Based Guidance. Proposed by Rysdyk [18].

Tracking: Simultaneously reduces heading and track error via "Good Helmsma" behavior.

Angular Coordination: Adjusts stand-off distance and/or speed thereby affecting clock angle rate.

2: Lyapunov Vector Field Guidance. Adapted from Lawrence and Frew [8].

Tracking: Adjusts heading vector to converge with a velocity limit cycle centered on the target.

Angular Coordination: Adjusts heading thereby affecting clock angle rate.

3: Controlled Collective Motion. Adapted from Klein and Morgansen [10].

Tracking: Matches UAV flock centroid with target position.

Angular Coordination: UAV's 'repel' one another so stay equally spaced about orbit.

Predictive Methods

4: Evolutionary Based Trajectory Paths and Path Following. Adapted from Pongpunwattana [16] work on ECoPS (see Chapter 4).

Tracking and Angular Coordination: Determines the best feasible path with an evolutionary algorithm to keep clock separation. Optimization flexible enough to concurrently account for other considerations, such as radar exposure minimization.

5: Model Predictive Control Solving Convex Optimization

Tracking and Angular Coordination: Predicts future target and other UAV positions. Adjusts heading to minimize future divergences in clock angle separation.

It is our intention that by implementing these considerably different methods, not only will a robust solution to the tracking problem be determined, but significant insight into the utility of each of these methods for autonomous, multi-vehicle control will also be revealed.

To complete this research, the following items will need to be addressed:

- Develop a consistent testing platform, non-real time
- Adapt and/or formulate methods 2 - 5
- Implement each method within the platform
- Investigate the performance of each method
- Investigate the robustness of each method to signal variations
- Develop a consistent testing platform that can test real-time

  Determine the effect of signal drop-out (communication loss) on coordination

- Adapt each method to actively use wind information (sensed) to account for wind effects on orbiting

Via this process, the following items will also be addressed:

- Merge tactical problem of tracking with strategic planning problem currently implemented in ECoPS.
- Use ECoPS to consider radar exposure minimization while cooperatively tracking a hostile target.

2.1 Research Completed to Date

The following items of the proposed research have been completed:

- A non-real time MATLAB© simulation has been developed for methods 1 - 3, 5. See Appendix A for details.
- The MATLAB© simulation has been modified for real-time using over-the-network communication. See Appendix A for details.
- Methods 1-3,5 have been formulated and adapted for simulation, in the non-real time, with results available in the following chapters. Active use of sensed wind data is not implemented yet.
- Methods 1-2 have been adapted to run on the real time simulation. Method 3 has been determined unsuitable for further consideration (See Chapter 3).
- A basic, point-mass model simulation interface has been implemented for ECoPS in C++, with post-visualization of results done in MATLAB©.
- ECoPS has been formulated for one vehicle tracking and has been simulated.

2.2 Research to be Completed

To complete the proposed research, the following outstanding issues will be addressed :

- Adapt each method to actively use wind information (sensed) to account for wind effects on orbiting

- Obtain real-time simulation results for Methods 1,2,4,5 to include effect of communication dropout on cooperation.

- Formulate ECoPS for cooperative tracking and merge strategic and tactical algorithms. Demonstrate with simulation.

- Add radar exposure cost to ECoPS.

2.2.1 Potential Research

To perform the proposed research, the following outstanding issues may also be addressed

- Consider the effect of mutation and recombination in ECoPS on the optimization of the path parameters.

- Implement Methods 1,2,4,5 with Hardware-in-the-Loop (HIL) or Piccolo© simulation versus the UAV model described in Section [1.3].

Chapter 3

REACTIVE METHODS

This chapter details the three methods that rely primarily on error feedback.

3.1 Helmsman Behavior Based Guidance Law

3.1.1 Description

The first method is an extension of path following as developed here by Rysdyk [18]. In this work, regaining the path consists of stabilizing two degrees of freedom to zero: course angle error, and cross track distance. The path geometry to be followed is determined based on the target current position and heading as well as a standoff distance.

Deviations from this path produce a course angle error and cross track distance. "Good Helmsman" behavior is employed, so-called as it smoothly transitions from current path to the commanded path by *simultaneously* bringing cross track distance and course angle error to zero:

$$\sigma_s = (\tilde{\chi} - \chi_s)\frac{e^{-ay_s/2} - 1}{e^{-ay_s/2} + 1} \tag{3.1}$$

The output, σ_s, is the commanded relative course with gain a. A PD control relates this to the course rate:

$$\begin{aligned} \chi_s &= s^{-}\kappa_1 \\ \tilde{\chi}_s &= \tilde{\chi} - \chi_s \\ y_s &= \int_0^t (V_c \sin(\tilde{\chi}_s)d\tau \\ s &= \int_0^t (V_c \cos(\tilde{\chi}_s)(1 - y_s^{-}\kappa_1))d\tau \\ \dot{\psi}_c \approx \dot{\chi}_c &= \left(k_p(\sigma_s - \tilde{\chi}_s) + k_i \int_0^t (\sigma_s - \tilde{\chi}_s)d\tau\right) - V_c\kappa_2 \end{aligned} \tag{3.2}$$

Here, κ_1 is the command nominal path curvature $(1/r_c)$ and κ_2 is the 'adjusted' path curvature. For two UAV coordination, the 'lead' aircraft holds a constant stand-off distance

from the target so κ_1 and κ_2 are equal. Based on the clock angle separation error (using another "Good Helmsman" transition function), the 'follower' aircraft adjusts κ_2 as well as adjusting the cross track error used to calculate σ_s: $y_s = y_s + (1/\kappa_2 - 1/\kappa_1)$. Optionally, both aircraft speeds can be adjusted (one up, one down) when speed control is available. This provides the necessary additional degree of freedom to achieve a desired angular separation between the two UAVs.

3.1.2 Non-Real Time Simulations

The following results are based on the platform described in Appendix [A.2.1].

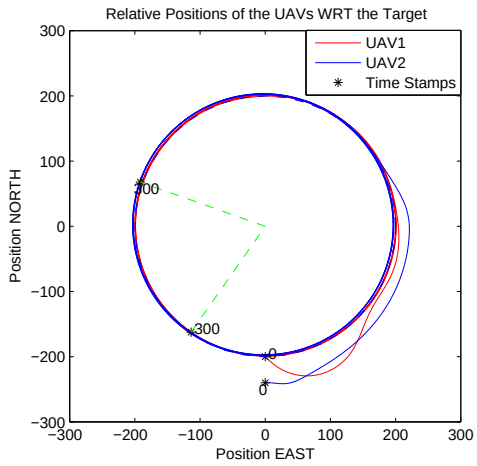


Figure 3.1: Helmsman: Tracking a Fixed Target With an Initial Condition Offset

Figures [3.1 - 3.7] result as the target and wind velocity profiles were changed per Section [A.1.2]. In each figure, UAV1 is the 'lead' UAV and UAV2 is the 'follower'.

In Figure [3.1], UAV1 has the initial heading of -45 degrees. Comparing to Figure[1.2], UAV1 should have a heading of zero degrees at this point in the orbit. An orbit correction occurs as a result, and UAV1 settles on the desired orbit. UAV2 has an initial distance from the target further than the commanded orbit radius of 200 meters. Further, the clock angle between UAV1 and UAV2 is zero whereas the commanded relative clock angle is 75 degrees. UAV2 also corrects and settles on the desired orbit with a correct separation clock

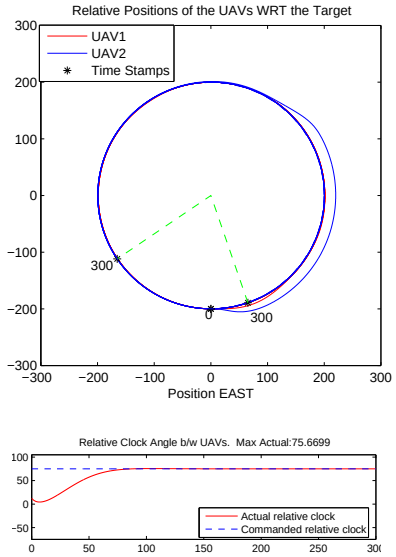


Figure 3.2: Helmsman - Tracking a Fixed Target

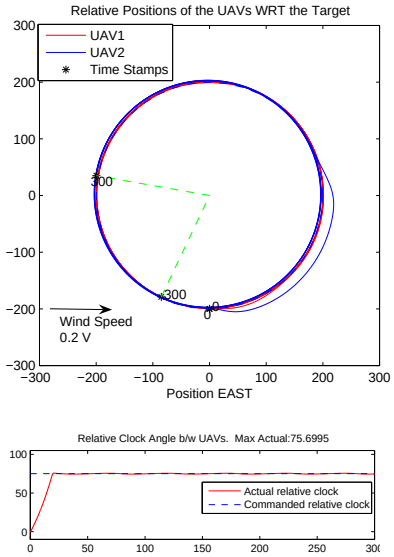


Figure 3.3: Helmsman - Tracking a Fixed Target With Wind

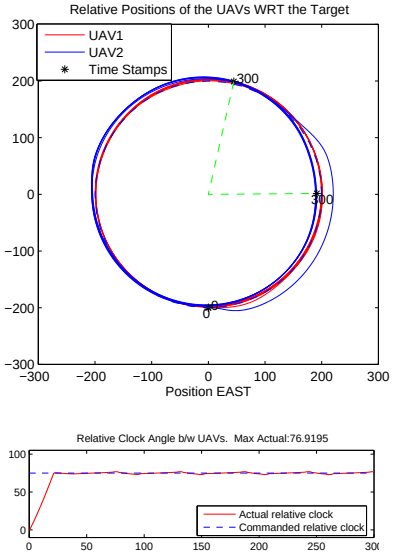


Figure 3.4: Helmsman - Tracking a Slow Target

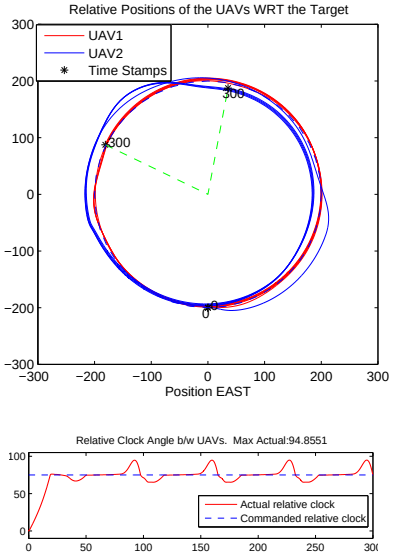


Figure 3.5: Helmsman - Tracking a Slow Target With Wind

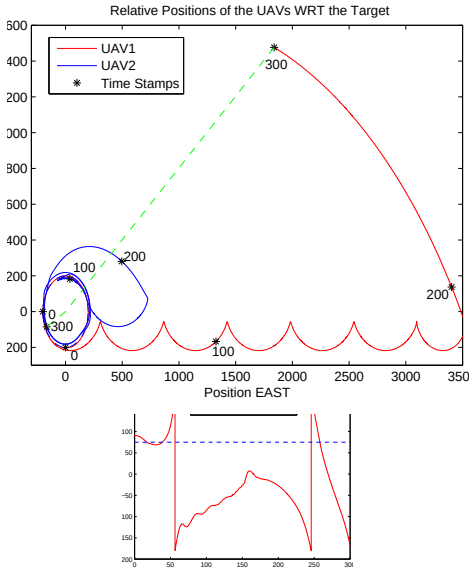


Figure 3.6: Helmsman: Tracking a Fast Target

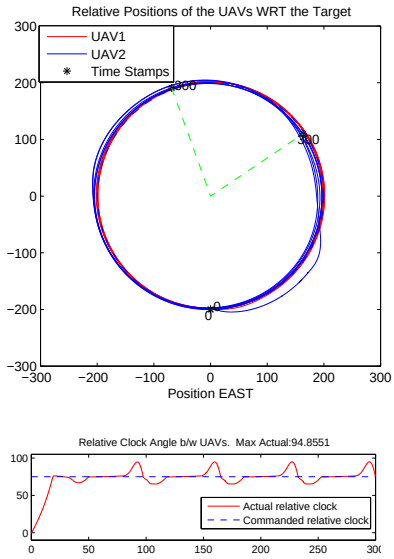


Figure 3.7: Helmsman: Tracking a Turning Target

angle. As this method is able to correct for initial condition errors, further simulations will all start with both UAVs having heading angles of 0 degrees and initial positions 200 meters to the South of the target.

