An Automata Theory Method for the Analysis of Unicycle Pursuit Problems

David Dovrat

Technion - Computer Science Department - Ph.D. Thesis PHD-2022-05 - 2022

An Automata Theory Method for the Analysis of Unicycle Pursuit Problems

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

David Dovrat

Submitted to the Senate
of the Technion — Israel Institute of Technology
Shvat 5782 Haifa January 2022

The research thesis was done under the supervision of Prof. Alfred M. Bruckstein, in the Faculty of Computer Science.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's doctoral research period, the most up-to-date versions of which being:

David Dovrat and Alfred M. Bruckstein. On gathering and control of unicycle a(ge)nts with crude sensing capabilities. *IEEE Intelligent Systems*, 32(6):40–46, 2017.

David Dovrat and Alfred M. Bruckstein. Antalate—a multi-agent autonomy framework. *Frontiers in Robotics and AI*, 8:264, 2021.

David Dovrat, Twinkle Tripathy, and Alfred M. Bruckstein. On tracking and capture in proportional-control bearing-only unicycle pursuit. *IEEE Control Systems Letters*, 6:2132–2137, 2022.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Tables

List of Figures

Abstract

The Pursuit Problem depicts a scenario where a target is chased by an agent, whose movement is prescribed by some defined policy. Examples of what can be regarded as solutions to the pursuit problem include the shape of the agent's trajectory, whether the agent ultimately captures the target, and the circumstances of the capture, including the time required for capture to be achieved.

The Unicycle Model is a popular simplification used to describe the kinematics of complex vehicular systems. An agent modeled as a unicycle has three degrees of freedom: the location on the plane, and the orientation of the agent. The agent is constrained to move only in the direction of the its orientation, and only two input signals are available to control the model: steering and speed.

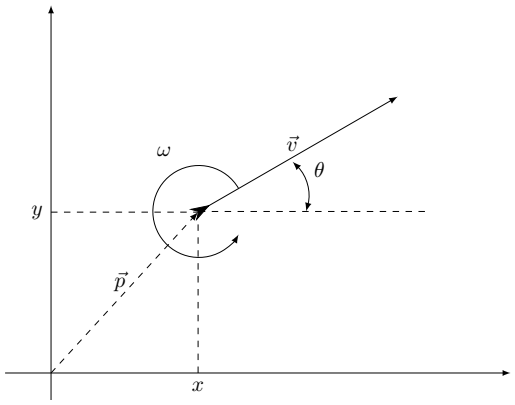
A Unicycle Pursuit Problem is a pursuit problem where the pursuing agent is modeled as a unicycle.

This thesis describes a method used for solving unicycle pursuit problems by mapping the properties of the pursuit to a directed graph, which can then be regarded as a state machine that describes the evolution of these properties in an abstract way.

The thesis details the analysis of two particular unicycle pursuit problems, to demonstrate results that were achieved by studying the traits and structure of the corresponding finite state machines generated by the method described here. In addition, we present a software framework used to implement a multi-agent version of one of these unicycle pursuit problems using Unmanned Aerial Vehicles (UAVs). We call this framework *AntAlate*. AntAlate allows software application developers to focus on their algorithms by abstracting away the UAV platform, enforcing safety measures, and providing a versatile interface for algorithm interaction.

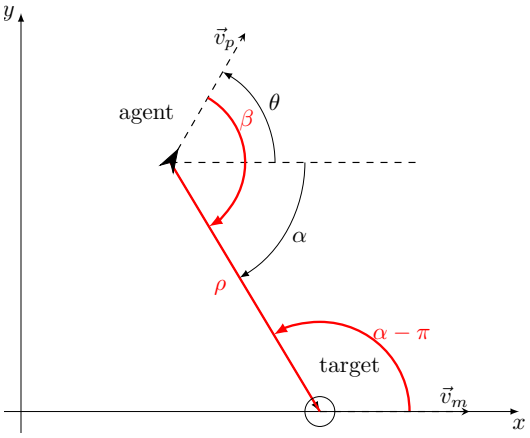
Abbreviations and Notations

PCBOUP	:	Proportional-Control Bearing-Only Unicycle Pursuit
UCSLSV	:	Unicycles with Crude Sensing over a Limited Sector of Visibility
DFSM	:	Deterministic Finite State Machine
QED	:	Quod Erat Demonstrandum (what was to be shown)
UAV	:	Unmanned Aerial Vehicle
MARS	:	Multi-Agent Robotic System
ROS	:	Robot Operating System
R&D	:	Research and Development
MC	:	Mission Control
HLC	:	High Level Control
OSC	:	Operator Station Client
BMA	:	Behavior Module Arbiter
LLC	:	Low Level Control
HTTP	:	Hypertext Transfer Protocol
TCP/IP	:	Transmission Control Protocol / Internet Protocol
API	:	Application Programming Interface
SITL	:	Software in the Loop

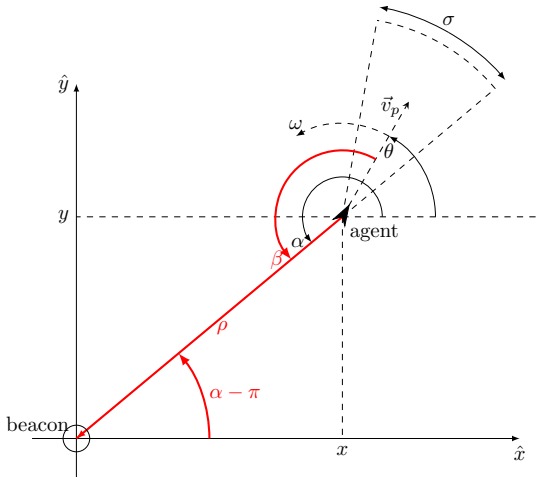


(a) The Unicycle Model.

v	:	The agent's (and target's) speed.
v_p	:	The agent's speed.
v_m	:	The target's speed.
$\vec{v}$	:	The agent's velocity vector.
$\vec{v}_p$	:	The agent's velocity vector.
$\vec{v}_m$	:	The target's velocity vector.
$\vec{p} = (x, y)^T$	:	The agent's vector of coordinates on the plane.
θ	:	The agent's orientation in the global frame of reference.
$\omega = \dot{\theta}$	:	The agent's turning rate.
α	:	The angle from the agent towards its target, in the global frame of reference.
$\beta = \alpha - \theta$	:	The angle from the agent towards its target, in the agent's frame.
ρ	:	The distance between the agent and its target.
σ	:	The central angle of the agent's visibility sector.
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	:	A directed graph.
$\mathcal{V}$	:	The vertex set of $\mathcal{G}$.
$\mathcal{E}$	:	The edge set of $\mathcal{G}$.
t_0	:	The moment in time at which a process begins, e.g. state entry.
t_n	:	A moment when a specific measurable or calculated event, indexed n, occurs.
t_*	:	An unspecified moment, usually during a process, e.g. $t_0 < t_* < t_1$.
τ_u	:	The *maximal dwell time* of state u.



(a) Notations used in the analysis of the problem presented in Chapter 3.



(b) Notations used in the analysis of the problem presented in Chapter 4.

Chapter 1

Introduction

Pursuit problems are centuries old, yet still are the subject of much academic activity [Shn98, Nah07, SG07, Shi07]. Form Pierre Bouguer, who is credited with formulating the problem in the 18th century (as a pirate ship trying to intercept a merchantman), to mathematicians, control theorists and roboticists today, pursuit has fascinated generations of thinkers. This might be due to the simplicity and elegance of the problem statement, contrasted with the sophistication of the possible solutions and methods of reaching them. This is coupled with the endless variation options in the dynamics of the pursuing agent, its controller, its maneuverability compared to that of the target, the sensing and computing capabilities required of the pursuing agent's hardware, etc.

Pursuit still poses some interesting problems that the current literature has not fully addressed. The unicycle-model variant of pursuit, for instance, has yet to be completely solved, to the best of our knowledge. The unicycle model is a popular and useful model for representing complex robotic and vehicular systems and their behaviors in various tasks[DB17, TS17], but it is an underactuated system, having three degrees of freedom (the unicycle's orientation, and two axes of its location on the plain) and only two actuators (forward speed and turn angle rate), which makes its control challenging.

The lack of a full analytic solution to all unicycle pursuit problems does not imply a lack of achievements towards this goal. Medagoda and Gibbens[MG10], for instance, propose a waypoint-following pursuit strategy wherein a virtual target moves linearly between path waypoints at a speed depending on parameters like the pursuer-target relative distance and the pursuer's speed. Tracking is achieved using an LQR based controller to pursue the virtual, linear agent. Following this work, Ratnoo et al.[RHGS15] present necessary conditions to achieve tracking (or tail chase) when the target is moving in a straight line. Resulting from a pure-pursuit guidance law that requires acquiring both bearing and distance to the target, their control policy aligns the pursuer with the target. Other versions of pure-pursuit and several other approaches have also been explored to address the tracking problem[ET 16, ST10, OAE16, BBR06, DLN21, JTWS19]. These versions of unicycle pursuit rely on sophisticated controllers, which in turn demand additional sensing and computing capabilities for their implementa-

tion and generally, also superior maneuverability of the pursuing agent compared to the target. Belkhouche and Belkhouche[BB04], for example, study capture conditions when the pursuer moves at a speed higher than that of the target.

This thesis explores a novel method to solve unicycle pursuit-like problems; in Chapter 3, we revisit the classic problem of pursuit when the motion of the pursuing agent is governed by unicycle kinematics. Instead of enhancing the pursuer's capabilities, we consider the case of an agent restricted to have a constant speed equal to that of the target. This hindrance has a double effect; by keeping the speed constant we underactuate the pursuer even more than it inherently is, since only the steering control can be used for the pursuit; by matching the speed to that of the target we keep the pursuer just barely capable of tracking it even in the case where the initial conditions put the pursuer in a good position to track the target. We further constrain our pursuing agent to only acquire information about the bearing towards the target, and act upon this information in a memory-less fashion; i.e. the pursuer does not remember the previous bearing readings, and has to control its parameters based on the instantaneous reading of the bearing only.

Our work contributes to the existing literature by guaranteeing convergence to either tracking or capture, under conditions we analyze in detail, from arbitrary initial conditions on the pursuer and the target using Lyapunov stability analysis[Lib03] in Section 3.2, and then again using our method in Section 3.3. We demonstrate, using our method, how we were able to achieve stronger results regarding the final distance between the agent and its target than was possible using the Lyapunov analysis; and in Section 3.4 we make predictions about which initial conditions may result in the agent capturing the target.

Though, admittedly, following the steps in our algorithm, presented in Chapter 2, is time consuming and sometimes difficult, the algorithm grants a way to attain a solution, arguably in a more straightforward manner than trying to guess a Lyapunov function. In Chapter 4 we revisit the *homing* problem with unicycle agents with crude sensing capabilities, and show how using our method we were able to swiftly re-solve the problem. In Chapter 5 we go into detail about the philosophy and function of a software framework we created, dubbed AntAlate, before concluding in future work and final remarks.

Chapter 2

Method

The adoption of ideas from the realm of the computer sciences and applying them to other fields, especially in the control systems discipline, is perhaps the foundation of robotics. Describing and analyzing complex systems can be overwhelming for the human mind, unless abstracted into generalized, qualitative, discrete, preferably linear systems which we can better understand. Once better understood, the same abstractions could be used for the design of automated management and control systems to govern the original, complex systems.

Examples of this notion could be found in the analysis of biological networks, as presented by Glass[Gla75]; Chaos theory, as illustrated by Moore[Moo90]; and of-course dynamical systems, as shown by Beer[Bee95] and Stursberg et al[SKHP97].

This chapter introduces our method of abstracting the agent-target relationship in unicycle pursuit-type problems into an automaton.

2.1 Dynamics of Pursuit

Consider the following set of kinematic equations describing the evolution of the agent's state,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2.1}$$

where $p^T = (x, y)^T$ is the agent's position, θ is the agent's orientation, $v > 0$ its speed, and ω its turning rate.

In general pursuit problems, the relationship between the agent and the target it pursues can be described by the distance between them ρ, the bearing angle towards the target as measured from the agent's frame β, and the bearing angle towards the agent from the target's frame, which is equivalent to $\alpha - \pi$, where α is the azimuth from the agent to the target in the global frame, see for example Figure 2.2, where we

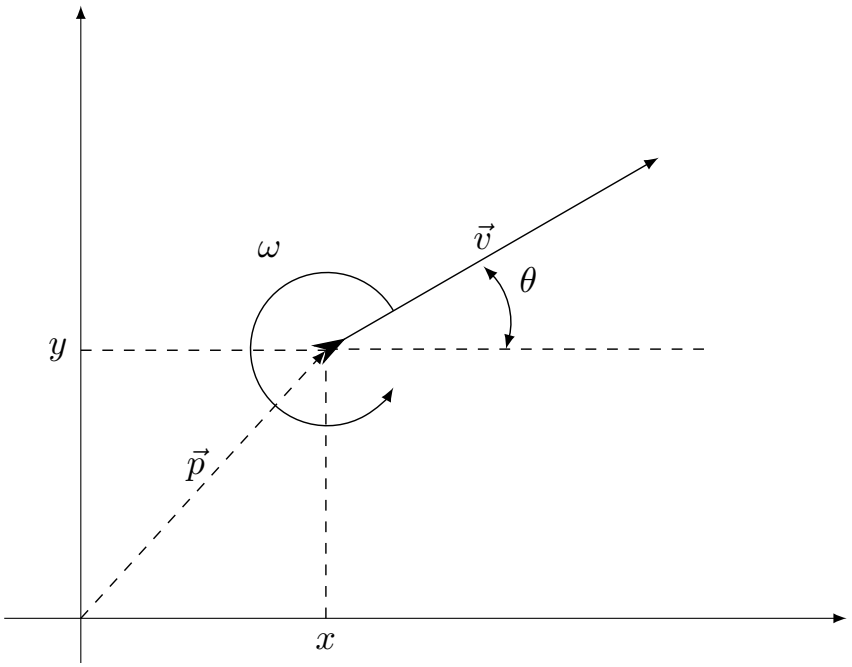
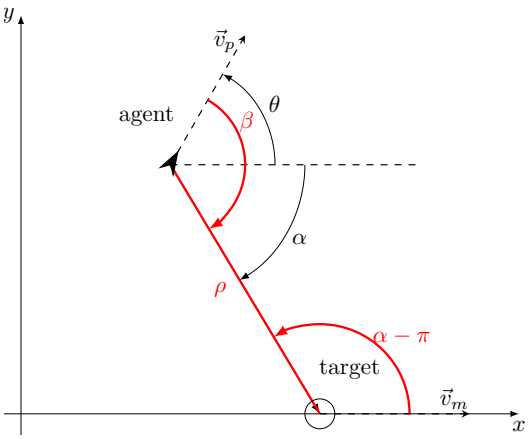


Figure 2.1: The Unicycle Model

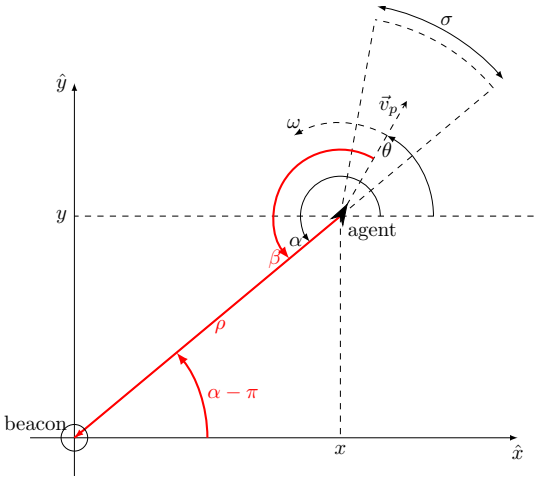
can also see that

$$\beta = \alpha - \theta. \tag{2.2}$$

by definition.



(a) A typical configuration of the problem analyzed in Chapter 3.



(b) A typical configuration of the problem analyzed in Chapter 4.

Figure 2.2: Examples of Unicycle Pursuit Problems and the relationship between the target and pursuing agent expressed by (ρ, α, β).

Looking at the target's motion from a frame attached to the agent, with its real axis pointing in the direction of the target's velocity (or any other global, arbitrary, yet stationary axis), we arrive at the complex representation of the position of the target

as observed by the agent

$$p_{target}^{agent} = p_{ta} = \rho e^{i\alpha},$$

where $\rho = |p_{ta}|$ is the distance form target to agent. Taking the time derivative

$$\frac{d}{dt}p_{ta} = \frac{d}{dt}\left(\rho e^{i\alpha}\right) = \left(\dot{\rho} + i\rho\dot{\alpha}\right)e^{i\alpha} = \left(\frac{\dot{\rho}}{\rho} + i\dot{\alpha}\right)p_{ta},$$

and comparing it with the target's velocity as observed by the agent

$$\left(\frac{\dot{\rho}}{\rho} + i\dot{\alpha}\right)\rho e^{i\alpha} = v_m e^0 - v_p e^{i\theta}$$

$$\Downarrow$$

$$\dot{\rho} + i\rho\dot{\alpha} = v_m e^{i(0-\alpha)} - v_p e^{i(\theta-\alpha)} = v_p\left(\frac{v_m}{v_p}e^{-i\alpha} - e^{-i\beta}\right),$$

leaves us with an expression for the rate of rotation

$$\rho\dot{\alpha} = v_p\left(\frac{v_m}{v_p}\sin(-\alpha) - \sin(-\beta)\right)$$

$$\Downarrow$$

$$\dot{\alpha} = \frac{v_p}{\rho}\left(\sin(\beta) - \frac{v_m}{v_p}\sin(\alpha)\right), \tag{2.3}$$

and the rate of change in distance

$$\dot{\rho} = v_p\left(\frac{v_m}{v_p}\cos(\alpha) - \cos(\beta)\right). \tag{2.4}$$

From (2.1), (2.2), and (2.3),

$$\dot{\beta} = \dot{\alpha} - \dot{\theta} = \frac{v_p}{\rho}\left(\sin(\beta) - \frac{v_m}{v_p}\sin(\alpha)\right) - \omega. \tag{2.5}$$

Now that we have expressed the dynamics of pursuit in terms of the connection between the target and the pursuing agent, we can analyze pursuit systems by categorizing this relationship into equivalence classes we call system states.

2.2 Automaton Generating Algorithm

The algorithm we use to solve Unicycle Pursuit problems (Algorithm 2.1) has two major steps: the first is to identify the different system states, and the second is to identify the transitions between the states and enforce a *maximal dwell time* for each state, i.e. if entering state u at t_0, then a transition must occur by $t_0 + \tau_u$ at the latest.

In order to generate the desired state machine, we start with an empty graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We consider the equations obtained in the previous section (Equations 2.4, 2.3, and 2.5),

$$\begin{cases} \dot{\rho} &= v_p \left(\frac{v_m}{v_p} \cos(\alpha) - \cos(\beta) \right) \\ \dot{\alpha} &= \frac{v_p}{\rho} \left(\sin(\beta) - \frac{v_m}{v_p} \sin(\alpha) \right) \\ \dot{\beta} &= \frac{v_p}{\rho} \left(\sin(\beta) - \frac{v_m}{v_p} \sin(\alpha) \right) - \omega, \end{cases}$$

describing the evolution of ρ, α, and β, corresponding to some defined control inputs v and ω. In the cases we analyze in the next chapters, v is held constant and ω is a function of ρ, α, and β.

With these expressions for the rates of change, we can map disjoint regions in (ρ, α, β) space to separate states in $\mathcal{V}$, characterized by conditions on $\left(\dot{\rho}, \dot{\alpha}, \dot{\beta}\right)$. For example, in Chapter 3, state $A \in \mathcal{V}$ is defined by the condition $0 \leq \alpha < \beta < \frac{\pi}{2}$, corresponding to the behavior $\dot{\rho} > 0$ and $\dot{\alpha} > 0$. Similarly, state $B \in \mathcal{V}$ is defined by the condition $0 < \beta \leq \alpha < \frac{\pi}{2}$, corresponding to the behavior $\dot{\rho} \leq 0$ and $\dot{\alpha} \leq 0$. In Chapter 4, the states denoted by capital letters are defined by conditions on ρ, α, and β that correspond to $\dot{\alpha} - \dot{\beta} = \frac{v}{R}$, i.e. $-\frac{\sigma}{2} < \beta < \frac{\sigma}{2}$. While the exact conditions depend on the explicit form of the equations for $\left(\dot{\rho}, \dot{\alpha}, \dot{\beta}\right)$, derived from the control inputs defined for the specific problem at hand, the general algorithm step is to observe these equations and deduce the characteristic conditions from them.

Once we have defined all of the system states, and populated the vertex set $\mathcal{V}$, we must find an upper bound to the dwell time for each state, to guarantee that the state exits in finite time. This can be achieved by calculating the bounds on the defining conditions for each state. If there are no transitions out of the state, the state is a *sink*, like the capture state in Chapter 3; if the bounds evolve to an inevitable transition to another state, such as the transition in Chapter 3 from State A to State B, then we readily have the longest possible time until transition occurs, i.e. the maximal dwell time.

Yet if the maximal dwell time is infinite, due for instance to an asymptotic decay of the state's defining condition towards its point of invalidation, such as in State C in Chapter 3, then we can manufacture a transition by arbitrarily choosing a convenient dwell time, τ_u, and calculate the bounds for $\left(\rho_{(t_0+\tau_u)}, \alpha_{(t_0+\tau_u)}, \beta_{(t_0+\tau_u)}\right)$; by doing so we create an exit condition from the state to itself. We can later use this type of timed self-loop to track the evolution of the bounds on ρ, α, and β, in cases where it is unclear whether the system evolves out of the state in question, as we do in the concluding theorems of both of the following chapters.

After completing the vertex-set $\mathcal{V}$ and finding an expiration time for every non-sink state in it by finding all the cases where the conditions that define the state cease to hold, and after adding the transition to the other state to the edge-set $\mathcal{E}$ for each of these cases, along with all the manufactured self-loops, we have finally generated a timed automaton, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, that faithfully describes the evolution of the dynamical

system. The resulting Deterministic State Machine (DSM) can now be used to make observations on the behavior of the dynamical system it describes.

Capitalizing on the structure of the generated automaton, we analyze the different paths and cycles in $\mathcal{G}$ to find the final regions in (ρ, α, β) space the dynamical system reaches with time. In the next two chapters we go into detail on how we used Algorithm 2.1 to analyze two pursuit problems and reach novel results.

Algorithm 2.1 A Unicycle Pursuit Problem becomes a Deterministic Finite State Machine

initialize the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = (\emptyset, \emptyset)$;
categorize the relationship between the target and agent according to disjoint regions in (ρ, α, β) space.
for each resulting state u **do**
  add u to $\mathcal{V}$.
end for
for each $u \in \mathcal{V}$ **do**
  calculate the bounds on the defining conditions of u.
  if at least one condition expires by time $t_0 + t_*$ **then**
    the maximal $\tau_u = t_* - t_0$ is the maximal dwell time for state u.
  else
    arbitrarily select a maximal dwell time τ_u for state u.
    calculate bounds for $\left(\rho_{(t_0+\tau_u)}, \alpha_{(t_0+\tau_u)}, \beta_{(t_0+\tau_u)}\right)$, and add $t = t_0 + \tau_u$ as an exit condition from state u to itself.
  end if
  for each exit condition e from state u to state w **do**
    add e to $\mathcal{E}$.
  end for
end for
return $\mathcal{G}$.

Chapter 3

Tracking and Capture in Proportional-Control Bearing-Only Unicycle Pursuit

This chapter shows a Lyapunov function[LaS60] we found when studying the Proportional - Control Bearing - Only Unicycle Pursuit (PCBOUP) problem, formally defined below in Section 3.1.2, which proves that α and β asymptotically go to zero, i.e. $\lim_{t\to\infty} \max\{|\beta(t)|, |\alpha(t)|\} = 0$, ensuring that the pursuing agent ultimately gets closer and closer to the target's trajectory. The analysis is presented in Section 3.2, yet bringing α and β asymptotically to zero is not enough to ensure that the target does not slip away into infinity, and therefore does not constitute *tracking*, which also requires the agent to maintain a finite final distance from the target.

However, using our method presented in Chapter 2, we were able to show that the distance between agent and target remains bounded forever, as shown in Section 3.3. We were also able to find a region on the plane that contains all initial conditions that may result in the pursuing agent capturing the target, as we present in Section 3.4.

3.1 Preliminaries

Before continuing to the main results, we start by laying some foundations.

3.1.1 Pure Pursuit

The classic pure pursuit problem involves a target (a merchantman) moving in a straight line with kinematics

$$p_m^T = (0, vt), \tag{3.1}$$

and a pursuing agent (a pirate ship) with kinematics

$$\dot{p}_p = v_p \frac{p_m - p_p}{|p_m - p_p|} \tag{3.2}$$

where p_m, p_p are the target's and agent's locations, respectively, v is the target's speed and v_p is the pursuing agent's speed. The problem is to find the *curve of pursuit* $y(x)$ given $p_p^T = (x(t), y(t))$ and $(x(0), y(0)) = (x_0, 0)$. Nahin dedicated a chapter in his highly recommended book[Nah07] to the formulation, history and solution to this classic problem.

3.1.2 Proportional-Control Bearing-Only Unicycle Pursuit

The Proportional-Control Bearing-Only Unicycle Pursuit (PCBOUP) problem involves an agent with constant speed unicycle kinematics, able to sense only the bearing towards the target it pursues. The agent's steering is proportional to this measured bearing-angle towards the target. The target's speed is equal to that of the chasing agent while it moves at a constant velocity.

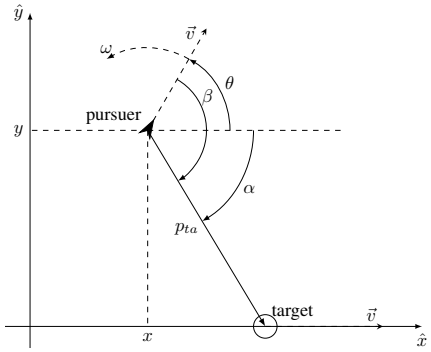


Figure 3.1: The Unicycle Pursuit Problem

This setup is similar to that of the Pure Pursuit problem (Section 3.1.1), if we take the target's trajectory to be

$$p_m^T = (vt, 0)\,, \tag{3.3}$$

causing α to be

$$\tan(\alpha) = \frac{y}{x - vt}, \tag{3.4}$$

and if we replace the pursuing agent's kinematics with Unicycle Kinematics (Equation 2.1). Since agents modeled as unicycles have an orientation, which must align with the bearing towards the target in order to mimic pure pursuit, we assign a proportional controller

$$\omega = \kappa\beta \tag{3.5}$$

where κ is a gain that amplifies the bearing angle β. Consequently, the pursuing agent applies the resulting signal ω as a steering command to its single controlled actuator, while keeping v constant.

Definition 3.1.1. the agent is *facing the target* when $|\beta| < \frac{\pi}{2}$;

Definition 3.1.2. the agent *falls behind the target* when $|\alpha| < \frac{\pi}{2}$;

Definition 3.1.3. the agent *captures* the target if the distance from the target is r_c or less, i.e. $\exists t_c \,|\rho(t_c) \leq r_c$ with $\rho(t)$ as the distance between the agent and the target;

Definition 3.1.4. the agent is said to be *tracking* the target if it reaches a trajectory which is very close to that of the target while remaining within a finite range from the target, i.e. $\exists R,\ 0 < R < \infty \mid \rho(t) < R\ ,\ \forall t$, and also given arbitrary $\varepsilon > 0$, $\exists t_\varepsilon \mid \forall t > t_\varepsilon\,,\ |\beta(t)| < \varepsilon$ and $|\alpha(t)| < \varepsilon$.

The physical meaning of the distance r_c could be the dimension of either the target, the agent, or any combination of both, while ε could be the acceptable measure of observational error of the agent's sensors.

Given the definitions above, we can now formally define the Proportional-Control Bearing-Only Unicycle Pursuit (PCBOUP) problem.

PCBOUP Problem Statement

A target with kinematics (3.3) is pursued by an agent with kinematics (2.1), (3.5), and a constant speed v equal to that of the target. Find κ_c such that if $\kappa > \kappa_c$, then either the agent captures the target or tracks it, i.e. $\exists t_c \,|\rho(t_c) \leq r_c$ or $\exists R,\ 0 < R < \infty \mid \rho(t) < R\ ,\ \forall t$, and $\forall \varepsilon > 0$, $\exists t_\varepsilon \mid \forall t > t_\varepsilon\,,\ |\alpha(t)| < \varepsilon$ and $|\beta(t)| < \varepsilon$.

3.2 Lyapunov Function Approach

In this section we perform Lyapunov stability analysis to identify a forward invariant region for α and β, and ultimately prove that given $\kappa > 2\frac{v}{r_c}$, $(\alpha, \beta) = (0, 0)$ is a stable equilibrium point in the Lyapunov sense.

The structure of this section's reasoning can be described as follows: Lemma 3.2.1 gives us a time frame after which the agent faces the target, and Corollary 3.1 emphasizes the fact that once the agent turns to face the target it always faces the target; with lemmas 3.2.2 and 3.2.3 we prove that there is a point in time at which the agent both falls behind and faces the target, if the agent has not captured the target by that time; Theorem 3.2 establishes the conclusion of all these lemmas, that the system configuration where $(\alpha, \beta) \in \left[\frac{\pi}{2}, -\frac{\pi}{2}\right] \times \left[\frac{\pi}{3}, -\frac{\pi}{3}\right]$ is forward invariant; after introducing the equilibrium configuration $(\alpha, \beta) = (0, 0)$ in Lemma 3.2.4, we conclude the section with Theorem 3.3, proving that the PCBOUP system must evolve to either capture the target or asymptotically reach the equilibrium configuration.

Lemma 3.2.1. *An agent (2.1) governed by the bearing-only control law (3.5) with $\kappa > 2\frac{v}{r_c}$, in pursuit of a target moving in a straight line (3.3), either captures the target or turns to face the target in finite time, such that $|\beta| \leq \frac{\pi}{3}$.*

Proof From (2.5) and (3.5),

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta. \tag{3.6}$$

$|\sin(\beta) - \sin(\alpha)| \leq 2$; therefore from (3.6),

$$-2\frac{v}{r_c} - \kappa\beta \leq \dot{\beta} \leq 2\frac{v}{r_c} - \kappa\beta; \tag{3.7}$$

and the bounds on β are

$$\begin{cases} \beta^+(t) &= 2\frac{v}{\kappa r_c} + \left(\beta(t_0) - 2\frac{v}{\kappa r_c}\right) e^{-\kappa(t-t_0)}; \\ \beta^-(t) &= -2\frac{v}{\kappa r_c} + \left(\beta(t_0) + 2\frac{v}{\kappa r_c}\right) e^{-\kappa(t-t_0)}; \end{cases} \tag{3.8}$$

i.e.

$$\beta^-(t) \leq \beta(t) \leq \beta^+(t).$$

If $\frac{\pi}{3} \leq \beta(t_0)$, then the upper bound on β decreases asymptotically to $\frac{2v}{\kappa r_c} < 1$, and reaches $\frac{\pi}{3} > 1$ by t_1,

$$\beta^+(t_1) = 2\frac{v}{\kappa r_c} + \left(\beta(t_0) - 2\frac{v}{\kappa r_c}\right) e^{-\kappa(t_1-t_0)} = \frac{\pi}{3}$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{1}{\kappa} \ln\left(\frac{\beta(t_0) - 2\frac{v}{\kappa r_c}}{\frac{\pi}{3} - 2\frac{v}{\kappa r_c}}\right). \tag{3.9}$$

Similarly, $\beta^-(t)$ reaches $-\frac{\pi}{3}$ if $\beta(t_0) \leq -\frac{\pi}{3}$, and if $\frac{\pi}{3} \leq |\beta(t_0)|$, then $\exists t_*$, $t_0 \leq t_* \leq t_1$, when either $r(t_*) = r_c$, or $|\beta(t_*)| \leq \frac{\pi}{3}$. ■

Corollary 3.1. *If an agent (2.1) governed by the bearing-only control law (3.5) with $\kappa > 2\frac{v}{r_c}$, in pursuit of a target moving in a straight line (3.3), faces the target, it will continue to face the target forever.*

$$\exists t_* \;\Big|\; |\beta(t_*)| \leq \frac{\pi}{3} \Rightarrow |\beta(t)| \leq \frac{\pi}{3} \;\forall t > t_*.$$

Proof β^+ and β^- from the previous proof are global bounds on $\beta(t)$, even for $|\beta| < \frac{\pi}{3}$.■

Notice that the higher the gain κ, the closer t_1 is to t_0 and the lower the actual upper bound on β is, at $\frac{2v}{\kappa r_c}$.

Lemma 3.2.2. *If an agent (2.1) governed by the bearing-only control law (3.5) with $\kappa > 2\frac{v}{r_c}$, in pursuit of a target moving in a straight line (3.3), falls behind the target, it will remain behind the target forever.*

$$\exists t_* \;\Big|\; |\alpha(t_*)| < \frac{\pi}{2} \Rightarrow |\alpha(t)| < \frac{\pi}{2} \;\forall t > t_*.$$

Proof If $\exists t_* \;\big|\; |\alpha(t_*)| < \frac{\pi}{2}$, then $x(t_*) < x_t(t_*) = vt_*$, and from (3.3) and (2.1), $\dot{x} = v\cos\theta \leq v \Rightarrow x(t) - x(t_*) \leq v\,(t - t_*) = x_t(t) - x_t(t_*) \Rightarrow x(t) + x_t(t_*) \leq x_t(t) + x(t_*)$; but $x(t_*) < x_t(t_*) \Rightarrow x(t) < x_t(t)\ \forall t \geq t_*$. ■

Lemma 3.2.3. *An agent (2.1) governed by the bearing-only control law (3.5) with $\kappa > 2\frac{v}{r_c}$, in pursuit of a target moving in a straight line (3.3), either captures the target or falls behind it in finite time.*

Proof According to Lemma 3.2.1 and Corollary 3.1, if $|\beta(t_0)| \geq \frac{\pi}{3}$, then either the target has been captured by t_1, or $|\beta(t)| < \frac{\pi}{3}\ \forall t > t_1$; and if $|\beta(t_0)| \leq \frac{\pi}{3}$, $|\beta(t)| < \frac{\pi}{3}\ \forall t > t_0$, and (2.4) becomes $\dot{\rho} \leq -v\cos(\beta) < -\frac{v}{2}$, until $|\alpha(t_*)| < \frac{\pi}{2}$. Seeing that for the case where $|\beta(t_0)| > \frac{\pi}{3}$, $t_0 \leq t \leq t_1$, $\dot{\rho} \leq v$,

$$\rho(t) < \rho^+(t) = \rho(t_0) + v\,(t_1 - t_0) - \frac{v}{2}\,(t - t_1)$$

$$= \rho(t_0) + v\left(\frac{3}{2}t_1 - t_0 - \frac{1}{2}t\right).$$

By t_2, the agent must therefore capture its target, unless $|\alpha| < \frac{\pi}{2}$ sometime before;

$$\rho^+(t_2) = r_c \Rightarrow \rho(t_0) + v\left(\frac{3}{2}t_1 - t_0 - \frac{1}{2}t_2\right) = r_c$$

$$\Downarrow$$

$$t_2 = t_0 + 2\frac{\rho(t_0) - r_c}{v} + \frac{3}{\kappa}\ln\left(\frac{\beta(t_0) - 2\frac{v}{\kappa r_c}}{\frac{\pi}{3} - 2\frac{v}{\kappa r_c}}\right). \tag{3.10}$$

Similarly, if $|\beta(t_0)| < \frac{\pi}{3}$, $\rho(t) < \rho(t_0) - \frac{v}{2}\,(t - t_0)$, and $t_2 = t_0 + 2\frac{\rho(t_0) - r_c}{v}$. In any case,

$$t_2 \leq t_0 + 2\frac{\rho(t_0) - r_c}{v} + \frac{3}{\kappa}\ln\left(\frac{\pi - 2\frac{v}{\kappa r_c}}{\frac{\pi}{3} - 2\frac{v}{\kappa r_c}}\right),$$

and if the agent has not captured the target by t_2, then

$$\exists t_*,\ t_0 \leq t_* \leq t_2 \ \Big|\ |\alpha(t_*)| < \frac{\pi}{2}\,.$$

Theorem 3.2. *An agent (2.1) governed by the bearing-only control law (3.5) with $\kappa > 2\frac{v}{r_c}$, in pursuit of a target moving in a straight line (3.3), either captures the target or reaches a forward invariant configuration region $(\alpha, \beta) \in \left[\frac{\pi}{2}, -\frac{\pi}{2}\right] \times \left[\frac{\pi}{3}, -\frac{\pi}{3}\right]$ in finite time.*

Proof Results directly from all previous statements. ■

Lemma 3.2.4. *If an agent (2.1) governed by the bearing-only control law (3.5), in pursuit of a target moving in a straight line (3.3), reaches the target's path with $\beta = 0$, then it tracks the target from that point on.*

Proof If $(\alpha, \beta) = (0, 0)$, then by (2.3), (2.4) and (3.6), $\left(\dot{\alpha}, \dot{\beta}, \dot{\rho}\right) = (0, 0, 0)$, and (α, β, ρ) become constant. ■

Theorem 3.3. *An agent (2.1) governed by the bearing-only control law (3.5) with $\kappa > 2\frac{v}{r_c}$, in pursuit of a target moving in a straight line (3.3), either captures the target or asymptotically reaches the target's path.*

Proof Consider the function

$$V(\alpha,\beta) = 2\left(\sin^2\left(\frac{\alpha(t)}{2}\right) + \sin^2\left(\frac{\beta(t)}{2}\right)\right) \geq 0.$$

$$\begin{aligned}\dot{V} &= \sin(\alpha)\,\dot{\alpha} + \sin(\beta)\,\dot{\beta} \\ &= \frac{v}{\rho}\left(\sin^2(\beta) - \sin^2(\alpha)\right) - \kappa\beta\sin(\beta) \\ &\leq \left(\frac{v}{\rho} - \kappa\right)\beta\sin(\beta) - \frac{v}{\rho}\sin^2(\alpha).\end{aligned}$$

If the agent captures the target, then $\exists t_c < \infty \mid \rho(t_c) \leq r_c$, and the pursuit concludes; otherwise, $\rho(t) > r_c$, and since we choose the gain such that $\kappa > 2\frac{v}{r_c}$,

$$\dot{V} \leq \left(\frac{v}{r_c} - 2\frac{v}{r_c}\right)\beta\sin(\beta) - \frac{v}{\rho}\sin^2(\alpha) \leq 0.$$

We proceed to find the points in which $\dot{V} = 0$.

$$\dot{V} = 0 \Rightarrow \frac{v}{\rho}\left(\sin^2(\beta) - \sin^2(\alpha)\right) - \kappa\beta\sin(\beta) = 0$$

$$\Downarrow$$

$$\sin^2(\alpha) = \beta\sin(\beta)\left(\frac{\sin(\beta)}{\beta} - \frac{\kappa\rho}{v}\right).$$

From Theorem 3.2, if the target was not captured by t_2, then $\forall t \geq t_2$, $(\alpha(t),\beta(t)) \in \left[\frac{\pi}{2},-\frac{\pi}{2}\right] \times \left[\frac{\pi}{3},-\frac{\pi}{3}\right]$. Using this fact, we notice that $\frac{\kappa\rho}{v} > 2$, and that if $|\beta(t)| \leq \frac{\pi}{3}$, then $\frac{\sin(\beta)}{\beta} \leq 1$, $\beta\sin(\beta) \geq 0$, and the right hand side of the last equation becomes non-positive, while the left hand side non-negative, and the only point in the forward invariant region $(\alpha,\beta) \in \left[\frac{\pi}{2},-\frac{\pi}{2}\right] \times \left[\frac{\pi}{3},-\frac{\pi}{3}\right]$ at which $\dot{V} = 0$ is the origin $(0,0)$. To conclude the proof,

1. $\forall t > t_2$, $(\alpha(t),\beta(t)) \in \left[\frac{\pi}{2},-\frac{\pi}{2}\right] \times \left[\frac{\pi}{3},-\frac{\pi}{3}\right]$,
2. $V(\alpha,\beta) > 0$, $\forall\,(\alpha,\beta) \in \left[\frac{\pi}{2},-\frac{\pi}{2}\right] \times \left[\frac{\pi}{3},-\frac{\pi}{3}\right] \setminus (0,0)$,
3. $\dot{V}(\alpha,\beta) < 0$, $\forall\,(\alpha,\beta) \in \left[\frac{\pi}{2},-\frac{\pi}{2}\right] \times \left[\frac{\pi}{3},-\frac{\pi}{3}\right] \setminus (0,0)$,
4. $V(0,0) = \dot{V}(0,0) = 0$;

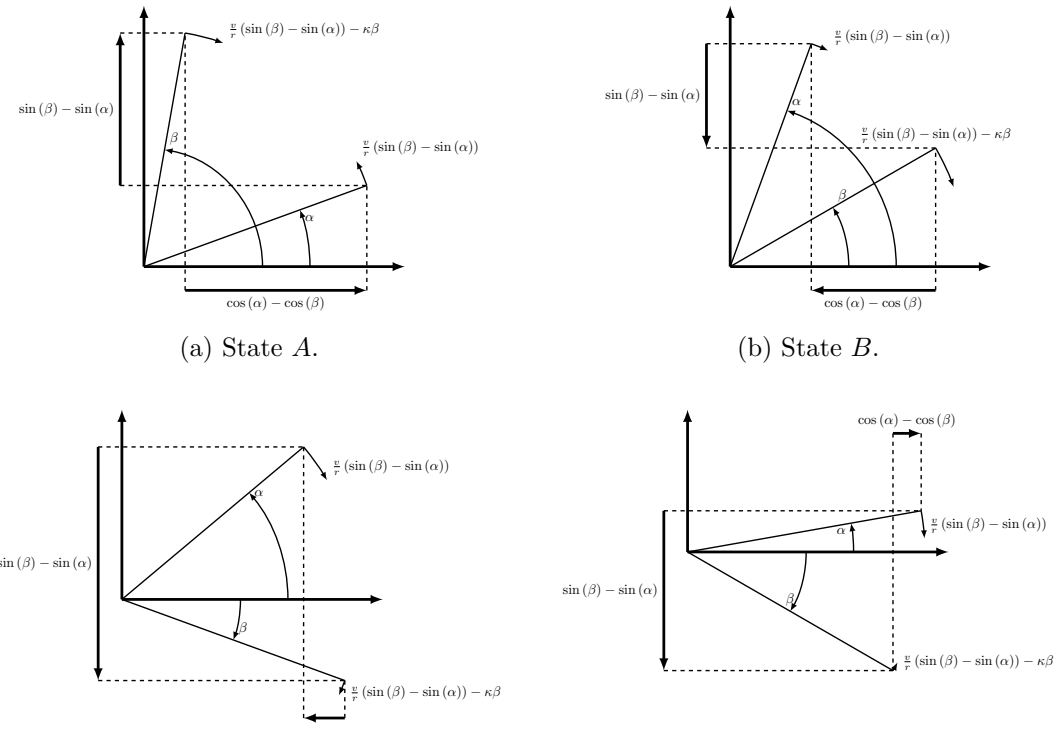
V is therefore a Lyapunov function[LaS60], and if the target was not captured in the process, then $\lim_{t\to\infty}(\alpha,\beta) = (0,0)$. ■

3.3 Tracking

Though Theorem 3.3 asserts that $\dot{\rho}(t \to \infty) = 0$, it does not necessarily guarantee that $\rho(t \to \infty) < \infty$. If the rate at which $|\beta|$ and $|\alpha|$ shrink is too low, the target may slip away to infinity.

This section addresses this issue and affirms that even if the target does not get captured, it does not get away. We achieve this result by applying Algorithm 2.1 to generate a deterministic finite state machine.

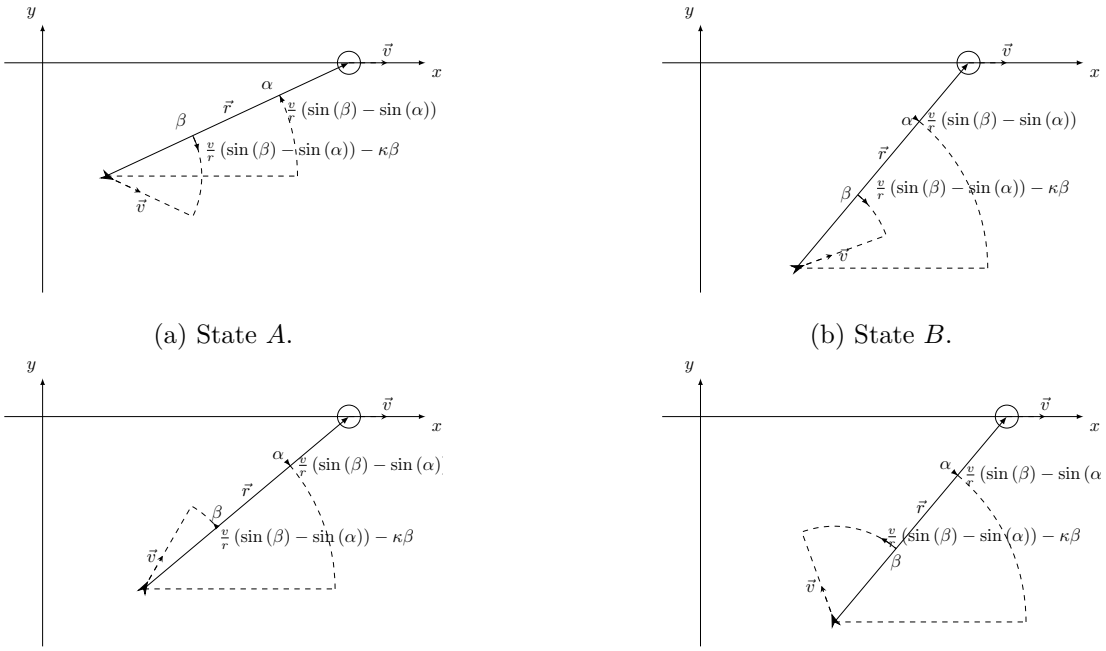
The first major stage in Algorithm 2.1 involves identifying the different system states. Figure 3.2 illustrates the defining condition of each state, while Figure 3.3 shows a typical configuration for each of the primary system states, and Table 3.1 summarizes the state and condition pairs.



(a) State A. (b) State B.

(c) State C. (d) State D.

Figure 3.2: The primary states, defined by the angle couple α, β, and their rate of change.



(a) State A.

(b) State B.

(c) State C.

(d) State D.

Figure 3.3: An illustration of a typical configuration on the plane for each system state.

The second major stage of Algorithm 2.1 is identifying the transitions from state to state by calculating the bounds on (ρ, α, β) upon exit. To this end we present the following series of lemmas.

Lemma 3.3.1 (States E, E^- Exit Condition)**.** *If* $\kappa > \frac{4}{3}\frac{v}{r_c}$, *and* $\rho(t_0) > r_c$, *then* $|\beta(t)| < \frac{\pi}{2}\ \forall t \left| t > t_0 + \frac{1}{\kappa}\ln\left(\frac{\beta(t_0)-2\frac{v}{\kappa r_c}}{\frac{\pi}{2}-2\frac{v}{\kappa r_c}}\right)\right.$.

Proof Using the same bounds on β from Equation 3.8, if $\frac{\pi}{2} \leq |\beta(t_0)| \leq \pi$, then the magnitude of the bounds on β shrink asymptotically to $2\frac{v}{\kappa r_c} < \frac{\pi}{2}$, and reach $\frac{\pi}{2}$ by t_1,

$$\beta^+(t_1) = 2\frac{v}{\kappa r_c} + \left(\beta(t_0) - 2\frac{v}{\kappa r_c}\right)e^{-\kappa(t_1-t_0)} = \frac{\pi}{2}$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{1}{\kappa}\ln\left(\frac{\beta(t_0) - 2\frac{v}{\kappa r_c}}{\frac{\pi}{2} - 2\frac{v}{\kappa r_c}}\right).$$

Since the bounds on β are global, if $|\beta(t_0)| < 2\frac{v}{\kappa r_c} < \frac{\pi}{2}$, then so must $|\beta(t_1)| < \frac{\pi}{2}$. ■

Lemma 3.3.2 (State A Exit Condition)**.** *If* $\kappa > \frac{v}{r_c}$, *and at time* $t = t_0$,

1. $0 \leq \alpha(t_0) < \beta(t_0) < \frac{\pi}{2}$,
2. $\rho(t_0) > r_c$;

then

State	System Configuration
A	$0 \leq \alpha(t) < \beta(t) < \frac{\pi}{2}$
A^-	$-\frac{\pi}{2} < \beta(t) < \alpha(t) \leq 0$
B	$0 < \beta(t) \leq \alpha(t) < \frac{\pi}{2}$
B^-	$-\frac{\pi}{2} < \alpha(t) \leq \beta(t) < 0$
C	$0 \leq -\beta < \alpha < \frac{\pi}{2}$
C^-	$0 \leq \beta(t) < -\alpha(t) < \frac{\pi}{2}$
D	$0 < \alpha \leq -\beta \leq \frac{\pi}{2}$
D^-	$0 < -\alpha(t) \leq \beta(t) < \frac{\pi}{2}$
E	$\frac{\pi}{2} \leq \beta \leq \pi$
E^-	$-\pi < \beta \leq -\frac{\pi}{2}$

Table 3.1: Path-Following States of Proportional-Control Unicycles with Bearing-Only Sensing in Pursuit of a Constant Velocity Target.

1. $t_1 = t_0 + \frac{1}{\kappa} \ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right)$,

2. $\beta^+(t_1) = \beta(t_0) e^{-\left(\kappa - \frac{v}{\rho(t_0)}\right)(t_1 - t_0)} - \frac{v}{\kappa\rho(t_0) - v} \sin\left(\alpha(t_0)\right) \left(1 - e^{-\left(\kappa - \frac{v}{\rho(t_0)}\right)(t_1 - t_0)}\right)$,

3. $\alpha^+(t_1) = \beta(t_0) + 2\,\mathrm{arccot}\left(\left(\cot\left(\frac{\alpha(t_0)-\beta(t_0)}{2}\right) - \tan\left(\beta(t_0)\right)\right) e^{\frac{v}{\rho(t_0)}\cos(\beta(t_0))(t_1 - t_0)} + \tan\left(\beta(t_0)\right)\right)$,

4. $\rho^+(t_1) = \rho(t_0) + \frac{v}{\kappa} \ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right) (\cos(\alpha(t_0)) - \cos(\beta(t_0)))$,

5. $\alpha^-(t_1) = \alpha(t_0) + \frac{v}{\kappa\rho^+(t_1)} \ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right)\left(\sin\left(\frac{\beta(t_0)-\alpha(t_0)}{2}\right) - \sin\left(\alpha^+(t_1)\right)\right)$,

6. $t_2 = t_0 + \frac{\rho(t_0)}{\kappa\rho(t_0)-v} \ln\left(\frac{\beta(t_0)+\frac{v}{\kappa\rho(t_0)-v}\sin(\alpha(t_0))}{\alpha^-(t_1)+\frac{v}{\kappa\rho(t_0)-v}\sin(\alpha(t_0))}\right)$,

and $\exists t_* \,|t_1 \le t_* \le t_2$ *such that* $\alpha(t_*) = \beta(t_*) \le \beta^+(t_1)$.

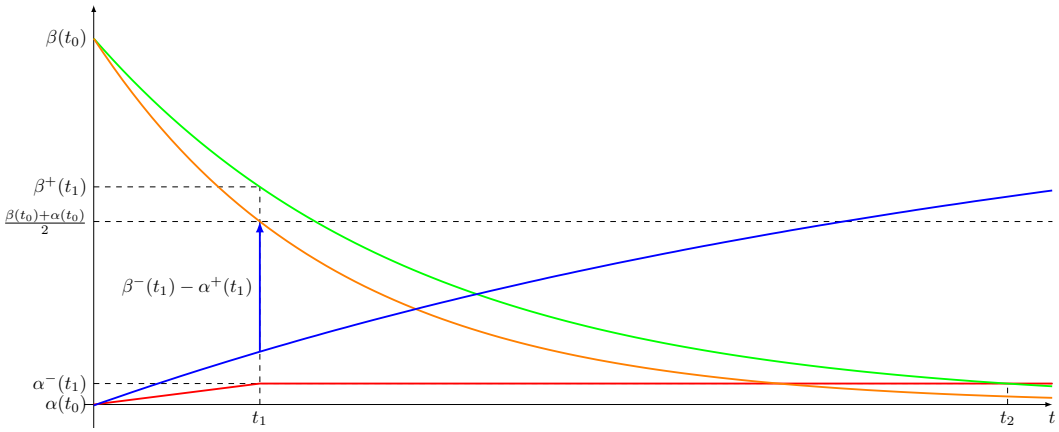


Figure 3.4: Lemma 3.3.2 proof outline. We start by finding β^+ and β^-, the bounds of β, and use β^- to calculate t_1 and $\beta^+(t_1)$. We then find α^+, an upper bound on α, and $\alpha^+(t_1)$. We continue to find the upper bound on $\rho(t_1)$, and with it we calculate $\alpha^-(t_1)$, the minimal possible value for α at time t_1. We conclude by finding time t_2, when $\beta^+(t_2) = \alpha^-(t_1)$, and note that $\beta = \alpha$ must happen between t_1 and t_2.

Figure 3.4 shows an outline of the proof.

Proof While in state A, $\sin(\beta) > \sin(\alpha)$ and $\cos(\alpha) > \cos(\beta)$, therefore

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) > 0,$$

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) > 0,$$

and α and ρ grow accordingly.

Meanwhile, since $\rho(t_0) > r_c$, $\dot{\rho} > 0$, and $\kappa > \frac{v}{r_c} > \frac{v}{\rho(t_0)}$,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta \le \frac{v}{\rho(t_0)}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta$$

$$< -\frac{v}{\rho(t_0)}\sin(\alpha) + \left(\frac{v}{\rho(t_0)} - \kappa\right)\beta < -\frac{v}{\rho(t_0)}\sin(\alpha(t_0)) - \left(\frac{\kappa\rho(t_0) - v}{\rho(t_0)}\right)\beta < 0.$$

So we see that while in state A, α grows and β shrinks.

$$-\kappa\beta < \dot{\beta} < -\frac{v}{\rho(t_0)}\sin(\alpha(t_0)) - \left(\kappa - \frac{v}{\rho(t_0)}\right)\beta$$

$$\Downarrow$$

$$\beta(t) > \beta^-(t) = \beta(t_0)e^{-\kappa(t-t_0)};$$

$$\beta(t) < \beta^+(t) = \beta(t_0)e^{-\left(\kappa-\frac{v}{\rho(t_0)}\right)(t-t_0)} - \frac{\frac{v}{\rho(t_0)}}{\kappa-\frac{v}{\rho(t_0)}}\sin\left(\alpha(t_0)\right)\left(1-e^{-\left(\kappa-\frac{v}{\rho(t_0)}\right)(t-t_0)}\right).$$

Since $\alpha(t_0) > 0$ and $\kappa > \frac{v}{\rho(t_0)}$, β^+ decays to a negative value.

Let t_1 such that

$$\beta^-(t_1) = \beta(t_0)e^{-\kappa(t_1-t_0)} = \frac{\beta(t_0)+\alpha(t_0)}{2}$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{1}{\kappa}\ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right),$$

and $\beta^+(t_1) = \left(\beta(t_0) + \frac{\frac{v}{\rho(t_0)}}{\kappa-\frac{v}{\rho(t_0)}}\sin\left(\alpha(t_0)\right)\right)e^{-\left(\kappa-\frac{v}{\rho(t_0)}\right)(t_1-t_0)} - \frac{\frac{v}{\rho(t_0)}}{\kappa-\frac{v}{\rho(t_0)}}\sin\left(\alpha(t_0)\right)$.

We now find an upper bound on α,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin\left(\beta\right) - \sin\left(\alpha\right)\right) \leq \frac{v}{\rho(t_0)}\left(\sin\left(\beta(t_0)\right) - \sin\left(\alpha\right)\right);$$

$$\sin(\alpha) = 2\sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\alpha}{2}\right) = \frac{2\sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\alpha}{2}\right)}{\sin^2\left(\frac{\alpha}{2}\right)+\cos^2\left(\frac{\alpha}{2}\right)} = \frac{2\tan\left(\frac{\alpha}{2}\right)}{1+\tan^2\left(\frac{\alpha}{2}\right)};$$

$$\Downarrow$$

$$\dot{\alpha} \leq \frac{v}{\rho(t_0)}\sin\left(\beta(t_0)\right) - \frac{v}{\rho(t_0)}\frac{2\tan\left(\frac{\alpha}{2}\right)}{1+\tan^2\left(\frac{\alpha}{2}\right)}$$

$$\Downarrow$$

$$\alpha(t) \leq \alpha^+(t);$$

$$\dot{\alpha} \leq \dot{\alpha}^+ = \frac{v}{\rho(t_0)}\sin\left(\beta(t_0)\right) - \frac{v}{\rho(t_0)}\frac{2\tan\left(\frac{\alpha^+}{2}\right)}{1+\tan^2\left(\frac{\alpha^+}{2}\right)}.$$

Substituting $u = \tan\left(\frac{\alpha^+}{2}\right)$,

$$\frac{du}{d\alpha^+} = \frac{1+u^2}{2}$$

$$\Downarrow$$

$$d\alpha^+ = \frac{2du}{1+u^2}$$

$$\Downarrow$$

$$\frac{d\alpha^+}{dt} = \frac{2}{1+u^2}\frac{du}{dt}$$

$$\Downarrow$$

$$\frac{2}{1+u^2}\frac{du}{dt} = \frac{v}{\rho(t_0)}\left(\sin\left(\beta(t_0)\right) - \frac{2u}{1+u^2}\right)$$

$$\Downarrow$$

$$\frac{2}{1+u^2}\frac{du}{dt} = \frac{v\sin\left(\beta(t_0)\right)}{\rho(t_0)}\frac{u^2 - \frac{2u}{\sin(\beta(t_0))} + 1}{1+u^2}$$

$$= \frac{v\sin\left(\beta(t_0)\right)}{\rho(t_0)}\frac{u^2 - \frac{2u}{\sin(\beta(t_0))} + \frac{1}{\sin^2(\beta(t_0))} - \frac{1}{\sin^2(\beta(t_0))} + 1}{1+u^2}$$

$$\Downarrow$$

$$\frac{2}{1+u^2}\frac{du}{dt} = \frac{v\sin\left(\beta(t_0)\right)}{\rho(t_0)}\frac{\left(u - \frac{1}{\sin(\beta(t_0))}\right)^2 + 1 - \frac{1}{\sin^2(\beta(t_0))}}{1+u^2}$$

$$\Downarrow$$

$$\int \frac{du}{\left(u - \frac{1}{\sin(\beta(t_0))}\right)^2 + 1 - \frac{1}{\sin^2(\beta(t_0))}} = \int \frac{v\sin\left(\beta(t_0)\right)}{2\rho(t_0)}dt$$

$$\Downarrow$$

$$\int \frac{du}{\left(u - \frac{1}{\sin(\beta(t_0))}\right)^2 - \cot^2\left(\beta(t_0)\right)} = \int \frac{v\sin\left(\beta(t_0)\right)}{2\rho(t_0)}dt$$

$$\Downarrow$$

$$\frac{\tan\left(\beta(t_0)\right)}{2}\left(\int \frac{du}{\left(u - \frac{1}{\sin(\beta(t_0))}\right) - \cot\left(\beta(t_0)\right)} - \int \frac{du}{\left(u - \frac{1}{\sin(\beta(t_0))}\right) + \cot\left(\beta(t_0)\right)}\right)$$

$$= \int \frac{v\sin\left(\beta(t_0)\right)}{2\rho(t_0)}dt$$

$$\Downarrow$$

$$\frac{\tan\left(\beta(t_0)\right)}{2}\left(\int \frac{du}{u - \frac{1+\cos(\beta(t_0))}{\sin(\beta(t_0))}} - \int \frac{du}{u - \frac{1-\cos(\beta(t_0))}{\sin(\beta(t_0))}}\right) = \int \frac{v\sin\left(\beta(t_0)\right)}{2\rho(t_0)}dt$$

$$\Downarrow$$

$$\int \frac{du}{u - \cot\left(\frac{\beta(t_0)}{2}\right)} - \int \frac{du}{u - \tan\left(\frac{\beta(t_0)}{2}\right)} = \int \frac{v}{\rho(t_0)}\cos\left(\beta(t_0)\right)dt$$

$$\Downarrow$$

$$\ln\left(u - \cot\left(\frac{\beta(t_0)}{2}\right)\right) - \ln\left(u - \tan\left(\frac{\beta(t_0)}{2}\right)\right) = \frac{v}{\rho(t_0)}\cos\left(\beta(t_0)\right)(t - t_0) + C_0$$

$$\Downarrow$$

$$\ln\left(\frac{u - \cot\left(\frac{\beta(t_0)}{2}\right)}{u - \tan\left(\frac{\beta(t_0)}{2}\right)}\right) = \frac{v}{\rho(t_0)}\cos\left(\beta(t_0)\right)(t - t_0) + C_0$$

$$\Downarrow$$

$$\ln\left(\frac{\tan\left(\frac{\alpha^+}{2}\right)-\cot\left(\frac{\beta(t_0)}{2}\right)}{\tan\left(\frac{\alpha^+}{2}\right)-\tan\left(\frac{\beta(t_0)}{2}\right)}\right)=\frac{v}{\rho(t_0)}\cos\left(\beta(t_0)\right)(t-t_0)+C_0$$

$$\Downarrow$$

$$\ln\left(\frac{\frac{\sin\left(\frac{\alpha^+}{2}\right)\sin\left(\frac{\beta(t_0)}{2}\right)-\cos\left(\frac{\alpha^+}{2}\right)\cos\left(\frac{\beta(t_0)}{2}\right)}{\cos\left(\frac{\alpha^+}{2}\right)\sin\left(\frac{\beta(t_0)}{2}\right)}}{\frac{\sin\left(\frac{\alpha^+}{2}\right)\cos\left(\frac{\beta(t_0)}{2}\right)-\sin\left(\frac{\beta(t_0)}{2}\right)\cos\left(\frac{\alpha^+}{2}\right)}{\cos\left(\frac{\alpha^+}{2}\right)\cos\left(\frac{\beta(t_0)}{2}\right)}}\right)=\frac{v}{\rho(t_0)}\cos\left(\beta(t_0)\right)(t-t_0)+C_0$$

$$\Downarrow$$

$$\ln\left(-\frac{\cos\left(\frac{\alpha^++\beta(t_0)}{2}\right)}{\sin\left(\frac{\alpha^+-\beta(t_0)}{2}\right)}\cot\left(\frac{\beta(t_0)}{2}\right)\right)=\frac{v}{\rho(t_0)}\cos\left(\beta(t_0)\right)(t-t_0)+C_0$$

$$\Downarrow$$

$$-\frac{\cos\left(\frac{\alpha^+-\beta(t_0)}{2}\right)\cos\left(\beta(t_0)\right)-\sin\left(\frac{\alpha^+-\beta(t_0)}{2}\right)\sin\left(\beta(t_0)\right)}{\sin\left(\frac{\alpha^+-\beta(t_0)}{2}\right)}\cot\left(\frac{\beta(t_0)}{2}\right)$$

$$=\left(\sin\left(\beta(t_0)\right)-\cot\left(\frac{\alpha^+-\beta(t_0)}{2}\right)\cos\left(\beta(t_0)\right)\right)\cot\left(\frac{\beta(t_0)}{2}\right)$$

$$=C_1e^{\frac{v}{\rho(t_0)}\cos(\beta(t_0))(t-t_0)}$$

$$\Downarrow$$

$$\cot\left(\frac{\alpha^+-\beta(t_0)}{2}\right)=\tan\left(\beta(t_0)\right)+C_2e^{\frac{v}{\rho(t_0)}\cos(\beta(t_0))(t-t_0)}$$

$$\Downarrow$$

$$\alpha^+(t)=\beta(t_0)+2\,\mathrm{arccot}\left(C_2e^{\frac{v}{\rho(t_0)}\cos(\beta(t_0))(t-t_0)}+\tan\left(\beta(t_0)\right)\right);$$

$$\alpha^+(t_0)=\alpha(t_0)$$

$$\Downarrow$$

$$C_2=\cot\left(\frac{\alpha(t_0)-\beta(t_0)}{2}\right)-\tan\left(\beta(t_0)\right)$$

$$\Downarrow$$

$$\alpha(t)\le\alpha^+(t),$$

$$\alpha^+(t)=\beta(t_0)$$

$$+2\,\mathrm{arccot}\left(\left(\cot\left(\frac{\alpha(t_0)-\beta(t_0)}{2}\right)-\tan\left(\beta(t_0)\right)\right)e^{\frac{v}{\rho(t_0)}\cos(\beta(t_0))(t-t_0)}+\tan\left(\beta(t_0)\right)\right).$$

For $t_0 \le t \le t_1$,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) \le v\left(\cos(\alpha(t_0)) - \cos(\beta(t_0))\right)$$

$$\Downarrow$$

$$\rho(t_1) \le \rho(t_0) + v\left(\cos(\alpha(t_0)) - \cos(\beta(t_0))\right)(t_1 - t_0)$$

$$\rho(t_0) + \frac{v}{\kappa}\ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right)\left(\cos(\alpha(t_0)) - \cos(\beta(t_0))\right) = \rho^+(t_1).$$

From these results we can find a lower bound on $\alpha(t)$, for all $t_0 < t < t_1$,

$$\dot{\alpha} \ge \frac{v}{\rho^+(t_1)}\left(\sin\left(\beta^-(t_1)\right) - \sin\left(\alpha^+(t_1)\right)\right)$$

$$\Downarrow$$

$$\alpha(t_1) \ge \alpha^-(t_1) = \alpha(t_0) + \frac{v}{\rho^+(t_1)}\left(\sin\left(\beta^-(t_1)\right) - \sin\left(\alpha^+(t_1)\right)\right)(t_1 - t_0)$$

Let t_2 be the moment at which $\beta^+(t_2) = \alpha^-(t_1)$,

$$\alpha^-(t_1) = -\frac{\frac{v}{\rho(t_0)}}{\kappa - \frac{v}{\rho(t_0)}}\sin\left(\alpha(t_0)\right) + \left(\beta(t_0) + \frac{\frac{v}{\rho(t_0)}}{\kappa - \frac{v}{\rho(t_0)}}\sin\left(\alpha(t_0)\right)\right)e^{-\left(\kappa - \frac{v}{\rho(t_0)}\right)(t_2 - t_0)}$$

$$t_2 = t_0 + \frac{1}{\kappa - \frac{v}{\rho(t_0)}}\ln\left(\frac{\beta(t_0) + \frac{\frac{v}{\rho(t_0)}}{\kappa - \frac{v}{\rho(t_0)}}\sin\left(\alpha(t_0)\right)}{\alpha^-(t_1) + \frac{\frac{v}{\rho(t_0)}}{\kappa - \frac{v}{\rho(t_0)}}\sin\left(\alpha(t_0)\right)}\right)$$

and $\exists t_*$, $t_1 \le t_* \le t_2$, such that $\beta(t_*) = \alpha(t_*)$. ■

Corollary 3.4 (State A^- Exit Condition)**.** *If $\kappa > \frac{v}{r_c}$, and at time $t = t_0$,*

1. $-\frac{\pi}{2} < \beta(t_0) < \alpha(t_0) \le 0$,
2. $\rho(t_0) > r_c$;

then

1. $t_1 = t_0 + \frac{1}{\kappa}\ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right)$,
2. $\beta^+(t_1) = \beta(t_0)e^{-\left(\kappa - \frac{v}{\rho(t_0)}\right)(t_1 - t_0)} - \frac{v}{\kappa\rho(t_0) - v}\sin\left(\alpha(t_0)\right)\left(1 - e^{-\left(\kappa - \frac{v}{\rho(t_0)}\right)(t_1 - t_0)}\right)$,
3. $\alpha^+(t_1) = \beta(t_0)$
   $+2\,\mathrm{arccot}\left(\left(\cot\left(\frac{\alpha(t_0)-\beta(t_0)}{2}\right) - \tan\left(\beta(t_0)\right)\right)e^{\frac{v}{\rho(t_0)}\cos(\beta(t_0))(t_1 - t_0)} + \tan\left(\beta(t_0)\right)\right)$,
4. $\rho^+(t_1) = \rho(t_0) + \frac{v}{\kappa}\ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right)\left(\cos(\alpha(t_0)) - \cos(\beta(t_0))\right)$,

5. $\alpha^-(t_1) = \alpha(t_0) + \frac{v}{\kappa\rho^+(t_1)} \ln\left(\frac{2}{1+\frac{\alpha(t_0)}{\beta(t_0)}}\right)\left(\sin\left(\frac{\beta(t_0)-\alpha(t_0)}{2}\right) - \sin\left(\alpha^+(t_1)\right)\right)$,

6. $t_2 = t_0 + \frac{\rho(t_0)}{\kappa\rho(t_0)-v} \ln\left(\frac{\beta(t_0)+\frac{v}{\kappa\rho(t_0)-v}\sin(\alpha(t_0))}{\alpha^-(t_1)+\frac{v}{\kappa\rho(t_0)-v}\sin(\alpha(t_0))}\right)$,

and $\exists t_* \,|t_1 \le t_* \le t_2$ *such that* $\alpha(t_*) = \beta(t_*) \ge \beta^+(t_1)$.

Lemma 3.3.3 (State B Exit Conditions)**.** *If at time* $t = t_0$

1. $0 < \beta(t_0) \le \alpha(t_0) < \frac{\pi}{2}$,

2. $\rho(t_0) > r_c$;

then

1. $\rho^-(t_0 + \frac{1}{\kappa}) = \rho(t_0) - \frac{v}{\kappa}\left(1 - \cos\left(\alpha(t_0)\right)\right)$,

2. $a = \kappa\frac{v}{\rho^-(t_0+\frac{1}{\kappa})} = \frac{v}{\rho(t_0)}\frac{1}{\kappa - \frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}$,

3. $b = \frac{\beta(t_0)+a\left(\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\frac{\beta(t_0)}{e}+a\left(\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}$,

4. $t_2 = t_0 + \frac{1}{\kappa}\ln(b)$,

5. $\alpha^-(t_2) = \alpha(t_0) + \left(\frac{1}{e}+b\right)\beta(t_0) + a\left(\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)\right)(1 - b - \ln(b))$,

6. $\beta^+(t_2) = \frac{\beta(t_0)}{b}$,

7. $t_3 = t_2 - \frac{1}{\kappa}\ln\left(\frac{\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}{\beta^+(t_2)+\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}\right)$;

and either $\exists t_* \,|t_2 \le t_* \le t_3$ *such that* $\beta(t_*) = 0$*, or* $\exists t_* \,|t_0 < t_* \le t_3$ *such that* $\rho(t_*) \le r_c$.

Figure 3.5 shows an outline of the proof.

Proof While in state B, $0 < \beta(t) \le \alpha(t) \le \frac{\pi}{2}$, and

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin\left(\beta\right) - \sin\left(\alpha\right)\right) \le 0,$$

$$\dot{\beta} = \dot{\alpha} - \kappa\beta < \dot{\alpha} \le 0,$$

therefore β shrinks faster than α. Also,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) \le 0,$$

shrinking ρ.

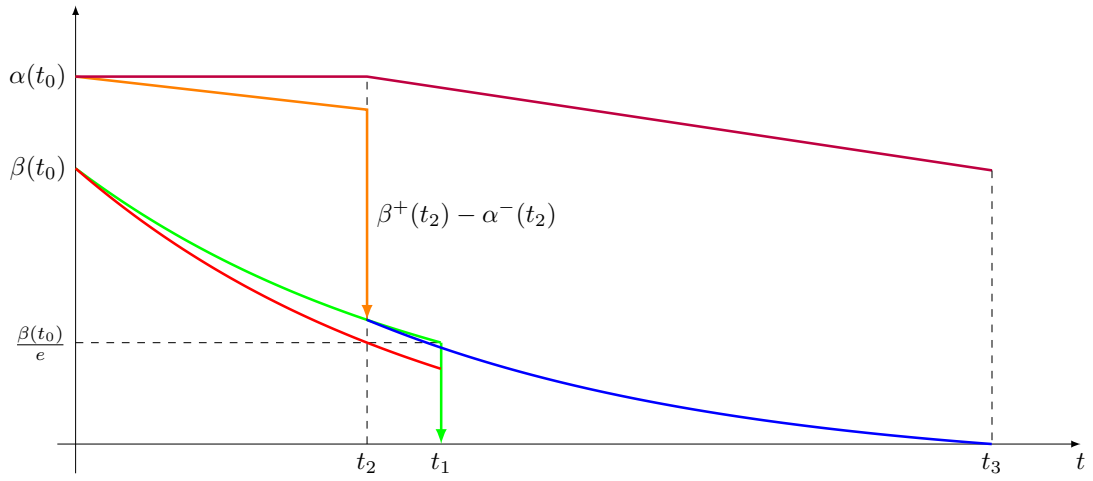


Figure 3.5: Lemma 3.3.3 proof outline. We start by finding an upper bound for β and using it to calculate t_1. We then use t_1 to find $\rho^-(t_1)$, which is needed in order to find the lower bound on β. We then proceed to calculate t_2, and find the lower bound on $\alpha(t_2)$. We conclude the proof by showing that the upper bound on β must reach 0 by t_3, and that therefore $\beta = 0$ must have happened between t_2 and t_3.

We will now find t_1, t_2, the upper and lower bounds respectively to the time required until $\beta = \frac{\beta(t_0)}{e}$.

$$\dot{\beta} \leq -\kappa\beta$$

$$\Downarrow$$

$$\beta(t) \leq \beta(t_0)e^{-\kappa(t-t_0)} = \beta^+(t).$$

Let t_1 be the moment at which the upper bound on β reaches $\frac{\beta(t_0)}{e}$,

$$\beta^+(t_1) = \beta(t_0)e^{-\kappa(t_1-t_0)} = \frac{\beta(t_0)}{e}$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{1}{\kappa}.$$

So $\frac{1}{\kappa}$ is the longest possible period of time required for $\beta \leq \frac{\beta(t_0)}{e}$.