Figure [3.2] shows that this method works when the target is stationary and no wind is present. Figure [3.3] demonstrates the effect of wind. An adverse wind with a speed of 20% of UAV speed results in a regular oscillation occurs for the relative clock angle between the UAVs, but remains bounded to within 2% of the command.

In Figure [3.4], target forward speed has a similar effect on the relative clock angle as the wind, i.e. a small, regular oscillation. Wind can be seen to skew the orbit patterns of the UAVs in Figure [3.5]. The compensation of UAV2 is also apparent as UAV2s radius regularly changes. The wind has also caused the clock angle to be affected in a regular pattern. The overshoot has increased to approximately 25%.

In Figure [3.6], the UAVs are too slow to remain near the target and fall behind. Considering the logic in Equation [3.2], and noting that cross-track error will grow large as the

target distance increases, the heading rate command will quickly grow to saturate the bank angle command. For UAV1, the integrator anti-windup is in effect, but UAV1 cannot reduce the cross track error until the target slows. Its heading shifts from the high angle limit to the low angle limit and it quickly loses ground. When the target slows, the command desaturates and progress can be made to reduce the lost distance. UAV2 also drops back, but due to its 'adjustable' path curvature, the heading rate command does not saturate and its heading is such that it looses much less ground than UAV1. When the target slows, UAV2 is still close and quickly regains the orbit.

Finally, Figure [3.7] demonstrates the UAVs following a target through a turn.

3.1.3 Real Time Simulations

The helmsman method has been modified, as in Appendix [A.3], to run in a real-time environment. Simulations in progress.

3.2 Lyapunov Vector Field Guidance Law

3.2.1 Description of the Vector Field

The second method builds upon the work done by Lawrence and Frew [8]. The guidance of the UAV to the observation 'orbit' is determined by building a vector field that has a stable limit cycle centered on the target position. The governing equations are as follows:

$$\begin{aligned} r &= \sqrt{\tilde{x}^2 + \tilde{y}^2} \\ vx &= -\tilde{x}(r^2 - r_c^2) - 2rr_d\tilde{y} \\ vy &= -\tilde{y}(r^2 - r_c^2) + 2rr_d\tilde{x} \\ \chi_{comVF} &= atan2(vy, vx) \\ \Delta_{chi\_com} &= \chi_{comVF} - ang(\tilde{\vec{V}}) \end{aligned} \tag{3.3}$$

Figures [3.8 - 3.10] visualize this vector field. Figure [3.8] shows the vector field surrounding a stationary target. It is clear that a path following the field vectors from any point will end up on a circle of radius equal to the specified stand-off distance (shown in black).

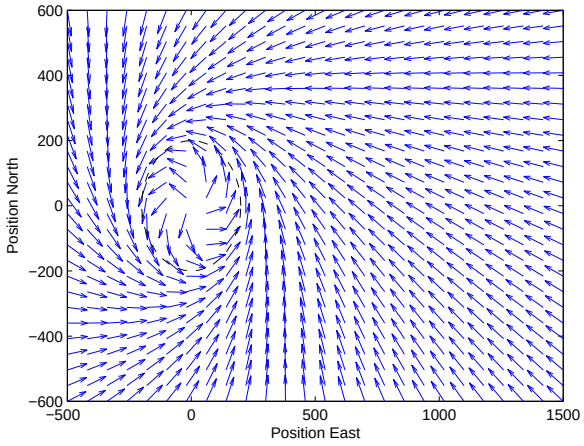


Figure 3.8: Stationary Target Vector Field

To understand how the target motion affects the velocity field, we can determine the rate of change of the gradient due to the target as follows: The area around the target is discretized into points at relative coordinates (i, j) (i.e. target relative location is (0,0)). The gradient is calculated at each discrete point resulting in the direction of the field vector $\chi_{(i,j)}(t)$. A new gradient vector is calculated at the same points (i,j), but with target position delayed in time by ΔT, resulting in a new field angle $\chi_{(i,j)}(t - \Delta T)$. Rate of change is:

$$\dot{\chi} = \frac{\chi_{(i,j)}(t) - \chi_{(i,j)}(t - \Delta T))}{\Delta T}$$

Figures [3.9–3.10] show the vector field locally surrounding a target moving Easterly. In each figure, a color map is used to indicate vector rotational rates as a percentage of the maximum possible heading rate. Far from the target, the color is constant and represents $\dot{\psi} < 0.25\dot{\psi}_{max}$. Each subsequent shade darkening of the map indicates an increase of $0.25\dot{\psi}_{max}$. A dark, dashed line indicates the relative orbit around the target. The maximum heading rate is based on Equation [1.7]. A heading rate of greater than 17.7 deg/sec will saturate the bank angle command, set at 45 degrees.

In Figure [3.9], the target's speed is 43% that of the UAV. No shade increase appears along the orbit and thus the maximum command along the orbit remains less than $0.25\dot{\psi}_{max}$.

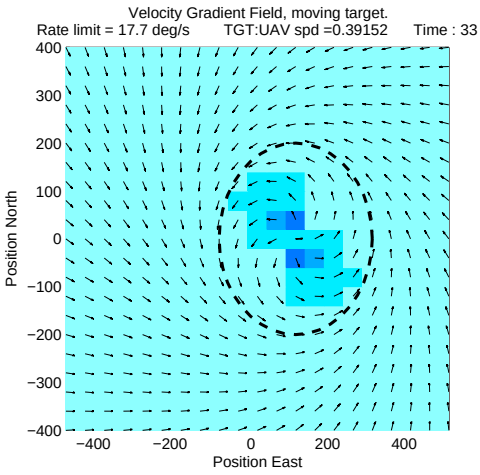


Figure 3.9: Slow Target Vector Field

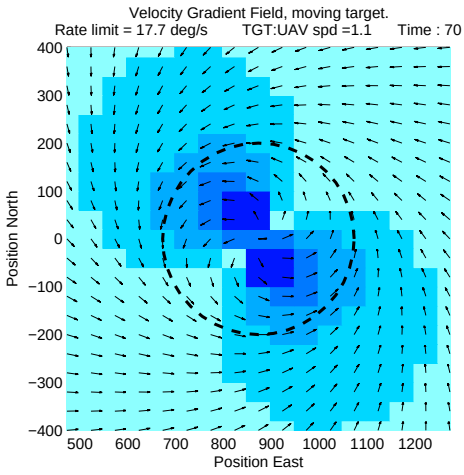


Figure 3.10: Fast Target Vector Field

In Figure [3.10], the target has speed up to 110% that of the UAV. As this is faster than the UAV, the UAV will never orbit the target in this regime. And yet, along the orbit, the shade has increased by only two increments indicating a command of $\dot{\chi_c} < 0.75\dot{\psi}_{max}$. Indeed, the command does not increase to greater than $\dot{\psi}_{max}$ unless the UAV is well within the specified stand-off distance. Therefore, assuming that no wind is present, the angle command given by the vector field is within the capability of the UAV to follow for all target speeds for which the UAV can orbit the target.

3.2.2 Application of the Vector Field for Control Guidance

To extend the vector field angle output to the required logic output we start with the PD control:

$$\dot{\psi}_c \approx \dot{\chi}_c = k_p \Delta_{chi\_com} + k_d \, ang(\tilde{\vec{V}}) \tag{3.4}$$

To ensure the heading rate command is still within the bounds of the UAV capability, the proportional constant, k_p, is kept less than unity and the derivative term is set to only reduce the heading rate command.

To be able to follow a target that goes faster than the UAV, a second command is

generated:

$$\dot{\chi}_{ct} = k_2 \Delta \left(ang(\tilde{\vec{V}}) - ang(\theta_{tan}) \right) \tag{3.5}$$

where $ang(\theta_{tan})$ is the angle of a line that connects the current UAV position to the tangent of a circle, centered on the target postion, of radius equal to the specified stand-off distance.

However, this command is heavily weighted, as a function of target distance, such that it only has an effect when the UAV is distant from the target. This is the case when Equation [3.4] is insufficient to maintain the UAV at stand-off distance, such as when the UAV becomes slower than the target. Conversely, the basic *vector field* command is weighted oppositely. This ensures that the vector field command is used whenever the UAV is near, and can keep up with, the target. The weighting functions are used as transitions to avoid hard switches.

3.2.3 UAV Coordination

For two UAV coordination, an additional course adjustment is made, based upon the relative clock angle, for the 'follower' aircraft. This amounts to adjusting the output of the vector field command in order to vary the UAV distance to the target, resulting in a change in the UAV's clock angle rate ($\dot{\Psi} = (|\mathbf{V}_{uav} - \mathbf{V}_t|)/r$).

Using radius adjustment as a means to regulate relative clock angle has a drawback. The reasoning follows that of Section [1.2.1]. If, for example, the 'follower' UAV is passing along side the target while the 'leader' has just passed by the front of the target. The 'leader' will have a faster clock angle rate than the 'follower', due to the 'leader's' increased relative speed with respect to the target. The 'follower' must then reduce its radius to increase its rotational rate about the target to catch up. If the target is fast, the reduction of radius will have less of an effect than if the target were slow ($\dot{\Psi} = (|\mathbf{V}_{uav} - \mathbf{V}_t|)/r$) and the 'follower' will lag well behind. When the 'follower' passes in front of the target at a reduced radius, the effect will be reversed, and the 'follower' UAV's high rotational rate about the target may cause it to end up ahead of the 'leader' UAV. Since the rate divergence increases with target speed as well as with the relative clock angle between the UAVs, the commanded relative clock angle between the two UAVs is reduced as target speed increases. Note that this speed correction will reduce the accuracy of the target location determination at higher

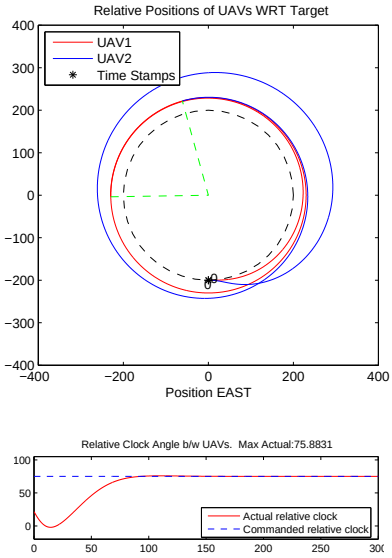


Figure 3.11: Lyapunov - Tracking a Fixed Target

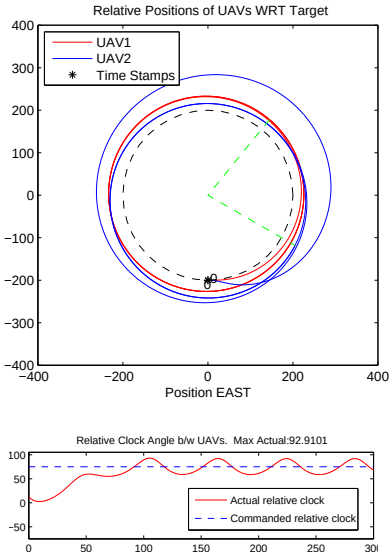


Figure 3.12: Lyapunov - Tracking a Fixed Target With Wind

target speeds.

3.2.4 Non-Real Time Simulations

The following results are based on the platform described in Appendix [A.2.1].

Figures [3.11 - 3.16] result as the target and wind velocity profiles were changed per Section [A.1.2]. In each figure, UAV1 is the 'lead' UAV and UAV2 is the 'follower'.

In Figure[3.11], it is clear that this method can meet the clock angle requirement while loitering around a stationary target with no wind present. Figure [3.12] demonstrates the effect of wind. An undamped oscillation occurs for the relative clock angle between the UAVs, but remains bounded to within 25% of the command.

In Figure [3.13], target forward speed has a similar effect on the relative clock angle as the wind, i.e. an undamped oscillation. Wind can be seen to skew the orbit patterns of the UAVs in Figure [3.14]. The median of the relative clock angle oscillation is also shifted below the command.

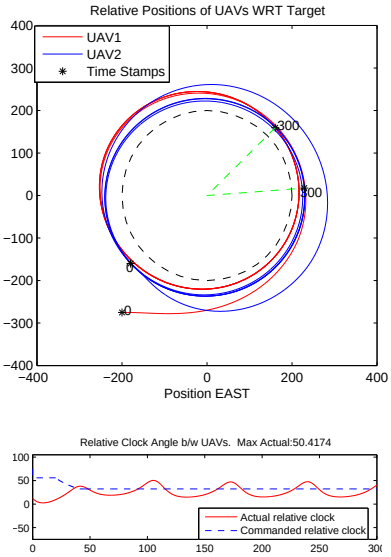


Figure 3.13: Lyapunov - Tracking a Slow Target

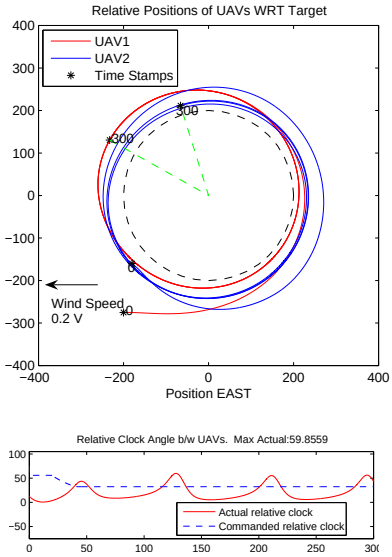


Figure 3.14: Lyapunov - Tracking a Slow Target With Wind

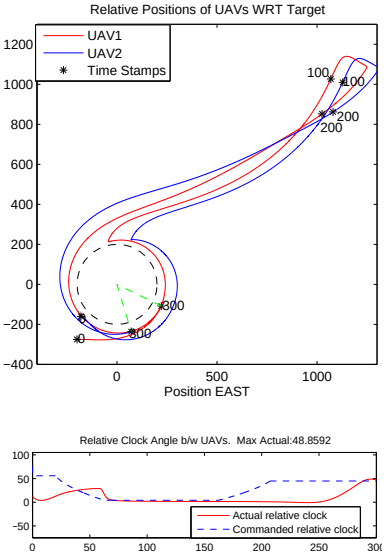


Figure 3.15: Lyapunov - Tracking a Fast Target

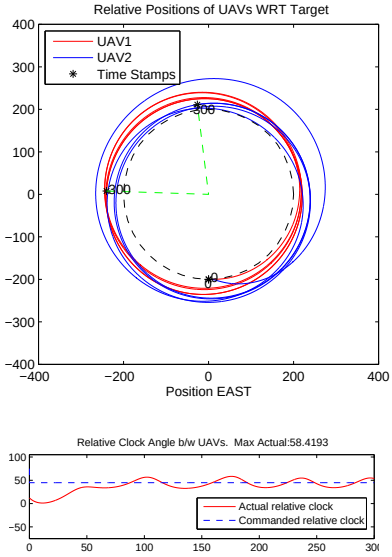


Figure 3.16: Lyapunov - Tracking a Turning Target

In Figure [3.15], the UAVs are too slow to remain near the target and fall behind. A line tangent to their backward-drawing path can be seen to fall ahead of the westerly moving target by about 300 meters. This is due to the command that preferrentially draws the UAV, as the distance to the target increases, to align its heading to a tangent to the circle, centered on the target, of stand-off distance radius as the distance to the target increases. When the target speed drops below the speed of the UAV near time 170 (see Figure[A.3]), the UAVs again gain ground on the target and transition back to their orbit. It should be noted that this result was obtained only after making the following modification:

> Equations [3.3] and [3.4] show that the heading rate command depends on the angle of the relative velocity vector. For $K > 0$
>
> Case 1: Let $\vec{V}_{rel1} = \vec{V}_{UAV} - \vec{V}_{tgt}$, where $|\vec{V}_{UAV}| = K \times |\vec{V}_{tgt}|$.
> Case 2: Let $\vec{V}_{rel2} = \vec{V}_{UAV} - \vec{V}_{tgt}$, where $|\vec{V}_{tgt}| = K \times |\vec{V}_{UAV}|$.
> It follows that $|\vec{V}_{rel1}| = |\vec{V}_{rel2}|$ but $ang_{rel2} = ang_{rel1} + \pi$.
>
> Case 1 was used all of the time, regardless of the relative speed, see Equation [3.4]. The result was an injection of an extra π in the calculated heading angle as the target speed up past the UAV speed. This saturated the error command and resulted in significant overshooting. As a solution to this problem, note that the relative angle can also be calculated by determining the difference in course angles:
>
> $$ang_{relb} = ang(\vec{V}_{UAV}) - ang(\vec{V}_{tgt})$$
>
> In all results shown for this method, ang_{relb} is used, preferrentially over $ang(\tilde{\vec{V}})$, at target speeds faster than the UAV. The transition between the two angle commands is provided by a weighting function resulting in a smooth relative angle command without overshoot.

Finally, Figure [3.16] shows that the UAVs can follow a target through a turn. A single, much wider path line of the 'follower' UAV corresponds to the compensation required as the turn occurs. However, the clock angle does not seem to suffer for it.

3.2.5 Real Time Simulations

The Lyapunov method has been modified, as in Appendix [A.3], to run in a real time environment. Simulations in progress.

3.3 Controlled Collective Motion

3.3.1 Description

The third method builds upon the work done by Klein and Morgansen [10]. In [10], the guidance of unit speed planar kinematic unicycles is commanded such that the centroid of the 'flock' of unicycles corresponds with the moving target position. Our extension on this work applies this method to UAV model described in Section [1.3], again attempting to impose a constant angular separation between two UAVs. The basic Equation for the control of $\dot{\chi}_j$, $j = 1 \ldots$number of UAVs, is

$$\begin{aligned} \dot{\chi}_j &= u_1 + u_2 + u_3 \\ u_1 &= uspac_j \\ u_2 &= \frac{K}{n}(\frac{d}{d\chi_j}(vek_j)u1_j) \\ u_3 &= -\frac{K}{n}(vek_j\frac{d}{d\chi_j}(tek_j)u2_j) \end{aligned} \tag{3.6}$$

Where:
$uspac_j$ calculates the control necessary to achieve a specified distance from UAV_j to the centroid,
$\frac{K}{n}(\frac{d}{d\chi_j}(vek_j)u1_j)$ describes the control necessary to match the target and centroid speed,
$-\frac{K}{n}(vek_j\frac{d}{d\chi_j}(tek_j)u2_j)$ describes the control necessary to match the centroid direction of travel with that of the target.

Knowing also that K, contained within Eqn [3.6], is an adjustable control gain we can immediately gain insight. Note that if K is zero, no control authority is given to match the centroid velocity (speed and direction) with the velocity of the target and thus we cannot expect the UAVs to track any target that moves. Note also that if $K \to \infty$, no control authority is given to keep the UAVs at a specified distance from the centroid. Thus we can

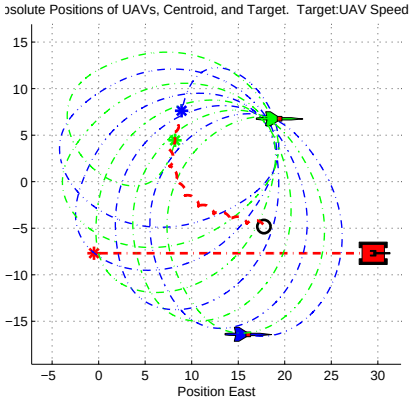


Figure 3.17: Centroid : Two Oscillators Tracking a Target

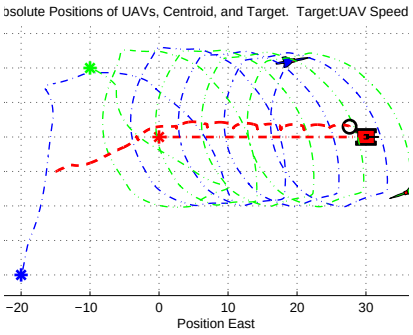


Figure 3.18: Centroid : Tracking With a Tangent Swarming Function

expect the UAVs to head, in radially opposite directions, towards infinity, as the centroid will still remain locally near the target. Thus there is a tradeoff between maintaining at a specified stand-off distance from the centroid and tracking the centroid with the target.

This tradeoff is of significant issue if there are only two vehicles. In fact, there is no hope of perfectly tracking the target velocity. Figure [3.17] shows that the UAVs, simply modeled as oscillators, increasingly lag behind a target that is only one tenth as fast as the UAV. If the UAV number increases to three, this problem is alleviated as the centroid of three vehicles can be adjusted either to the left or right of the centroid of two. This extra authority over the centroid position, based on the position of the third vehicle, is enough to drive the centroid to meet the target and stay there as in Figure [3.19]. Note however, in the case of two UAVs, the orbit spacing is always maintained by default (the UAVs are always separated by a clock angle of 180^o with respect to the centroid), as long as the centroid is maintained on the target position. In the case of three vehicles, the 'adjustment' above to keep the centroid tracking the target penalizes the spacing requirement. As each vehicle will have differing distances from the centroid, each will also have differing rotational rates around the centroid. This will result in non-constant clock angle separation.