While in state B,

$$\dot{\rho} = v\left(\cos\left(\alpha\right) - \cos(\beta)\right) > v\left(\cos\left(\alpha(t_0)\right) - 1\right),$$

and the lower bound on ρ is

$$\rho^-(t) = \rho(t_0) + v\left(\cos\left(\alpha(t_0)\right) - 1\right)(t - t_0) < \rho(t).$$

Using $\rho^-(t) < \rho(t)\ \forall t\,|t_0 < t \leq t_1$, the lower bound to the minimal possible value for ρ

at t_1 becomes

$$\rho^-(t_1) = \rho(t_0) + v\left(\cos\left(\alpha(t_0)\right) - 1\right)(t_1 - t_0) = \rho(t_0) - \frac{v}{\kappa}\left(1 - \cos\left(\alpha(t_0)\right)\right).$$

We can now calculate the lower bound on $\beta(t)\ \forall t\,|t_0 < t < t_1$,

$$\dot{\beta} > \frac{v}{\rho^-(t_1)}\left(\sin\left(\frac{\beta(t_0)}{e}\right) - \sin\left(\alpha(t_0)\right)\right) - \kappa\beta$$

$$\Downarrow$$

$$\beta > \beta^-(t) = -\frac{v}{\rho(t_0)}\frac{\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa - \frac{v}{\rho(t_0)}\left(1 - \cos\left(\alpha(t_0)\right)\right)} +$$

$$\left(\beta(t_0) + \frac{v}{\rho(t_0)}\frac{\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa - \frac{v}{\rho(t_0)}\left(1 - \cos\left(\alpha(t_0)\right)\right)}\right)e^{-\kappa(t-t_0)}$$

and find t_2, the first possible moment at which $\beta = \frac{\beta(t_0)}{e}$.

$$\beta^-(t_2) = -\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa - \frac{v}{\rho(t_0)}\left(1 - \cos\left(\alpha(t_0)\right)\right)}$$

$$+\left(\beta(t_0) + \frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa - \frac{v}{\rho(t_0)}\left(1 - \cos\left(\alpha(t_0)\right)\right)}\right)e^{-\kappa(t_2-t_0)} = \frac{\beta(t_0)}{e}$$

$$\Downarrow$$

$$t_2 = t_0 + \frac{1}{\kappa}\ln\left(\frac{\beta(t_0) + \frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0)) - \sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa - \frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}{\frac{\beta(t_0)}{e} + \frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0)) - \sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa - \frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}\right).$$

During this time, $t_0 < t < t_2$, β shrunk faster than α and the two angles drifted away from one another.

$$\frac{d}{dt}(\alpha - \beta) = \dot{\alpha} - \dot{\beta} = \kappa\beta;$$

$$\kappa\beta^-(t) < \kappa\beta \leq \kappa\beta^+(t) = \kappa\beta(t_0)e^{-\kappa(t-t_0)}$$

$$\Downarrow$$

$$-\kappa\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa - \frac{v}{\rho(t_0)}\left(1 - \cos\left(\alpha(t_0)\right)\right)}$$

$$+\left(\beta(t_0) + \frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa - \frac{v}{\rho(t_0)}\left(1 - \cos\left(\alpha(t_0)\right)\right)}\right)\kappa e^{-\kappa(t-t_0)} < \frac{d}{dt}\left(\dot{\alpha} - \dot{\beta}\right)$$

$$\Downarrow$$

$$\alpha(t)-\beta(t) > \alpha(t_0)+\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa-\frac{v}{\rho(t_0)}\left(1-\cos\left(\alpha(t_0)\right)\right)}\left(1-\kappa\left(t-t_0\right)-e^{-\kappa(t-t_0)}\right)-\beta(t_0)e^{-\kappa(t-t_0)}$$

and we can now calculate the lower bound on $\alpha(t_2)-\beta(t_2)$,

$$\alpha(t_2)-\beta(t_2) > \alpha(t_0)+\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa-\frac{v}{\rho(t_0)}\left(1-\cos\left(\alpha(t_0)\right)\right)}\left(1-\kappa\left(t-t_0\right)-e^{-\kappa(t-t_0)}\right)-\beta(t_0)e^{-\kappa(t-t_0)}$$

$$\Downarrow$$

$$\alpha(t_2) > \alpha(t_0)+\beta(t_2)$$
$$+\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa-\frac{v}{\rho(t_0)}\left(1-\cos\left(\alpha(t_0)\right)\right)}\left(1-\kappa\left(t_2-t_0\right)-e^{-\kappa(t_2-t_0)}\right)-\beta(t_0)e^{-\kappa(t_2-t_0)}$$

$$=\alpha(t_0)+\beta(t_2)+\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa-\frac{v}{\rho(t_0)}\left(1-\cos\left(\alpha(t_0)\right)\right)}\left(1+\ln\left(\frac{\frac{\beta(t_0)}{e}+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}{\beta(t_0)+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}\right)\right)$$
$$-\left(\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa-\frac{v}{\rho(t_0)}\left(1-\cos\left(\alpha(t_0)\right)\right)}+\beta(t_0)\right)\left(\frac{\frac{\beta(t_0)}{e}+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}{\beta(t_0)+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}\right)$$

and

$$\alpha(t_2) \geq \alpha(t_0)+\frac{\beta(t_0)}{e}$$
$$+\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa-\frac{v}{\rho(t_0)}\left(1-\cos\left(\alpha(t_0)\right)\right)}\left(1+\ln\left(\frac{\frac{\beta(t_0)}{e}+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}{\beta(t_0)+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}\right)\right)$$
$$-\left(\frac{\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\kappa-\frac{v}{\rho(t_0)}\left(1-\cos\left(\alpha(t_0)\right)\right)}+\beta(t_0)\right)\left(\frac{\frac{\beta(t_0)}{e}+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}{\beta(t_0)+\frac{v}{\rho(t_0)}\frac{\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)}{\kappa-\frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}}\right)=\alpha^-(t_2)$$

After t_2, and since $\alpha-\beta$ grows while α and β independently shrink in state B, we can calculate new upper bounds.

$$\dot{\alpha}=-\frac{v}{\rho}\left(\sin\left(\alpha\right)-\sin\left(\beta\right)\right)=-2\frac{v}{\rho}\cos\left(\frac{\alpha+\beta}{2}\right)\sin\left(\frac{\alpha-\beta}{2}\right)$$

$$<-2\frac{v}{\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)<0;$$

$$\dot{\beta} = \dot{\alpha} - \kappa\beta < -2\frac{v}{\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right) - \kappa\beta < 0,$$

and $\forall t \geq t_2$,

$$\alpha(t) < \alpha^+(t) = \alpha(t_0) - \frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)(t-t_2)\,;$$

$$\beta(t) < \beta_2^+(t) = -\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)$$
$$+\left(\beta^+(t_2) + \frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)\right)e^{-\kappa(t-t_2)},$$

and now we can find the time t_3, when $\beta_2^+(t_3) = 0$.

$$\beta_2^+(t_3) = -\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)$$
$$+\left(\beta^+(t_2) + \frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)\right)e^{-\kappa(t_3-t_2)} = 0$$

$$\Downarrow$$

$$t_3 = t_2 - \frac{1}{\kappa}\ln\left(\frac{\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}{\beta^+(t_2)+\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}\right)$$

To conclude, since $0 < \beta < \beta_2^+$ while in state B and past t_2, and since $\beta_2^+(t_3) = 0$, then $\exists t\,|t_2 \leq t \leq t_3$ such that $\beta(t) = 0$. By t_3, the lowest possible value for $\rho(t)$ is

$$\rho^-(t_3) = \rho(t_0) - v\,(1 - \cos(\alpha(t_0)))\,(t_3 - t_0)\,;$$

concluding the proof. ■

Corollary 3.5 (State B^- Exit Conditions)**.** *If at time $t = t_0$*

1. $-\frac{\pi}{2} < \alpha(t_0) \leq \beta(t_0) < 0$,
2. $\rho(t_0) > r_c$;

then

1. $\rho^-(t_0 + \frac{1}{\kappa}) = \rho(t_0) - \frac{v}{\kappa}\,(1 - \cos\,(\alpha(t_0)))$,
2. $a = \kappa\frac{v}{\rho^-(t_0+\frac{1}{\kappa})} = \frac{v}{\rho(t_0)}\frac{1}{\kappa - \frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}$,
3. $b = \frac{\beta(t_0)+a\left(\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\frac{\beta(t_0)}{e}+a\left(\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}$,
4. $t_2 = t_0 + \frac{1}{\kappa}\ln\,(b)$,

5. $\alpha^-(t_2) = \alpha(t_0) + \left(\frac{1}{e} + b\right)\beta(t_0) + a\left(\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)\right)(1 - b - \ln(b))$,

6. $\beta^+(t_2) = \frac{\beta(t_0)}{b}$,

7. $t_3 = t_2 - \frac{1}{\kappa}\ln\left(\frac{\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}{\beta^+(t_2)+\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}\right)$;

and either $\exists t_* \,|t_2 \le t_* \le t_3$ *such that* $\beta(t_*) = 0$, *or* $\exists t_* \,|t_0 < t_* \le t_3$ *such that* $\rho(t_*) \le r_c$.

Lemma 3.3.4 (State C Exit Conditions)**.** *If* $\kappa > 2\frac{v}{r_c}$, *and at time* $t = t_0$, $0 \le -\beta(t_0) < \alpha(t_0) < \frac{\pi}{2}$, *then*

1. $t_1 = t_0 + \frac{\rho(t_0)}{v}\ln\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c}\frac{\alpha(t_0)-\beta(t_0)}{2}\right)}\right)$,

2. $t_2 = t_0 + \frac{r_c}{2v}\ln\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c}\frac{\alpha(t_0)-\beta(t_0)}{2}\right)}\right)$,

3. $t3 = t_0 + \frac{r_c}{2v}\ln\left(-\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\beta(t_0)}{2}\right)}\right)$;

and either

1. $0 \le -\beta(t_1) < \alpha(t_1) < \frac{v}{\kappa r_c}(\alpha(t_0) - \beta(t_0))$, *or*

2. $\exists t_*,\ t_2 < t_* \le t_1 \,|\, -\beta(t_*) = \alpha(t_*)$, *and if* $\beta(t_0) \le \frac{v}{v-\kappa r_c}\alpha(t_0)$, *then* $0 < \alpha(t_*) < \frac{v}{\kappa r_c}(\alpha(t_0) - \beta(t_0))$ *or*

3. $\exists t_*,\ t_3 < t_* \le t_1 \,|\, -\beta(t_*) = \alpha(t_*)$, *and if* $\beta(t_0) > \frac{v}{v-\kappa r_c}\alpha(t_0)$, *then*

$$0 < \alpha(t_*) < \frac{v}{\kappa r_c}(\alpha(t_0) - \beta(t_0)) - \left(\beta(t_0) + \frac{v}{\kappa r_c}(\alpha(t_0) - \beta(t_0))\right) e^{-\frac{\kappa}{2}\frac{r_c}{v}\ln\left(-\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\beta(t_0)}{2}\right)}\right)}$$

or

4. $\exists t_*,\ t_0 \le t_* \le t_1 \,|\, \rho(t_*) \le r_c$.

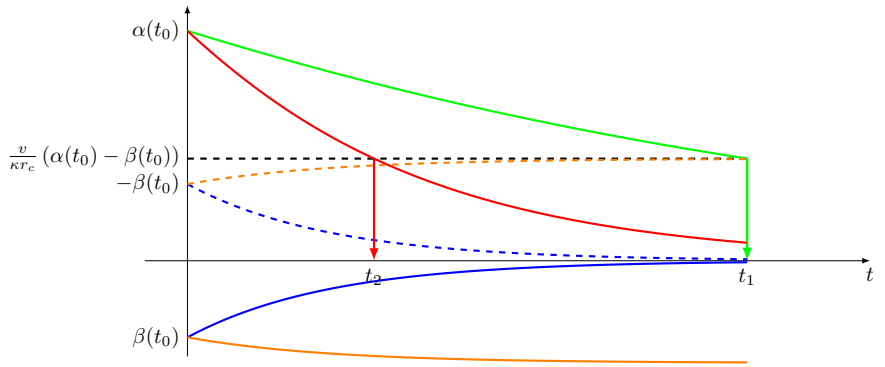
Figure 3.6 shows an outline of the proof.

Proof While in state C, $0 \le -\beta < \alpha \le \frac{\pi}{2}$, and

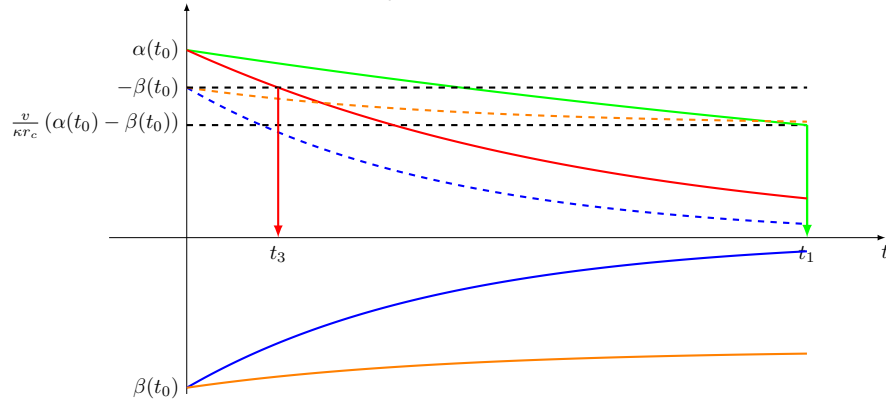
$$\dot{\rho} = v(\cos(\alpha) - \cos(\beta)) < 0,$$

$$\dot{\alpha} - \dot{\beta} = \kappa\beta \le 0,$$

$$\dot{\alpha} = \frac{v}{\rho}(\sin(\beta) - \sin(\alpha)) \le -\frac{v}{\rho}\sin(\alpha) < -\frac{v}{\rho(t_0)}\sin(\alpha) < 0,$$



(a) If $\beta(t_0) \geq \frac{v}{\kappa r_c}(\beta(t_0) - \alpha(t_0))$ and exiting the state with $\alpha(t_*) = -\beta(t_*)$, then $t_2 \leq t_* \leq t_1$ and $\alpha(t_*) \leq \frac{v}{\kappa r_c}(\alpha(t_0) - \beta(t_0))$



(b) If $\beta(t_0) < \frac{v}{\kappa r_c}(\beta(t_0) - \alpha(t_0))$ and exiting the state with $\alpha(t_*) = -\beta(t_*)$, then $t_3 \leq t_* \leq t_1$ and $\alpha(t_*) < -\beta^-(t_3)$

Figure 3.6: Lemma 3.3.4 proof outline. We find lower and upper bounds on α and β, and use the bounds on α to calculate t_1, t_2, and t_3. By t_1, either the system re-enters state C or exits the state with $\alpha = -\beta$, or $\rho = r_c$.

In other words, ρ, α and $\alpha - \beta$ are positive and shrinking.

$$\sin(\alpha) = 2\sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\alpha}{2}\right) = \frac{2\sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\alpha}{2}\right)}{\sin^2\left(\frac{\alpha}{2}\right)+\cos^2\left(\frac{\alpha}{2}\right)} = \frac{2\tan\left(\frac{\alpha}{2}\right)}{1+\tan^2\left(\frac{\alpha}{2}\right)}.$$

$$\Downarrow$$

$$\dot{\alpha} < -\frac{v}{\rho(t_0)}\frac{2\tan\left(\frac{\alpha}{2}\right)}{1+\tan^2\left(\frac{\alpha}{2}\right)}.$$

Substituting $u = \tan\left(\frac{\alpha}{2}\right)$,

$$\frac{du}{d\alpha} = \frac{1+u^2}{2}$$

$$\Downarrow$$

$$d\alpha = \frac{2du}{1+u^2}$$

$$\Downarrow$$

$$\frac{d\alpha}{dt} = \frac{2}{1+u^2}\frac{du}{dt}$$
$$\Downarrow$$
$$\frac{2}{1+u^2}\frac{du}{dt} < -\frac{v}{\rho(t_0)}\frac{2u}{1+u^2}$$
$$\Downarrow$$
$$\int \frac{du}{u} < -\frac{v}{\rho(t_0)}\int dt$$
$$\Downarrow$$
$$\ln(u) < C_1 - \frac{v}{\rho(t_0)}(t-t_0)$$
$$\Downarrow$$
$$\tan\left(\frac{\alpha}{2}\right) < \tan\left(\frac{\alpha(t_0)}{2}\right)e^{-\frac{v}{\rho(t_0)}(t-t_0)}$$
$$\Downarrow$$
$$\alpha(t) < \alpha^+(t) = 2\arctan\left(\tan\left(\frac{\alpha(t_0)}{2}\right)e^{-\frac{v}{\rho(t_0)}(t-t_0)}\right).$$

Now that we found the upper bound on α, we can find the last possible moment in time when

$$\alpha(t) = \frac{v}{\kappa r_c}\left(\alpha(t_0) - \beta(t_0)\right).$$
$$\alpha^+(t_1) = \frac{v}{\kappa r_c}\left(\alpha(t_0) - \beta(t_0)\right)$$
$$\Downarrow$$
$$2\arctan\left(\tan\left(\frac{\alpha(t_0)}{2}\right)e^{-\frac{v}{\rho(t_0)}(t_1-t_0)}\right) = \frac{v}{\kappa r_c}\left(\alpha(t_0) - \beta(t_0)\right)$$
$$\Downarrow$$
$$t_1 = t_0 + \frac{\rho(t_0)}{v}\ln\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c}\frac{\alpha(t_0)-\beta(t_0)}{2}\right)}\right).$$

We continue and calculate a lower bound on α,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) \geq -2\frac{v}{r_c}\sin(\alpha)$$
$$\Downarrow$$
$$\alpha(t) \geq \alpha^-(t) = 2\arctan\left(\tan\left(\frac{\alpha(t_0)}{2}\right)e^{-\frac{2v}{r_c}(t-t_0)}\right),$$

and the bounds on β,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta \leq -\kappa\beta$$

$$\Downarrow$$

$$\beta(t) \leq \beta^{+}(t) = \beta(t_0)e^{-\kappa(t-t_0)}$$

and

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta = -2\frac{v}{\rho}\left(\cos\left(\frac{\alpha+\beta}{2}\right)\sin\left(\frac{\alpha-\beta}{2}\right)\right) - \kappa\beta$$

$$\Downarrow$$

$$\dot{\beta} \geq -2\frac{v}{r_c}\sin\left(\frac{\alpha(t_0)-\beta(t_0)}{2}\right) - \kappa\beta \geq -2\frac{v}{r_c}\left(\frac{\alpha(t_0)-\beta(t_0)}{2}\right) - \kappa\beta$$

$$\Downarrow$$

$$\beta(t) \geq \beta^{-}(t) = -\frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right) + \left(\beta(t_0) + \frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right)\right)e^{-\kappa(t-t_0)}.$$

Next, we calculate time t_2, when

$$\alpha^{-}(t_2) = \frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right)$$

$$\Downarrow$$

$$2\arctan\left(\tan\left(\frac{\alpha(t_0)}{2}\right)e^{-\frac{2v}{r_c}(t2-t_0)}\right) = \frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right)$$

$$\Downarrow$$

$$t2 = t_0 + \frac{r_c}{2v}\ln\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c}\frac{\alpha(t_0)-\beta(t_0)}{2}\right)}\right),$$

and t_3, when

$$\alpha^{-}(t_3) = -\beta(t_0)$$

$$\Downarrow$$

$$2\arctan\left(\tan\left(\frac{\alpha(t_0)}{2}\right)e^{-\frac{2v}{r_c}(t3-t_0)}\right) = -\beta(t_0)$$

$$\Downarrow$$

$$t3 = t_0 + \frac{r_c}{2v}\ln\left(-\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\beta(t_0)}{2}\right)}\right).$$

If $\beta(t_0) \leq \frac{v}{v-\kappa r_c}\alpha(t_0)$, then $\forall t,\ |\ t_0 \leq t < t_2$, it is guaranteed that $-\beta(t) \leq \frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right) < \alpha(t)$, and therefore if $\exists t_* \mid \alpha(t_*) = \beta(t_*)$, then $\alpha(t_*) \leq \frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right)$. Similarly, if $\beta(t_0) > \frac{v}{v-\kappa r_c}\alpha(t_0)$, then $\forall t,\ -\dot{\beta}^{-}(t) < 0$ and therefore if $\exists t_* \mid \alpha(t_*) = \beta(t_*)$, then

$\alpha(t_*) \leq -\beta^-(t_3)$;

$$\beta^-(t_3) = -\frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right) + \left(\beta(t_0) + \frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right)\right) e^{-\frac{\kappa}{2}\frac{r_c}{v}\ln\left(-\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\beta(t_0)}{2}\right)}\right)};$$

concluding the proof. ■

Corollary 3.6 (State C^- Exit Conditions)**.** *If $\kappa > 2\frac{v}{r_c}$, and at time $t = t_0$, $0 \leq \beta(t_0) < -\alpha(t_0) < \frac{\pi}{2}$, then*

1. $t_1 = t_0 + \frac{\rho(t_0)}{v}\ln\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c}\frac{\alpha(t_0)-\beta(t_0)}{2}\right)}\right)$,
2. $t_2 = t_0 + \frac{r_c}{2v}\ln\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c}\frac{\alpha(t_0)-\beta(t_0)}{2}\right)}\right)$,
3. $t3 = t_0 + \frac{r_c}{2v}\ln\left(-\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\beta(t_0)}{2}\right)}\right)$;

and either

1. $\frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right) < \alpha(t_1) < -\beta(t_1) \leq 0$, *or*
2. $\exists t_*,\ t_2 < t_* \leq t_1 \mid -\beta(t_*) = \alpha(t_*)$, *and if* $\beta(t_0) \geq \frac{v}{v-\kappa r_c}\alpha(t_0)$, *then* $\frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right) < \alpha(t_*) < 0$ *or*
3. $\exists t_*,\ t_3 < t_* \leq t_1 \mid -\beta(t_*) = \alpha(t_*)$, *and if* $\beta(t_0) < \frac{v}{v-\kappa r_c}\alpha(t_0)$, *then*

   $$\frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right) - \left(\beta(t_0) + \frac{v}{\kappa r_c}\left(\alpha(t_0)-\beta(t_0)\right)\right) e^{-\frac{\kappa}{2}\frac{r_c}{v}\ln\left(-\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\beta(t_0)}{2}\right)}\right)} < \alpha(t_*) < 0$$

   or
4. $\exists t_*,\ t_0 \leq t_* \leq t_1 \mid \rho(t_*) \leq r_c$.

Lemma 3.3.5 (State D Exit Conditions)**.** *If $\kappa > 2\frac{v}{r_c}$ and at time $t = t_0$,*

1. $0 < \alpha(t_0) \leq -\beta(t_0) < \frac{\pi}{2}$,
2. $\rho(t_0) > r_c$;

then

1. $t_1 = t_0 + \frac{\rho(t_0)}{v}\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\alpha(t_0)}{4}\right)} - 1\right)$;
2. $t_2 = t_0 + \frac{1}{\kappa}\ln\left(-\frac{\beta(t_0)}{\alpha(t_0)}\right)$;

3. $t_3 = t_0 + \frac{\rho(t_0)}{v} \ln\left(1 - \frac{\alpha(t_0)}{\sin(\beta(t_0))}\right)$;

4. $\rho(t) > \rho(t_0)\ \forall t\ |t_0 < t \leq t_1$.

and either

1. $0 < \alpha(t_1) \leq \frac{\alpha(t_0)}{2}$ *and*

   $0 < \alpha(t_1) \leq -\beta(t_1) \leq -\beta(t_0) e^{-\left(\kappa \frac{\rho(t_0)}{v} - 2\right)\left(\frac{1+\tan^2\left(\frac{\alpha(t_0)}{4}\right)}{1-\tan^2\left(\frac{\alpha(t_0)}{4}\right)}\right)} < -\beta(t_0)$; *or*

2. $\exists t_*\ |t_2 \leq t_* \leq t_1$ *such that* $0 \leq -\beta(t_*) < \alpha(t_*) < \alpha(t_0)$; *or*

3. $\exists t_*\ |t_3 \leq t_* \leq t_1$ *such that* $\beta(t_0) e^{-\left(\kappa \frac{\rho(t_0)}{v} - 2\right) \ln\left(1 - \frac{\alpha(t_0)}{\sin(\beta(t_0))}\right)} \leq \beta(t_*) < \alpha(t_*) = 0$.

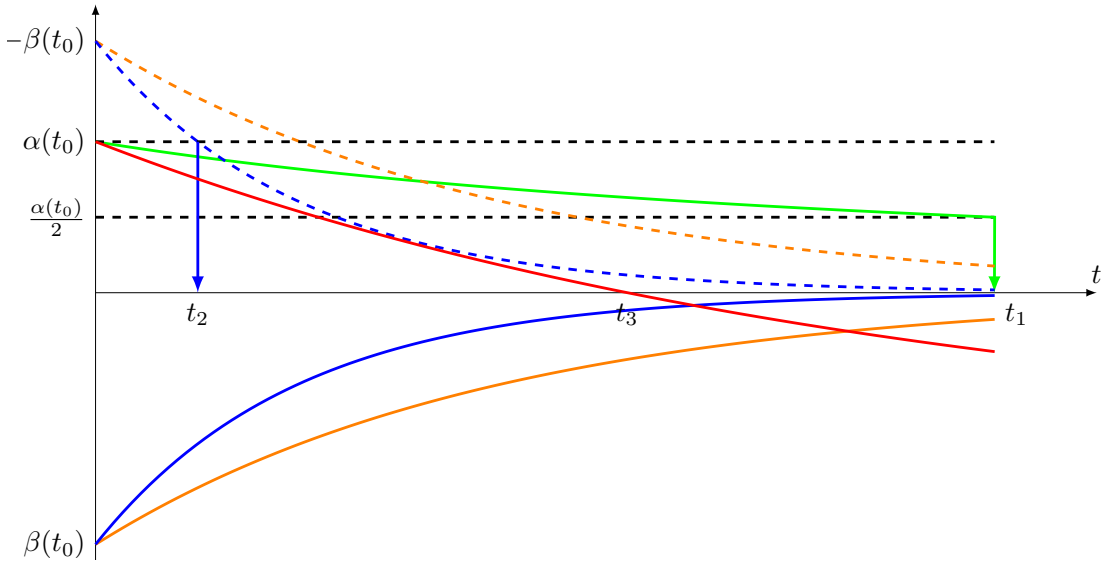


Figure 3.7: Lemma 3.3.5 proof outline. After finding an upper bound on ρ, we use this upper bound to find an upper bound for α and the moment t_1 where the upper bound reaches $\frac{\alpha(t_0)}{2}$. Then, we find the bounds on β, and show that both grow monotonously. Since $|\beta|$ shrinks with time, we can find the lower bound on α, and show that unless leaving the state with $\alpha > -\beta$ or $\alpha = 0$, the angles α, β continue to shrink.

Figure 3.7 shows an outline of the proof.

Proof While in state D, $0 < \alpha \leq -\beta \leq \frac{\pi}{2}$, and therefore

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) \geq 0,$$

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) \leq 0,$$

$$\dot{\alpha} - \dot{\beta} = \kappa\beta < 0,$$

and α shrinks as ρ grows, and the difference between α and β shrinks as well. Since $-\frac{\pi}{2} \leq \beta \leq -\alpha$ and $0 < \alpha$, we can find an upper bound on ρ,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) \leq v\left(1 - 0\right) = v$$

$$\Downarrow$$

$$\rho(t) \leq \rho^{+}(t) = \rho(t_0) + v\left(t - t_0\right)$$

We can now find an upper bound for α,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) \leq \frac{v}{\rho^{+}(t)}\left(\sin(\beta) - \sin(\alpha)\right) \leq -\frac{v}{\rho(t_0) + v\left(t - t_0\right)}\sin(\alpha)$$

$$\Downarrow$$

$$\frac{d\alpha}{dt} \leq -\frac{v}{\rho(t_0) + v\left(t - t_0\right)}\sin(\alpha)$$

$$\Downarrow$$

$$\int \frac{d\alpha}{\sin(\alpha)} \leq -\int \frac{v}{\rho(t_0) + v\left(t - t_0\right)}dt.$$

Substituting $w = \rho(t_0) + v\left(t - t_0\right)$, $\frac{dw}{dt} = v$, and

$$-\int \frac{v}{\rho(t_0) + v\left(t - t_0\right)}dt = -\int \frac{v}{w}\frac{dw}{v}$$

$$\Downarrow$$

$$\ln\left(\tan\left(\frac{\alpha}{2}\right)\right) \leq -\int \frac{dw}{w} = C_0 - \ln\left(\rho(t_0) + v\left(t - t_0\right)\right)$$

$$\Downarrow$$

$$\tan\left(\frac{\alpha}{2}\right) \leq e^{(C_0)}\left(e^{(\ln(\rho(t_0)+v(t-t_0)))}\right)^{-1} = \frac{C_1}{\rho(t_0) + v\left(t - t_0\right)}$$

$$\Downarrow$$

$$\alpha(t) \leq \alpha^{+}(t) = 2\arctan\left(\frac{\rho(t_0)\tan\left(\frac{\alpha(t_0)}{2}\right)}{\rho(t_0) + v\left(t - t_0\right)}\right).$$

Let t_1 such that $\alpha(t_1) \leq \frac{\alpha(t_0)}{2}$,

$$\frac{\alpha(t_0)}{2} = \alpha^{+}(t_1) = 2\arctan\left(\frac{\rho(t_0)\tan\left(\frac{\alpha(t_0)}{2}\right)}{\rho(t_0) + v\left(t_1 - t_0\right)}\right)$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{\rho(t_0)}{v}\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\alpha(t_0)}{4}\right)} - 1\right) = t_0 + \frac{\rho(t_0)}{v}\left(\frac{1 + \tan^2\left(\frac{\alpha(t_0)}{4}\right)}{1 - \tan^2\left(\frac{\alpha(t_0)}{4}\right)}\right)$$

We can bound $\beta(t)$,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta \leq -\kappa\beta$$

$$\Downarrow$$

$$\beta(t) \leq \beta^{+}(t) = \beta(t_0)e^{-\kappa(t-t_0)},$$

and since $\rho(t_0)$ is the lower bound on ρ,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta \geq \frac{v}{\rho(t_0)}\left(\sin(\beta) - \sin(-\beta)\right) - \kappa\beta \geq \left(\frac{2v}{\rho(t_0)} - \kappa\right)\beta.$$

$$\Downarrow$$

$$\beta(t) \geq \beta^{-}(t) = \beta(t_0)e^{\left(\frac{2v}{\rho(t_0)}-\kappa\right)(t-t_0)}. \tag{3.11}$$

We can now find t_2, a moment before the first opportunity for $-\beta < \alpha$,

$$\alpha(t_0) = -\beta^{+}(t_2) = -\beta(t_0)e^{-\kappa(t_2-t_0)}$$

$$\Downarrow$$

$$t_2 = t_0 + \frac{1}{\kappa}\ln\left(-\frac{\beta(t_0)}{\alpha(t_0)}\right).$$

If $\kappa > 2\frac{v}{r_c}$ and $\rho(t_0) > r_c$, then $\rho(t_0) > \frac{2v}{\kappa}$, and it is guaranteed that $|\beta(t)|$ shrinks with time; in particular,

$$\beta(t_0)e^{-\left(\kappa\frac{\rho(t_0)}{v}-2\right)\left(\frac{1+\tan^2\left(\frac{\alpha(t_0)}{4}\right)}{1-\tan^2\left(\frac{\alpha(t_0)}{4}\right)}\right)} \leq \beta(t_1) \leq \beta(t_0)e^{-\kappa\frac{\rho(t_0)}{v}\left(\frac{1+\tan^2\left(\frac{\alpha(t_0)}{4}\right)}{1-\tan^2\left(\frac{\alpha(t_0)}{4}\right)}\right)},$$

and therefore

$$|\beta(t_0)| > |\beta(t_1)|.$$

Since $\beta(t) > \beta(t_0)\ \forall t > t_0$,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) \geq \frac{v}{\rho(t_0)}\left(\sin(\beta(t_0)) - \alpha\right)$$

$$\Downarrow$$

$$\alpha(t) \geq \alpha^{-}(t) = \sin(\beta(t_0)) + \left(\alpha(t_0) - \sin(\beta(t_0))\right)e^{-\frac{v}{\rho(t_0)}(t-t_0)},$$

and the first opportunity for $\alpha = 0$ must come after time t_3,

$$0 = \sin(\beta(t_0)) + \left(\alpha(t_0) - \sin(\beta(t_0))\right)e^{-\frac{v}{\rho(t_0)}(t_3-t_0)}$$

$$\Downarrow$$

$$t_3 = t_0 + \frac{\rho(t_0)}{v} \ln\left(1 - \frac{\alpha(t_0)}{\sin(\beta(t_0))}\right),$$

at which time

$$\beta > \beta^-(t_3) = \beta(t_0) e^{\left(\frac{2v}{\rho(t_0)} - \kappa\right)(t_3 - t_0)} = \beta(t_0) e^{-\left(\kappa \frac{\rho(t_0)}{v} - 2\right) \ln\left(1 - \frac{\alpha(t_0)}{\sin(\beta(t_0))}\right)};$$

concluding the proof. ■

Corollary 3.7 (State D^- Exit Conditions)**.** *If $\kappa > 2\frac{v}{r_c}$ and at time $t = t_0$,*

1. $0 < -\alpha(t_0) \leq \beta(t_0) < \frac{\pi}{2}$,
2. $\rho(t_0) > r_c$;

then

1. $t_1 = t_0 + \frac{\rho(t_0)}{v}\left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\alpha(t_0)}{4}\right)} - 1\right)$;
2. $t_2 = t_0 + \frac{1}{\kappa}\ln\left(-\frac{\beta(t_0)}{\alpha(t_0)}\right)$;
3. $t_3 = t_0 + \frac{\rho(t_0)}{v}\ln\left(1 - \frac{\alpha(t_0)}{\sin(\beta(t_0))}\right)$;
4. $\rho(t) > \rho(t_0)\ \forall t\ |t_0 < t \leq t_1$.

and either

1. $0 < -\alpha(t_1) \leq -\frac{\alpha(t_0)}{2}$ *and*
   $0 < -\alpha(t_1) \leq \beta(t_1) \leq \beta(t_0) e^{-\left(\kappa \frac{\rho(t_0)}{v} - 2\right)\left(\frac{1+\tan^2\left(\frac{\alpha(t_0)}{4}\right)}{1-\tan^2\left(\frac{\alpha(t_0)}{4}\right)}\right)} < \beta(t_0)$; *or*
2. $\exists t_*\,|t_2 \leq t_* \leq t_1$ *such that* $0 \leq \beta(t_*) < -\alpha(t_*) < -\alpha(t_0)$; *or*
3. $\exists t_*\,|t_3 \leq t_* \leq t_1$ *such that* $0 = \alpha(t_*) < \beta(t_*) \leq \beta(t_0) e^{-\left(\kappa \frac{\rho(t_0)}{v} - 2\right)\ln\left(1 - \frac{\alpha(t_0)}{\sin(\beta(t_0))}\right)}$.

After analyzing all the exit conditions for all states, we can update Table 3.1 to create a new transition table, see Table 3.2.

Noticing that states B, C, B^-, and C^- can lead to capture, we make another observation, that there is a limit to how long the agent can fall behind the target and still be able to capture it.

Lemma 3.3.6. *$\alpha(t_0) = 0$, $\rho(t_0) > r_c \Rightarrow \rho(t) > r_c\ \forall t > t_0$.*

Proof The target and pursuer agent have the same speed. If the agent is on the target's path ($\alpha = 0$), then the shortest path towards the target is on the target's straight path, therefore the best the agent can do in terms of pursuing the target is to stay on the straight path, resulting in a constant ρ and never capturing the target. Any other course of action taken by the agent will increase ρ, and again result in never capturing the target. ■

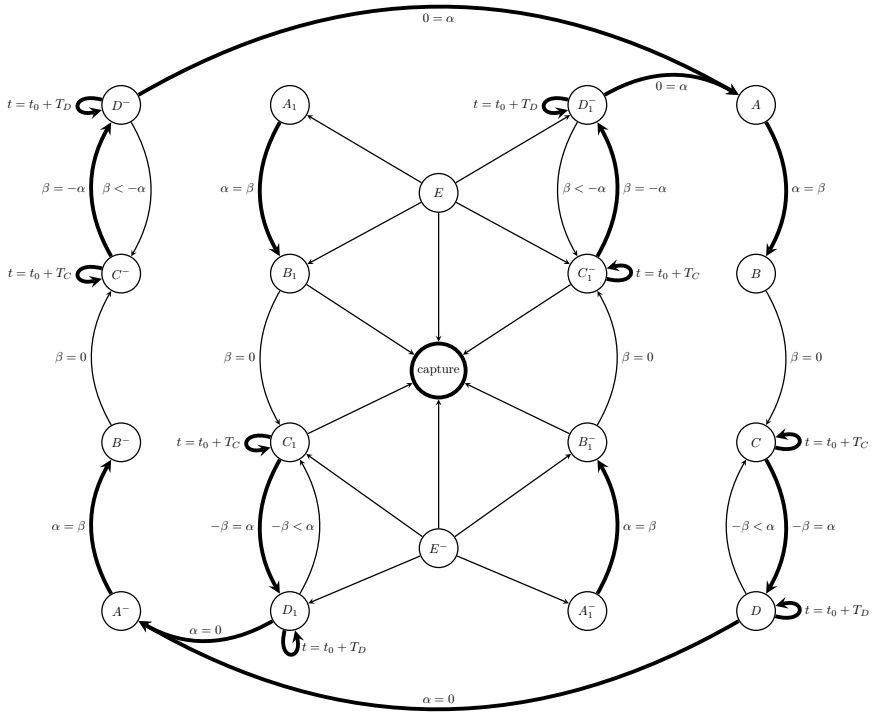


Figure 3.8: The Pursuit graph $\mathcal{G}$ is a DFSM.