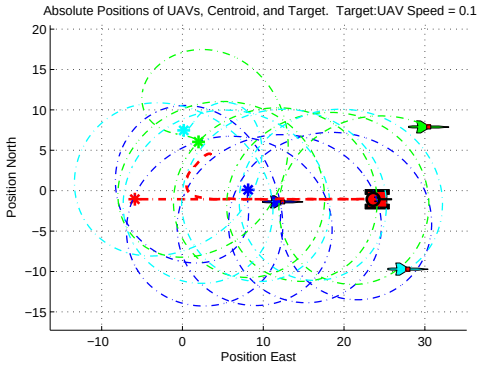


Figure 3.19: Centroid : Three Oscillators Tracking a Target

As our problem focuses on two vehicles, a solution to target tracking must be found in order to further consider this method. We employ a *swarming* function, similar to the tangent function of Section [3.2.2]. This function directs the UAV along a path tangent to a circle, centered on the target position, of radius equal to the specified standoff distance. The total command is thus

$$\dot{\chi_j}_{tot} = (1 - e^{\frac{-d_c}{K}})\delta_{\chi_{jt}} + e^{\frac{-d_c}{K}}\dot{\chi_j} \qquad j = 1, 2 \tag{3.7}$$

Where d_c is the distance from the centroid to the target and K is an adjustable gain.

Equation [3.7] swarms, or brings all of the UAVs toward the target, by setting the commanded heading rate to be zero only if the vehicle heading is pointing along the tangent line to the target. The exponential weighting functions avoid the use of switching by providing the emphasis on the original function (trade off vehicle spacing and centroid-target convergence) only when the distance from the centroid to the target is small. This implies that one can reduce the gain in Equation [3.6] to achieve good distance separation as the swarming function complements the centroid-target convergence by definition. See Figure [3.18].

3.3.2 Non-Real Time Simulations

The following results are based on the platform described in Appendix [A.2.1].

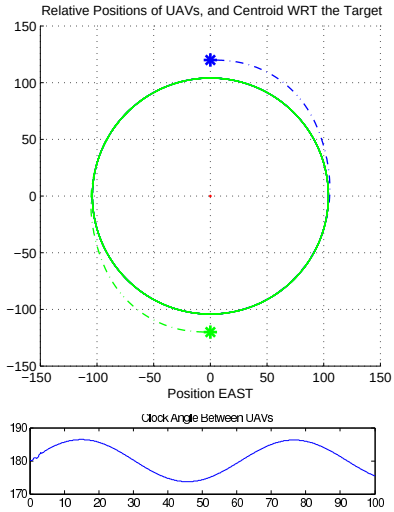


Figure 3.20: Centroid - Tracking a Fixed Target

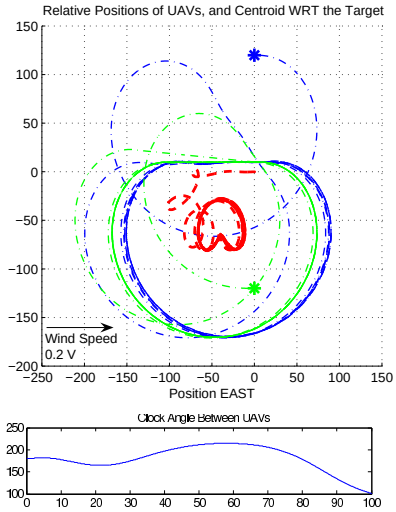


Figure 3.21: Centroid - Tracking a Fixed Target With Wind

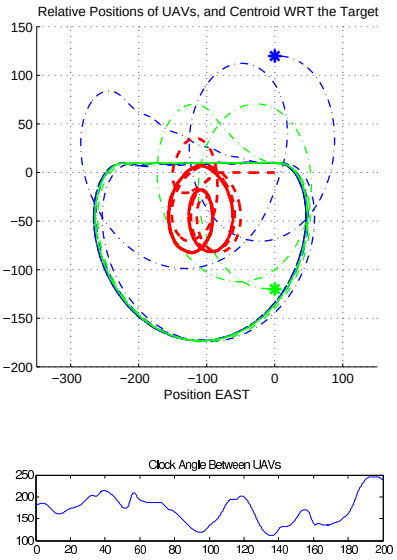


Figure 3.22: Centroid - Tracking a Slow Target

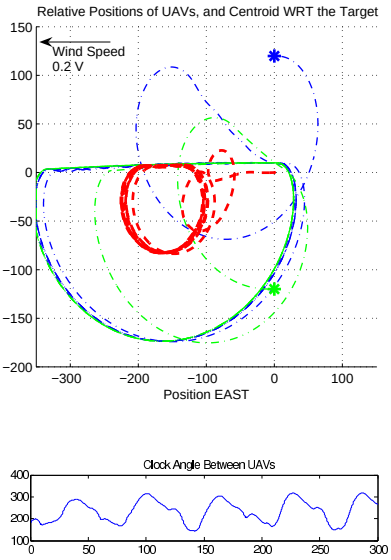


Figure 3.23: Centroid - Tracking a Slow Target With Wind

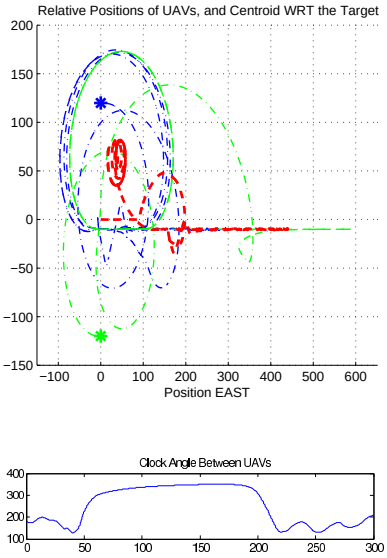


Figure 3.24: Centroid - Tracking a Fast Target

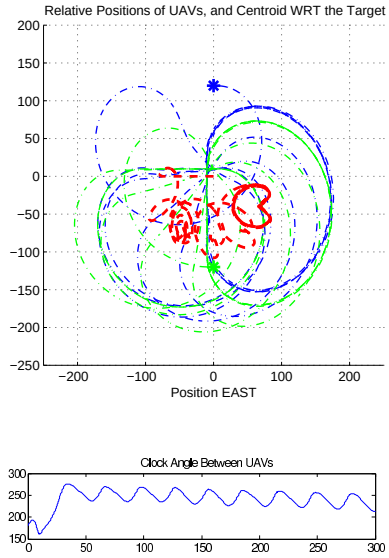


Figure 3.25: Centroid - Tracking a Turning Target

Figures [3.20-3.25] result as the target and wind velocity profiles were changed per Section [A.1.2]. In each figure, the path of UAV1 is in blue, the path of UAV2 is in green, and the path of the centroid is in red.

Figure [3.20] demonstrates the loitering capability of this method. A slight oscillation in the clock angle occurs. The wind, as seen in Figure [3.21], significantly alters the orbit patterns of the UAVs. This moves the centroid away from the target position, detrimentally to the clock angle separation.

Figures [3.22 - 3.25], as opposed to Figures [3.17 - 3.18], demonstrate the considerable effect of adding kinematics, dynamics, and limits to the oscillator model upon which this method is based. In each case, the variations of the clock angle separation with respect to the target is quite large. This is due to the 'traveling' of the centroid with respect to the target. Maintaining the centroid on the target is the key to keeping angular separation of the UAVs with respect to the target.

Notice that the tangent *swarming* function keeps the UAVs within range of the fast

target in Figure [3.24] and allows them the regain their orbit when the target slows per the profile in Figure [A.3]. Also note that in Figure [3.25], the slower target speed has reduced the clock angle oscillation amplitude, although now a clear drift in average clock angle separation can be detected.

Figures [3.21] - [3.25] demonstrate that this method does not maintain the UAV clock angle constant, or even nearly constant. Rather, larger variations in relative clock angle occur due to both target motion and wind effects, contrary to goals of cooperative target tracking. Further, the steady-state angular separation is a function of the number of vehicles ($2\pi/n$) and cannot be commanded. This precludes achieving the optimal angle of separation discussed in Section [1.1]. Therefore, this method will not be considered in further work.

Chapter 4

PREDICTIVE METHODS

This chapter details two methods that utilize future path determination.

4.1 Evolutionary Based Trajectory Paths and Path Following Commands

4.1.1 Introduction to ECoPS

Evolution-Based Cooperative Planning System (ECoPS) was developed at the University of Washington Autonomous Flight System Laboratory (AFSL) as a strategic, cooperative planner. See Pongpunwattana [16]. ECoPS uses evolutionary-based computation to determine, in real time, a feasible vehicle path based on a selection and mutation scheme of a plan consisting of multiple potential paths.

Real-Time Operation

To operate in real-time, ECoPS uses model predictive methods to evaluate potential future paths. The process begins by determining a fixed initial vehicle trajectory, offline (or prior to vehicle movement). This fixed trajectory has a specified length, as measured in time, called the *planning horizon time* (PHT). From the initial time of movement, t_o, until time $t_o + PHT$, the algorithm uses evolutionary-based computation to evolve future paths, all of which start at time $t_o + PHT$. At time $t_o + PHT$, the first PHT seconds of the best future path becomes the fixed trajectory and the future paths are evolved starting at time $t_o + 2 \times PHT$. This is repeated until the problem is over.

Process of Evolution

Path evolution, depicted in Figure [4.1], starts with a set of plans. Each plan consists of a possible future vehicle path and the *fitness* of that path. Each path consists of a number of simple segments (straight lines and curves) that, when joined, create a total path joining

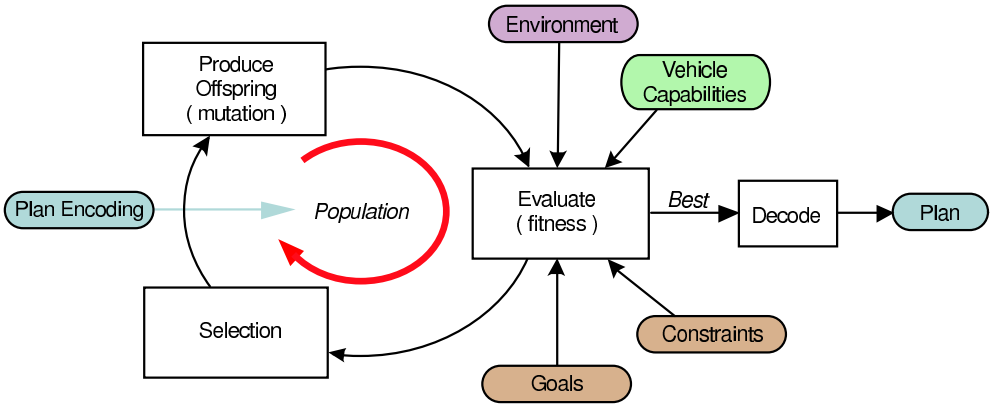


Figure 4.1: Evolutionary Algorithmic Cycle

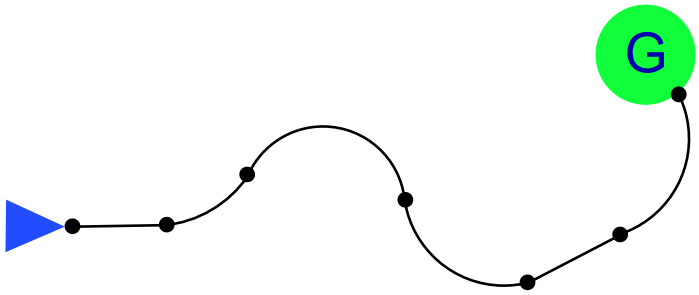


Figure 4.2: Creating a Path From Segments

the beginning and end states of the problem, see Figure [4.2]. Each segment is created such that all maneuvers necessary to 'fly' the segment are within the capabilities of the aircraft. As a path is created by joining feasible segments, every path represents a feasible solution.

Next, the *fitness* of each path is determined. *Fitness* is a relative measure of how a given path compares to another. ECoPS is designed to handle optimization with respect to multiple criteria. Each criterion is associated with a function that outputs a parameter that can be related to a parameter associated by other criteria. The function and relation must be specified, but the structure is not limited (i.e. linear, non-linear, probabilistic, etc).

With many possible plans, ECoPS moves on to a selection scheme. A tournament selection is used to determine the 'better' paths in the plan based on each plan's *fitness*.

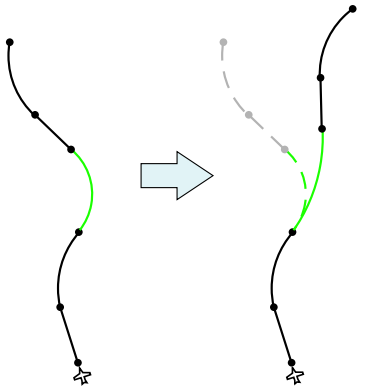


Figure 4.3: Mutation of a Path

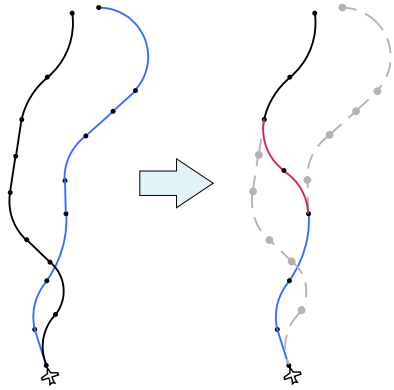


Figure 4.4: Combining Two Paths

Selection orders the plans by fitness level and 'keeps' some of the plans with higher fitness levels.

Finally, with a reduced number of 'better' plans, the population of plans is refilled (or repopulated) to original capacity by mutating particular paths or combining multiple paths, see Figures [4.3] and [4.4]. The evolution process runs continuously. At any time, the best path in the population is available to be extracted.

4.1.2 Transitioning ECoPS From Strategic to Tactical

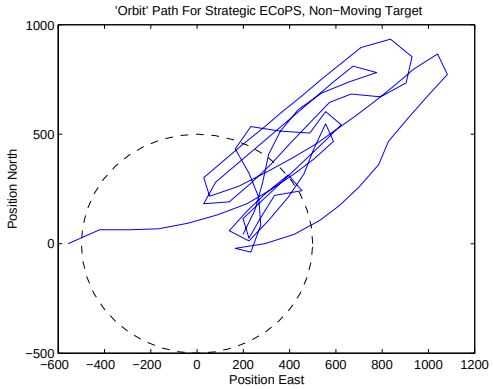


Figure 4.5: 'Orbiting' with ECoPS in Strategic Mode

Our main goal is to expand ECoPS to be able to handle both strategic and tactical problems, including the transition between tem. Figure[4.5] demonstrates the inadequacy of the strategic formulation of ECoPS to maintain an orbit, even of a non-moving target. Normally, when the vehicle is near enough to the target, the strategic task of locating the target (searching) is considered complete. However, the tactical task of remaining near the target is not complete, so ECoPS replans. The result is a vehicle path that keeps looping around itself as the vehicle tries to get close enough to complete the task.

By improving the structural framework while using the same evolutionary ideas already in place, we have increased the modularity of ECoPS to be able to generate paths for multiple mission types, simply by logically switching to an appropriate set of evolutionary *plan* types and *fitness* methods as the following description details.

Increasing Computational Efficiency

ECoPS, as it stood, was solving an inherently strategic problem. The algorithm considered the long-term logic required for path planning while integrating dynamically changing inputs to modify the plan. The strategic nature of the planning process implied that the dynamic rate of change of the problem was much slower than the calculational demand of the algorithm.

Our first focus was on reducing the computational load to meet the shorter time scale of a tactical problem. This started with logic to switch between computational methods based on the accuracy needed (such as flat earth versus round earth) and optimizing/rewriting methods to reduce unnecessary calculations (which also resulted in less computational error).

Our next focus was on modularizing ECoPS. As ECoPS now stands, the logic to switch from strategic to tactical is made at the highest level. The knowledge of this decision is then propagated downward. In most cases, we worked to integrate the decision as seamlessly as possible into the existing lower level methods such that the appropriate calculational methods were used. In some cases, a second version of an existing method was created for use when the problem was deemed tactical. These few new methods were identified as modules, as they each had similar underlying framework and output, but had significantly

different calculational methods based on the requirements of the problem. With this framework identified, future types of tactical problems (other than target tracking with clock angle separation) can be also tackled without writing a whole new program. Rather, the requirements of the problem need just be integrated and coded within the framework.

Modification of the Evolution Process

- Segment Scaling: For the strategic problem, the scaling of the constituent segments was based on the size of the strategic problem such that each path had nearly the same number of segments. In the tactical problem, the constituent segments are rescaled based on the size of the tactical problem, which depends on a variable orbit stand-off distance.

- Path Creation: The end of the strategic mission is the goal state and all paths joined to the goal. For a tactical problem, there is no apparent 'end' state to join the 'begin' state of the tactical problem to. To this end, all tactical paths are forced to have the same length, as measured in time. To make all paths have the same length, a basic total path time is calculated: $T = c\frac{l_{ave}}{v_{ave}}$, where l_{ave}, v_{ave} are the average segment length and vehicle velocity, and c is some constant that gives the number of average segments in the path. The segments are scaled based on the tactical problem as before. When a path is created by segments, the time elapsed by each subsequent segment is calculated as δt_i. Additional segments are added until

$$T > \sum_{i=1}^{k} \delta t_i, \qquad T < \sum_{i=1}^{k+1} \delta t_i$$

  Segment $k+1$ is then split such that $T = \sum_{i=1}^{k+1} \delta t_i$.

- Fitness: In strategic ECoPS, each plan's path is assigned a fitness score based on the following measures: path length, probability of task accomplishment along the path, and the probability of vehicle loss. For the tactical problem, path length remains important but vehicle loss may not be a factor and task accomplishment is not applicable, as the task is ongoing. For target tracking, more appropriate measures are

distance maintained along the path to the target (or the target's predicted future position) and clock angle separation based on the predicted future position of the other vehicle.

Distance fitness (1 is best, 0 is worst):

$r = r_o \rightarrow \text{score} = 1$

$r_o < r < r_{max} \rightarrow \text{score} = \frac{r_o}{r}$

$r > r_{max} \rightarrow \text{score} = (\frac{r_o}{r})^2$

$r_{min} < r < r_o \rightarrow \text{score} = \frac{r}{r_o}$

$r < r_{min} \rightarrow \text{score} = (\frac{r}{r_o})^2$

- Tactical Bounding Box: To ensure that the vehicle stays near the target, a bounding box, that is centered on and moves with the target, is created and scaled based on the command orbit radius. Only paths that remain in this bounding box are considered feasible. To ensure that the paths remain bounded to a moving target, an iterative process is followed. The path is discretized into time increments. At each time increment ($t_i* = t_{pathbegin} + \delta t_i$), the target's position is estimated and a Bounding Box is created centered on this position. Then, the path segment is selected that contains the time ($t_{begin} \leq t_i* < t_{end}$), where t_{begin} and t_{end} are the beginning and end times of the path segment. Whether or not the segment Bounding Box is contained within the target Bounding Box is determined. A yes/no boolean is associated. A no boolean will exclude this path from succeeding to the next evolutionary cycle. As such, the path initialization procedures are modified such that every plan in the initial population (start of problem) is contained within the Tactical Bounding Box.

- Repopulation:

  Combination versus Mutation: Continual combination of two existing paths tends to result in, over time, a path with the best that the two existing paths have to offer, i.e. the optimal combination of the existing constituent path segments. Mutation of

a particular path, on the other hand, changes the constituent segments. Therefore, mutation tends to keep the population of paths varied. In terms of optimization, combination alone may result in a local minimum, and not a global minimum. Mutation will not get stuck in a local minimum, but it tends to take more time to reach a solution as minimized as using combination alone. A strategic problem uses both but, as the time scale is longer, focuses on mutation to prevent overlooking a potentially better solution. As tactical problems change on a shorter time scale than strategic problems, the focus must be shifted toward quickly optimizing a solution, and combinatorial methods are more important. To this end, we are adding a second joining method to ECoPS and reducing the number of mutative methods from four to three.

Effect of a constant total path time on repopulation: As combination or mutation will, in general, vary the path time, each path is subsequently renormalized (segments are added or subtracted from the end of the path to keep path time constant). A basic mutation in ECoPS is to expand the path by adding segments. The normalization effectively cancels this mutation. Another basic mutation in ECoPS is to shrink the path. The path normalization adds new segments back to keep path time constant. The end effect is that the end segments are randomly switched out for new ones.

Single Vehicle Tactical Paths

Comparing Figure[4.5] to Figures[4.6 - 4.7] demonstrates that the tactical formulation results in much improved orbiting. However, note the jaggedness of the paths. This is a result of two effects. The first is that figures represent paths and not actual trajectories flown by aircraft with dynamics, which would tend to smooth the path. The second, more subtle reason, is that during repopulation, one or more segments are replaced by new, random segments. The end state of a new replacement segment may not be an exact state match to the next segment. To ensure continuity, a joining segment is used. See [16]. These joining segments are designed to be as short as possible and still be within the maneuvering capabilities of the vehicle. If the states are very close, the join segment will have a correspondingly sharp turn. This is exacerbated as there is currently no minimum length, in time, of a segment.

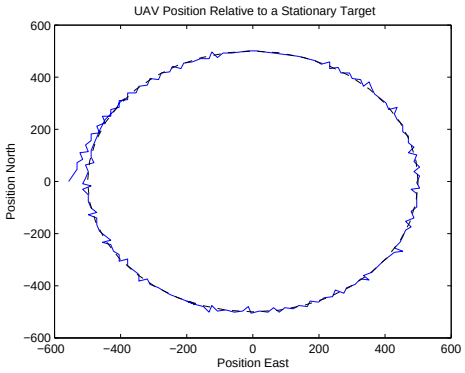


Figure 4.6: Tactical ECoPS for Stationary Target Tracking

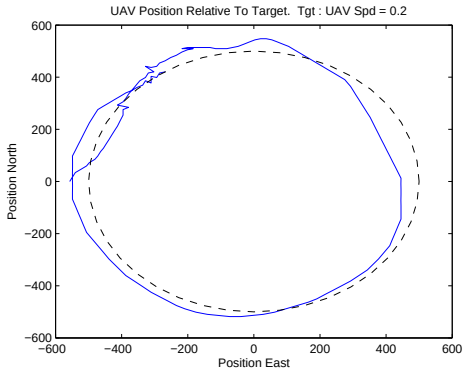


Figure 4.7: Tactical ECoPS for Moving Target Tracking

This means that the path may consist of very many short segments, many of which need to be joined.