As a conclusion form the previous lemma, if there is a transition from D or D^- to A^- or A, then a future capture is impossible. To adjust for this fact, we doubled the transition table from Table 3.2 such that E and E^- transition to prime states, marked by subscript 1, and the new D_1 or D_1^- transition to A^- or A. Figure 3.8 shows the result of this doubling.

State	System Configuration	Exit Condition	Transition
A	$0 \leq \alpha(t) < \beta(t) < \frac{\pi}{2}$	$\alpha = \beta$	B
B	$0 < \beta(t) \leq \alpha(t) < \frac{\pi}{2}$	$\rho = r_c$	Capture
		$\beta = 0$	C
C	$0 \leq -\beta < \alpha < \frac{\pi}{2}$	$t = t_0 + \frac{\rho(t_0)}{v} \ln \left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c} \frac{\alpha(t_0)-\beta(t_0)}{2}\right)} \right)$	C
		$\rho = r_c$	Capture
		$\alpha = \beta$	D
D	$0 < \alpha \leq -\beta \leq \frac{\pi}{2}$	$t = t_0 + \frac{\rho(t_0)}{v} \left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\alpha(t_0)}{4}\right)} - 1 \right)$	D
		$-\beta < \alpha$	C
		$\alpha = 0$	A^-
A^-	$-\frac{\pi}{2} < \beta(t) < \alpha(t) \leq 0$	$\alpha = \beta$	B^-
B^-	$-\frac{\pi}{2} < \alpha(t) \leq \beta(t) < 0$	$\rho = r_c$	Capture
		$\beta = 0$	C^-
C^-	$0 \leq \beta(t) < -\alpha(t) < \frac{\pi}{2}$	$t = t_0 + \frac{\rho(t_0)}{v} \ln \left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{v}{\kappa r_c} \frac{\alpha(t_0)-\beta(t_0)}{2}\right)} \right)$	C^-
		$\rho = r_c$	Capture
		$\alpha = \beta$	D^-
D^-	$0 < -\alpha(t) \leq \beta(t) < \frac{\pi}{2}$	$\beta < -\alpha$	C^-
		$t = t_0 + \frac{\rho(t_0)}{v} \left(\frac{\tan\left(\frac{\alpha(t_0)}{2}\right)}{\tan\left(\frac{\alpha(t_0)}{4}\right)} - 1 \right)$	D^-
		$\alpha = 0$	A
E, E^-	$\frac{\pi}{2} \leq \lvert\alpha\rvert \leq \pi$ *or* $\frac{\pi}{2} \leq \lvert\beta\rvert \leq \pi$	$0 \leq \alpha(t) < \beta(t) < \frac{\pi}{2}$	A
		$0 < \beta(t) \leq \alpha(t) < \frac{\pi}{2}$	B
		$0 \leq -\beta < \alpha < \frac{\pi}{2}$	C
		$0 < \alpha \leq -\beta < \frac{\pi}{2}$	D
		$-\frac{\pi}{2} < \beta(t) < \alpha(t) \leq 0$	A^-
		$-\frac{\pi}{2} < \alpha(t) \leq \beta(t) < 0$	B^-
		$0 \leq \beta(t) < -\alpha(t) < \frac{\pi}{2}$	C^-
		$0 < -\alpha(t) \leq \beta(t) < \frac{\pi}{2}$	D^-

Table 3.2: Transition Table for the PCBOUP problem.

Theorem 3.8 (Tracking)**.** *If*

1. $\gamma(t) = \max\{|\alpha(t)|, |\beta(t)|\}$,
2. $\kappa > 2\frac{v}{r_c}$, *and*
3. $\rho(t_0) > r_c$,

then either

1. $\exists T > 0 \mid \rho(T) \leq r_c$, *or*

2. $\exists R,\ 0 < R < \infty \mid \rho(t) < R\ ,\ \forall t,\ and\ \forall \varepsilon \mid 0 < \varepsilon < \frac{\pi}{2}\ , \exists T > 0 \mid T < t \Rightarrow \gamma(t) < \varepsilon\ .$

In other words, a unicycle agent (Equation 2.1) in pursuit of a target moving in a straight line (Equation 3.3) with the bearing-only control law (Equation 3.5) governing its steering, either captures the target or asymptotically reaches the target's path, and the distance between the target and agent is bound from above by a finite value.

Figure 3.8 shows $\mathcal{G}$, the resulting DFSM of the discussion so far. Each state has a time limit that results in exiting the state when the time limit expires. The bold edges represent transitions that entail a diminishing of γ before exiting the state. Since no loop on the graph is possible without traversing a bold edge, Then γ must shrink every loop. Summing the time upper bounds for each state traversed until the eventual $\gamma < \varepsilon$ results in T. The maximal added distance from states A, D, and their prime and symmetrical counterparts, accumulates to a finite value less than some finite R.

Proof Except for the capture state at which the pursuit is concluded, each of the system states has a finite time limit, and the state must transition when the limit lapses, even if re-entering the same state. Any initial condition other than $0 \leq |\alpha| < \frac{\pi}{2}$, $0 \leq |\beta| < \frac{\pi}{2}$ falls into one of the prime states (the inner states with the subscript 1) in $\mathcal{G}$ (Figure 3.8) in finite time, resulting in either capture, remaining in a loop between states C_1, D_1 or their symmetrical states C_1^-, D_1^-, or an eventual exit towards the outer states with the transition into state A or A^- with $\alpha = 0$.

Entering State A with $\rho = \rho_0$, $\alpha = 0$ and $\beta = \beta_0$ results in an exit with γ smaller than $\beta^+(t_1)$,

$$\gamma_A \leq \beta^+(t_1) = \beta_0 e^{-\left(\kappa - \frac{v}{\rho_0}\right)(t_1 - t_0)} - \frac{\frac{v}{\rho_0}}{\kappa - \frac{v}{\rho_0}} \sin(0) \left(1 - e^{-\left(\kappa - \frac{v}{\rho_0}\right)(t_1 - t_0)}\right)$$

$$= \beta_0 e^{-\left(\kappa - \frac{v}{\rho_0}\right)(t_1 - t_0)} = \beta_0 e^{-\left(\kappa - \frac{v}{\rho_0}\right)\left(t_0 + \frac{1}{\kappa}\ln\left(\frac{2}{1 + \frac{0}{\beta_0}}\right) - t_0\right)} = \frac{\beta_0}{2^{\frac{1}{\kappa}\left(\kappa - \frac{v}{\rho_0}\right)}}$$

$$\Downarrow$$

$$\gamma_A \leq 2^{\frac{v}{\kappa\rho_0}} \frac{\beta_0}{2}. \tag{3.12}$$

and the state transitions to B with $\alpha = \beta < \gamma_A$.

Exiting state B, γ is less than γ_A, and $\rho < \rho^+(t_2)$, see Figure 3.9.

$$\rho^+(t_2) = \rho_A + v\left(\cos\left(\alpha^-(t_2)\right) - \cos\left(\beta^+(t_2)\right)\right)(t_2 - t_0)$$

and the state transitions to state C with $\alpha = \gamma_B < \gamma_A$ and $\beta = 0$. Figure 3.10 shows the evolution of α and β after entering state C from state B.

Any consecutive loop between state C and itself follows the schema presented in Figure 3.6. Given $\kappa > 2\frac{v}{r_c}$, each iteration shrinks γ, the greater the κ the greater the

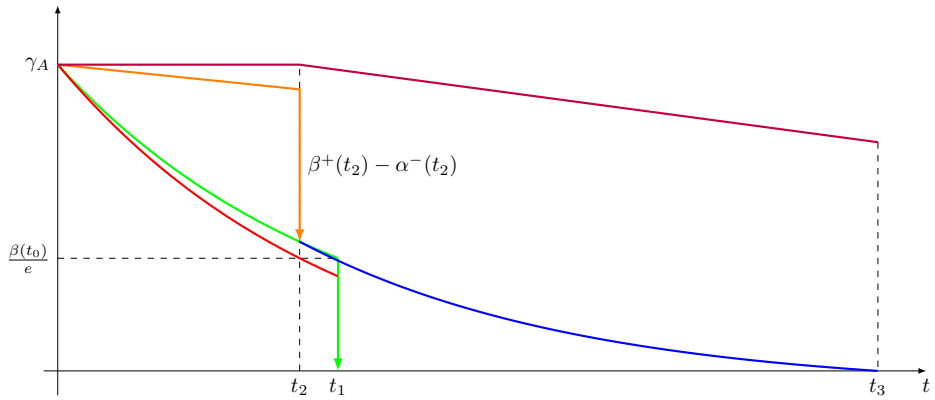


Figure 3.9: State B. Similar to Figure 3.5, the initial condition $\alpha_0 = \beta_0 = \gamma_A$ from the transition from state A results in exiting state B with $\gamma_B < \gamma_A$.

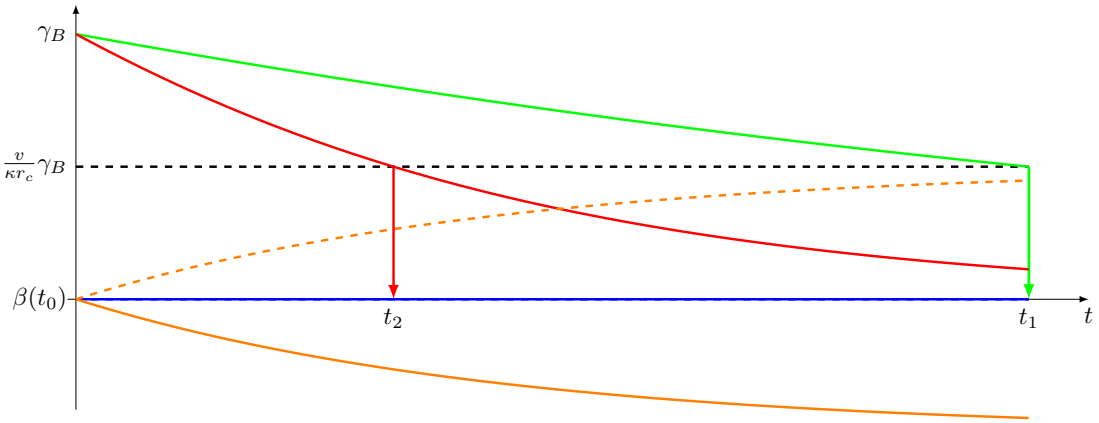


Figure 3.10: State C after state B. Similar to Figure 3.6, with the initial conditions $\alpha_0 = \gamma_B$ and $\beta_0 = 0$. By t_1, a transition must have happened, either to state D or re-entry to state C, with $\gamma_C \leq \frac{v}{\kappa r_c}\gamma_B$.

step in each iteration,

$$\gamma_C^{i+1} < 2\frac{v}{\kappa r_c}\gamma_C^i < \gamma_C^i.$$

Ultimately, $\gamma_C < \frac{v}{\kappa r_c}\gamma_B < \frac{v}{\kappa r_c}\gamma_A$ for any amount of self-loops, and since $\dot{\alpha} < \dot{\beta}$ while in state C, a transition to state D must occur in finite time, with $\alpha = -\beta = \gamma_C < \frac{v}{\kappa r_c}\gamma_A$.

As shown in Figure 3.11, a transition back to state C is possible immediately, with $\gamma_D^C = \gamma_C$. Yet on return from state C, $\gamma < 2\frac{v}{\kappa r_c}\gamma_C$, as discussed above, and the result of the $D \to C \to D$ loop is $\gamma_C^{i+1} < 2\frac{v}{\kappa r_c}\gamma_D^C = 2\frac{v}{\kappa r_c}\gamma_C^i$ for each return to state D from C. If not returning to state C, the first opportunity to transition out of D is to state A^-, which happens at t_3, when

$$\gamma_D^A < -\beta^-(t_3) = -\beta(t_0)e^{-\left(\kappa\frac{\rho(t_0)}{v}-2\right)\ln\left(1-\frac{\alpha(t_0)}{\sin(\beta(t_0))}\right)} = \gamma_C\left(\frac{-\sin(\gamma_C)}{-\sin(\gamma_C)-\gamma_C}\right)^{\left(\kappa\frac{\rho(t_0)}{v}-2\right)}$$

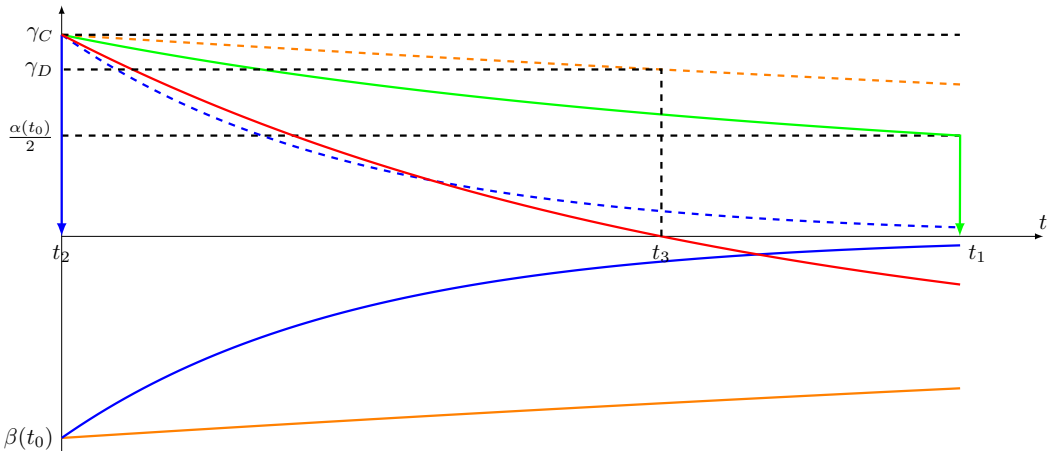


Figure 3.11: State D after state C. Similar to Figure 3.7, with the initial conditions $\alpha_0 = -\beta_0 = \gamma_C$. By t_1, a transition must have happened, either to state A, back to C, or re-entry to state D.

$$= \gamma_C \left(\frac{\sin(\gamma_C)}{\sin(\gamma_C)+\gamma_C}\right)^{\left(\kappa\frac{\rho(t_0)}{v}-2\right)} < 2^{\left(2-\kappa\frac{\rho(t_0)}{v}\right)}\gamma_C = \frac{4}{2^{\kappa\frac{\rho(t_0)}{v}}}\gamma_C < \frac{4}{2^{2\frac{\rho(t_0)}{r_c}}}\gamma_C$$

$$\Downarrow$$

$$\gamma_D^A < 4^{-\frac{\rho(t_0)-r_c}{r_c}}\gamma_C < \gamma_C. \tag{3.13}$$

The self loop of state D happens at time t_1, and results in

$$\gamma_D^{i+1} < -\beta^-(t_1) = \gamma_D^i e^{-\left(\kappa\frac{\rho(t_0)}{v}-2\right)\left(\frac{1+\tan^2\left(\frac{\alpha(t_0)}{4}\right)}{1-\tan^2\left(\frac{\alpha(t_0)}{4}\right)}\right)} < \gamma_D^i e^{-2\frac{\rho_D^i-r_c}{r_c}} < \gamma_D^i. \tag{3.14}$$

The loop $A \to B \to C \to D \to A^- \to B^- \to C^- \to D^- \to A$, as seen in Figure 3.8, therefore results in

$$\gamma_A^{i+1},$$

and according to Equation 3.12,

$$\gamma_A^{i+1} \le 2^{\frac{v}{\kappa\rho_D}}\frac{\gamma_D^A}{2} = 2^{-\left(1-\frac{v}{\kappa\rho_D}\right)}\gamma_D^A;$$

according to Equation 3.13,

$$\gamma_A^{i+1} \le 2^{-\left(1-\frac{v}{\kappa\rho_D}\right)}2^{\left(2-\kappa\frac{\rho_C}{v}\right)}\gamma_C = 2^{\left(2-\kappa\frac{\rho_C}{v}\right)-\left(1-\frac{v}{\kappa\rho_D}\right)}\gamma_C = 2^{\left(1+\frac{v}{\kappa\rho_D}-\kappa\frac{\rho_C}{v}\right)}\gamma_C,$$

and ultimately,

$$\gamma_A^{i+1} \le 2^{\left(1+\frac{v}{\kappa\rho_D}-\kappa\frac{\rho_C}{v}\right)}\frac{v}{\kappa r_c}\gamma_A^i < 2^{\left(\frac{v}{\kappa\rho_D}-\kappa\frac{\rho_C}{v}\right)}\gamma_A^i < 2^{\left(\frac{r_c}{2\rho_D}-\frac{2\rho_C}{r_c}\right)}\gamma_A^i = 2^{\left(\frac{r_c^2-4\rho_C\rho_D}{2r_c\rho_D}\right)}\gamma_A^i$$

$$\Downarrow$$

$$\gamma_A^{i+1} < 2^{\left(\frac{r_c^2-4r_c^2}{2r_c\rho_D}\right)}\gamma_A^i = 2^{-\frac{3}{2}\frac{r_c}{\rho_D}}\gamma_A^i \leq \gamma_A^i, \tag{3.15}$$

and we have shown that any of cycle in graph $\mathcal{G}$ must include at least one type of loop that decreases in γ, $\forall\kappa > 2\frac{v}{r_c}$.

Notice that the only states where the distance between the agent and the target grows are states A, D and their variations A^-, A_1, D_1^- etc. While the transition from state A is exclusive to state B, state D can transition towards state A^-, state C, and self-loop to state D.

Leaving state D increases the distance between the pursuing agents and the target by the maximum relative velocity times the maximal time spent in state D. Assuming all other states have no effect on ρ and γ, when actually γ is non-increasing in all states, we isolate and amplify the contribution of state D to the overall increase in ρ by analyzing the increase in ρ as result of a perpetual self loop in state D.

Let ρ_D^i be the distance on entering the self-loop iteration and ρ_D^{i+1} when exiting,

$$\rho_D^{i+1} < \rho_D^i + v\left(\cos(0) - \cos\left(\gamma_D^i\right)\right)(t_1 - t_0)$$

$$= \rho_D^i + v\left(1 - \cos\left(\gamma_D^i\right)\right)\frac{\rho_D^i}{v}\left(\frac{\tan\left(\frac{\alpha_D^i}{2}\right)}{\tan\left(\frac{\alpha_D^i}{4}\right)} - 1\right)$$

$$\leq \rho_D^i + \rho_D^i\left(1 - \cos\left(\gamma_D^i\right)\right)\left(\frac{1 + \tan\left(\frac{\gamma_D^i}{4}\right)}{1 - \tan\left(\frac{\gamma_D^i}{4}\right)}\right)$$

$$= \rho_D^i\left(1 + \left(1 - \cos\left(\gamma_D^i\right)\right)\left(\frac{1 + \left(\frac{1-\cos\left(\frac{\gamma_D^i}{2}\right)}{1+\cos\left(\frac{\gamma_D^i}{2}\right)}\right)}{1 - \left(\frac{1-\cos\left(\frac{\gamma_D^i}{2}\right)}{1+\cos\left(\frac{\gamma_D^i}{2}\right)}\right)}\right)\right)$$

$$= \rho_D^i\left(1 + \left(1 - \cos\left(\gamma_D^i\right)\right)\left(\frac{\frac{1+\cos\left(\frac{\gamma_D^i}{2}\right)+1-\cos\left(\frac{\gamma_D^i}{2}\right)}{1+\cos\left(\frac{\gamma_D^i}{2}\right)}}{\frac{1+\cos\left(\frac{\gamma_D^i}{2}\right)-1+\cos\left(\frac{\alpha_D^i}{2}\right)}{1+\cos\left(\frac{\gamma_D^i}{2}\right)}}\right)\right)$$

$$= \rho_D^i \left(1 + \left(1 - \cos\left(\gamma_D^i\right)\right) \frac{2}{2\cos\left(\frac{\gamma_D^i}{2}\right)}\right) = \rho_D^i \left(1 + \frac{1 - \cos\left(\gamma_D^i\right)}{\cos\left(\frac{\gamma_D^i}{2}\right)}\right)$$

$$= \rho_D^i \left(\frac{\cos\left(\frac{\gamma_D^i}{2}\right) + 1 - \left(1 - 2\sin^2\left(\frac{\gamma_D^i}{2}\right)\right)}{\cos\left(\frac{\gamma_D^i}{2}\right)}\right) = \rho_D^i \left(\frac{\cos\left(\frac{\gamma_D^i}{2}\right) + 2\sin^2\left(\frac{\gamma_D^i}{2}\right)}{\cos\left(\frac{\gamma_D^i}{2}\right)}\right)$$

$$\Downarrow$$

$$\rho_D^{i+1} < \rho_D^i \left(1 + 2\sin\left(\frac{\gamma_D^i}{2}\right)\tan\left(\frac{\gamma_D^i}{2}\right)\right) \tag{3.16}$$

We used t_1, the longest time possible to remain in state D, and $\alpha_0 = \gamma$ since any smaller α results in a smaller exit ρ. The addition to ρ as result of the $i+1$ iteration of the self loop is therefore

$$\rho_D^{i+1} - \rho_D^i < \rho_D^i \left(1 + 2\sin\left(\frac{\gamma_D^i}{2}\right)\tan\left(\frac{\gamma_D^i}{2}\right)\right) - \rho_D^i = 2\rho_D^i \sin\left(\frac{\gamma_D^i}{2}\right)\tan\left(\frac{\gamma_D^i}{2}\right),$$

and with $\gamma_D < \frac{\pi}{2}$,

$$\rho_D^{i+1} - \rho_D^i < \rho_D^i \gamma_D^i = d_i.$$

The total contribution of state D to ρ is therefore

$$L_D = \sum_{i=0}^{\infty} d_i$$

Applying d'Alembert's ratio test on the series produces the following result,

$$\lim_{n\to\infty}\left|\frac{d_{n+1}}{d_n}\right| = \lim_{n\to\infty}\left|\frac{\rho_D^{n+1}\gamma_D^{n+1}}{\rho_D^n\gamma_D^n}\right|,$$

and with equations 3.16 and 3.14,

$$\lim_{n\to\infty}\left|\frac{\rho_D^{n+1}\gamma_D^{n+1}}{\rho_D^n\gamma_D^n}\right| < \lim_{n\to\infty}\left|\frac{\rho_D^n\left(1 + 2\sin\left(\frac{\gamma_D^n}{2}\right)\tan\left(\frac{\gamma_D^n}{2}\right)\right)\gamma_D^n e^{-2\frac{\rho_D^n - r_c}{r_c}}}{\rho_D^n\gamma_D^n}\right|$$

$$= \lim_{n\to\infty}\left|\left(1 + 2\sin\left(\frac{\gamma_D^n}{2}\right)\tan\left(\frac{\gamma_D^n}{2}\right)\right)e^{-2\frac{\rho_D^n - r_c}{r_c}}\right| < \lim_{n\to\infty}\left|\left(1 + \gamma_D^n\right)e^{-2\frac{\rho_D^0 - r_c}{r_c}}\right|$$

$$= \lim_{n\to\infty}\left|\left(1 + \gamma_D^0 e^{-2n\frac{\rho_D^0 - r_c}{r_c}}\right)e^{-2\frac{\rho_D^0 - r_c}{r_c}}\right| = \lim_{n\to\infty} e^{-2\frac{\rho_D^0 - r_c}{r_c}} < 1,$$

therefore a finite L_D exists.

We shall now assess the contribution of state A to the growth of ρ. Let us assume

that all other states in $\mathcal{G}$ (Figure 3.8) make no contribution to γ and ρ, and if any changes occur, they happen exclusively in state A. In this case there is no actual meaning to leaving the state, other than having α miraculously return to 0 for the next iteration. If at the initial entry, $\rho = \rho_0$, $\beta = \beta_0$, and $\alpha = 0$, then

$$\gamma(t) < \beta^+(t) = \beta_0 e^{-\left(\kappa - \frac{v}{\rho_0}\right)(t-t_0)} - \frac{\frac{v}{\rho_0}}{\kappa - \frac{v}{\rho_0}} \sin(0) \left(1 - e^{-\left(\kappa - \frac{v}{\rho_0}\right)(t-t_0)}\right)$$

$$= \beta_0 e^{-\left(\kappa - \frac{v}{\rho_0}\right)(t-t_0)} < \beta_0 e^{-\left(\kappa - \frac{v}{r_c}\right)(t-t_0)} = \gamma_A^+(t).$$

Since the $\gamma_A^+(t)$ rate of decay is constant, and remains the same regardless of the initial conditions on entering the state other than $\beta_0 = \gamma_A^+(t_0)$, we can arbitrarily choose when to leave and re-enter the state, so we choose $t_1 = t_0 + \frac{1}{\kappa}\ln(2)$, and the change per iteration for γ becomes

$$\gamma_n^A = 2^{\left(\frac{v}{\kappa r_c} - 1\right)} \gamma_{n-1}^A = 2^{n\left(\frac{v}{\kappa r_c} - 1\right)} \beta_0. \tag{3.17}$$

The upper bound on the distance between pursuing agent and target on the nth iteration becomes

$$\rho_n^A = \rho_{n-1}^A + \frac{v}{\kappa}\left(1 - \cos\left(\gamma_{n-1}^A\right)\right)\ln(2),$$

and the distance gained per iteration is

$$\rho_n^A - \rho_{n-1}^A = \frac{v}{\kappa}\left(1 - \cos\left(\gamma_{n-1}^A\right)\right)\ln(2) = 2\ln(2)\frac{v}{\kappa}\sin^2\left(\frac{\gamma_{n-1}^A}{2}\right)$$

$$< 2\ln(2)\frac{v r_c}{2v}\left(\frac{\gamma_{n-1}^A}{2}\right)^2 = \frac{r_c}{4}\ln(2)\left(2^{(n-1)\left(\frac{v}{\kappa r_c} - 1\right)}\beta_0\right)^2 = \frac{r_c}{4}\ln(2)\, 2^{2(n-1)\left(\frac{v}{\kappa r_c} - 1\right)}\beta_0^2$$

$$< \frac{r_c}{4}\ln(2)\, 2^{2(n-1)\left(\frac{1}{2} - 1\right)}\beta_0^2 = \frac{r_c}{4}\ln(2)\,\beta_0^2 2^{(1-n)} = \frac{\ln(2)}{2} r_c \beta_0^2 \frac{1}{2^n} = A_n.$$

Summing all A_n will result in an upper bound to all the contribution to the growth of ρ by all variances of state A, on any possible path on the graph $\mathcal{G}$ (Figure 3.8). Let us denote this bound by L_A,

$$L_A = \sum_{n=0}^{\infty} A_n = \rho_\infty^A - \rho_0^A.$$

$$\sum_{n=0}^{\infty} A_n = \sum_{n=0}^{\infty}\left(\frac{\ln(2)}{2} r_c \beta_0^2 \frac{1}{2^n}\right) = \frac{\ln(2)}{2} r_c \beta_0^2 \sum_{n=0}^{\infty}\frac{1}{2^n}.$$

Notice that

$$\frac{1}{2}\sum_{n=1}^{\infty}\frac{1}{2^n} = \sum_{n=1}^{\infty}\frac{1}{2^{n+1}} = \sum_{n=2}^{\infty}\frac{1}{2^n} = \sum_{n=1}^{\infty}\frac{1}{2^n} - \frac{1}{2}$$

$$\Downarrow$$

$$\frac{1}{2} = \frac{1}{2}\sum_{n=1}^{\infty}\frac{1}{2^n}$$
$$\Downarrow$$
$$\sum_{n=1}^{\infty}\frac{1}{2^n} = 1$$
$$\Downarrow$$
$$\sum_{n=0}^{\infty}\frac{1}{2^n} = 2,$$

and therefore

$$L_A = \ln(2)\, r_c \beta_0^2. \tag{3.18}$$

Given the previous analysis of graph $\mathcal{G}$ (Figure 3.8), the only states that add to the initial distance $\rho(t=0)$ are states A, D, and their variants. The maximal contribution of any of those states combined is less than $R = L_A + L_D$, therefore,

$$\exists R,\ 0 < R < \infty \mid \rho(t) < R\ ,\ \forall t,$$

concluding the proof. ■

Corollary 3.9. *If*

1. $\gamma(t) = \max\{|\alpha(t)|, |\beta(t)|\}$,
2. $\kappa > 4\frac{v}{r_c}$, *and*
3. $\rho(t_0) > r_c$,

then $\forall t$, $\rho(t) - \rho(t_0) < 4r_c$.

Proof Returning to the proof of Theorem 3.8, if $\kappa > 4\frac{v}{r_c}$, then from Equation 3.11, the global upper bound on γ while in state D is

$$-\beta(t_0)e^{-\left(\kappa - 2\frac{v}{\rho(t_0)}\right)(t-t_0)} < \gamma_D^+(t_0)e^{-\left(\kappa-2\frac{v}{r_c}\right)(t-t_0)} = \gamma_D^+(t).$$

If we sample $\gamma_D^+(t)$ once every $\frac{1}{\kappa}\ln(2)$ time, we generate the series

$$\gamma_n^D = \gamma_{n-1}^D e^{-\left(\kappa-2\frac{v}{r_c}\right)\frac{1}{\kappa}\ln(2)} = \gamma_{n-1}^D 2^{\left(2\frac{v}{\kappa r_c}-1\right)} = -\beta(t_0)2^{n\left(2\frac{v}{\kappa r_c}-1\right)},$$

and the added distance per iteration is

$$\rho_n^D - \rho_{n-1}^D = v\left(\cos(0) - \cos(\gamma_{n-1}^D)\right)\frac{1}{\kappa}\ln(2) = \frac{v}{\kappa}\left(1 - \cos(\gamma_{n-1}^D)\right)\ln(2)$$

$$= 2\ln(2)\frac{v}{\kappa}\sin^2\left(\frac{\gamma_{n-1}^D}{2}\right) < 2\ln(2)\frac{r_c}{4}\left(\frac{-\beta(t_0)2^{(n-1)\left(2\frac{v}{\kappa r_c}-1\right)}}{2}\right)^2$$

$$= \frac{r_c}{8}\ln(2)\beta^2(t_0)2^{-2(n-1)\left(1-2\frac{v}{\kappa r_c}\right)} < \frac{r_c}{8}\ln(2)\beta^2(t_0)\frac{1}{2^{(n-1)}} = \frac{r_c}{4}\ln(2)\beta^2(t_0)\frac{1}{2^n} = d_n.$$

Summing all d_n, we arrive at an upper bound to the contribution of state D to the growth of ρ,

$$L_D = \sum_{n=0}^{\infty} d_n = \rho_\infty^D - \rho_0^D.$$

$$\sum_{n=0}^{\infty} d_n = \frac{r_c}{4}\ln(2)\beta^2(t_0)\sum_{n=0}^{\infty}\frac{1}{2^n} = \frac{r_c}{2}\ln(2)\beta^2(t_0).$$

Adding L_A (Equation 3.18), and since $\beta^2(t_0) < \frac{\pi}{2}$,

$$L_A + L_D < r_c\ln(2)\frac{\pi^2}{4} + \frac{r_c}{2}\ln(2)\frac{\pi^2}{4} = \frac{3}{8}r_c\ln(2)\pi^2.$$

and if $\rho(t_0)$ is the distance between the target and the agent when first leaving one of the states E or E^-, then the maximal distance between the agent and the target is $\rho(t_0) + \frac{3}{8}r_c\ln(2)\pi^2$. If we take the initial conditions such that the system remains in state E or E^- for the longest time possible, (see Lemma 3.3.1), we get

$$R < v\left(\cos(0) - \cos(\pi)\right)\left(\frac{1}{\kappa}\ln\left(\frac{\pi - 2\frac{v}{\kappa r_c}}{\frac{\pi}{2} - 2\frac{v}{\kappa r_c}}\right)\right) + \frac{3}{8}r_c\ln(2)\pi^2$$

$$< \frac{r_c}{2}\ln\left(\frac{\pi-1}{\frac{\pi}{2}-1}\right) + \frac{3}{8}r_c\ln(2)\pi^2 = \frac{r_c}{2}\left(\ln\left(\frac{\pi}{\pi-2}+1\right) + \frac{3}{4}\ln(2)\pi^2\right) < 4r_c$$

Q.E.D. ■

3.4 Capture

This section extends the previous analysis to explore the *capture regions*, i.e. the initial conditions in (ρ, α, β) space from which the pursuing agent may capture the target.

Problem Statement: Given r_c, v, κ, find $\Gamma \subset ([0,\infty), (-\pi,\pi], (-\pi,\pi])$ such that if

1. $\exists t_c \mid \rho(t_c) \leq r_c$
2. $p_a(t_0) = -\left(\rho_0\cos(\alpha_0), \rho_0\sin(\alpha_0)\right)^T$, and
3. $\theta(t_0) = \alpha_0 - \beta_0$,

then $(\rho_0, \alpha_0, \beta_0) \in \Gamma$.

To solve the problem stated above, we extend the transition table (Table 3.2) by splitting state E, previously assigned to all $\cos(\alpha) \leq 0$ or $\cos(\beta) \leq 0$, to the mutually exclusive states detailed in Table 3.3. The cases where $\alpha(t_0) = \pi$ and either $\sin(\beta(t_0)) = 0$ or $\sin(\beta(t_0)) \neq 0$, where left out of the discussion due to the results of the lemmas 3.4.1 and 3.4.2.

State	System Configuration
F	$\frac{\pi}{2} \leq \beta \leq \pi$
F^-	$-\pi < \beta \leq -\frac{\pi}{2}$
W	$\cos(\alpha) \leq 0$ and $0 < \sin(\alpha) \leq \sin(\beta) < 1$
W^-	$\cos(\alpha) \leq 0$ and $-1 < \sin(\beta) \leq \sin(\alpha) < 0$
X	$\cos(\alpha) \leq 0$ and $0 \leq \sin(\beta) < \sin(\alpha) \leq 1$
X^-	$\cos(\alpha) \leq 0$ and $-1 \leq \sin(\alpha) < \sin(\beta) \leq 0$
Y	$-\frac{\pi}{2} < \beta \leq -\frac{\pi}{3}$, and $\frac{\pi}{2} \leq \alpha < \pi$
Y^-	$\frac{\pi}{3} \leq \beta < \frac{\pi}{2}$, and $-\pi < \alpha \leq -\frac{\pi}{2}$
Z	$-\frac{\pi}{3} < \beta < 0$, and $\frac{\pi}{2} \leq \alpha < \pi$
Z^-	$0 < \beta < \frac{\pi}{3}$, and $-\pi < \alpha \leq -\frac{\pi}{2}$

Table 3.3: Capturing Extension States of PCBOUP.

Lemma 3.4.1. *If $\alpha(t_0) = \pi$ and $\beta(t_0) = 0$, then $\alpha(t_0) \equiv \pi$, $\beta \equiv 0$, and the agent captures the target in $T = \frac{\rho(t_0) - r_c}{2v}$ time.*

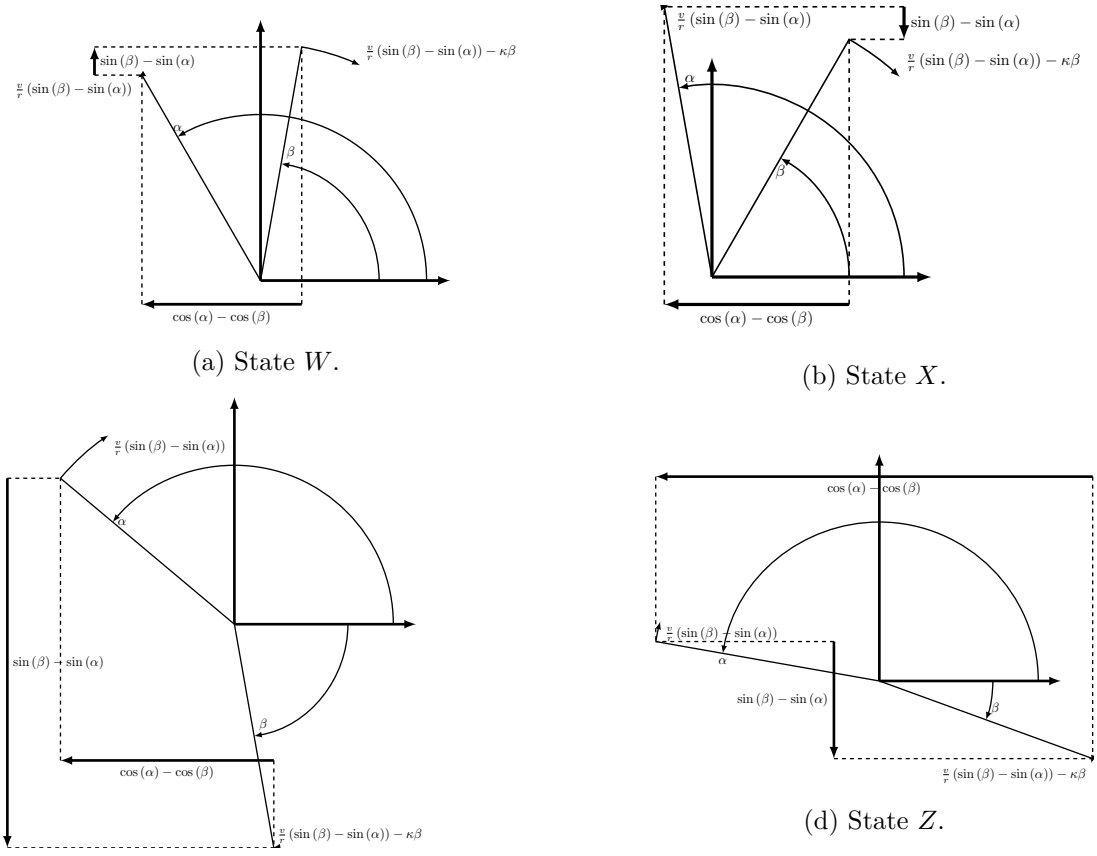
Proof Equations 2.3, 2.5 ensure $\alpha(t_0) \equiv \pi$ and $\beta \equiv 0$. From 2.4,

$$\int \dot{\rho} dt = \int v\left(\cos(\alpha) - \cos(\beta)\right) dt = -2v \int dt$$

$$\Downarrow$$

$$\rho(t) = \rho(t_0) - 2v(t - t_0)$$

$$\Downarrow$$



(a) State W.