4.1.3 Radar Avoidance

The structure of ECoPS makes it possible to add and/or subtract an unrestricted range of fitness functions of any type with fitness parameters. Minimizing radar signature while cooperatively tracking is an application that exposes this flexibility and has practical application. One scenario is counter-detection by a target with radar. Another is detection by hostile sites near to the target's path.

Efforts to solve the path planning problem to minimize radar exposure include the use of a minimax optimal control problem[14], and sequential quadratic programming[12]. For this research, we will focus on the radar model and scenario presented in [14], i.e. a strategic path planning problem, to demonstrate the capability of ECoPS. Then, the radar minimization problem will be incorporated as a tactical consideration for the tracking problem.

Probability of Detection and UAV Loss

In [14], the radar is associated with a hostile, surface-to-air missile threat. For the radar-guided missile to hit the aircraft, the radar must track and hold track for enough time to gain a good track ($\delta_t = T_{resp}$) and hold track during the missile flyout ($\delta_t = T_{fo}$). That is,

for a total time of $T = T_{resp} + T_{fo}$. This minimum time duration is known as the threat window. If the radar loses track during the response time, another T_{resp} seconds must elapse prior to missile launch. Also, if the radar loses track during the missile flyout, the aircraft must again be tracked for another T_{resp} seconds before another missile launch.

At any instant, the probability of tracking is given by

$$P_t = \frac{1}{1 + (c_2 R^4/\sigma)^{c_1}} \tag{4.1}$$

where R is the slant range and c_1 and c_2 are radar model specific constants (dependent on

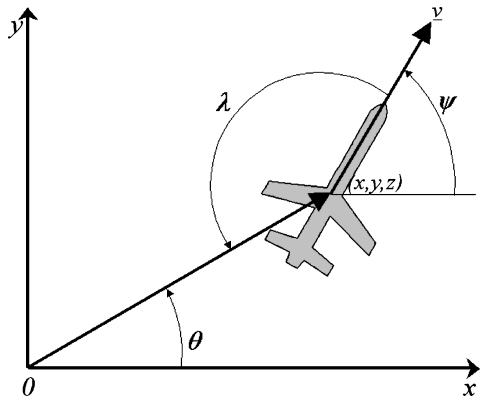


Figure 4.8: Angles - Heading (ψ), Aspect(λ), Azimuth(θ)

power, processing etc.). By setting $c_1 = 1$, we can define $c_2 = 1/R_p$. R_p is called the radar power parameter and is defined as the range at which $P_t = 0.5$ when $\sigma = 1$. Via integration, we get the probability of continuous detection as

$$P_t(t) = \frac{1}{\Delta T} \int_{t-\Delta T}^{T} P_t(\tau) d\tau \tag{4.2}$$

As the radar must hold continuous track to down the aircraft, the probability of aircraft loss ($P_d(t)$)is bounded above by the probability of continuous track:

$$P_d(t) \leq \frac{1}{T} \int_{t-T}^{T} P_t(\tau) d\tau \tag{4.3}$$

Therefore, the problem is to minimize $P_d(t)$ over all intervals over which continuous track is held.

Radar Modeling

Figure [4.8] defines the the heading, aspect, and azimuth angles. Assuming no altitude changes and given position (x, y, z), azimuth is given by: $\theta = tan^{-1}(y/x)$,
aspect: $\lambda = \theta - \psi + \pi$, elevation: $\mu = tan^{-1}(z/\sqrt{x^2 + y^2})$, bank angle: $\phi = tan^{-1}(\frac{v}{g}\dot{\psi})$.

In [14], an ellipsoid models the radar cross-section (RCS) is used as follows:

$$\begin{aligned} \sigma(\lambda, \phi, \mu) &= \frac{\pi a^2 b^2 c^2}{\left(a^2 \sin^2 \lambda_e \cos^2 \phi_e + b^2 \sin^2 \lambda_e \sin^2 \phi_e + c^2 \cos^2 \lambda_e\right)^2} \\ \lambda_e &= \cos^{-1}(\cos \mu \cos \lambda) \\ \phi_e &= \phi - tan^{-1}\left(\frac{\tan \mu}{\sin \lambda}\right) \end{aligned} \tag{4.4}$$

where a, b, c are dependent on the radar return levels of the aircraft as a function of aspect and bank angle.

Application to the ECoPS Fitness Function

In order to determine the fitness of a particular path with respect to radar avoidance, we need to determine the maximum probability of the aircraft being downed $P_d(t)$. This consists of three parts. The first is to determine during which time segments of the path the aircraft is being detected. The second part is to determine which of those time segments meets or exceeds the time span requirement of the threat window. The third is to determine the maximum probability of aircraft downing for all time segments whose duration meets or exceeds the threat window.

Let t_o be time now, T_{path} be the total path time (time now until path end time), T_{tw} be the threat window duration, and N be some integer number such that $\delta_t = T_{path}/N << T_{tw}$.

- For $t_o, t_1 = t_o + \delta_t \ldots t_i = t_{i-1} + \delta_t \ldots T_{path}$, determine the instantaneous probability of detection, $P_t(t_i)$, using Equations [4.1] and [4.4]. This is possible as ECoPS forward predicts the UAV and target motion in time.

- Determine the minimum set D consisting of $k \leq N$ time segments for which during the time span of segment, no time step has an instantaneous probability of detection of zero, ignoring any one member sets. For example, if $P_t(t_o), P_t(t_1), P_t(t_2) \neq 0$, then

the longest set $\{P_t(t_o), P_t(t_1), P_t(t_2)\}$ is a member of the minimum set D, and not either of the subsets $(\{P_t(t_o), P_t(t_1)\} + \{P_t(t_2)\}), (\{P_t(t_o), P_t(t_2)\} + \{P_t(t_1)\})$, etc.

- Determine the subset $W \subset D$ consisting of $m \leq k$ time segments whose duration is greater than T_{tw}. Note that if $W = \emptyset$ then the probability of downing for the path is considered zero.

- Determine the probability of downing for each member of W : $P_d(v) \quad v = 1 \ldots m$. This can be done by approximating the integral in Equation [4.2] by

$$P_d(v) \leq \frac{1}{n-1} \sum_{k=r}^{s-1} \frac{P_t(t_{k+1}) + P_t(t_k)}{2} \quad v = 1 \ldots m$$

  where n is the number of time instances in the member set, r is the index of the earliest time, s is the index of the latest time.

- The maximum probability of downing is $\max_m(P_d(v))$.

- As ECoPS fitness functions consider 1 best, 0 worst, then the fitness level for radar avoidance is $1 - \max_m(P_d(v))$

4.2 Model Predictive Control By Convex Optimization

There are today many very efficient methods to solve a problem that is formulated as a convex optimization problem [2]. This method converts cooperative tracking into a framework under which a convex optimization routine is used to determine the control input.

The method uses the 'Leader-Follower' approach. The 'Leader' operates to maintain a constant distance from the target. The 'Follower' UAV calculates its control input such that its position meets the constant relative clock angle requirement.

4.2.1 Feedforward Using Model Predictive Control

Let time now $= T$ and the current command be given by $\theta_c(T)$. To determine the best command in the future $(\theta(T + t_f))$ will take a calculation time of δ_t. If $\delta_t < t_f$ then at each integer multiple of t_f, a new command will be available for use $(\theta(T + n\ t_f))$. See Figure

[4.9].

A zero-order hold can then be used to create a piece-wise constant command $\theta_c(t)$.

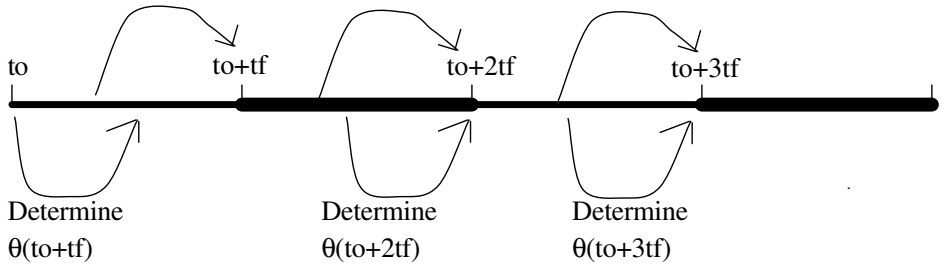


Figure 4.9: Calculation and Data Flow

4.2.2 Description of the Leader Dynamics

The guidance for the leader will be determined by the vector field described in Section [3.2]. If we assume that the leader can follow the vector field exactly, then we can replace the position dynamics of the leader with the vector field dynamics. Also, we assume that target motion will follow, at least for some time:

$$\begin{aligned} \dot{x}_t(t) &= v_{xt} = v_{xtr} V_o \\ \dot{y}_t(t) &= v_{yt} = v_{ytr} V_o \\ x_t(t+k) &\approx x_t(t) + k\ v_{xtr}\ V_o \\ y_t(t+k) &\approx y_t(t) + k\ v_{ytr}\ V_o \end{aligned} \tag{4.5}$$

Where V_o is the constant speed of the UAVs and v_{xt} and v_{yt} are assumed constant and known (as a percentage of UAV speed) over the interval $T \leq t \leq T + t_f$.

Further, over a short time period ($k = \delta t$)

$$x(t+k) \approx x(t) + k\dot{x(t)}$$

Therefore, for the leader, we can approximate the dynamics of the leader (with respect to the target) with:

$$r(t) = \sqrt{\tilde{x_L(t)}^2 + \tilde{y_L(t)}^2}$$

$$
\begin{aligned}
C(t) &= \frac{-V_o}{r(t)(r(t)^2 + r_c^2)} \\
R(t) &= \begin{pmatrix} r(t)^2 - r_c^2 & 2r(t)r_c \\ -2r(t)r_c & r(t)^2 - r_c^2 \end{pmatrix} \\
\begin{pmatrix} \tilde{x_L}(t+k) \\ \tilde{y_L}(t+k) \end{pmatrix} &= (I + kCR) \begin{pmatrix} \tilde{x_L}(t) \\ \tilde{y_L}(t) \end{pmatrix} - kV_o \begin{pmatrix} v_{xtr} \\ v_{ytr} \end{pmatrix}
\end{aligned} \tag{4.6}
$$

Note that at each time step, the clock angle of the leader can be determined as:

$$\tan \Psi_L(t) = \frac{\tilde{y_L}(t)}{\tilde{x_L}(t)} \tag{4.7}$$

4.2.3 Description of the Follower Dynamics

A simplified model of the dynamics of the follower vehicle is:

$$
\begin{aligned}
\tilde{x_F}(t) &= x_F(t) - x_t(t) \\
\tilde{y_F}(t) &= y_F(t) - y_t(t) \\
\begin{pmatrix} \tilde{x}_F(t+k) \\ \tilde{y}_F(t+k) \end{pmatrix} &= \begin{pmatrix} \tilde{x}_F(t) \\ \tilde{y}_F(t) \end{pmatrix} + kV_o \begin{pmatrix} \cos\psi_c - v_{xtr} \\ \sin\psi_c - v_{ytr} \end{pmatrix}
\end{aligned} \tag{4.8}
$$

Note that at each time step, the clock angle of the follower can be determined as:

$$tan\Psi_F(t) = \frac{\tilde{y_F}(t)}{\tilde{x_F}(t)} \tag{4.9}$$

Note also that the dynamics propagation in the follower reflects that a constant heading angle is to be used during the time step.