(b) State X.

(c) State Y.

(d) State Z.

Figure 3.12: The primary capture states, defined by the angle couple α, β, and their angular velocities.

$$\rho(t_0 + T) = \rho(t_0) - 2v(t_0 + T - t_0) = \rho(t_0) - 2vT = r_c$$

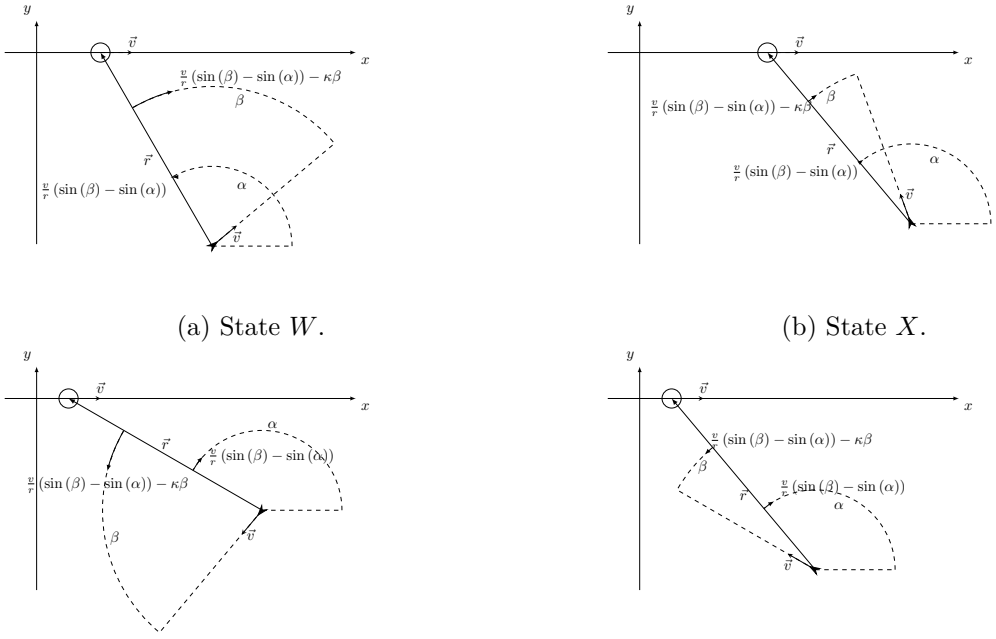
$$\Downarrow$$

$$T = \frac{\rho(t_0) - r_c}{2v}$$

Lemma 3.4.2. *If $\alpha(t_0) = \pi$ and $\beta(t_0) \neq 0$, then $\dot{\alpha}(t_0) \neq 0$ and the system transitions to state Z or Z^- in infinitesimal time.*

Proof Immediate from Equation 2.3. ■

Figure 3.12 shows illustrations of the extension states, and Figure 3.13 shows their interpretation as the system's configuration on the plane.



(a) State W.

(b) State X.

(c) State Y.

(d) State Z.

Figure 3.13: An illustration of a typical configuration on the plain for each primary system state.

Having identified the new states, we advance to the next step in Algorithm 2.1, and identify the transitions out of these states.

Lemma 3.4.3 (State F)**.** *If* $\kappa > 2\frac{v}{r_c}$*, and* $\cos(\beta(t_0)) \leq 0$ *then* $\cos(\beta(t)) > 0\ \forall t > t_0 + \frac{r_c}{2v}\ln\left(\frac{2\beta(t_0)-2}{\pi-2}\right)$.

Proof

$$\cos(\beta(t_0)) \leq 0$$

$$\Downarrow$$

$$|\beta(t_0)| \geq \frac{\pi}{2}.$$

From 2.5,

$$-2\frac{v}{r_c} - \kappa\beta \leq -2\frac{v}{\rho} - \kappa\beta \leq \dot{\beta} \leq 2\frac{v}{\rho} - \kappa\beta \leq 2\frac{v}{r_c} - \kappa\beta$$

$$\Downarrow$$

$$\beta^+ = 2\frac{v}{\kappa r_c} + \left(\beta(t_0) - 2\frac{v}{\kappa r_c}\right)e^{-\kappa(t-t_0)};$$

$$\beta^- = -2\frac{v}{\kappa r_c} + \left(\beta(t_0) + 2\frac{v}{\kappa r_c}\right)e^{-\kappa(t-t_0)};$$

$$\beta^- \leq \beta(t) \leq \beta^+.$$

Without loss of generality, assume $\beta(t_0) \geq \frac{\pi}{2}$, then if

$$\beta^+(t) = \frac{\pi}{2}$$

$$2\frac{v}{\kappa r_c} + \left(\beta(t_0) - 2\frac{v}{\kappa r_c}\right) e^{-\kappa(t-t_0)} = \frac{\pi}{2}$$

$$\Downarrow$$

$$t = t_0 + \frac{1}{\kappa} \ln\left(\frac{\beta(t_0) - 2\frac{v}{\kappa r_c}}{\frac{\pi}{2} - 2\frac{v}{\kappa r_c}}\right) < t_0 + \frac{r_c}{2v} \ln\left(\frac{2\beta(t_0) - 2}{\pi - 2}\right)$$

then $\beta(t) < \frac{\pi}{2}\ \forall t > t_0 + \frac{r_c}{2v} \ln\left(\frac{2\beta(t_0)-2}{\pi-2}\right)$. ■

Lemma 3.4.4 (State W)**.** *If* $\kappa > 2\frac{v}{r_c}$*, and*

1. $\rho(t_0) > r_c$,
2. $\cos(\alpha(t_0)) < 0$,
3. $\cos(\beta(t_0)) > 0$,
4. $0 < \sin(\alpha(t_0)) \leq \sin(\beta(t_0)) < 1$;

then

1. $t_1 = t_0 + \frac{1}{\kappa} \ln\left(\frac{\beta(t_0)}{\pi-\alpha(t_0)}\right)$,
2. $t_2 = t_0 + \frac{1}{\kappa - \frac{v}{r_c}} \ln\left(\frac{2\beta(t_0)}{\pi-\alpha(t_0)}\right)$,
3. $t_3 = t_0 + \frac{r_c}{v\cos(\beta(t_0))} \ln\left(\frac{\cot\left(\frac{\beta(t_0)-\pi}{2}\right)+\tan(\beta(t_0))}{\cot\left(\frac{\beta(t_0)-\alpha(t_0)}{2}\right)+\tan(\beta(t_0))}\right)$;

and either

1. $\exists t_* \,|t_0 < t_* \leq t_2$ *such that* $\rho(t_*) = r_c$*, or*
2. $\exists t_* \,|t_1 < t_* \leq t_2$ *such that* $\sin(\alpha(t_*)) > \sin(\beta(t_*))$*, or*
3. $\exists t_* \,|t_3 < t_* < t_2$ *such that* $\sin(\alpha(t_*)) < 0$*, or*
4. $\sin(\beta(t_2)) < \sin\left(\frac{\alpha(t_0)}{2}\right)$.

Proof From Equation 2.4,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) < 0,$$

From Equation 2.5,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta \leq \left(\frac{v}{r_c} - \kappa\right)\beta;$$

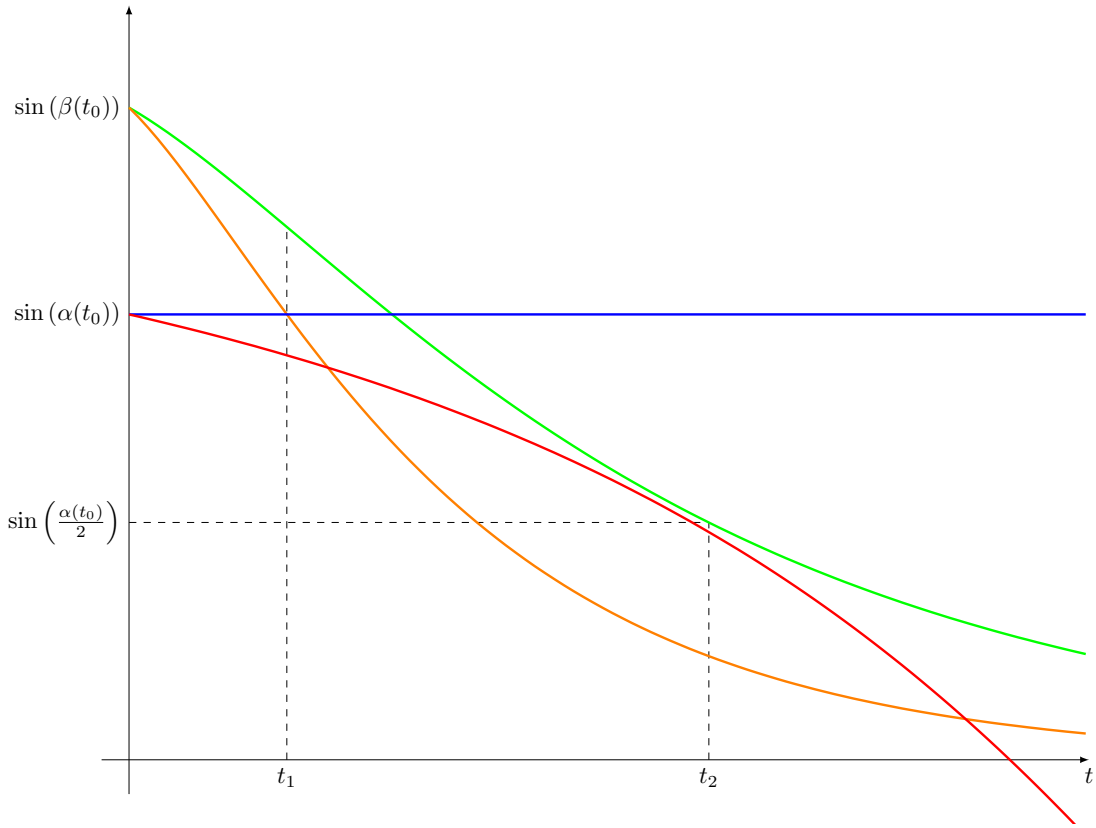


Figure 3.14: Lemma 3.4.4 proof outline. Having ρ shrink at this state, we can find the bounds on β, which shrinks, and on α which grows. Then we can find t_1, when a transition to state X becomes possible, and t_2, where the state must exit. Depending on $\rho(t_0)$, $\alpha(t_0)$ and $\beta(t_0)$, state W can transition into either the capture state, X, Y^-, Z^-, or re-enter W.

$$\dot{\beta} \geq -\kappa\beta;$$

$$\Downarrow$$

$$\beta(t_0)e^{-\kappa(t-t_0)} = \beta^-(t) \leq \beta(t) \leq \beta^+(t) = \beta(t_0)e^{\left(\frac{v}{r_c}-\kappa\right)(t-t_0)} \tag{3.19}$$

From Equation 2.3,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) \leq \frac{v}{r_c}\left(\sin(\beta(t_0)) - \sin(\alpha)\right)$$

$$\Downarrow$$

$$\alpha(t_0) \leq \alpha(t) \leq \alpha^+(t),$$

$$\alpha^+(t) = \\ \beta(t_0) - 2\,\mathrm{arccot}\left(\left(\cot\left(\frac{\beta(t_0)-\alpha(t_0)}{2}\right) + \tan\left(\beta(t_0)\right)\right) e^{\frac{v}{r_c}\cos(\beta(t_0))(t-t_0)} - \tan\left(\beta(t_0)\right)\right). \tag{3.20}$$

The state exits when $\alpha(t) = \pi$, which could only happen after $\alpha^+(t_3) = \pi$,

$$\alpha^+(t_3) = \pi$$

$$= \beta(t_0) - 2\,\mathrm{arccot}\left(\left(\cot\left(\frac{\beta(t_0)-\alpha(t_0)}{2}\right) + \tan\left(\beta(t_0)\right)\right) e^{\frac{v}{r_c}\cos(\beta(t_0))(t_3-t_0)} - \tan\left(\beta(t_0)\right)\right)$$

$$\Downarrow$$

$$\cot\left(\frac{\beta(t_0)-\pi}{2}\right) + \tan\left(\beta(t_0)\right) = \left(\cot\left(\frac{\beta(t_0)-\alpha(t_0)}{2}\right) + \tan\left(\beta(t_0)\right)\right) e^{\frac{v}{r_c}\cos(\beta(t_0))(t_3-t_0)}$$

$$\Downarrow$$

$$t_3 = t_0 + \frac{r_c}{v\cos\left(\beta(t_0)\right)} \ln\left(\frac{\cot\left(\frac{\beta(t_0)-\pi}{2}\right) + \tan\left(\beta(t_0)\right)}{\cot\left(\frac{\beta(t_0)-\alpha(t_0)}{2}\right) + \tan\left(\beta(t_0)\right)}\right)$$

Let t_1 be the first opportunity for $\sin(\alpha) = \sin(\beta)$,

$$\pi - \alpha(t_0) = \beta(t_0)e^{-\kappa(t_1-t_0)}$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{1}{\kappa}\ln\left(\frac{\beta(t_0)}{\pi-\alpha(t_0)}\right),$$

and t_2 the moment at which the upper bound on β reaches $\frac{\alpha(t_0)}{2}$

$$\frac{\pi-\alpha(t_0)}{2} = \beta(t_0)e^{\left(\frac{v}{r_c}-\kappa\right)(t_2-t_0)}$$

$$\Downarrow$$

$$t_2 = t_0 + \frac{1}{\frac{v}{r_c}-\kappa}\ln\left(\frac{\pi-\alpha(t_0)}{2\beta(t_0)}\right).$$

At time t_1,

$$\beta^+(t_1) = \beta(t_0)e^{\left(\frac{v}{r_c}-\kappa\right)(t_1-t_0)} = \beta(t_0)e^{\left(\frac{v}{r_c}-\kappa\right)\left(t_0+\frac{1}{\kappa}\ln\left(\frac{\beta(t_0)}{\pi-\alpha(t_0)}\right)-t_0\right)} = \beta(t_0)e^{\left(\frac{v}{\kappa r_c}-1\right)\ln\left(\frac{\beta(t_0)}{\pi-\alpha(t_0)}\right)}$$

$$= \beta(t_0)\left(\frac{\beta(t_0)}{\pi-\alpha(t_0)}\right)^{\left(\frac{v}{\kappa r_c}-1\right)} = \beta(t_0)\left(\frac{\pi-\alpha(t_0)}{\beta(t_0)}\right)^{\left(1-\frac{v}{\kappa r_c}\right)}$$

Lemma 3.4.5 (State X)**.** *If $\kappa > 2\frac{v}{r_c}$, and*

1. *$\rho(t_0) > r_c$,*

2. $\cos(\alpha(t_0)) \leq 0$,

3. $\cos(\beta(t_0)) \geq 0$,

4. $0 \leq \sin(\beta(t_0)) < \sin(\alpha(t_0)) \leq 1$;

then

1. $t_1 = t_0 + \frac{1}{\kappa}$,

2. $t_2 = t_0 + \frac{1}{\kappa}\left(1 + \ln\left(\frac{\frac{\beta(t_0)}{e}\frac{\kappa\rho(t_0)}{v}}{\sin(\alpha(t_0)) - \sin\left(\frac{\beta(t_0)}{e}\right)} - 1\right)\right)$,

3. $t_3 = t_0 + \frac{\rho(t_0)}{v}\frac{\alpha(t_0) - \frac{\pi}{2}}{\sin(\alpha(t_0)) - \sin\left(\frac{\beta(t_0)}{e}\right)} + \frac{1}{\kappa}$;

and either

1. $\exists t_* \,|t_0 < t_* \leq t_2 + t_3$ *such that* $\rho(t_*) = r_c$, *or*

2. $\exists t_* \,|t_0 < t_* \leq t_2$ *such that* $\beta(t_*) = 0$, *or*

3. $\exists t_* \,|t_0 < t_* < t_3$ *such that* $\alpha(t_*) < \frac{\pi}{2}$.

Proof While in state X, from Equation 2.4,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) < 0,$$

from Equation 2.3,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) < 0,$$

and from Equation 2.5,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta < \dot{\alpha} < 0.$$

Also,

$$\dot{\beta} < -\kappa\beta$$

$$\Downarrow$$

$$\beta(t) < \beta(t_0)e^{-\kappa(t-t_0)},$$

and $\forall t \geq t_1 = t_0 + \frac{1}{\kappa}$,

$$\dot{\beta} < \frac{v}{\rho(t_0)}\left(\sin\left(\frac{\beta(t_0)}{e}\right) - \sin(\alpha(t_0))\right) - \kappa\beta$$

$$\Downarrow$$

$$\beta(t) < \frac{v\left(\sin\left(\frac{\beta(t_0)}{e}\right) - \sin(\alpha(t_0))\right)}{\kappa\rho(t_0)}$$

$$+\left(\frac{\beta(t_0)}{e}-\frac{v\left(\sin\left(\frac{\beta(t_0)}{e}\right)-\sin(\alpha(t_0))\right)}{\kappa\rho(t_0)}\right)e^{-\kappa(t-t_1)}.$$

Solving for t_2, when $\beta(t_2)<0$,

$$0=\frac{v\left(\sin\left(\frac{\beta(t_0)}{e}\right)-\sin(\alpha(t_0))\right)}{\kappa\rho(t_0)}$$

$$+\left(\frac{\beta(t_0)}{e}-\frac{v\left(\sin\left(\frac{\beta(t_0)}{e}\right)-\sin(\alpha(t_0))\right)}{\kappa\rho(t_0)}\right)e^{-\kappa(t_2-t_1)}$$

$$\Downarrow$$

$$\frac{v}{\kappa\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)$$

$$=\left(\frac{\beta(t_0)}{e}-\frac{v}{\kappa\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)\right)e^{-\kappa(t_2-t_1)}$$

$$\Downarrow$$

$$\frac{v}{\kappa\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)$$

$$=\left(\frac{\beta(t_0)\kappa\rho(t_0)-ve\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{e\kappa\rho(t_0)}\right)e^{-\kappa(t_2-t_1)}$$

$$\Downarrow$$

$$\frac{\beta(t_0)\kappa\rho(t_0)-ve\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{ve\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}=e^{\kappa(t_2-t_1)}$$

$$\Downarrow$$

$$t_2=t_0+\frac{1}{\kappa}\left(1+\ln\left(\frac{\frac{\beta(t_0)}{e}\frac{\kappa\rho(t_0)}{v}}{\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)}-1\right)\right).$$

After t_1,

$$\dot{\alpha}<\frac{v}{\rho(t_0)}\left(\sin\left(\frac{\beta(t_0)}{e}\right)-\sin\left(\alpha(t_0)\right)\right),$$

until t_3 when

$$\alpha^{+}(t_3)=\alpha(t_0)-\frac{v}{\rho(t_0)}\left(\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)\right)(t_3-t_1)=\frac{\pi}{2}$$

$$\Downarrow$$

$$t_3=t_0+\frac{\rho(t_0)}{v}\frac{\alpha(t_0)-\frac{\pi}{2}}{\sin\left(\alpha(t_0)\right)-\sin\left(\frac{\beta(t_0)}{e}\right)}+\frac{1}{\kappa}.$$

Q.E.D. ■

Lemma 3.4.6 (State Y). *If* $\kappa > 2\frac{v}{r_c}$, *and*

1. $\rho(t_0) = \rho_0 > r_c$,
2. $-\frac{\pi}{2} < \beta(t_0) \leq -\frac{\pi}{3}$,
3. $\frac{\pi}{2} \leq \alpha(t_0) < \pi$;

then

1. $t_1 = t_0 + \frac{1}{\kappa} \ln\left(\frac{3|\beta(t_0)|}{\pi}\right)$,
2. $t_2 = t_0 + \frac{1}{\kappa} \ln\left(\frac{2\frac{v}{\kappa r_c} + \beta(t_0)}{2\frac{v}{\kappa r_c} - \frac{\pi}{3}}\right)$,
3. $t_3 = t_0 + \frac{r_c}{v}\left(\frac{\alpha(t_0)}{2} - \frac{\pi}{4}\right)$,
4. $t_4 = t_0 + \frac{\rho_0}{v} \ln\left(\tan\left(\frac{\alpha(t_0)}{2}\right)\right)$;

and either

1. $\exists t_* \, | t_0 \leq t_* \leq \max\{t_2, t_4\}$ *such that* $\rho(t_*) = r_c$, *or*
2. $\exists t_* \, | t_1 < t_* \leq t_2$ *such that* $\beta(t_*) > -\frac{\pi}{3}$, *or*
3. $\exists t_* \, | t_3 < t_* < t_4$ *such that* $\alpha(t_*) < \frac{\pi}{2}$.

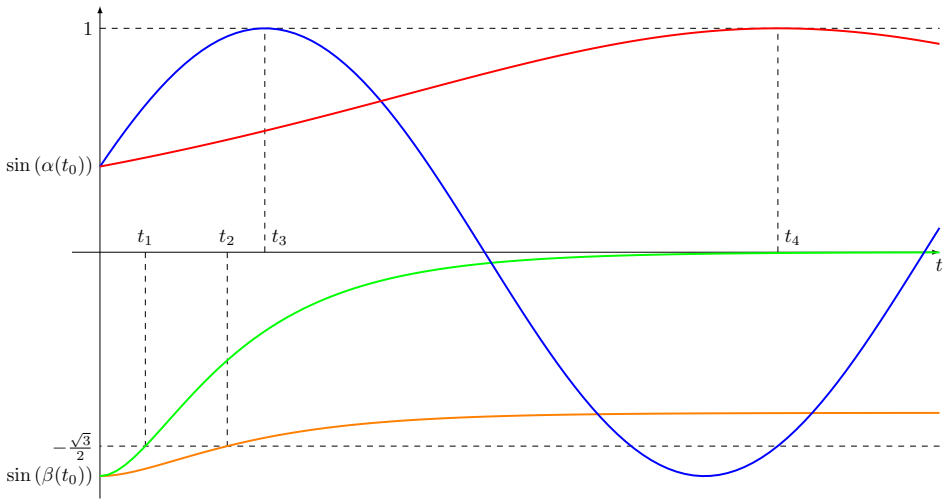


Figure 3.15: Lemma 3.4.6 proof outline. t_1 and t_2 are calculated by the bounds on $\sin(\beta)$, while t_3 and t_4 are calculated by the bounds on $\sin(\alpha)$. A transition to state Z must happen sometime between t_1 and t_2, while a transition to state C_1 must happen between t_3 and t_4, unless the agents captures the target before the transition.

Proof From Equation 2.4,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) < 0,$$

From Equation 2.5,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta \leq -\kappa\beta \leq 0;$$

$$\dot{\beta} > -2\frac{v}{r_c} - \kappa\beta;$$

$$\Downarrow$$

$$-2\frac{v}{\kappa r_c} + \left(\beta(t_0) + 2\frac{v}{\kappa r_c}\right)e^{-\kappa(t-t_0)} = \beta^-(t) < \beta(t) \leq \beta^+(t) = \beta(t_0)e^{-\kappa(t-t_0)}.$$

Therefore, β grows, and may reach $-\frac{\pi}{3}$ by t_1,

$$\beta^+(t_1) = \beta(t_0)e^{-\kappa(t_1-t_0)} = -\frac{\pi}{3}$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{1}{\kappa}\ln\left(\frac{3\left|\beta(t_0)\right|}{\pi}\right);$$

and by t_2, $\beta(t_2)$ must be greater than $-\frac{\pi}{3}$,

$$\beta(t_2) > \beta^-(t_2) = -2\frac{v}{\kappa r_c} + \left(\beta(t_0) + 2\frac{v}{\kappa r_c}\right)e^{-\kappa(t_2-t_0)} = -\frac{\pi}{3}$$

$$\Downarrow$$

$$t_2 = t_0 + \frac{1}{\kappa}\ln\left(\frac{2\frac{v}{\kappa r_c} + \beta(t_0)}{2\frac{v}{\kappa r_c} - \frac{\pi}{3}}\right) \tag{3.21}$$

From Equation 2.3,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) < -\frac{v}{\rho_0}\sin(\alpha) \leq 0;$$

$$\dot{\alpha} > -2\frac{v}{r_c};$$

$$\Downarrow$$

$$\alpha(t_0) - 2\frac{v}{r_c}\left(t - t_0\right) = \alpha^-(t) < \alpha(t) < \alpha^+(t) = -2\,\mathrm{arccot}\left(-\cot\left(\frac{\alpha(t_0)}{2}\right)e^{\frac{v}{\rho_0}(t-t_0)}\right).$$

We can now find t_3, the earliest point at which α might cross below $\frac{\pi}{2}$.

$$\alpha^-(t_3) = \alpha(t_0) - 2\frac{v}{r_c}\left(t_2 - t_0\right) = \frac{\pi}{2}$$

$$\Downarrow$$

$$t_3 = t_0 + \frac{r_c}{v}\left(\frac{\alpha(t_0)}{2} - \frac{\pi}{4}\right);$$

and t_4, after which α must be less than $\frac{\pi}{2}$,

$$\alpha^+(t_4) = -2\,\mathrm{arccot}\left(-\cot\left(\frac{\alpha(t_0)}{2}\right)e^{\frac{v}{\rho_0}(t_4-t_0)}\right) = \frac{\pi}{2}$$

$$\Downarrow$$

$$\cot\left(\frac{\alpha(t_0)}{2}\right)e^{\frac{v}{\rho_0}(t_4-t_0)} = \cot\left(\frac{\pi}{4}\right) = 1$$

$$\Downarrow$$

$$t_4 = t_0 + \frac{\rho_0}{v}\ln\left(\tan\left(\frac{\alpha(t_0)}{2}\right)\right),$$

Q.E.D. ■

Lemma 3.4.7 (State Z)**.** *If $\kappa > 2\frac{v}{r_c}$, and*

1. $\rho(t_0) > r_c$,
2. $-\frac{\pi}{3} < \beta(t_0) < 0$,
3. $\frac{\pi}{2} \leq \alpha(t_0) < \pi$;

then

1. $t_1 = t_0 + \frac{r_c}{v}\left(\frac{\alpha(t_0)}{2} - \frac{\pi}{4}\right)$,
2. $t_2 = t_0 + \frac{\rho(t_0)}{v}\ln\left(\tan\left(\frac{\alpha(t_0)}{2}\right)\right)$;

and either

1. $\exists t_* \,|t_0 \leq t_* \leq t_2$ *such that* $\rho(t_*) = r_c$, *or*
2. $\exists t_* \,|t_1 < t_* < t_2$ *such that* $\alpha(t_*) < \frac{\pi}{2}$.

Proof From Equation 2.4,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) < 0,$$

From Equation 2.5,

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) - \kappa\beta \leq -\kappa\beta \leq 0;$$

$$\dot{\beta} > -2\frac{v}{r_c} - \kappa\beta;$$

$$\Downarrow$$

$$-2\frac{v}{\kappa r_c} + \left(\beta(t_0) + 2\frac{v}{\kappa r_c}\right)e^{-\kappa(t-t_0)} = \beta^-(t) < \beta(t) \leq \beta^+(t) = \beta(t_0)e^{-\kappa(t-t_0)}.$$

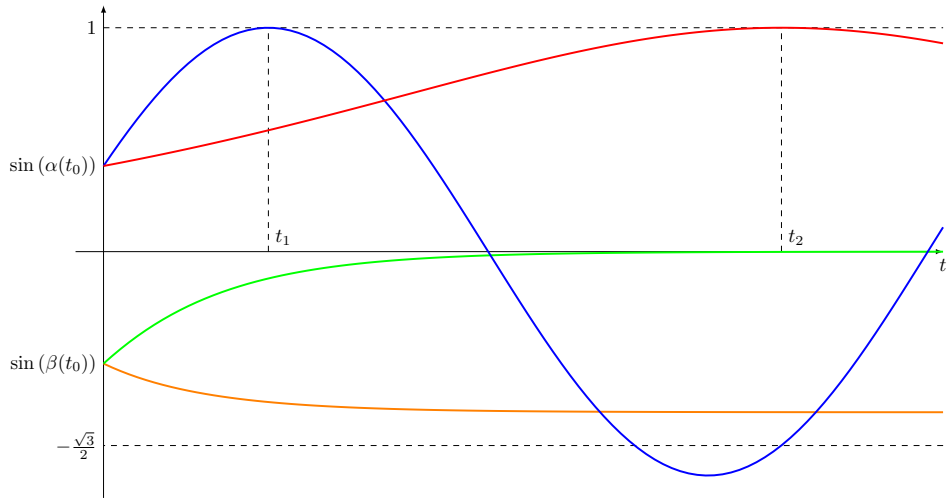


Figure 3.16: Lemma 3.4.7 proof outline. We calculate t_1 and t_2 from the bounds on α. By t_2, the system must transition to state C_1.

Therefore, β is asymptotically locked between 0 and $-2\frac{v}{\kappa r_c} > -1$, and

$$-\frac{\sqrt{3}}{2} < -\sin(1) < -\sin\left(2\frac{v}{\kappa r_c}\right) < \sin(\beta) < 0.$$

From Equation 2.3,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\alpha)\right) < -\frac{v}{\rho_0}\sin(\alpha) < 0;$$

$$\dot{\alpha} > -2\frac{v}{r_c};$$

$$\Downarrow$$

$$\alpha(t_0) - 2\frac{v}{r_c}(t - t_0) = \alpha^-(t) < \alpha(t) < \alpha^+(t) = -2\,\mathrm{arccot}\left(-\cot\left(\frac{\alpha(t_0)}{2}\right)e^{\frac{v}{\rho_0}(t-t_0)}\right).$$

Let t_1, t_2 the earliest and latest points at which α can cross below $\frac{\pi}{2}$.

$$\alpha^-(t_1) = \alpha(t_0) - 2\frac{v}{r_c}(t_1 - t_0) = \frac{\pi}{2}$$

$$\Downarrow$$

$$t_1 = t_0 + \frac{r_c}{v}\left(\frac{\alpha(t_0)}{2} - \frac{\pi}{4}\right);$$

$$\alpha^+(t_2) = -2\,\mathrm{arccot}\left(-\cot\left(\frac{\alpha(t_0)}{2}\right)e^{\frac{v}{\rho_0}(t_2-t_0)}\right) = \frac{\pi}{2}$$

$$\Downarrow$$

$$\cot\left(\frac{\alpha(t_0)}{2}\right)e^{\frac{v}{\rho_0}(t_2-t_0)} = \cot\left(\frac{\pi}{4}\right) = 1$$

$$\Downarrow$$

$$t_2 = t_0 + \frac{\rho_0}{v} \ln\left(\tan\left(\frac{\alpha(t_0)}{2}\right)\right).$$

With the lemmas above we can extend Table 3.2 to include the new states and their transitions, and generate a new transition table, Table 3.4. Figure 3.17 illustrates how these adjustments fit into the graph $\mathcal{G}$.

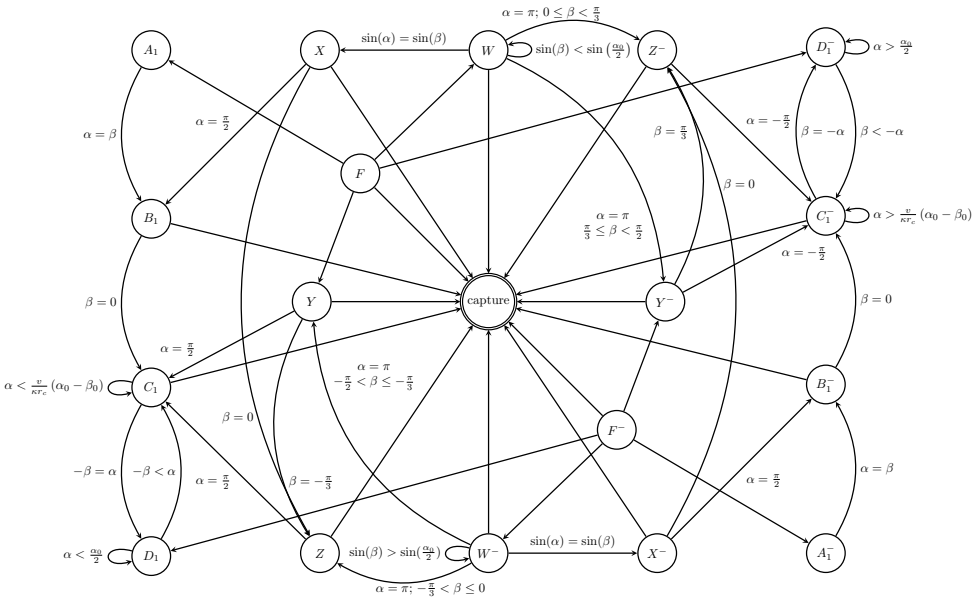


Figure 3.17: All states that have a path to the capture state.

State	System Configuration	Exit Condition	Transition
F	$\frac{\pi}{2} \leq \beta \leq \pi$	$\rho = r_c$	Capture
		$0 \leq \alpha(t) < \beta(t) < \frac{\pi}{2}$	A
		$0 < -\alpha(t) \leq \beta(t) < \frac{\pi}{2}$	D^-
		$\beta(t) = \frac{\pi}{2}$	W
		$\beta = -\frac{\pi}{2}$	Y
W	$0 < \sin(\alpha) \leq \sin(\beta) < 1$	$\rho = r_c$	Capture
		$\sin(\beta) < \sin\left(\frac{\alpha_0}{2}\right)$	W
		$\sin(\alpha) = \sin(\beta)$	X
		$\alpha = \pi,\ \frac{\pi}{3} \leq \beta < \frac{\pi}{2}$	Y^-
		$\alpha = \pi,\ 0 \leq \beta < \frac{\pi}{3}$	Z^-
X	$0 \leq \sin(\beta) < \sin(\alpha) \leq 1$	$\rho = r_c$	Capture
		$\beta = 0$	Z
		$\alpha = \frac{\pi}{2}$	B_1
Y	$-\frac{\pi}{2} < \beta \leq -\frac{\pi}{3}$	$\rho = r_c$	Capture
		$\alpha = \frac{\pi}{2}$	C_1
		$\beta = -\frac{\pi}{3}$	Z
Z	$-\frac{\pi}{3} < \beta < 0$	$\rho = r_c$	Capture
		$\alpha = \frac{\pi}{2}$	C_1

Table 3.4: Capturing Extension Transitions of PCBOUP. The system configuration for these states includes $\frac{\pi}{2} \leq |\alpha| < \pi$.

Figure 3.17 shows all possible transitions between states that may eventually lead to capture; yet capture is not guaranteed, as can be seen in Figure 3.8, where the outer states have no path back to the capture state. To discover which initial configurations may lead to the capture state, we reverse the direction of the edges of the graph $\mathcal{G}$ (Figure 3.17), and by doing so we reverse the transitions between systems states, in a manner that flows from the capture state to all possible initial conditions. We denote the reverse graph $\mathcal{G}^-$. There is no path in $\mathcal{G}^-$ from the capture state to the outer states A, B, C, D, A^-, B^-, C^-, and D^-, so we discard these states.

Since we are dealing with bounds on the actual kinematics of the pursuing agent, we treat each reverse state as an addition of a *region of uncertainty* to Γ, i.e. for every state in $\mathcal{G}^-$, when entering the state, we add the volume trapped between the bounds on (ρ, α, β) while in this state to Γ, since the configuration that led to capture must be in this volume, alongside configurations that did not result in capture. The uncertainty is a result of having the entire volume added to Γ instead of only those configurations that ultimately resulted in capture.

Lemma 3.4.8 (Reverse State F)**.** *Entering state F in $\mathcal{G}^-$ dilates the area of uncertainty by a circle with radius $r_c \ln\left(1 + \frac{\pi}{\pi-2}\right) < \frac{4}{3} r_c$.*

Proof We have shown in Lemma 3.4.3 that the maximal time spent in state F is $T_F = \frac{r_c}{2v} \ln\left(\frac{2\beta_0 - 2}{\pi - 2}\right)$, where in this case β_0 is taken so T_F could assume the maximal possible

value, i.e. $\beta_0 = \pi$, and therefore,

$$T_F = \frac{r_c}{2v} \ln\left(\frac{2\pi - 2}{\pi - 2}\right) = \frac{r_c}{2v} \ln\left(\frac{\pi - 2 + \pi}{\pi - 2}\right) = \frac{r_c}{2v} \ln\left(1 + \frac{\pi}{\pi - 2}\right).$$

While in state F, $\cos(\beta) \leq 0$ and we have no information regarding α, therefore from Equation 2.4,

$$\dot{\rho} < v\,(1 - \cos(\beta)) < 2v,$$

and

$$\Delta\rho < 2vT_F = r_c \ln\left(1 + \frac{\pi}{\pi - 2}\right)$$

is the maximal addition to ρ while the systems is in this state. The reverse flow terminates at this state and with $\beta = \pi$. Since F and its symmetric state F^- are the only sink states in $\mathcal{G}^-$ (Fig. 3.17), then any traversal on the graph must end with either, and with an addition of $\Delta\rho e^{i\Delta\alpha}$, $-\pi < \Delta\alpha \leq \pi$, to all points in the area of uncertainty, resulting in a dilation of the area of uncertainty by $r_c \ln\left(1 + \frac{\pi}{\pi-2}\right)$. ■

Lemma 3.4.9 (Reverse State W)**.** *If the system enters state W in $\mathcal{G}^-$ at time $t_0 + T$ with α_1, β_1, and ρ_1, and exits the state with α_0, β_0, and ρ_0, then*

1. $\alpha_1 - 0.15\pi < \alpha_0 < \alpha_1$
2. $\rho_0 < \rho_1 - 2v\cos(\alpha_1) \ln\left(\frac{\frac{\pi}{2}}{\pi - \alpha_1}\right)$.