Equation [1.4] applied to the two UAVs will demonstrate that a constant heading command to the follower over any time step *cannot* maintain the relative clock angle constant, even if the target has a constant speed. This is because the leader's heading is not constant. This naturally leads to an optimization to find the heading command to minimize the change in relative clock angle over the time step.

4.2.4 Optimization Problem

Let time now be given as $t = T$. Given a prediction of the target's future position (using Equations [4.5]) and the leader's future position (using Equations [4.6]) from time future

time ($t = T$) to some future time ($t = T + t_f$).

Determine ψ_c where ψ_c is a member of the solution set that that minimizes

$$\sum_{t=T}^{T+t_f} (\Psi_c - (\Psi_L(t) - \Psi_F(t)))^2$$

such that

$$\begin{aligned} \begin{pmatrix} \tilde{x}_F(t+k) \\ \tilde{y}_F(t+k) \end{pmatrix} &= \begin{pmatrix} \tilde{x}_F(t) \\ \tilde{y}_F(t) \end{pmatrix} + kV_o \begin{pmatrix} \cos\psi_c - v_{xtr} \\ \sin\psi_c - v_{ytr} \end{pmatrix} \\ tan\Psi_F(t) &= \frac{\tilde{y_F}(t)}{\tilde{x_F}(t)} \qquad t = T, T + t_f/N, \ldots T + t_f \end{aligned} \tag{4.10}$$

where N is some fixed, integer number of steps forward in time. Note that t_f, the prediction horizon time is specified.

The control input to the 'follower' dynamics at time $t = T + t_f$ is ψ_c.

4.2.5 Alternate Optimization Problem

By restricting the above minimization problem to a relative command clock angle of less than 90 degrees, the objective function can be recast as:

$$\sum_{t=T}^{T+t_f} \left[\tan(\Psi_c) - \tan\left(\Psi_L(t) - \Psi_F(t)\right)\right]^2 \tag{4.11}$$

Using the identity:

$$\tan(s - t) = \frac{\tan s - \tan t}{1 + \tan s \tan t}$$

and the definitions for $\psi_F(t)$ and $\psi_L(t)$, the objective function becomes:

$$\begin{aligned} J &= \sum_{t=T}^{T+t_f} \left[\tan(\Psi_c) - \frac{\frac{\tilde{y}_L(t)}{\tilde{x}_L(t)} - \frac{\tilde{y}_F(t)}{\tilde{x}_F(t)}}{1 + \frac{\tilde{y}_L(t)}{\tilde{x}_L(t)}\frac{\tilde{y}_F(t)}{\tilde{x}_F(t)}}\right]^2 \\ &= \sum_{k=T}^{T+t_f} \left[\tan(\Psi_c) - \frac{\tilde{y}_L(t)\tilde{x}_F(t) - \tilde{y}_F(t)\tilde{x}_L(t)}{\tilde{x}_L(t)\tilde{x}_F(t) + \tilde{y}_L(t)\tilde{y}_F(t)}\right]^2 \end{aligned} \tag{4.12}$$

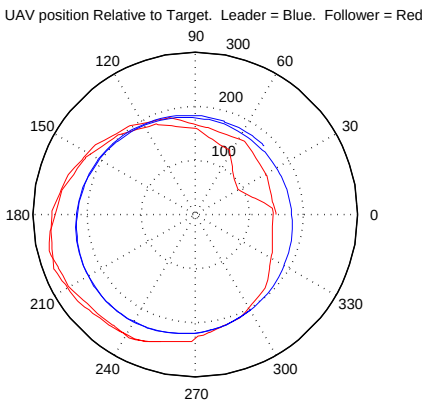


Figure 4.10: UAV Position Relative to Tgt

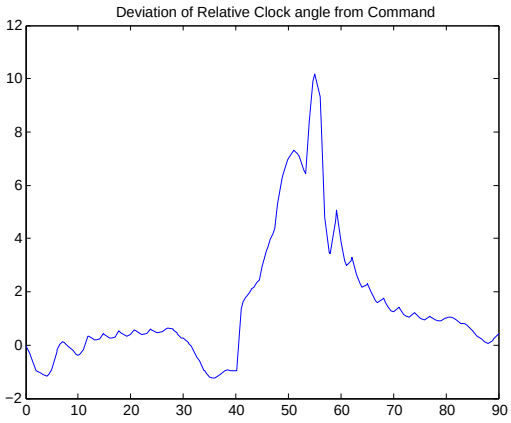


Figure 4.11: Deviation in Clock Angle From Command

So that the optimization problem is determine θ_c, where θ_c is a member of the solution set that that minimizes J in Equation (4.12) such that

$$\begin{pmatrix} \tilde{x}_F(t+k) \\ \tilde{y}_F(t+k) \end{pmatrix} = \begin{pmatrix} \tilde{x}_F(t) \\ \tilde{y}_F(t) \end{pmatrix} + kV_o \begin{pmatrix} \cos\psi_c - v_{xtr} \\ \sin\psi_c - v_{ytr} \end{pmatrix}$$
$$t = T, T + t_f/N, \ldots T + t_f \tag{4.13}$$

4.2.6 Non-Real Time Simulation

The MATLAB© function "fmincon"was used on the "Alternate Optimization Problem." See Appendix [A.2.2]. Note that the leader and follower are assumed to follow the same dynamics as in the model prediction, i.e. the models are assumed perfect. The problem is to minimize $|\dot{\psi}_L(t) - \dot{\psi}_F(t)|$ so as to maintain the relative clock angle constant. Initially, the relative clock angle is set to equal the command relative clock angle.

Figures[4.10 - 4.11] result. Note that the deviation remain relatively small until time 30. The command calculated for this point, started when the follower had a clock angle of about 225^o while the leader had a clock angle of 270^o. At this point, the leader has a lesser relative velocity (in fact its lowest). The optimization thus attempts to slow down the follower in anticipation of the divergence in relative speed leading to a divergence in relative clock angle. Indeed, the clock angle still decreases. In the next time step, the situation has

reversed and the follower has the least relative speed. All the optimization can do is reduce the radius, which happens, but not before the relative speed difference leads to a significant increase in clock angle difference.

The command does attempt to significantly reduce the radius at this point, but it saturates the *Angle Saturation Block*. It is not until the 'Follower' passes $zero-degree-clock-angle$ that a significant stride towards reducing the clock angle divergence can be made, and is, eventually resettling to near zero.

Chapter 5

SUMMARY

This chapter summarizes the goals of the proposed research and presents a timeline for completion.

5.1 Summary of Proposed Contributions

The following list summarizes the contribution for the proposed research:

- Robust solution to the cooperative tracking problem
- In depth investigation as the the utility of multiple methods both as applied to the cooperative tracking problem and as a generalization to the problem of autonomous cooperation
- Development of a consistent, real-time platform for testing cooperative tracking algorithms
- Study of the effect of signal drop-out (communication loss) on coordination
- Merge tactical problem of tracking with strategic planning problem currently implemented in ECoPS.
- Use ECoPS to consider radar exposure minimization while cooperatively tracking a hostile target.

5.2 Publication - Pending

Wise, R. and Rysdyk, R. "UAV Coordination for Autonomous Target Tracking", AIAA Guidance, Navigation, and Control Conference and Exhibit, August 21-24 2006,Keystone, CO

5.3 Research Time Line

Tasks	2006						
	JUN	JUL	AUG	SEP	OCT	NOV	DEC
General Exam	◆						
ECoPS Radar Avoidance							
Methods 1, 2, 4 Real-Time							
ECoPS Coordination							
Communication Dropout							

Tasks	2007				
	JAN	FEB	MAR	APR	MAY
ECoPS Merge Strategic and Tactical					
Dissertation/Final Exam					

Figure 5.1: Research Time Line

BIBLIOGRAPHY

[1] A.P.Press. Aerial drones being used for civilian applications. http://kvoa.com/Global/story.asp?S=4860789&nav=menu216_2, May 2006.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. reprinted 2005.

[3] Mark E. Campbell and Jarurat Ousingsawat. On-line estimation and path planning for multiple vehicles in an uncertain environment. In *AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, August 2002.

[4] Brian Capozzi and Juris Vagners. Evolving (semi)-autonomous vehicles. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. IEEE Publications, 2001.

[5] Brian Capozzi and Juris Vagners. An evolution-based alternative for autonomous motion planning. In *Proceedings of the 1st AIAA Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference*. 2002.

[6] K. Misovec et al. Low observable nonlinear trajectory generation for unmanned air vehicles. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 3103 – 3110. IEEE Publications, Piscataway, NJ, 2003.

[7] David B. Fogel. *Evolutionary Computation - Toward a New Philosophy of Machine Intelligence*. IEEE Press, 2 edition, 2000.

[8] Eric W. Frew and Dale Lawrence. Cooperative stand-off tracking of moving targets by a team of autonomous aircraft. In *AIAA Guidance, Navigation, and Control Conference*, San Francisco, CA, August 2005.

[9] Dan Klein. Controlled collective motion for multivehicle trajectory tracking. Master's Thesis - University of Washington, 2005.

[10] Dan Klein and Kristi Morgansen. Controlled collective motion for trajectory tracking. In *(Submitted to) American Control Conference*, 2006.

[11] Dale A. Lawrence. Lyapunov fields for uav flock coordination. In *2nd AIAA Unmanned Unlimited Systems, Technologies and Operations Conference*, San Diego, CA, September 2003.

[12] Frank W. Moore. Radar cross-section reduction via route planning and intelligent control. *IEEE Transactions on Control Systems Technology*, 10(5):696 – 700, September 2002.

[13] Office of the Secretary of Defense. Unmanned aircraft systems roadmap 2005 - 2030. Technical report, United States Department of Defense, 2005.

[14] Semyon M. Keerkov Pierre T. Kabamba and Frederick H. Zeitz III. Optimal path planning for unmanned combat aerial vehicles to defeat radar tracking. *Journal of Guidance, Control, and Dynamics*, 29(2):279 – 288, March – April 2006.

[15] Anawat Pongpunwattana. Real-time planning for teams of autonomous vehicles in dynamic uncertain environments. Doctoral Thesis - University of Washington, 2004.

[16] Anawat Pongpunwattana and Rolf Rysdyk. Real-time planning for multiple autonomous vehicles in dynamic uncertain environments. *Journal of Aerospace Computing, Information, and Control*, 1:580 – 604, December 2004.

[17] David Rathburn and Brian Capozzi. Evolutionary approaches to path planning through uncertain environments. In *Proceedings of the 7th Internation Joint Conference on Artificial Intelligence*. 2001.