Figure 3.18 shows the maximal difference in ρ as function of the minimal α.

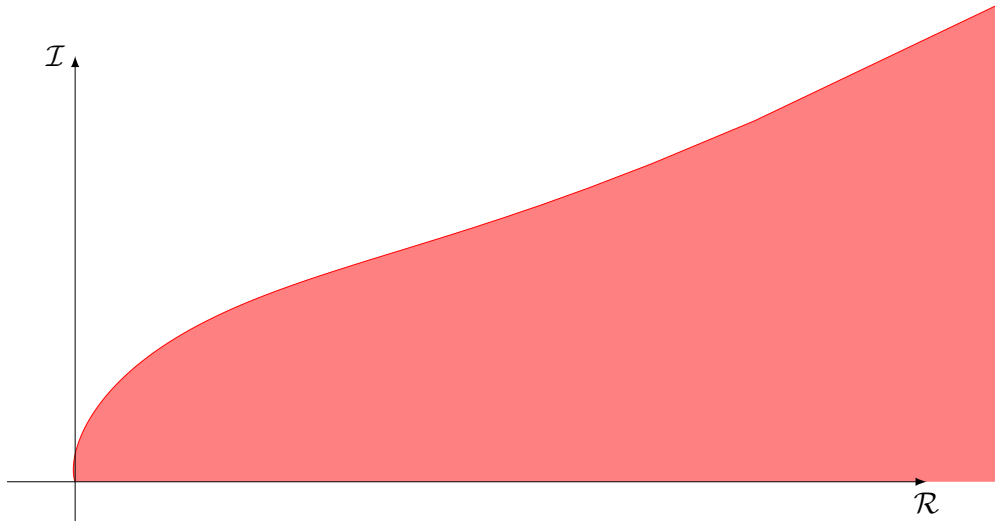


Figure 3.18: Reverse state W. Maximal magnitude and minimal α_0, $(\rho_0 - \rho_1)e^{i(\pi - \alpha_0)}$.

Proof While in state W, $\alpha > \pi - \beta$ If entering state W from state X, then $\alpha_1 = \pi - \beta_1$, otherwise, $\alpha_1 > \pi - \beta_1$.

Let T be the maximal possible time spent in state W; then by Eq. 3.19,

$$\beta(t_0+T) = \beta(t_0)e^{\left(\frac{v}{r_c}-\kappa\right)T}$$

$$\Downarrow$$

$$T = \frac{r_c}{\kappa r_c - v}\ln\left(\frac{\beta_0}{\pi-\alpha_1}\right) < \frac{r_c}{v}\ln\left(\frac{\frac{\pi}{2}}{\pi-\alpha_1}\right).$$

Arbitrarily selecting $\hat{\alpha}_1 = (\pi - \frac{\pi}{2}e^{-1})$ we get an upper bound on the time spent in state W for any $\frac{\pi}{2} < \alpha_1 \leq \hat{\alpha}_1$,

$$\hat{T} = \frac{r_c}{v}\ln\left(\frac{\frac{\pi}{2}}{\pi-\hat{\alpha}_1}\right) = \frac{r_c}{v}\ln\left(\frac{\frac{\pi}{2}}{\pi-\pi\left(\frac{2e-1}{2e}\right)}\right) = \frac{r_c}{v}.$$

With $\hat{T}$ we can calculate the minimal possible α_0 such that $\alpha_1 = \pi - \beta_1$,

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\hat{\alpha}_1)\right) < \frac{v}{r_c}\left(1 - \sin(\hat{\alpha}_1)\right)$$

$$\Downarrow$$

$$\alpha_1 < \alpha_0 + \frac{v}{r_c}\left(1-\sin(\hat{\alpha}_1)\right)\hat{T} = \alpha_0 + \left(1-\sin(\hat{\alpha}_1)\right)$$

$$\Downarrow$$

$$\alpha_0 > \alpha_1 + \sin(\hat{\alpha}_1) - 1 > \alpha_1 - 0.15\pi.$$

Also,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) \geq v\left(\cos(\alpha_1) - \cos(\beta_1)\right) = v\left(\cos(\alpha_1) - \cos(\pi-\alpha_1)\right) = 2v\cos(\alpha_1)$$

$$\Downarrow$$

$$\rho_1 \geq \rho_0 + 2v\cos(\alpha_1)T,$$

and for $\frac{\pi}{2} < \alpha_1 \leq \hat{\alpha}_1$,

$$\rho_1 > \rho_0 + 2v\cos(\alpha_1)\hat{T} = \rho_0 + 2r_c\cos(\alpha_1)$$

$$\Downarrow$$

$$\rho_0 - \rho_1 < -2r_c\cos(\alpha_1).$$

If $\hat{\alpha}_1 < \alpha_1$, then $\beta < \frac{\pi}{2e}$, and if we restart the clock when $\beta = \frac{\pi}{2e}$, then $\beta_0 < \frac{\pi}{2e}$, and $\hat{\alpha}_1 - 0.15\pi < \alpha_0$. If we let another $\hat{T}$ go by, the maximal possible α_1 becomes

$$\hat{T} = \frac{r_c}{v} = \frac{r_c}{v}\ln\left(\frac{\frac{\pi}{2}e^{-1}}{\pi-\hat{\alpha}_2}\right)$$

$$\Downarrow$$

$$\hat{\alpha}_2 = \pi - \frac{\pi}{2}e^{-2},$$

$$\dot{\alpha} = \frac{v}{\rho}\left(\sin(\beta) - \sin(\hat{\alpha}_2)\right) < \frac{v}{r_c}\left(\sin\left(\frac{\pi}{2}e^{-1}\right) - \sin\left(\frac{\pi}{2}e^{-2}\right)\right)$$

$$\Downarrow$$

$$\alpha_1 < \alpha_0 + \frac{v}{r_c}\left(\sin\left(\frac{\pi}{2}e^{-1}\right) - \sin\left(\frac{\pi}{2}e^{-2}\right)\right)\hat{T} = \alpha_0 + \left(\sin\left(\frac{\pi}{2}e^{-1}\right) - \sin\left(\frac{\pi}{2}e^{-2}\right)\right)$$

$$\Downarrow$$

$$\alpha_0 > \alpha_1 + \sin\left(\frac{\pi}{2}e^{-2}\right) - \sin\left(\frac{\pi}{2}e^{-1}\right) > \alpha_1 - 0.11\pi.$$

Also,

$$\dot{\rho} = v\left(\cos(\alpha) - \cos(\beta)\right) \geq 2v\cos(\alpha_1)$$

$$\Downarrow$$

$$\rho_1 \geq \rho_0 + 2v\cos(\alpha_1)T,$$

and for $\hat{\alpha}_1 < \alpha_1 \leq \hat{\alpha}_2$,

$$\rho_1 > \rho_0 + 2v\cos(\alpha_1)\hat{T} = \rho_0 + 2r_c\cos(\alpha_1)$$

$$\Downarrow$$

$$\rho_0 - \rho_1 < -2r_c\cos(\alpha_1).$$

We can now restart the clock again and again until eternity, and the following will remain true:

1. $-0.15\pi < \alpha_0 - \alpha_1 < 0$
2. $\rho_0 - \rho_1 < -2v\cos(\alpha_1)\ln\left(\frac{\frac{\pi}{2}}{\pi-\alpha_1}\right)$. ■

Lemma 3.4.10 (Reverse State X)**.** *If the system enters state X in $\mathcal{G}^-$ at time $t_0 + T$ with α_1, β_1, and ρ_1, and exits the state at time t_0 with α_0, β_0, and ρ_0, then*

1. $\frac{\pi}{2} < \alpha_0 < \alpha_1$, *and*
2. $\rho_0 < 2.2\rho_1 + 0.33r_c$.

Proof State X exits, according to Lemma 3.4.5, in either t_3, t_2, or sooner if capture occurs. For convenience, we choose t_2, though it might take longer than t_3.

$$t_3 = t_0 + \frac{\rho(t_0)}{v}\frac{\alpha(t_0) - \frac{\pi}{2}}{\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)} + \frac{1}{\kappa}.$$

During this time α shrinks,

$$\frac{\pi}{2} \leq \alpha_1 < \alpha_0$$

and so does ρ,

$$\dot{\rho} = v\left(\cos\left(\alpha\right) - \cos\left(\beta\right)\right) < v\left(\cos\left(\frac{\pi}{2}\right) - \cos\left(\beta(t_0)\right)\right) = -v\cos\left(\beta(t_0)\right),$$

$$\dot{\rho} \geq v\left(\cos\left(\alpha(t_0)\right) - \cos\left(0\right)\right) = v\left(\cos\left(\alpha(t_0)\right) - 1\right)$$

$$\Downarrow$$

$$v\left(\cos\left(\alpha(t_0)\right) - 1\right)\left(t_3 - t_0\right) \leq \rho_1 - \rho_0 < -v\cos\left(\beta(t_0)\right)\left(t_3 - t_0\right).$$

$$\Downarrow$$

$$\rho_0 \leq \rho_1 + v\left(1 - \cos\left(\alpha(t_0)\right)\right)\left(t_3 - t_0\right)$$

The only entry into state X from another state happens from state W, in the moment $\beta(t_0) < \pi - \alpha(t_0)$, just after $\beta(t_0) = \pi - \alpha(t_0)$. Exit will happen either to Z with $\beta = 0$, B_1 with $\alpha = \frac{\pi}{2}$, or capture. In any of these cases, a transition will happen before the time it takes for the case in which $\alpha_0 = \pi - \beta_0 = \frac{3}{4}\pi$, since $\dot{\beta} < \dot{\alpha} < 0$ while in this state, and for any $\alpha_0 = \pi - \beta_0 > \frac{3}{4}\pi$, β will reach zero faster, and for any $\alpha_0 = \pi - \beta_0 > \frac{3}{4}\pi$, α will reach $\frac{\pi}{2}$ faster.

Therefore,

$$\rho_0 \leq \rho_1 + v\left(1 - \cos\left(\alpha(t_0)\right)\right)\left(t_3 - t_0\right)$$

$$= \rho_1 + v\left(1 - \cos\left(\frac{3}{4}\pi\right)\right)\left(t_0 + \frac{\rho(t_0)}{v}\frac{\alpha(t_0) - \frac{\pi}{2}}{\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)} + \frac{1}{\kappa} - t_0\right)$$

$$\Downarrow$$

$$= \rho_1 + \left(2 - \sqrt{2}\right)\frac{\frac{\pi}{4}\rho_0}{\sqrt{2} - 2\sin\left(\frac{\pi}{4e}\right)} + \frac{v}{2\kappa}\left(2 - \sqrt{2}\right)$$

$$\rho_0\left(1 + \frac{\pi}{4}\frac{\sqrt{2} - 2}{\sqrt{2} - 2\sin\left(\frac{\pi}{4e}\right)}\right) \leq \rho_1 + \frac{v}{2\kappa}\left(2 - \sqrt{2}\right) < \rho_1 + \frac{r_c}{4}\left(2 - \sqrt{2}\right)$$

$$\Downarrow$$

$$\rho_0 < \frac{4\sqrt{2} - 8\sin\left(\frac{\pi}{4e}\right)}{\left(4 + \pi\right)\sqrt{2} - 8\sin\left(\frac{\pi}{4e}\right) - 2\pi}\rho_1 + \frac{\left(\sqrt{2} - 2\sin\left(\frac{\pi}{4e}\right)\right)\left(2 - \sqrt{2}\right)}{\left(4 + \pi\right)\sqrt{2} - 8\sin\left(\frac{\pi}{4e}\right) - 2\pi}r_c$$

$$< 2.2\rho_1 + 0.33r_c$$

Q.E.D. ■

Lemma 3.4.11 (Reverse State Y)**.** *If the system enters state Y in $\mathcal{G}^-$ at time $t_1 = t_0 + T$ with α_1, β_1, and ρ_1, and exits the state at time t_0 with α_0, β_0, and ρ_0, then*

1. $t_1 - t_0 = \frac{1}{\kappa}\ln\left(\frac{2\frac{v}{\kappa r_c} - \frac{\pi}{2}}{2\frac{v}{\kappa r_c} - \frac{\pi}{3}}\right)$,

2. $\frac{\pi}{2} \leq \alpha\left(t_0+T\right)=\alpha_1<\alpha(t)<\alpha(t_0)=\alpha_0<\pi$, *and*

3. $\rho_0<\rho_1+2r_c$.

Proof The maximal time spent in state Y is the time required for the lower bound on $\beta(t)$, starting at $-\frac{\pi}{2}$, to reach $-\frac{\pi}{3}$, i.e. $\beta_1>-\frac{\pi}{3}$. From Eq. 3.21,

$$T=\frac{1}{\kappa}\ln\left(\frac{2\frac{v}{\kappa r_c}-\frac{\pi}{2}}{2\frac{v}{\kappa r_c}-\frac{\pi}{3}}\right).$$

α shrinks, therefore

$$\frac{\pi}{2}\leq\alpha\left(t_0+T\right)<\alpha(t)<\alpha(t_0)<\pi.$$

The maximal ρ_0 as a function of ρ_1 can be computed,

$$\dot{\rho}>v\left(\cos\left(\alpha_0\right)-\cos\left(-\frac{\pi}{3}\right)\right)=v\left(\cos\left(\alpha_0\right)-\frac{1}{2}\right)$$

$$\Downarrow$$

$$\rho_1>\rho_0+v\left(\cos\left(\alpha_0\right)-\frac{1}{2}\right)T$$

$$\Downarrow$$

$$\begin{aligned}\rho_0&<\rho_1-v\left(\cos\left(\alpha_0\right)-\frac{1}{2}\right)\frac{1}{\kappa}\ln\left(\frac{2\frac{v}{\kappa r_c}-\frac{\pi}{2}}{2\frac{v}{\kappa r_c}-\frac{\pi}{3}}\right)\\&<r\left(t_0+T\right)-\frac{r_c}{2}\left(\cos\left(\alpha(t_0)\right)-\frac{1}{2}\right)\ln\left(\frac{1-\frac{\pi}{2}}{1-\frac{\pi}{3}}\right)\end{aligned}$$

$$\Downarrow$$

$$\rho_0-\rho_1<\frac{r_c}{2}\left(\frac{1}{2}-\cos\left(\alpha(t_0)\right)\right)\ln\left(\frac{1-\frac{\pi}{2}}{1-\frac{\pi}{3}}\right).$$

Since $\alpha_0<\pi$,

$$\rho_0<\rho_1+\frac{3}{4}\ln\left(\frac{1-\frac{\pi}{2}}{1-\frac{\pi}{3}}\right)r_c<\rho_1+\frac{3}{4}\ln\left(\frac{1-\frac{\pi}{2}}{1-\frac{\pi}{3}}\right)r_c<\rho_1+2r_c$$

Lemma 3.4.12 (Reverse State Z)**.** *If the system enters state Z in $\mathcal{G}^-$ at time $t_1=t_0+T$ with α_1, β_1, and r_1, and exits the state with α_0, β_0, and r_0, then $\alpha_0>2\operatorname{arccot}\left(\cot\left(\frac{\alpha_1}{2}\right)e^{-2}\right)$.*

Proof The maximal possible time to remain in state Z is the amount of time to capture the target if the state never exits.

$$\dot{\rho}=v\left(\cos(\alpha)-\cos(\beta)\right)<v\left(0-\cos\left(\frac{\pi}{3}\right)\right)=-\frac{v}{2}$$

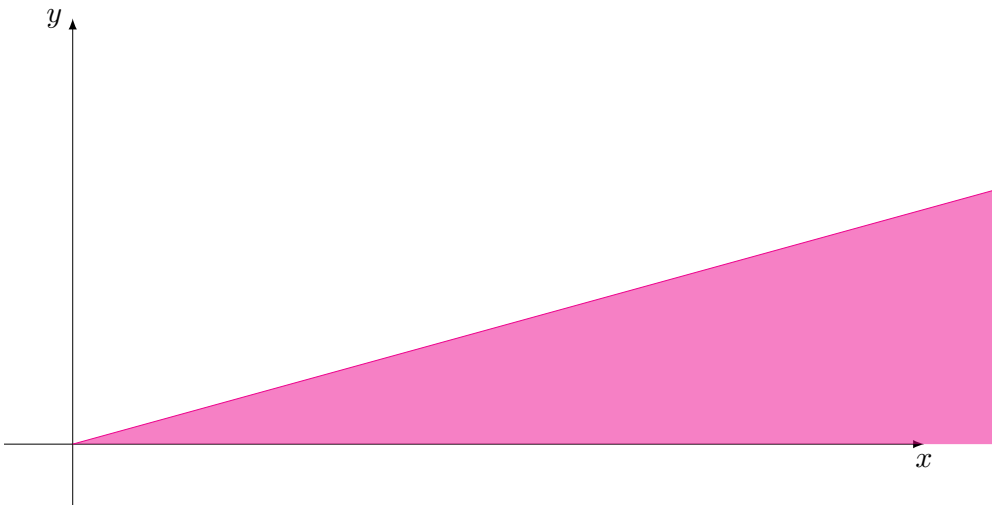


Figure 3.19: Reverse state Z. Maximal magnitude and minimal α_0, $(\rho_0 - \rho_1)e^{i(\pi-\alpha_0)}$.

$$\Downarrow$$

$$T = 2\frac{\rho_0}{v}.$$

According to the proof of Lemma 3.4.7,

$$\alpha_1 < \alpha^+(t) = -2\,\mathrm{arccot}\left(-\cot\left(\frac{\alpha\left(t_0\right)}{2}\right)e^2\right).$$

$$\Downarrow$$

$$2\,\mathrm{arccot}\left(\cot\left(\frac{\alpha_1}{2}\right)e^{-2}\right) < \alpha_0.$$

Corollary 3.10 (Reverse State B_1)**.** *If the system enters state B_1 in $\mathcal{G}^-$ at time t_0+T with α_1, β_1, and ρ_1, and exits the state at time t_0 with α_0, β_0, and ρ_0, then*

1. $\rho^-(t_0 + \frac{1}{\kappa}) = \rho(t_0) - \frac{v}{\kappa}\left(1 - \cos\left(\alpha(t_0)\right)\right)$,
2. $a = \kappa\frac{v}{\rho^-(t_0+\frac{1}{\kappa})} = \frac{v}{\rho(t_0)}\frac{1}{\kappa - \frac{v}{\rho(t_0)}(1-\cos(\alpha(t_0)))}$,
3. $b = \frac{\beta(t_0)+a\left(\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}{\frac{\beta(t_0)}{e}+a\left(\sin(\alpha(t_0))-\sin\left(\frac{\beta(t_0)}{e}\right)\right)}$,
4. $t_2 = t_0 + \frac{1}{\kappa}\ln\left(b\right)$,
5. $\alpha^-(t_2) = \alpha(t_0) + \left(\frac{1}{e}+b\right)\beta(t_0) + a\left(\sin\left(\alpha(t_0)\right) - \sin\left(\frac{\beta(t_0)}{e}\right)\right)\left(1 - b - \ln\left(b\right)\right)$,
6. $\beta^+(t_2) = \frac{\beta(t_0)}{b}$,
7. $t_3 = t_2 - \frac{1}{\kappa}\ln\left(\frac{\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}{\beta^+(t_2)+\frac{2v}{\kappa\rho(t_0)}\cos\left(\frac{\alpha(t_0)+\beta^+(t_2)}{2}\right)\sin\left(\frac{\alpha^-(t_2)-\beta^+(t_2)}{2}\right)}\right)$,

8. $0 < \alpha_0 < \frac{\pi}{2}$, *and*

9. $\rho_0 < \rho_1 + v\,(t_3 - t_0)$.

Proof From the proof of Lemma 3.3.3. ■

Note. State A_1, which is only accessible from State B_1 in $\mathcal{G}^-$, increases ρ, i.e. $\rho_0 < \rho_1$, and therefore does not contribute to Γ.

Lemma 3.4.13 (Reverse State C_1)**.** *If the system enters state C_1 in $\mathcal{G}^-$ at time t_0+T with α_1, β_1, and ρ_1, and exits the state at time t_0 with α_0, β_0, and ρ_0, then*

1. $\alpha_0 = \frac{\pi}{2}$, *and*

2. $\rho_0 < 14 r_c$.

Proof When in State C_1, α shrinks. Let $T = t_1 - t_0$ be a constant time interval, then

$$\dot{\rho} = v\,(\cos(\alpha) - \cos(\beta)) > v\,(\cos(\alpha_0) - \cos(0))$$

$$\Downarrow$$

$$\rho_1 > \rho_0 + v\,(\cos(\alpha_0) - 1)\,T = \rho_0 - 2v\sin^2\left(\frac{\alpha_0}{2}\right)T > \rho_0 - v\tan\left(\frac{\alpha_0}{2}\right)T.$$

As discovered in Lemma 3.3.4,

$$\dot{\alpha} = \frac{v}{\rho}\,(\sin(\beta) - \sin(\alpha)) \leq -\frac{v}{\rho_0}\sin(\alpha)$$

$$\Downarrow$$

$$\tan\left(\frac{\alpha(t)}{2}\right) < \tan\left(\frac{\alpha_0}{2}\right)e^{-\frac{v}{\rho_0}(t-t_0)}$$

$$\Downarrow$$

$$\tan\left(\frac{\alpha_1}{2}\right) < \tan\left(\frac{\alpha_0}{2}\right)e^{-\frac{v}{\rho_0}T}.$$

Arbitrarily choosing $T = \frac{\rho_0}{v}\ln(4)$, we get

$$\tan\left(\frac{\alpha_1}{2}\right) < \tan\left(\frac{\alpha_0}{2}\right)e^{-\frac{v}{\rho_0}\frac{\rho_0}{v}\ln(4)} = \frac{1}{4}\tan\left(\frac{\alpha_0}{2}\right).$$

Note that since ρ shrinks in State C, then $e^{-\frac{v}{\rho_i}\frac{\rho_0}{v}\ln(4)} < \frac{1}{4}$, $\forall i$, and therefore,

$$\rho_2 > \rho_1 - v\tan\left(\frac{\alpha_1}{2}\right)T > \rho_0 - v\tan\left(\frac{\alpha_0}{2}\right)T - v\tan\left(\frac{\alpha_1}{2}\right)T$$

$$= \rho_0 - \rho_0\ln(4)\left(\tan\left(\frac{\alpha_0}{2}\right) + \tan\left(\frac{\alpha_1}{2}\right)\right) > \rho_0 - \rho_0\ln(2)\tan\left(\frac{\alpha_0}{2}\right)\left(1 + \frac{1}{4}\right)$$

$$\Downarrow$$

$$\rho_n > \rho_0 \left(1 - \ln(2) \tan\left(\frac{\alpha_0}{2}\right) \sum_{i=0}^{n-1} \frac{1}{4^i}\right)$$

$$\Downarrow$$

$$\rho_\infty > \rho_0 \left(1 - \ln(2) \tan\left(\frac{\alpha_0}{2}\right) \sum_{i=0}^{\infty} \frac{1}{4^i}\right) = \rho_0 \left(1 - \frac{4}{3} \ln(2) \tan\left(\frac{\alpha_0}{2}\right)\right),$$

and the maximal contribution of State C_1 to Γ is if $\alpha_0 = \frac{\pi}{2}$ and the system remained in State C_1 until the end of time. Note that transitioning to D_1 would not contribute to Γ, as ρ increases in State D_1, therefore

$$\rho_\infty = r_c > \rho_0 \left(1 - \frac{4}{3} \ln(2) \tan\left(\frac{\pi}{4}\right)\right)$$

$$\Downarrow$$

$$\rho_0 < \frac{r_c}{\left(1 - \frac{4}{3} \ln(2) \tan\left(\frac{\pi}{4}\right)\right)} < 14 r_c$$

Q.E.D. ■

Theorem 3.11. *A bounded subspace* $\Gamma(\kappa, v, r_c) \subset ([0, \infty), (-\pi, \pi], (-\pi, \pi])$ *exists, such that if an agent (2.1) governed by the bearing-only control law (3.5) with* $\kappa > 2\frac{v}{r_c}$*, in pursuit of a target moving in a straight line (3.3) ultimately captures the target, then the agent's initial configuration* $(\rho_0, \alpha_0, \beta_0) \in \Gamma$*.*

Proof Lemmas 3.4.1 - 3.4.13 prove that for each state that the system passed through on the way to reach the capture state, the entry coordinates were limited as a function of the exit coordinates. Each of the states adds an area to Γ. According to the previous lemmas, Figure 3.20 shows all possible paths in $\mathcal{G}^-$. Having a finite set of paths, where each of the nodes contributes a bounded addition to Γ, concludes this proof. ■

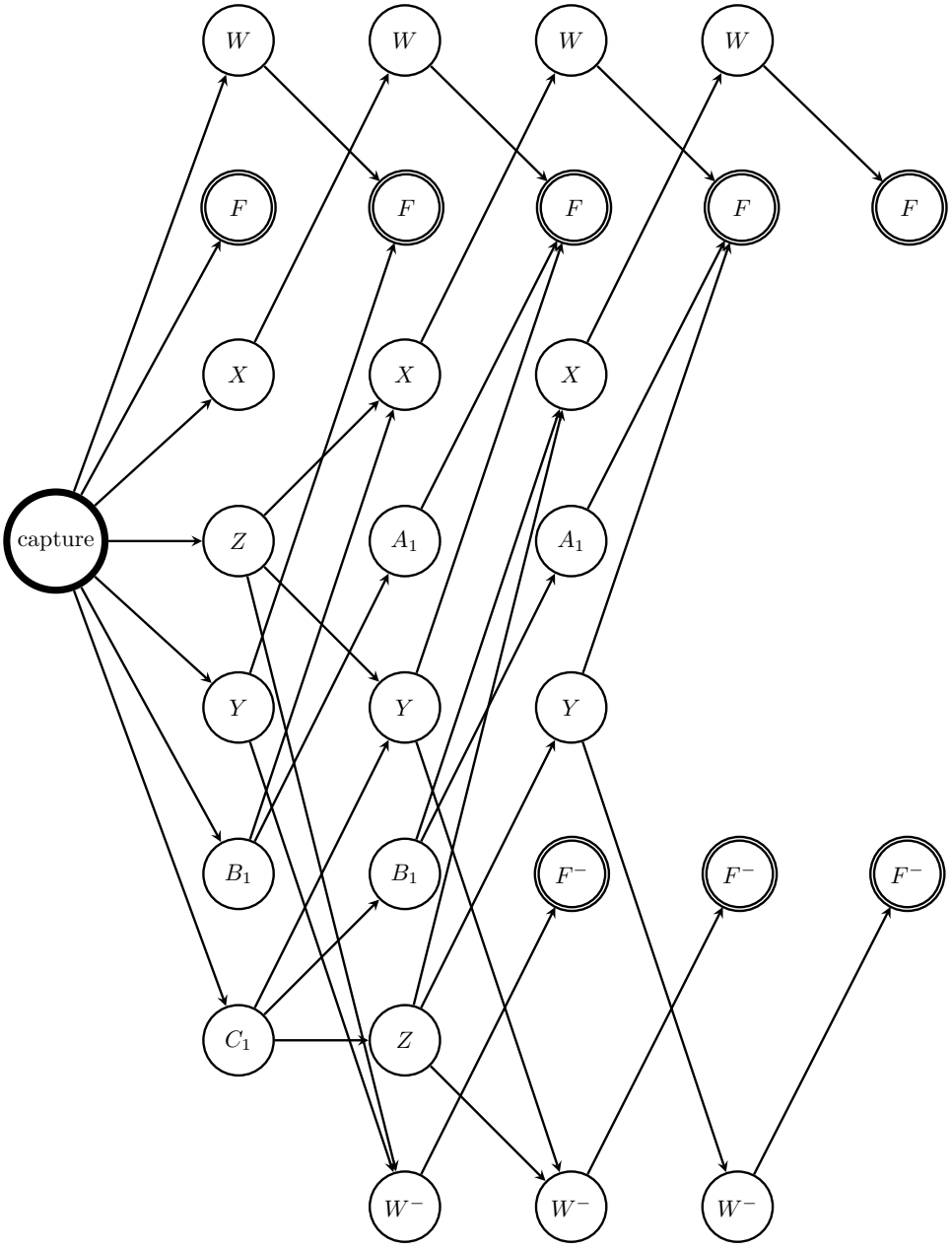
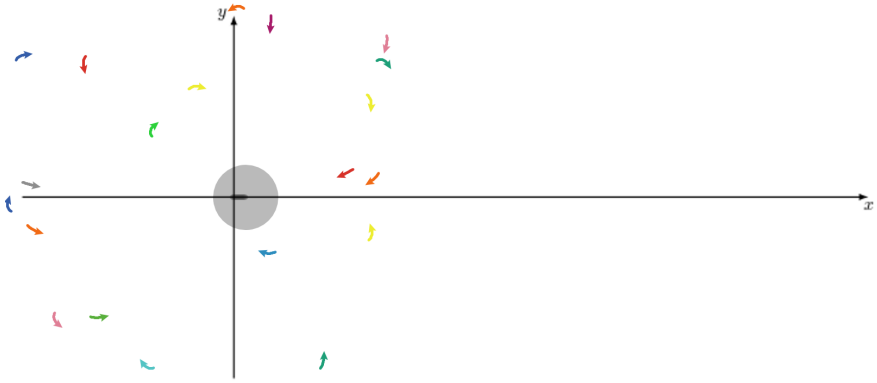
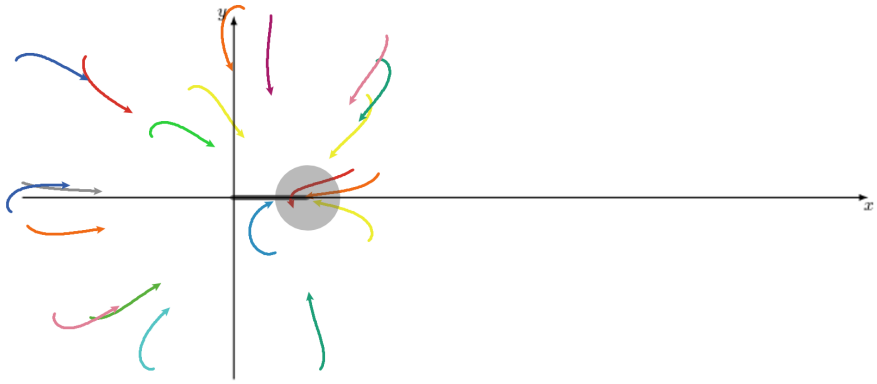


Figure 3.20: All Possible Paths in the Reverse Flow Graph $\mathcal{G}^{-}$.

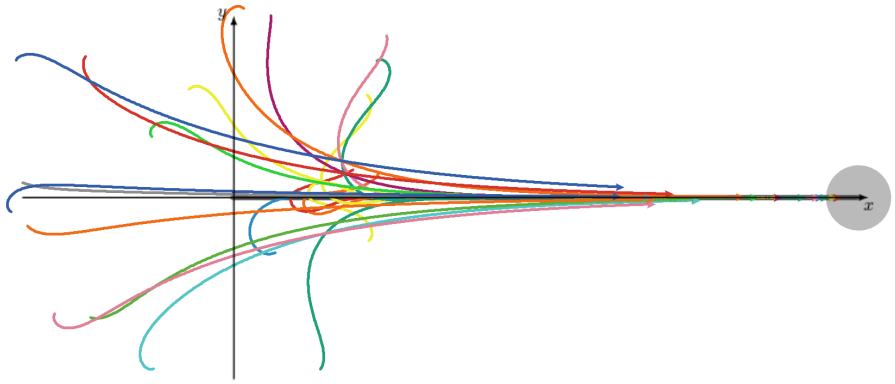
3.5 Simulation



(a) The agents turn to face the target.



(b) Some agents capture the target.



(c) Agents that fell behind the target asymptotically align their velocities with that of the target.

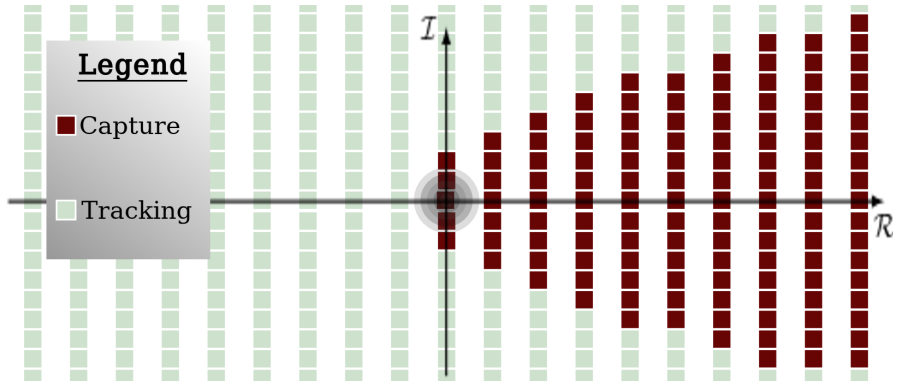
Figure 3.21: Simulation results showing that with $\kappa = \frac{2v}{r_c}$, all agents either capture the target or track it.

Using NetLogo[Wil99] to simulate our model (available at our laboratory's website[1]), we can illustrate our findings from the previous sections. Fig. 3.21 shows a simulation run with 20 agents with randomly generated initial conditions $(\rho(t_0), \alpha(t_0), \beta(t_0))$ and uniform parameters $(v, r_c, \kappa) = \left(1, 30, \frac{1}{15}\right)$ in pursuit of a single target. The target

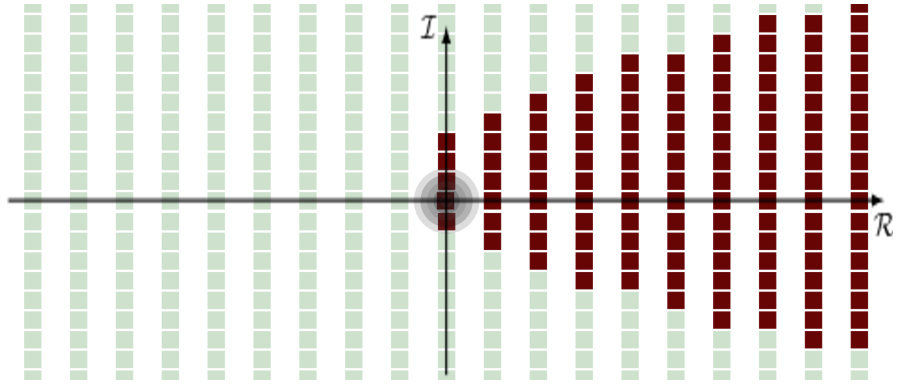
[1] https://mars.cs.technion.ac.il/simulations/

is a circle with radius r_c. The simulation confirms our analysis, that those agents that do not capture the target fall behind it and gradually align their velocity with the target's velocity. The figure shows one simulation run at three consecutive moments. Fig. 3.21a shows the pursuers near their initial positions, scattered randomly on the plane. Fig. 3.21b shows the first agents capturing the target, while Fig. 3.21c shows the pursuers settling into tracking the target.

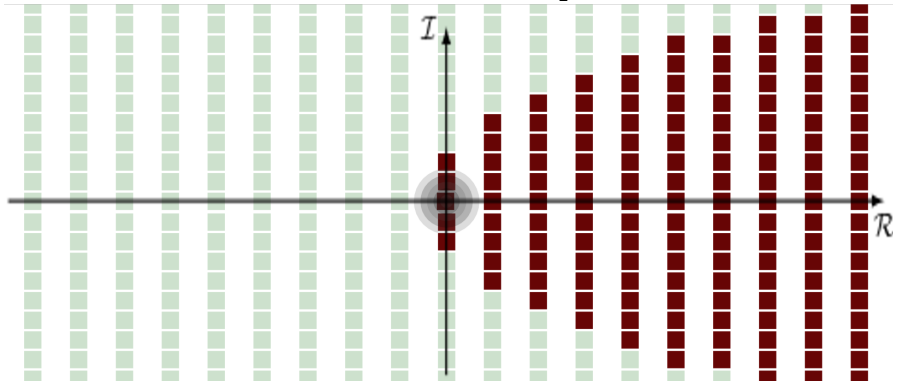
To investigate which initial conditions lead to capturing the target rather than tracking it, we generated the simulation presented in Fig. 3.22. In the simulation we sent 400 agents to chase after the same linearly moving target with uniform parameters $(v, r_c, \kappa) = \left(1, 30, \frac{1}{15}\right)$ and initial conditions $(\alpha(t_0), \beta(t_0), r(t_0))$, such that all agents share the same $\beta(t_0)$ and have varying $\alpha(t_0)$ and $r(t_0)$ according to their initial position. The darker regions mark the initial positions (determined by $\alpha(t_0)$ and $r(t_0)$) of agents that eventually captured the target, while lighter regions those of agents that ended up tracking the target instead. We noticed that $\beta(t_0)$ plays an important role in shaping the capture region, albeit to a lesser extent than $\alpha(t_0)$ and $r(t_0)$. For instance, in the simulation that resulted in Fig. 3.22a, all the pursuing agents were initialized with $\beta(t_0) = 0$, meaning the target was straight ahead of every agent at $t = t_0$. Notice that the capture region for this experiment is smaller, perhaps counter-intuitively, than the capture region of the experiment in Fig. 3.22c, where all agents were initialized with a heading opposite to where the target was. Arguably less surprising is the result shown in Fig. 3.22b, where the initial bias of 90° caused the agents beginning their chase directly under the target to go in the opposite direction of the target, missing their chance to capture, while the agents starting directly above the target started their chase with their velocity parallel to that of the target, allowing them to adjust their alignment without falling far behind the target. The overall observed result in the $\beta(t_0) = \frac{\pi}{2}$ case is a bias of the entire capture region towards the upper part of the experiment arena.



(a) $\beta(t_0) = 0$.



(b) $\beta(t_0) = \frac{\pi}{2}$.



(c) $\beta(t_0) = \pi$.

Figure 3.22: Capture Regions. The markers are drawn in the initial locations $p_{at}(t_0) = r(t_0)e^{i(\alpha(t_0)-\pi)} = -p_{ta}(t_0)$ of each of the agents participating in the experiment. Dark markers represent initial conditions that resulted in capture of the target, while light markers represent initial conditions that resulted in tracking of the target instead. The target's initial location is in the center of the arena.

Chapter 4

Homing of Unicycle Agents with Crude Sensing Capabilities

In this chapter we use our method, presented in Chapter 2, to generate a state machine with which we solve the homing problem of Unicycle-Agents with Crude Sensing over a Limited Sector of Visibility (UCSLSV). The homing problem is a variation of the pursuit problem, discussed in the previous chapter, where the target is stationary, and referred to as a beacon. The agent in this chapter does not acquire an exact reading of the bearing towards the beacon, but instead has only a crude sensor only capable of detecting whether the beacon is within a limited sector centered at the agent's velocity direction, i.e. front.

4.1 Unicycle-Agents with Crude Sensing over a Limited Sector of Visibility

An agent with constant speed unicycle kinematics switches between two rotation rates: a mild rotation rate if the agent detects a stationary beacon in its limited visibility sector, and a severe rotation rate if it doesn't.

Figure 4.1 illustrates the following description. Let a stationary beacon reside on the origin. An agent with a forward facing sensor, with a sensing sector with a central angle σ, $0 < \sigma < \pi$, can either sense the beacon, if $0 \leq |\beta| < \frac{\sigma}{2}$, or not if $\frac{\sigma}{2} \leq |\beta| \leq \pi$. The agent's steering is controlled by the following, simple, equation,

$$\omega = \begin{cases} \frac{v}{R} & 0 \leq |\beta| < \frac{\sigma}{2} \\ \frac{v}{r} & \frac{\sigma}{2} \leq |\beta| \leq \pi \end{cases} \quad 0 < r < R. \tag{4.1}$$

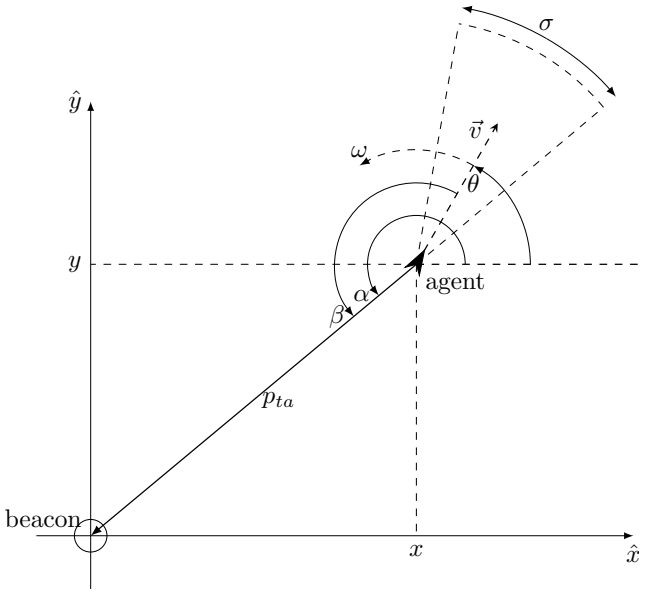


Figure 4.1: The Homing Problem.

This variation of the Unicycle Pursuit problem poses the question that is always of interest in pursuit problems: under what conditions is it guaranteed that the agent captures the target, or in this case, reaches the beacon. It turns out that most cases do not result in actually reaching the beacon, yet if $0 < \sigma < \pi$ and $0 < r < R < \infty$, then the agent falls into and remains within a compact space containing the beacon in finite time, as formalized in the following statement.

4.1.1 UCSLSV Problem Statement

An agent with kinematics (2.1) with constant speed v switches its rotation rate according to its perception of a stationary beacon (4.1). Describe the agent's trajectory and find T and a compact subspace $D \subset \mathbb{R}^2$ containing the agent's trajectory $\forall t > T$, such that

$$\left(1 - \cos\left(\frac{\sigma}{2}\right)\right) r \leq \rho(t) \leq \left(1 + \cos\left(\frac{\sigma}{2}\right)\right) r.$$

4.2 System States

The first step, prescribed by Algorithm 2.1, is to categorize the relationship between the agent and its target, the beacon in this instance, by unique conditions on (ρ, α, β), which vary form the those of the previous chapter due to the different control law, and the target being stationary.

For instance, the definition for α remains the same, but the expression changes to

$$\tan(\alpha) = \frac{y}{x}; \tag{4.2}$$

also,

$$\dot{\alpha} = \frac{v}{\rho}\sin(\beta), \tag{4.3}$$

$$\dot{\rho} = -v\cos(\beta), \tag{4.4}$$

and

$$\dot{\beta} = \dot{\alpha} - \dot{\theta} = \frac{v}{\rho}\sin(\beta) - \omega$$

$$\Downarrow$$

$$\dot{\beta} = \begin{cases} \frac{v}{\rho}\sin(\beta) - \frac{v}{R} & 0 \leq |\beta| < \frac{\sigma}{2} \\ \frac{v}{\rho}\sin(\beta) - \frac{v}{r} & \frac{\sigma}{2} \leq |\beta| \leq \pi. \end{cases} \tag{4.5}$$

To categorize the conditions on (ρ, α, β), we start with $\dot{\beta}$ to see when β grows or shrinks.

Assuming $0 \leq |\beta| < \frac{\sigma}{2}$, we test when $\dot{\beta} = 0$,

$$\dot{\beta} = \frac{v}{\rho}\sin(\beta) - \frac{v}{R} = 0$$

$$\Downarrow$$

$$0 \leq \rho = R\sin(\beta) < R\sin\left(\frac{\sigma}{2}\right),$$

and we classify all configurations with $0 \leq \rho < R\sin\left(\frac{\sigma}{2}\right)$ and $0 \leq |\beta| < \frac{\sigma}{2}$ as State A, while all configurations with $R\sin\left(\frac{\sigma}{2}\right) \leq \rho$ and $0 \leq |\beta| < \frac{\sigma}{2}$ are classified as State B.

Similarly for $\frac{\sigma}{2} \leq |\beta| \leq \pi$,

$$\dot{\beta} = \frac{v}{\rho}\sin(\beta) - \frac{v}{r} = 0$$

$$\Downarrow$$

$$0 \leq \rho = r\sin(\beta) \leq r,$$

we classify all configurations with $0 \leq \rho \leq r$ and $\frac{\sigma}{2} \leq |\beta| \leq \pi$ as State a, while all configurations with $r \leq \rho$ and $\frac{\sigma}{2} \leq |\beta| \leq \pi$ are classified as State b.

Table 4.1 lists the system states.

4.3 State Transitions

The following lemmas detail the exit conditions of the states in Table 4.1.

Lemma 4.3.1 (State B Exit Conditions)**.** *If*

1. $R\sin\left(\frac{\sigma}{2}\right) \leq \rho(t_0)$, *and*
2. $-\frac{\sigma}{2} < \beta(t_0) < \frac{\sigma}{2}$;

State	System Configuration
A	$0 \leq \rho < R\sin\left(\frac{\sigma}{2}\right)$ and $-\frac{\sigma}{2} < \beta < \frac{\sigma}{2}$
a	$0 \leq \rho < r$ and $\frac{\sigma}{2} \leq \lvert\beta\rvert \leq \pi$
B	$R\sin\left(\frac{\sigma}{2}\right) \leq \rho$ and $-\frac{\sigma}{2} < \beta < \frac{\sigma}{2}$
b	$r \leq \rho$ and $\frac{\sigma}{2} \leq \lvert\beta\rvert \leq \pi$

Table 4.1: System States of the Unicycle-Agents with Crude Sensing over a Limited Sector of Visibility Homing Problem.

then

1. $t_1 = t_0 + \frac{R}{v}\sigma$,

and

1. $\exists t_*,\ t_0 < t_* \leq t_1,\ \mid \beta(t_*) = -\frac{\sigma}{2}$, *or*
2. $\exists t_*,\ t_0 < t_* \leq t_1,\ \mid \rho(t_*) = R\sin\left(\frac{\sigma}{2}\right)$.

Proof while in State B, β shrinks,

$$\rho > R\sin\left(\frac{\sigma}{2}\right);\ \beta < \frac{\sigma}{2}$$

$$\Downarrow$$

$$\dot{\beta} = \frac{v}{\rho}\left(\sin\left(\beta\right) - \frac{\rho}{R}\right) < \frac{v}{\rho}\left(\sin\left(\beta\right) - \sin\left(\frac{\sigma}{2}\right)\right) < 0,$$

and ρ shrinks as well,

$$0 < \cos\left(\frac{\sigma}{2}\right) < \cos\left(\beta\right) \leq 1,$$

$$\Downarrow$$

$$\rho(t_0) - v(t - t_0) \leq \rho(t) < \rho(t_0) - v\cos\left(\frac{\sigma}{2}\right)(t - t_0) < \rho(t_0).$$

Exiting State B is therefore possible only if $\beta < -\frac{\sigma}{2}$, or $\rho < R\sin\left(\frac{\sigma}{2}\right)$.

The maximal time spent in state B is the maximal amount of time the beacon can stay in the agent's sector of visibility, i.e., the time it takes for the beacon to cross from

one side of the sector to the other while still in State B. Let t_0 and t_1 denote the entry time and exit time such that $T_B = t_1 - t_0$ is maximal,

$$\dot{\theta} = \dot{\alpha} - \dot{\beta} = \frac{v}{R}$$

$$\Downarrow$$

$$\theta(t) = \theta(t_0) + \frac{v}{R}(t - t_0);$$
$$\theta(t_1) - \theta(t_0) = \sigma$$

$$\Downarrow$$

$$\frac{v}{R}(t_1 - t_0) = \frac{v}{R}T_B = \sigma$$

$$\Downarrow$$

$$T_B = \frac{R}{v}\sigma.$$

If entering State B with $\beta(t_0) = \frac{\sigma}{2}$ and leaving with $\beta(t_1) = -\frac{\sigma}{2}$ and $\rho(t_1) \geq R\sin\left(\frac{\sigma}{2}\right)$, then

$$\rho(t_0) - \rho(t_1) = 2R\sin\left(\frac{\sigma}{2}\right),$$

see Figure 4.2. ■

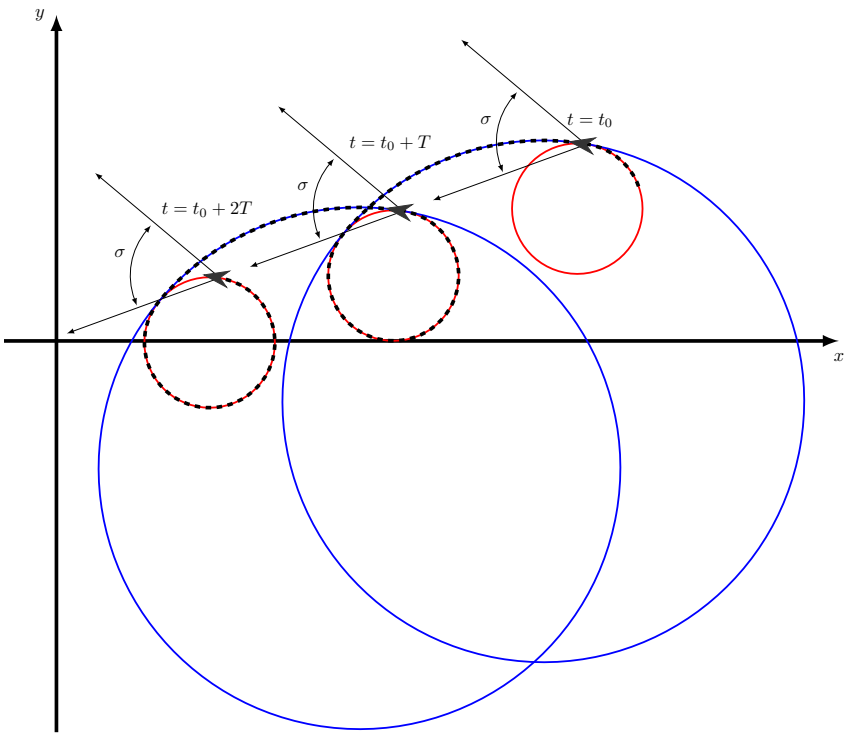


Figure 4.2: Transitions from states B and b, while $\rho(t) > R\sin\left(\frac{\sigma}{2}\right)$.

Lemma 4.3.2 (State b Exit Conditions)**.** *If*

1. $r \leq \rho(t_0)$, *and*
2. $\frac{\sigma}{2} \leq |\beta(t_0)| \leq \pi$*;*

then

1. $t_1 = t_0 + \frac{r}{v}(2\pi - \sigma)$,

and

1. $\exists t_*,\ t_0 < t_* \leq t_1,\ |\ 0 < \beta(t_*) < \frac{\sigma}{2}$, *or*
2. $\exists t_*,\ t_0 < t_* \leq t_1,\ |\ \rho(t_*) < r$, *or*
3. $\exists t_2,\ t_0 < t_2 \leq t_1,\ |\ \beta(t_2) = \frac{\pi}{2}$ *and* $\rho(t_2) = r$, *and* $\beta(t) = \frac{\pi}{2}$ *and* $\rho(t) = r\ \forall t \geq t_2$.

Proof while in State b, β almost always shrinks,

$$\rho \geq r;\ \frac{\sigma}{2} \leq |\beta| \leq \pi$$

$$\Downarrow$$

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \frac{\rho}{r}\right) \leq 0,$$

Exiting State b is therefore possible only if $\beta < \frac{\sigma}{2}$, or $\rho < r$. However, if at some time t_2, $\rho(t_2) = r$ and $\beta(t_2) = \frac{\pi}{2}$, then

$$\dot{\beta} = \frac{v}{r}\left(\sin\left(\frac{\pi}{2}\right) - \frac{r}{r}\right) = 0,$$

$$\dot{\rho} = -v\cos\left(\frac{\pi}{2}\right) = 0,$$

and the system never leaves State b, with $\rho \equiv r$ and $\beta \equiv \frac{\pi}{2}$ from t_2 on.

If State b ever exits, then the maximal time spent in state b is the maximal amount of time the beacon can stay outside the agent's sector of visibility, i.e., the time it takes for the agent's orientation, θ, to complete a $\pi - \sigma$ turn. Let t_0 and t_1 denote the entry time and exit time such that $T_b = t_1 - t_0$ is maximal,

$$\dot{\theta} = \dot{\alpha} - \dot{\beta} = \frac{v}{r}$$

$$\Downarrow$$

$$\theta(t) = \theta(t_0) + \frac{v}{r}(t - t_0);$$

$$\theta(t_1) - \theta(t_0) = \pi - \sigma$$

$$\Downarrow$$

$$\frac{v}{r}(t_1 - t_0) = \frac{v}{r}T_b = \pi - \sigma$$

$$\Downarrow$$

$$T_b = \frac{r}{v}(\pi - \sigma).$$

If entering State b with $\beta(t_0) = -\frac{\sigma}{2}$ and leaving with $\beta(t_1) = \frac{\sigma}{2}$ and $\rho(t_1) \geq r$, then

$$\rho(t_1) - \rho(t_0) = 2r \sin\left(\frac{\sigma}{2}\right),$$

see Figure 4.2. ■

Lemma 4.3.3 (State a Exit Conditions)**.** *If*

1. $0 \leq \rho(t_0) < r$, *and*
2. $\frac{\sigma}{2} \leq |\beta(t_0)| \leq \pi$;

then

1. $t_1 = t_0 + 2\frac{r}{v}\pi$,

and either

1. $\exists t_*,\ t_0 < t_* \leq t_1,\ |\ 0 < \beta(t_*) < \frac{\sigma}{2}$, *or*
2. $\exists t_*,\ t_0 < t_* \leq t_1,\ |\ r \leq \rho(t_*)$.

Proof In State a, ρ can grow or shrink, depending on β. In addition, β can either grow or shrink; it grows when

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \frac{\rho}{r}\right) > 0 \ \Rightarrow\ \rho < r\sin(\beta),$$

and shrinks when

$$\frac{v}{\rho}\left(\sin(\beta) - \frac{\rho}{r}\right) < 0 \ \Rightarrow\ \rho > r\sin(\beta).$$

Yet if $\beta = -\frac{\sigma}{2}$, then $r\sin(\beta) = -r\sin\left(\frac{\sigma}{2}\right) < \rho$, $\forall\rho$, and therefore the only transition due to β is if $\beta = \frac{\sigma}{2}$ and $\rho > r\sin\left(\frac{\sigma}{2}\right)$.

Note that there must be a transition out of the state before the agent has completed a full circle with $\dot{\theta} = \frac{v}{r}$,

$$\theta(t_1) - \theta(t_0) = 2\pi = \frac{v}{r}(t_1 - t_0)$$

$$\Downarrow$$

$$t_1 = t_0 + 2\frac{v}{r}\pi,$$

since if $\rho(t_0) < r$, and the agent completes a full circle, then there must be some point along the agent's orbit in which $\rho \geq r$, and the state exits. ■

Lemma 4.3.4 (State A Exit Conditions)**.** *If*

1. $0 \leq \rho(t_0) < R\sin\left(\frac{\sigma}{2}\right)$, *and*
2. $-\frac{\sigma}{2} < \beta(t_0) < \frac{\sigma}{2}$

then

1. $t_1 = t_0 + \frac{R}{v}\sigma$,

and

1. $\exists t_*,\ t_0 < t_* \leq t_1,\ |\ \beta(t_*) = -\frac{\sigma}{2}$ *and* $\rho(t_*) \leq \rho(t_0)$, *or*
2. $\exists t_*,\ t_0 < t_* \leq t_1,\ |\ \beta(t_*) = \frac{\sigma}{2}$ *and* $\rho(t_*) \leq \rho(t_0)$.

Proof From Equation 4.4,

$$\dot{\rho} = -v\cos(\beta);$$

while in State A,

$$0 < \cos\left(\frac{\sigma}{2}\right) < \cos(\beta) \leq 1,$$

and ρ shrinks,

$$\rho(t_0) - v(t - t_0) \leq \rho(t) < \rho(t_0) - v\cos\left(\frac{\sigma}{2}\right)(t - t_0) < \rho(t_0).$$

However, β can either grow or shrink; it grows when

$$\dot{\beta} = \frac{v}{\rho}\left(\sin(\beta) - \frac{\rho}{R}\right) > 0 \ \Rightarrow\ \rho < R\sin(\beta),$$

and shrinks when

$$\frac{v}{\rho}\left(\sin(\beta) - \frac{\rho}{R}\right) < 0 \ \Rightarrow\ \rho > R\sin(\beta).$$

Q.E.D. ■

Table 4.2 summarizes the lemmas above.

4.4 Paths and Cycles in $\mathcal{G}$

Having followed Algorithm 2.1 to generate $\mathcal{G}$, we can now make the following observations based on the paths and cycles in $\mathcal{G}$, which ultimately lead us to the solution to the UCSLSV problem, in the form of Theorem 4.1.

Lemma 4.4.1 (Cycles with $\beta = -\frac{\sigma}{2}$)**.** *Between every consecutive instances when* $\beta(t_0) = \beta(t_1) = -\frac{\sigma}{2}$,

1. $t_1 = t_0 + \frac{r}{v}(2\pi - \sigma) + \frac{R}{v}\sigma$,

State	System Configuration	Exit Condition	Transition
A	$0 \leq \rho < R\sin\left(\frac{\sigma}{2}\right)$ and $\lvert\beta\rvert < \frac{\sigma}{2}$	$\beta = \frac{\sigma}{2}$, $\rho < r$	a
		$\beta = -\frac{\sigma}{2}$, $\rho < r$	a
		$\beta = -\frac{\sigma}{2}$, $\rho \geq r$	b
		$\beta = \frac{\sigma}{2}$, $\rho \geq r$	b
a	$0 \leq \rho < r$ and $\frac{\sigma}{2} \leq \lvert\beta\rvert \leq \pi$	$r \leq \rho$	b
		$\beta < \frac{\sigma}{2}$, $\rho < R\sin\left(\frac{\sigma}{2}\right)$	A
		$\beta < \frac{\sigma}{2}$, $\rho \geq R\sin\left(\frac{\sigma}{2}\right)$	B
b	$r \leq \rho$ and $\frac{\sigma}{2} \leq \lvert\beta\rvert \leq \pi$	$\beta < \frac{\sigma}{2}$, $\rho < R\sin\left(\frac{\sigma}{2}\right)$	A
		$\beta < \frac{\sigma}{2}$, $\rho \geq R\sin\left(\frac{\sigma}{2}\right)$	B
		$\rho(t_0) = \rho(t_1)$ and $\beta(t_0) = \beta(t_1)$	b
		$\rho < r$	a
B	$R\sin\left(\frac{\sigma}{2}\right) \leq \rho$ and $\lvert\beta\rvert < \frac{\sigma}{2}$	$\beta = -\frac{\sigma}{2}$, $\rho \geq r$	b
		$\beta = -\frac{\sigma}{2}$, $\rho < r$	a
		$\rho = R\sin\left(\frac{\sigma}{2}\right)$	A

Table 4.2: Transition Table for the UCSLSV problem.

2. $\rho(t_1) = \rho(t_0) - 2\left(R - r\right)\sin\left(\frac{\sigma}{2}\right)$.

Figure 4.2 illustrates the following proof.

Proof At t_0, the system exits states A or B with $\beta = -\frac{\sigma}{2}$, and the agent rotates around a fixed point at distance r from it, until completing a $2\pi - \sigma$ arc, placing the agent at time $t_2 = t_0 + \frac{r}{v}\left(2\pi - \sigma\right)$ at a distance $\rho(t_2) = \rho(t_0) + 2r\sin\left(\frac{\sigma}{2}\right)$ with $\beta(t_2) = \frac{\sigma}{2}$. At t_2 the system transitions back to either A or B, and the agent rotates around a new fixed point, at distance R from it, until completing a σ arc at $t_1 = t_2 + \frac{R}{v}\sigma$ with $\beta(t_1) = -\frac{\sigma}{2}$ and $\rho(t_1) = \rho(t_2) - 2R\sin\left(\frac{\sigma}{2}\right)$. ■

Lemma 4.4.2 (Short cycles with $\beta = \frac{\sigma}{2}$)**.** *If the system transitions from States A or B to a or b at t_0 with $\beta(t_0) = \frac{\sigma}{2}$ and $r\sin\left(\frac{\sigma}{2}\right) \leq \rho(t_0) \leq R\sin\left(\frac{\sigma}{2}\right)$, then*

1. $t_1 = t_0 + \frac{\rho(t_0)}{v\cos\left(\frac{\sigma}{2}\right)} - \frac{r}{v}\tan\left(\frac{\sigma}{2}\right) \leq t_0 + \frac{R-r}{v}\tan\left(\frac{\sigma}{2}\right)$,

2. $\rho(t_1) = r\sin\left(\frac{\sigma}{2}\right)$,

3. $\beta(t_1) = \frac{\sigma}{2}$,

and $\forall t \geq t_1$, $\lvert\beta(t)\rvert \geq \frac{\sigma}{2}$.

Proof Since $\beta(t_0) = \frac{\sigma}{2}$ and $r\sin\left(\frac{\sigma}{2}\right) \leq \rho(t_0)$,

$$\dot{\beta}(t_0) = \frac{v}{\rho}\left(\sin\left(\frac{\sigma}{2}\right) - \frac{\rho}{r}\right) \leq \frac{v}{r}\left(1 - 1\right) = 0$$

and States a or b transition immediately back to States A or B, where $\rho(t_0) \leq R\sin\left(\frac{\sigma}{2}\right)$ and

$$\dot{\beta}(t_0) = \frac{v}{\rho}\left(\sin\left(\frac{\sigma}{2}\right) - \frac{\rho}{R}\right) \geq \frac{v}{R}\left(1 - 1\right) = 0$$

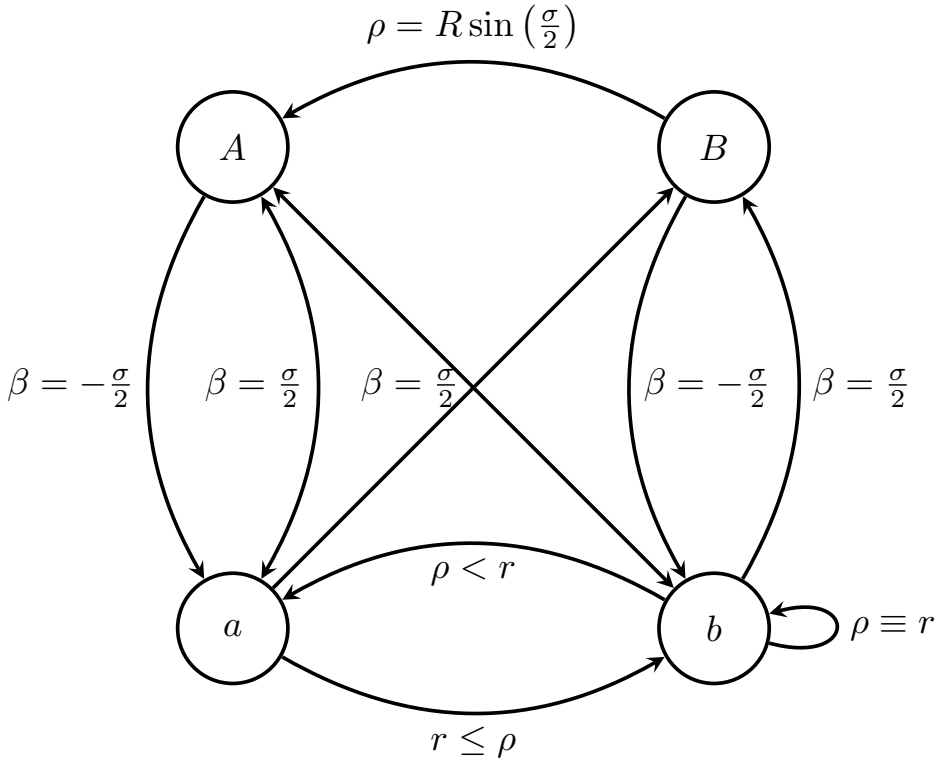


Figure 4.3: UCSLSV DFSM

and the system transitions back to either a or b and so forth and so on, keeping $\beta = \frac{\sigma}{2}$, while

$$\dot{\rho} = -v\cos(\beta) = -v\cos\left(\frac{\sigma}{2}\right)$$

$$\Downarrow$$

$$\rho(t) = \rho(t_0) - v\cos\left(\frac{\sigma}{2}\right)(t - t_0)\,,$$

until $\rho(t_1) = r\sin\left(\frac{\sigma}{2}\right)$,

$$t_1 = t_0 + \frac{\rho(t_0) - r\sin\left(\frac{\sigma}{2}\right)}{v\cos\left(\frac{\sigma}{2}\right)} < t_0 + \frac{R-r}{v}\tan\left(\frac{\sigma}{2}\right),$$

while keeping $\beta(t_1) = \frac{\sigma}{2}$. From that point on, the system only transitions between states a and b, see Figure 4.4. ■

Lemma 4.4.3 (Long cycles with $\beta = \frac{\sigma}{2}$)**.** *If the system transitions from States A or B to a or b at t_0 with $\beta(t_0) = \frac{\sigma}{2}$ and $0 \leq \rho(t_0) < r\sin\left(\frac{\sigma}{2}\right)$, then*

1. $k = \left\lceil \frac{r}{R-r} \right\rceil$,
2. $t_\Omega = t_0 + \left\lceil \frac{k}{2} \right\rceil \frac{r}{v}(2\pi - \sigma) + \left\lfloor \frac{k}{2} \right\rfloor \frac{R}{v}\sigma + \frac{R-r}{v}\tan\left(\frac{\sigma}{2}\right)$,
3. $\exists t_*,\ t_1 \leq t_* \leq t_\Omega,\ |\ \beta(t_*) = \frac{\sigma}{2}$, *and* $\rho(t_*) = r\sin\left(\frac{\sigma}{2}\right)$,

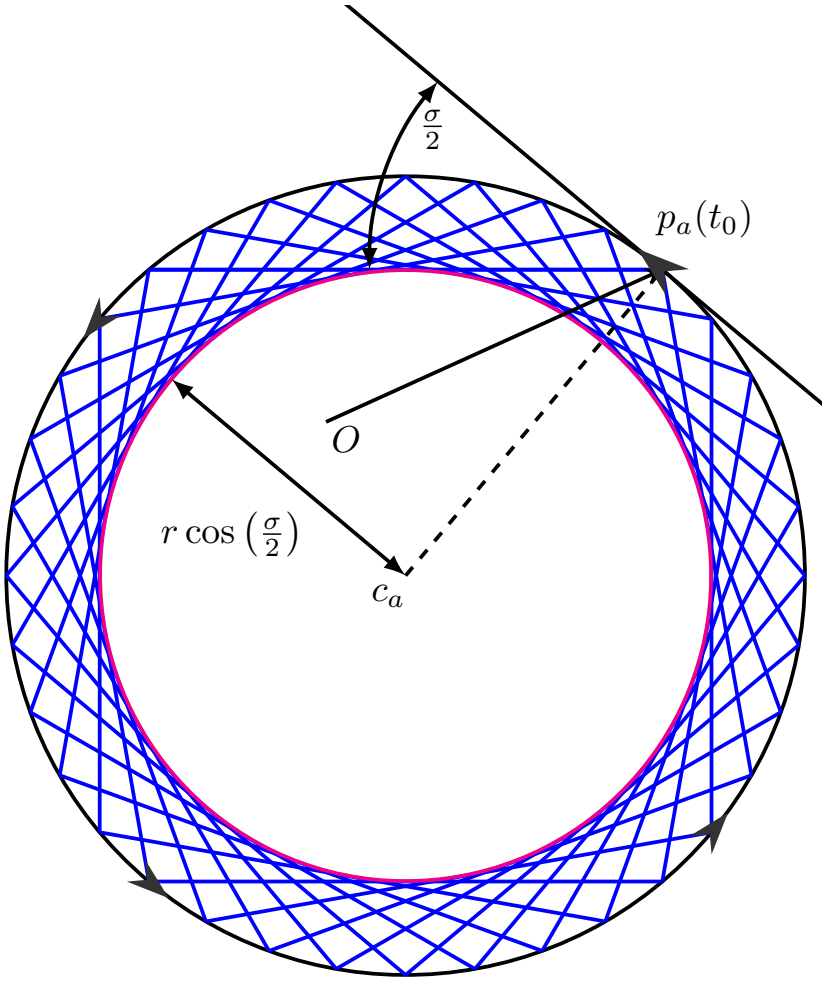


Figure 4.4: An agent located at $p_a(t_0)$, traveling upon the r radius circle with its center at c_a, never perceives the beacon (located at O) if it is on or inside a concentric circle with radius $r\cos\left(\frac{\sigma}{2}\right)$.

and $\forall t \geq t_\Omega$, $|\beta(t)| \geq \frac{\sigma}{2}$.

Proof Since $\beta(t_0) = \frac{\sigma}{2}$ and $0 \leq \rho(t_0) < r\sin\left(\frac{\sigma}{2}\right)$,

$$\dot{\beta}(t_0) = \frac{v}{\rho(t_0)}\left(\sin\left(\frac{\sigma}{2}\right) - \frac{\rho(t_0)}{r}\right) > \frac{v}{r}(1-1) = 0,$$

and at time $t_1 < t_0 + \frac{r}{v}(2\pi - \sigma)$, $\beta(t_1) = \frac{\sigma}{2}$, and $\rho(t_1) = 2r\sin\left(\frac{\sigma}{2}\right) - \rho(t_0)$, see Figure 4.5. Notice that since $0 \leq \rho(t_0) < r\sin\left(\frac{\sigma}{2}\right)$,

$$r\sin\left(\frac{\sigma}{2}\right) < \rho(t_1) = 2r\sin\left(\frac{\sigma}{2}\right) - \rho(t_0) \leq 2r\sin\left(\frac{\sigma}{2}\right) < 2R\sin\left(\frac{\sigma}{2}\right).$$

If also $r \leq \frac{R}{2}$, then $\rho(t_1) \leq R\sin\left(\frac{\sigma}{2}\right)$, and the conditions of Lemma 4.4.2 apply at t_1;

otherwise, $\frac{R}{2} < r < R$, and $t_2 \leq t_1 + \frac{R}{v}\sigma$, $\beta(t_2) = \frac{\sigma}{2}$, and

$$\rho(t_2) = 2R\sin\left(\frac{\sigma}{2}\right) - \rho(t_1) = 2(R-r)\sin\left(\frac{\sigma}{2}\right) + \rho(t_0)$$

$$\Downarrow$$

$$0 < \rho(t_2) < R\sin\left(\frac{\sigma}{2}\right).$$

If, in addition, $\rho(t_2) \geq r\sin\left(\frac{\sigma}{2}\right)$, then the conditions of Lemma 4.4.2 apply at t_2. If, on the other hand, $\rho(t_2) < r\sin\left(\frac{\sigma}{2}\right)$, then

$$2(R-r)\sin\left(\frac{\sigma}{2}\right) + \rho(t_0) < r\sin\left(\frac{\sigma}{2}\right)$$

$$\Downarrow$$

$$0 \leq \rho(t_0) < (3r-2R)\sin\left(\frac{\sigma}{2}\right)$$

$$\Downarrow$$

$$\frac{2}{3}R \leq r < R,$$

$t_3 \leq t_2 + \frac{r}{v}(2\pi - \sigma)$, $\beta(t_3) = \frac{\sigma}{2}$, and

$$r\sin\left(\frac{\sigma}{2}\right) < \rho(t_3) = 2r\sin\left(\frac{\sigma}{2}\right) - \rho(t_2) = (4r-2R)\sin\left(\frac{\sigma}{2}\right) - \rho(t_0) < 2R\sin\left(\frac{\sigma}{2}\right).$$

If $\rho(t_3) \leq R\sin\left(\frac{\sigma}{2}\right)$ then the conditions of Lemma 4.4.2 apply at t_3, otherwise,

$$R\sin\left(\frac{\sigma}{2}\right) < (4r-2R)\sin\left(\frac{\sigma}{2}\right) - \rho(t_0)$$

$$\Downarrow$$

$$0 \leq \rho(t_0) < (4r-3R)\sin\left(\frac{\sigma}{2}\right)$$

$$\Downarrow$$

$$\frac{3}{4}R < r < R.$$

Let $t_4 \leq t_3 + \frac{R}{v}\sigma$ be the moment when $\beta = \frac{\sigma}{2}$ and

$$\rho(t_4) = 2R\sin\left(\frac{\sigma}{2}\right) - \rho(t_3) = (4R-4r)\sin\left(\frac{\sigma}{2}\right) + \rho(t_0) < R\sin\left(\frac{\sigma}{2}\right).$$

If $\rho(t_4) \geq r\sin\left(\frac{\sigma}{2}\right)$ then the conditions of Lemma 4.4.2 apply at t_4, otherwise,

$$0 \leq \rho(t_0) < (5r-4R)\sin\left(\frac{\sigma}{2}\right),$$

and so forth and so on, until

$$r \leq \frac{k}{k+1} R$$
$$\Downarrow$$
$$\frac{r}{R-r} \leq k.$$

Choosing the minimal k, we arrive at $k = \left\lceil \frac{r}{R-r} \right\rceil$, and $t_k < t_0 + \left\lceil \frac{k}{2} \right\rceil \frac{r}{v} (2\pi - \sigma) + \left\lfloor \frac{k}{2} \right\rfloor \frac{R}{v} \sigma$. ■

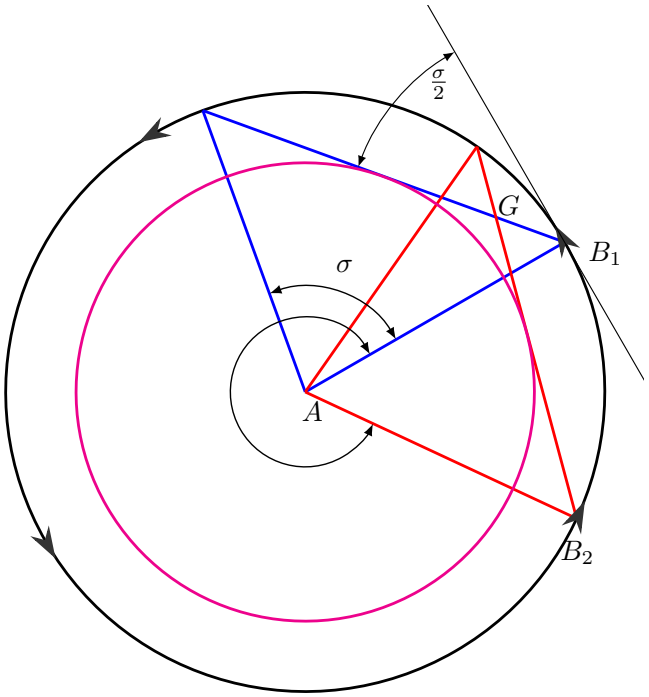


Figure 4.5: Entering State a with $\beta = \frac{\sigma}{2}$ and $\rho < r \sin\left(\frac{\sigma}{2}\right)$ transitions to State a with $\rho > r \sin\left(\frac{\sigma}{2}\right)$, via State b. An agent traveling form point B_1 senses the beacon, located at G, only when reaching point B_2.