[18] Rolf Rysdyk. UAV path following for constant line of sight. In *2nd AIAA Unmanned Unlimited Systems, Technologies and Operations Conference*, San Diego, CA, September 2003.

[19] Branko Soucek and the IRIS Group. *Dynamic, Genetic, and Chaotic Programming*. John Wiley & Sons, 1992.

[20] Michael D. Vose. *The Simple Genetic Algorithm*. MIT Press, 1999.

[21] Steve Wegener. Uav over-the-horizon disaster management demonstration projects. NASA Ames Research Center, 2001.

Appendix A

SIMULATIONS SETUPS

This chapter details the simulation set-ups used for comparison of performance.

A.1 Setup

A.1.1 Input/Output Block Diagram

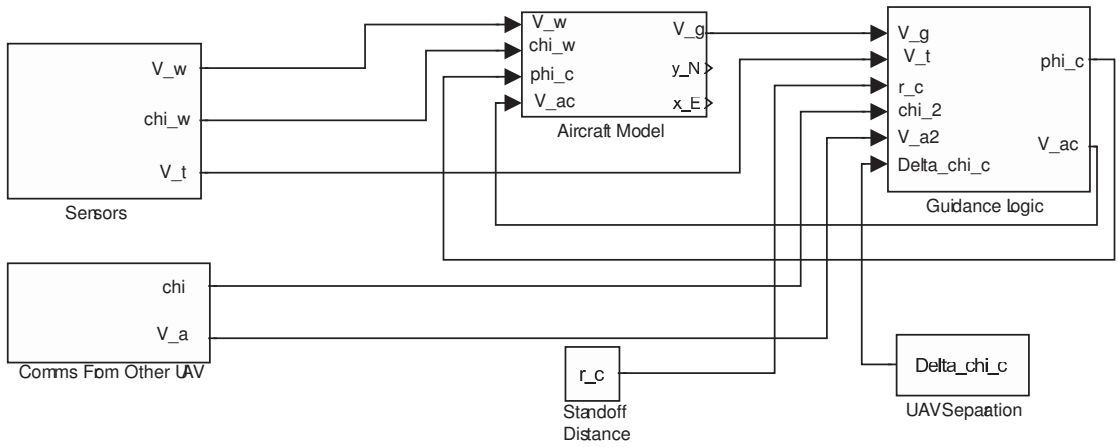


Figure A.1: Information Flow To and From the Guidance Logic

Figure [A.1] is a diagram of inputs and outputs relating the control logic to the UAV motion. Note that the the control logic will take in (at least) UAV and target ground velocity (or position) and orbit parameters (radius and UAV clock angle separation) and output a course rate command and velocity command. Sensed wind information may also be used by the controller (not shown). Also note the necessity of vehicle communication as the speed and clock angle of both UAVs are inputs.

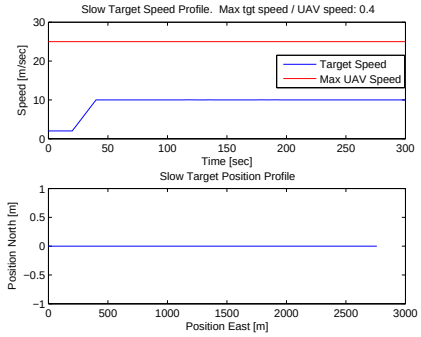


Figure A.2: Slow Target

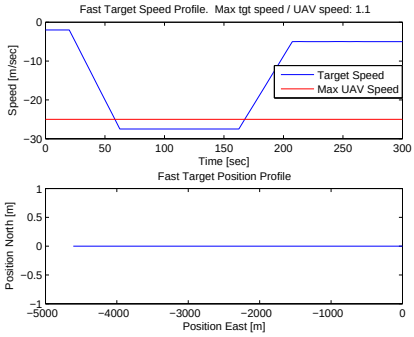


Figure A.3: Fast Target

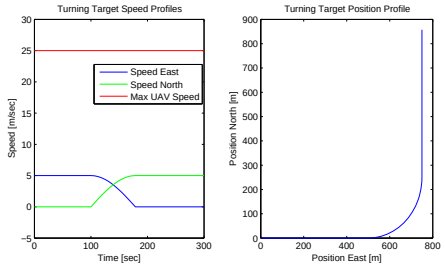


Figure A.4: Turning Target

A.1.2 Comparative Simulations

For all simulations, two UAVs will be required to cooperate. The following gives a list of target motion profiles for the UAVs to track:

1. No Wind, No Target Motion (control)
2. Constant wind, Wind Speed:UAV Speed = 0.2, Direction: from West, No Target Motion
3. No Wind, Slow Target Profile in Figure[A.2]
4. No Wind, Fast Target Profile in Figure[A.3]
5. Wind Speed:UAV Speed = 0.2, Wind Direction with Target, Slow Target Profile as in Figure [A.2]
6. No Wind, Target Turning in Figure [A.4]

A.2 Non - Real Time

A.2.1 Methods 1 - 3

The set-up for non - real time simply connects two Simulink© models, as in Figure [A.1], together. Each UAV receives identical wind and target data. The communication of clock angle and speed is continuous and unbroken. The problem is simulated at a time scale determined by the Simulink© solver, which in general is not real time. See Figure [A.5].

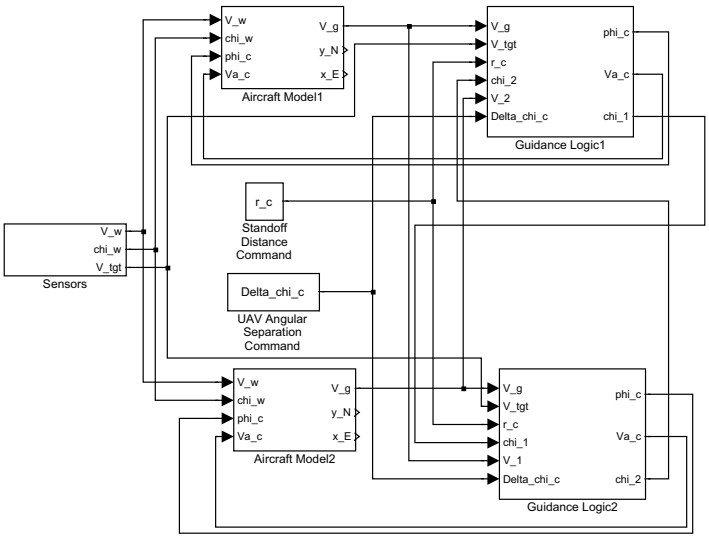


Figure A.5: Information Flow To and Between UAVs, Not Real Time

A.2.2 Convex Optimization Method

Figure [A.6] is a block diagram of the Simulink© setup for simulating the convex optimization method. The convex optimization problem is solved in the *feedforward* block. The block is controlled by a trigger. When the trigger fires, the current states of the leader, follower, and target are inputted as initial conditions for the convex optimization problem. The solution is used by the follower. In between trigger events, the optimization block output is a constant, equal to the last triggered solution.

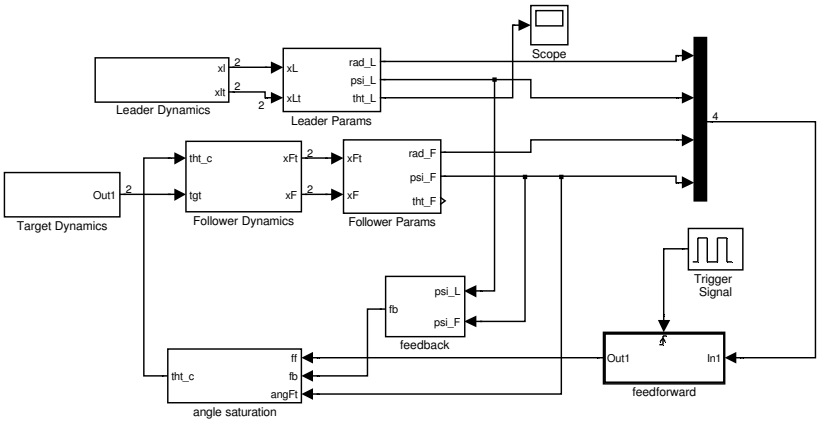


Figure A.6: Simulation Block Diagram for Convex Optimization Method

Note also that feedback is used. In this case, proportional to the divergence in the relative clock angle from the command relative clock angle. The feedback is to keep the divergence of relative clock angle small between subsequent calculations of the heading command. This is important as the feedforward control assumes that the relative clock angle is exactly equal to the command at the beginning of the calculation. This also helps to set $|\dot{\psi}_L(t) - \dot{\psi}_F(t)|$ exactly equal to zero.

An *Angle Saturation Block* is used to add position dependent limits so that the feedback does not cause the command to result in the UAV turning too quickly, or turning back on its previous path.

A.3 Real Time

The set-up for real time employs the networking *OPC toolbox* of Simulink©. OPC (Open Platform Control) utilizes Windows© network protocoling to set up a client on a network data server. This allows Simulink© to read (write) data to(from) the network by mapping the data (the map is determined a priori) to be read from (written to) a specific data member on the data server. Further, Simulink© will *enforce* real time. For example, if the simulation step takes t_s seconds, the simulation will not continue on to the next step until the computer clock time has elapsed t_s seconds as well. This does not preclude the

real possibility that the simulation runs *slower* than real time. This might happen due to computational load, or inefficiency, or network delays in retrieving server data. See Figure [A.7]. Note that the "environment" and the two UAV simulations are running on physically different computers.

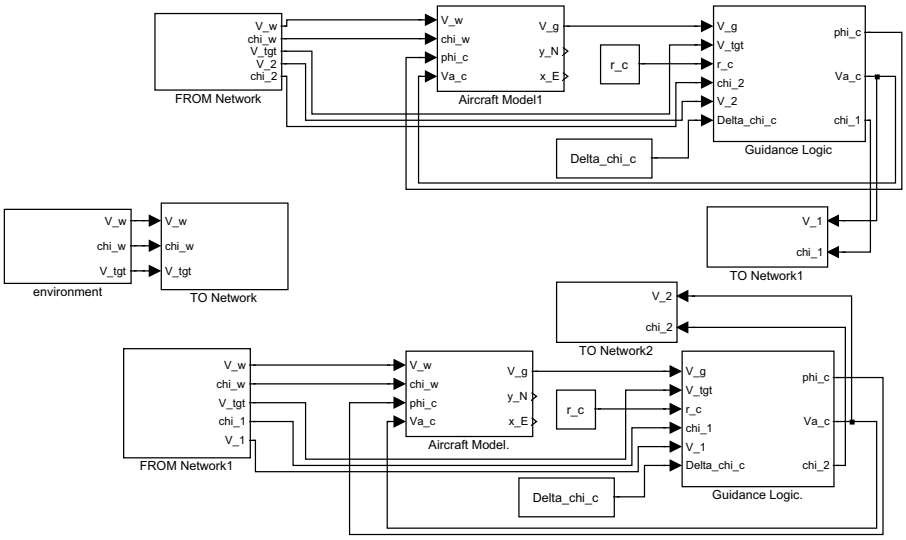


Figure A.7: Information Flow To and Between UAVs, Real Time