Theorem 4.1 (UCSLSV)**.** *An agent with kinematics (2.1) with constant speed v which switches its rotation rate according to its perception of a stationary beacon (4.1) located at a distance of $\rho(t_0)$ from the agent, reaches a circular orbit with $\omega = \frac{v}{r}$ such that the distance between the beacon and its center of rotation is not greater than $r \cos\left(\frac{\sigma}{2}\right)$ in less than*

$$T = \left(1 + \left\lceil \frac{\rho(t_0) + 2R}{2(R-r)\sin\left(\frac{\sigma}{2}\right)} \right\rceil\right) \left(\frac{2\pi r + (R-r)\sigma}{v}\right)$$
$$+ \left\lceil \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rceil \frac{r}{v} (2\pi - \sigma) + \left\lfloor \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rfloor \frac{R}{v} \sigma + \frac{R-r}{v} \tan\left(\frac{\sigma}{2}\right)$$

time.

Proof If not starting with initial conditions such that the beacon is inside the r radius circle the agent is traveling on, as illustrated in Figure 4.4, then the agent will reach a configuration where $\beta = \frac{\sigma}{2}$ in less than $t_1 = t_0 + \frac{r}{v}(2\pi - \sigma) + \frac{R}{v}\sigma$. After that time the distance from the beacon must be less than $\rho(t_1) = \rho(t_0) + 2R$, and the system will engage in cycles with $\beta = -\frac{\sigma}{2}$, as described in Lemma 4.4.1, until getting close enough to the beacon at t_2,

$$\rho(t_1) = \rho(t_0) + 2R \leq 2n\,(R-r)\sin\left(\frac{\sigma}{2}\right)$$

$$\Downarrow$$

$$n = \left\lceil \frac{\rho(t_0) + 2R}{2\,(R-r)\sin\left(\frac{\sigma}{2}\right)} \right\rceil;$$

$$t_2 = t_1 + n\left(\frac{r}{v}(2\pi - \sigma) + \frac{R}{v}\sigma\right) = t_0 + (1+n)\left(\frac{2\pi r + (R-r)\sigma}{v}\right);$$

at which time the system begins, at the latest, to follow the behavior described in Lemma 4.4.3, until t_3.

$$t_3 = t_2 + \left\lceil \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rceil \frac{r}{v}(2\pi - \sigma) + \left\lfloor \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rfloor \frac{R}{v}\sigma + \frac{R-r}{v}\tan\left(\frac{\sigma}{2}\right),$$

and finally,

$$T = t_3 - t_0 = t_2 - t_0 + \left\lceil \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rceil \frac{r}{v}(2\pi - \sigma) + \left\lfloor \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rfloor \frac{R}{v}\sigma + \frac{R-r}{v}\tan\left(\frac{\sigma}{2}\right)$$

$$= \left(1 + \left\lceil \frac{\rho(t_0) + 2R}{2\,(R-r)\sin\left(\frac{\sigma}{2}\right)} \right\rceil\right)\left(\frac{2\pi r + (R-r)\sigma}{v}\right)$$

$$+ \left\lceil \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rceil \frac{r}{v}(2\pi - \sigma) + \left\lfloor \frac{\left\lceil \frac{r}{R-r} \right\rceil}{2} \right\rfloor \frac{R}{v}\sigma + \frac{R-r}{v}\tan\left(\frac{\sigma}{2}\right)$$

Q.E.D. ■

Chapter 5

AntAlate

AntAlate is a software framework for Unmanned Aerial Vehicle (UAV) autonomy, designed to streamline and facilitate the work of application developers, particularly in deployment of Multi-Agent Robotic Systems (MARS). We created AntAlate in order to bring our research in the field of multi-agent systems from theoretical results to both advanced simulations and to real-life demonstrations. Creating a framework capable of catering to MARS applications requires support for distributed, decentralized, control using local sensing, performed autonomously by groups of identical anonymous agents. Though mainly interested in the emergent behavior of the system as a whole, we focused on the single agent and created a framework suitable for a system of systems approach, while minimizing the hardware requirements of the single agent. Global observers or even a centralized control can be added on top of AntAlate, but the framework does not require a global actor to finalize an application. The same applies to a human in the loop, and fully autonomous UAV applications can be written in as straightforward a way as can semi-autonomous applications. In this paper we describe the AntAlate framework and demonstrate its utility and versatility.

5.1 Introduction

Unmanned mobile robotic platforms overcame barriers to reach an ever-growing user-base in recent years. From the military to the civilian domain, from graduate school laboratories to grade school classes, and from the highly specialized professional's grasp to the enthusiastic hobbyist's reach, applications of unmanned ground, surface, underwater and aerial vehicles have become widespread. The growing availability of low-power-high-performance mini-computers and micro-controllers, as well as the level of maturity and popularity reached by open-source software systems such as the Robot Operating System (*ROS*[1]), *ArduPilot*[2], and *Dronekit*[3], have made this prevalence pos-

[1] https://www.ros.org/
[2] http://ardupilot.org/
[3] https://dronekit.io/

sible. In their survey, Lim et al.[LPLK12] explain and demonstrate how open-source UAV projects can empower the UAV application developer; the emergence of reliable software frameworks for UAV application development allows both professional and hobbyist developers to focus their efforts on the distinctive features of their own application while leaving the necessary yet onerous task of infrastructure development to the framework maintainers.

Demeyer et al.[DMNS97] describe frameworks as "semi-finished programs"; the applications being finalized by application developers that use the framework. The more functionality the framework offers, the more constraints it imposes on the future application developers. The framework designer must therefore resolve the conflicting tension between cross-context reuse and ease of adoption and adaptation. To balance the tension, Demeyer et al. offer guidelines to enhance three open system requirements: *Interoperability*, or the ability to run on various configurations; *Distribution*, or the ability to reliably run over a set of physically distributed nodes; and *Extensibility*, or the ability to finalize the application with added customization without having to change any of the framework's internal modules. One of the primary dilemmas encountered by anyone trying to create a useful framework for multi-agent robotic systems (MARS) incorporating UAVs, is how much emphasis must be given to the particular UAV aspects of the framework; another dilemma is how to incorporate the swarm enabling multi-agent interaction mechanisms. Too much emphasis on UAV applications might leave the framework unfit for other platforms such as ground vehicles, while not giving the UAV platform enough consideration might leave the framework too high-level, requiring extensive tailoring from the ultimate application developer. Balancing the emphasis on swarm-enabling mechanisms is perhaps even more problematic, since any mechanism built into the framework limits the use of alternatives by future applications.

Chamanbaz et al.[CMZ+17], for instance, recently created the *Marabunta*[4] framework built for enabling swarming capabilities to general purpose robotic systems, and demonstrated their framework's capabilities in classic swarming scenarios for ground and surface vehicles. Preferring interoperability over distribution and extensibility to some extent, much of the implementation is left to the final application developer, and the framework's synchronous calls to abstract functions from a single-threaded main loop per robot might become unfit for a UAV given a resource-demanding behavior. On the other hand, Preiss et al.[PHSA17] described *Crazyswarm*[5], a framework for indoor swarm applications using the *Crazyflie*[6] platform, and the highly popular ROS middleware, used in conjunction with a global object tracker such as *VICON*[7] for external feedback. While Crazyswarm applications perform most of their in-flight computation on-board the Crazyflie platform, a base station is required in order to calculate and

[4]https://github.com/david-mateo/marabunta
[5]https://github.com/USC-ACTLab/crazyswarm
[6]https://www.bitcraze.io/crazyflie-2-1
[7]https://www.vicon.com

broadcast pose estimates and is therefore an integral part of the Crazyswarm system architecture. Crazyswarm is therefore an example of a specialized framework willing to sacrifice generality for performance, as demonstrated in an impressive video featuring a swarm of 49 Crazyflies[8]. Arguably finding a middle ground between generality and specialization, Sanchez-Lopez et al.[SLMB+17] presented *Aerostack*[9] - a framework designed as a set of components organized in a multi-layered model. Ultimate application developers can create their own application by selecting a set of components from the Aerostack component library and modifying or adding additional components as needed, as long as the developers adhere to the Aerostack conventions, thus satisfying the extensibility framework requirement. The interoperability and distribution requirements are achieved inherently by using ROS as underlying middleware for the single agent's inter-process communication. Aerostack's swarming capabilities are enabled by the framework's social layer interface contracts, yet the mechanics of inter-agent communication is left for the application developer to finalize. A few examples of swarming solutions embedded into frameworks can be seen in the *Voltron* (Mottola et al. 2014[MMWG14]), *Buzz* (Pinciroli and Beltrame, 2016[PB16]), and *CyPhyHouse*[10]'s *Koord* (Ghosh et al. 2020[GJH+20]) programming languages. Though varying in implementation details, the development framework provided by each of these languages allows the ultimate application developer to write an application from a swarm (or a sub-group of a swarm's agents or super-group of sub-groups...) perspective with relative ease; this is done by including an underlying mechanism that propagates coordinating information between agents. Yet in applications where inter-agent communication is not required or even possible, these strengths become irrelevant, and with an increased number of agents the task of maintaining a distributed shared memory becomes a problem rather than a remedy.

For the past 20 years, our research team at the *Technion MARS laboratory*[11] has been focusing on developing algorithms that address a variety of global tasks with swarms of simple mobile agents. Our paradigm defines agents as anonymous (i.e. not specifically addressable by an identifier), oblivious (have little or no memory), identical hardware platforms, that rely on locally acquired information provided by simple sensors such as local pheromone level detectors [WLB96] [WLB99] [EB12a], proximity sensors [GEB08] [EB12b], or limited vision [BB17] [DB17] for their motion control decisions. Our work during these years led us to develop several types of local interaction-based motion rules for autonomous mobile agents in swarms deployed in various types of environments that achieve global tasks such as patrolling an area, gathering into a cohesive but flexible "cloud" of agents, coverage of regions for intruder detection, equitable distribution of workload, and path planning. See for example the works of our

[8] https://www.youtube.com/watch?v=D0CrjoYDt9w
[9] https://github.com/Vision4UAV/Aerostack
[10] https://cyphyhouse.github.io/index.html
[11] https://mars.cs.technion.ac.il/

team members [WB97], [YWB03], [FSAB06], [GEB08], [OYWB08], [EB14], [EB16], [BB17], [DB17], [APB18], [MB18], [AB19], [BDMB21], and [FB21]. We also addressed the issue of achieving guidance and steering of cohesive mobile agent swarms using some global "broadcast control" ideas, as presented in works by Segall and Bruckstein[SB16], Dovrat and Bruckstein[DB17], and Barel et al.[BMB18]; where the broadcast signal is often assumed to be acquired by only a random set of the swarm's agents. These ideas create a wealth of possibilities to deploy swarms of autonomous agents that can self-organize into cohesive, adaptive, and flexible-shaped constellations. These swarms can then be guided by a single user that communicates with the entire swarm via global broadcast signals based on observing the swarm's location, but without having precise information on any particular agent of the swarm. It is easy to imagine the wealth of applications such a system can address, from site surveillance to disaster relief to space exploration. Yet the fundamental capabilities and limitations of swarms of such agents are rather difficult to analyze theoretically, so novel mathematical approaches are often needed to prove task accomplishment and termination, to evaluate the time span necessary to do the work, and to assess the effects of random or byzantine failures of agents. As examples of our team's efforts we refer the reader to papers by Bruckstein et al. [BCE91], [Bru93], [AB11], [OB12], [EB12b], [SB16], [BMB16], and [BDMB21].

We created *AntAlate*[12] to deploy swarms of agents that perform our algorithms in the real world. Considering its usefulness beyond implementing our algorithms, we hereby offer the framework to the multi-agent robotics R&D community, to facilitate the implementation of systems demonstrating various types of interesting swarming behaviors. AntAlate expresses our preference of UAV platforms, particularly copters, over others, since copters can emulate the behavior of wheeled and fixed-winged platforms to a greater extent than vice versa. AntAlate enforces an orderly execution of behaviors by means of a mission control (MC) module which interfaces with the high-level control (HLC) of the UAV and an operator module. Though we recommend a human at a control station as the operator, a centralized (or distributed) control server, or an on-board node for fully autonomous applications will also do. We included an operator station HTTP client (OSC) in the framework for all but the fully autonomous operator agents to use, and an operator station server for inter-agent and human interface as a complementary project[13]. By design, the swarming mechanism in AntAlate is amorphous, and can either emerge in a bottom-up fashion from the single agent's behavior-set; be determined top-down by an operator; or any mixture of these approaches.

The remainder of this paper provides an in-depth description of AntAlate in Section 5.2, followed by a comparison of workflows when implementing the same algorithm to create ROS-based and AntAlate-based applications in Section 5.3, before concluding in Section 5.4.

[12]https://gitlab.com/nemala/alate
[13]https://gitlab.com/nemala/operator-station

5.2 Method

A good framework provides the final application developers a convenient trade-off between the freedom to write their own application and the constraints imposed by having useful functionality they will find unnecessary to modify. Though any part of the code in an open-source project can be edited, the parts which the framework authors deem immutable can be considered as the framework's *core* - developers are not required to alter these sections in order to write their own application. Hence, the framework core is generally where the benefits of using the framework present themselves. The framework's extendable parts should be well defined by *framework contracts* and mechanisms such as abstract classes that allow future developers to write modules that fit into the framework without significant overhead. The degrees of freedom the framework presents to the application developers can be thought of as a *design space*, where the framework contracts represent the axes, and different applications with different configurations can be represented by points in this space.

5.2.1 AntAlate Core

The core functionality of AntAlate is to coordinate between an operator, a set of payloads, sensors and algorithms running onboard the UAV, and the UAV's autopilot. Figure 5.1 shows a diagram of AntAlate's core components. Each component is a *NeMALA dispatcher*[14] node, communicating with other nodes by publishing messages to topics other interested nodes subscribe to. *NeMALA*[15] is a set of supporting projects for AntAlate, with *core* components for dispatching, publishing, and handling messages, and *tools*[16] to log and manage NeMALA dispatcher nodes and proxies. The dispatcher nodes are implemented in C++, utilizing *Boost*[17] and *ZeroMQ*[18], allowing nodes to communicate locally via inter-process communication, or TCP/IP if distributed over different computers. The ultimate application developers have control over the *distribution* of nodes, and can configure the method of communication between nodes by setting up *NeMALA proxies* catering their own project's architecture and requirements, adding to the framework's overall *interoperability*. AntAlate's core components are the Mission Control (MC, 5.2.1), High-Level Control (HLC, 5.2.1), Behavioral Module Arbiter (BMA, 5.2.1), and Operator Station Client (OSC, 5.2.1). Any future application requires only components of these four types, and some applications could do with less. Each of these components' executables expect a NeMALA proxy name and a configuration file path as arguments when run from the command line (except the BMA, which is special in its requirement of its own node name instead of a proxy name). The configuration file contains the node number used for each node, as well as the proxy endpoints

[14] https://gitlab.com/nemala/core
[15] https://gitlab.com/nemala
[16] https://gitlab.com/nemala/tools
[17] https://www.boost.org/
[18] https://zeromq.org/

and topics used. Interoperability and distribution are therefore easily achieved by using one or many proxies described in one or many configuration files, without the need to alter code, recompile the application, or even edit configuration text-files, but only by calling the core executables with different arguments instead. The configuration file is also where behaviors, autopilots, and operator servers are specified, giving the ultimate application developers control over the AntAlate design space.

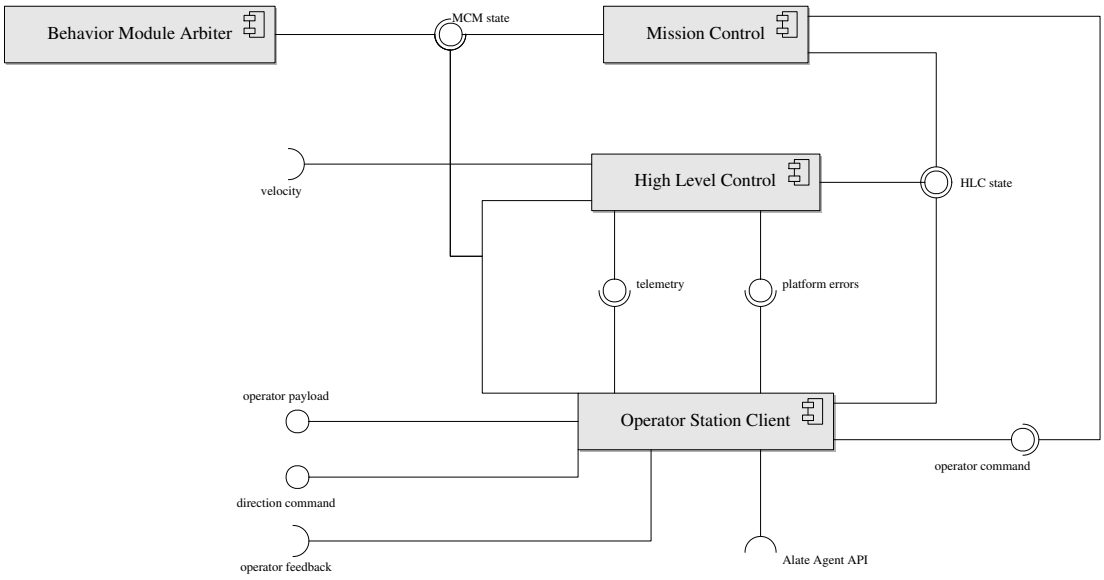


Figure 5.1: AntAlate core components and their interfaces.

MC - Mission Control

The mission control component provides logic to coordinate all other AntAlate components by maintaining a state machine, illustrated in figure 5.2, and publishing its state. The state machine's inputs are operator commands and the HLC component's state (see section 5.2.1); its output is the current mission state, which is provided as feedback to the operator, is used to initiate HLC processes, and perhaps most importantly from the framework point of view, governs the activation of sensors, payloads and algorithms by the Behavior Module Arbiter (BMA) component (see section 5.2.1).

Upon initialization, the mission control synchronizes with the HLC component's state machine and transitions itself to standby. An operator command to takeoff transitions the state machine to the autonomous states of taking off, performing a mission, returning to the launch site, and landing. A manual override initiated by a human pilot changes the HLC's state to manual, causing a transition in the mission control state as well. The mission control can then return to standby only after the HLC returns to its ready state, meaning the UAV's motors are disabled. If at any point the HLC reports that it is in its error state due to the autopilot shutting down, the mission state machine transitions to an unrecoverable error state.

HLC - High-Level Control

The HLC component is an abstraction of the UAV platform. The HLC subscribes to the MC state topic and a velocity command topic, and publishes its state, telemetry and error data. The HLC state is decided by a state machine, illustrated in figure 5.3, which coordinates the UAV autopilot abstraction with the MC state (see section 5.2.1).

When both autopilot and MC are up, the HLC enters its ready state. The HLC responds to a MC transition to its taking off state by a making an attempt to arm the UAV's motors while transitioning to the takeoff state. Failure to arm the motors brings the HLC state back to ready; success brings about a transition to gaining altitude, where the autopilot attempts to gain enough altitude to be considered airborne before running out of time and being forced to land by a transition to the landing state. While airborne, an MC transition to its landing or return to launch states causes the HLC to follow suit. A manual override is possible in any of the autonomous states. After landing and disarming the motors, the HLC returns to ready mode from either manual or autonomous states. If at any point the HLC loses communication with its autopilot, the state machine transitions to the unrecoverable state LLC down to inform the MC and Operator that the vehicle is about to crash. Low battery transitions the state machine to a low battery state.

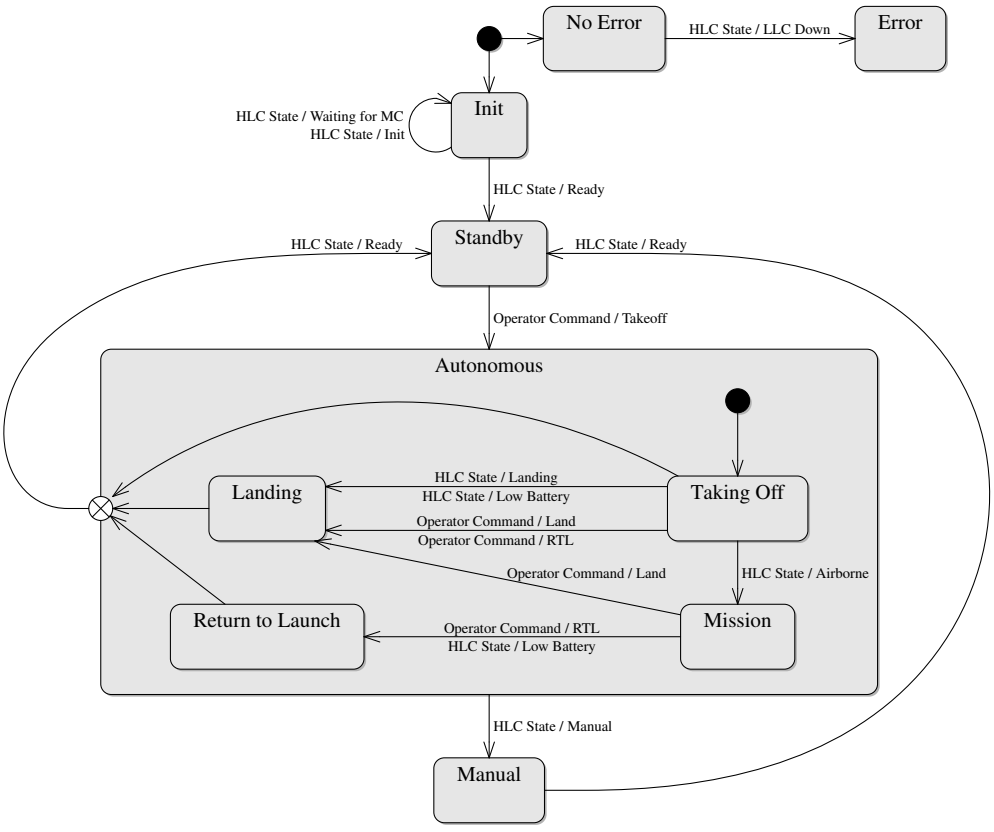


Figure 5.2: Mission Control State Machine.

BMA - Behavior Module Arbiter

The Behavior Module Arbiter activates behavior-sets according to a state topic it subscribes to. Multiple BMAs can be cascaded such that the root BMA subscribes to the MC state, and publishes its own arbitrary state for other BMAs to subscribe to. The BMA gives AntAlate an added degree of freedom in the distribution of behaviors over separate nodes, as well as being key to AntAlate's extensibility by activating plugin behaviors (see section 5.2.2). The configuration file given as an argument to the BMA executable tells the node which behavior set to run, and which proxies to subscribe to.

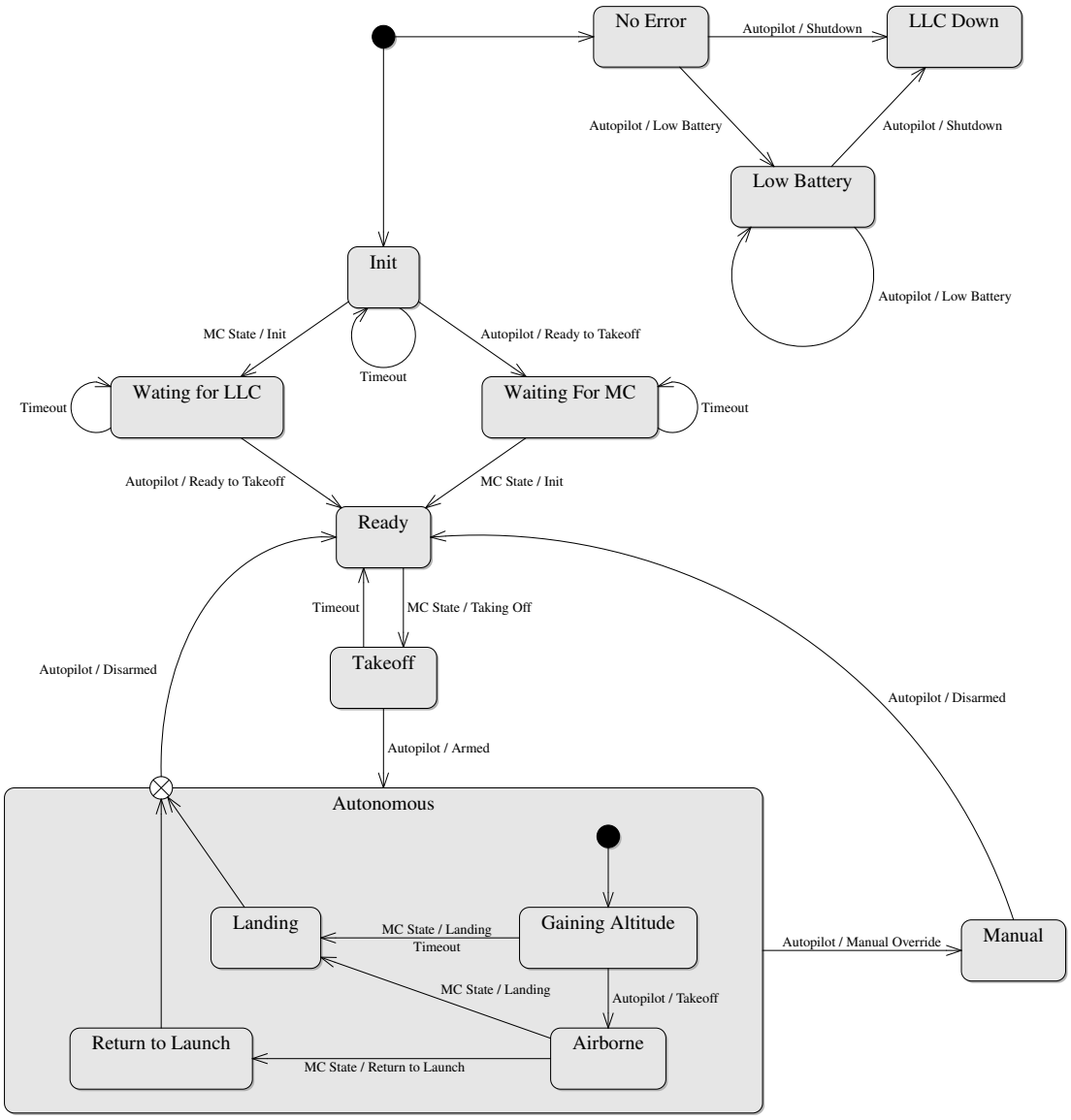


Figure 5.3: High-Level Control State Machine.

OSC - Operator Station Client

AntAlate requires an operator node to publish commands such as takeoff or land to the operator command topic. For example, a minimal operator node could be a BMA which publishes a takeoff command to the operator command topic every time the MC enters its standby state, as can be seen in Figure 5.4. Yet, in order to facilitate inter-agent, human-agent or human-swarm communication, we added an Operator Station Client that serves as an anonymous client to a server via HTTP, as can be seen in Figure 5.1. The OSC accepts tokens from the server, so anonymous protocols can stay anonymous, but any protocol requiring agents to be labeled is also supported. The OSC subscribes to the MC and HLC topics and forwards the messages to the server. The server replies with an operator command or a direction command if one was recently given, and the OSC publishes the commands received to their appropriate topics. In addition, outgoing operator payload and incoming feedback topics are left for the final application developers to use as they see fit. The server IP address and port are specified in the application's configuration file, which the OSC reads at runtime while setting up the node.

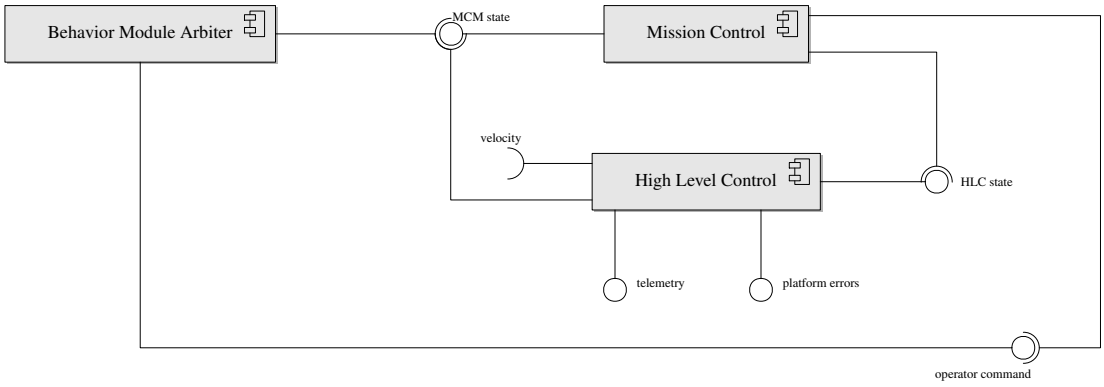


Figure 5.4: AntAlate minimal deployment.

5.2.2 AntAlate Design Space

Swarming protocols generally differ not only in the way their agents behave, but also in the way their agents sense the environment and communicate among themselves or with an external operator. UAV systems usually differ in type of flying platform and the autopilot providing low-level control over the flying platform. The AntAlate design space therefore is composed of three major axes: *Behavior* (5.2.2), *Autopilot* (5.2.2), and *Communication* (5.2.2, with *Sensing* split in implementation between these major axes.

Behaviors

We created AntAlate in order to easily implement and test new swarming behaviors on UAVs; during the design process, though, we found that there are other uses for the behavior mechanism other than swarming protocols, such as single-UAV autonomy, payload management, and sensing. Ultimately, the behavior mechanism can be used as a building block to create almost any type of UAV application.

Figure 5.5 shows the reusable design in the form of a class diagram; a BMA node is a NeMALA dispatcher that has a McStateMessageHandler which handles incoming McStateMessage type messages that arrive via a topic the BMA registers to. These messages encapsulate an instance of an enumeration type that represents a mission-state. The handler has an Arbiter which maps mission states with concrete behavior classes, and activates or deactivates its behaviors accordingly whenever a message containing a recognized state is received.

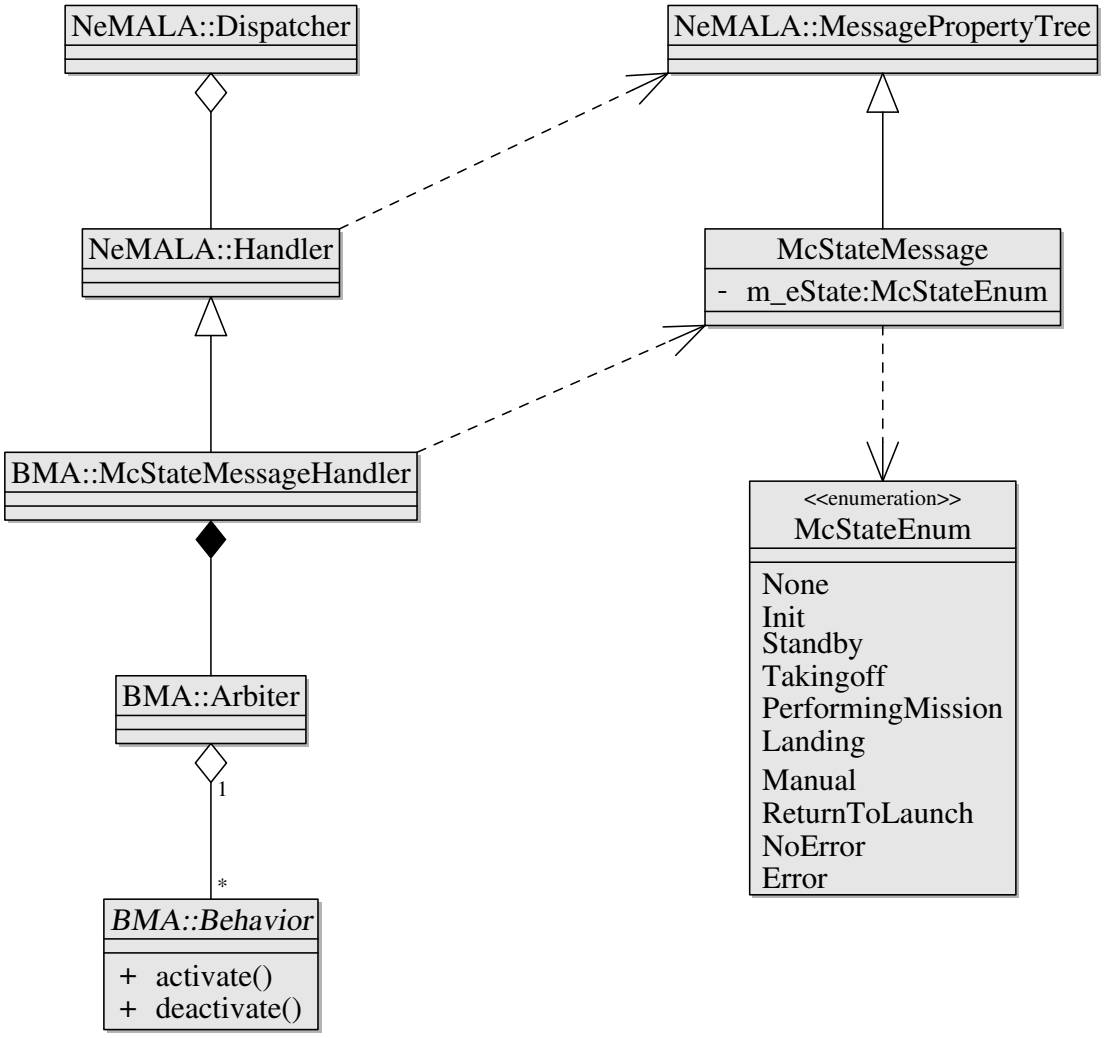


Figure 5.5: AntAlate's Behavior Module Arbiter class diagram. The behavior abstract class is a framework contract.

Using a plugin mechanism to populate arbiters with behaviors, the BMA generically controls the activation of behaviors while leaving the behavior specifics to future programmers. Behavior interaction is made easy by adding topics to publish and subscribe to, and BMAs can be cascaded by having behaviors publish mission-state messages to designated topics other than that of the original MC. To create a new behavior, one must create a shared library containing a class derived from the behavior abstract class and a concrete factory class to which the BMA delegates the construction and configuration of the behavior, along with its integration into the BMA node.

To add a behavior to an application, all that is required is that the application's configuration file includes an entry for that behavior, including which mission states activate and deactivate the behavior, the plugin library name, and to which topics the behavior subscribes and publishes to. No need to recompile the framework to change the configuration, even when adding new behaviors.

Autopilots

Autopilots, in this discussion's scope, are the hardware/software components that serve as an intermediate between the the actual UAV platform and AntAlate logic, including behaviors, operators and the mission control. The HLC and its autopilot class diagram are shown in Figure 5.6. AntAlate's modular design allows the extension through inheritance of the autopilot interface class to fit to a specific UAV API. The HLC's concrete autopilot class implements a Python-C++ bridge to facilitate the integration of python based APIs such as *Dronekit*[19] and *Tello*[20]. By using a bridge we can extend the concrete autopilot class without recompiling the AntAlate code-base; additional autopilot APIs can be covered by the same concrete autopilot class by adding Python implementations and updating a factory python script. Ultimate application developers can then choose which autopilot API to use by altering the application's configuration file.

Communication

The OSC (Section 5.2.1) provides some degree of freedom to the design space by means of the operator payload and feedback topics, yet imposes a specific HTTP request and expects the server's reply to be formatted in a specific way. We deliberately excluded a server from the framework to emphasize that the *server we produced*[21] only represents one example out of infinite possibilities. We encourage future application developers to use the OSC without alteration, and to write their own server, tailored for their application's behaviors, users, and use cases.

[19] https://dronekit.io/
[20] https://github.com/dji-sdk/Tello-Python
[21] https://gitlab.com/nemala/operator-station

Though we find it useful, the OSC is not the only extra-agent channel of communication allowed in AntAlate, and is indeed not even the only form of operator that falls into the constraints of the framework. Other operators can be implemented using the behavior mechanism, and other communication methods can be added as behaviors as well. In this context, inter-agent communication can be regarded as an AntAlate behavior, with a BMA node using any type of hardware/software communication stack, protocol, etc. NeMALA proxies, topics and messages can be used as well as general building blocks for future applications.

5.3 Results

We used AntAlate to implement a swarming algorithm we previously described and implemented using *ROS*[22] and *TurtleBot2*[23] platforms [DB17]. In this section, we will present our workflow using AntAlate and compare it with our ROS workflow, which may be familiar to most readers. We usually start our workflow with *NetLogo*[24] [Wil99] simulations to help refine our algorithms and to quickly make observations to base our theoretical hypotheses on. Figure 5.7 shows a *NetLogo simulation of our algorithm*[25] where a swarm of five agents are manipulated by the user taking over

[22] https://www.ros.org/
[23] https://www.turtlebot.com/turtlebot2/
[24] http://ccl.northwestern.edu/netlogo/
[25] http://ccl.northwestern.edu/netlogo/models/community/dovrat2017

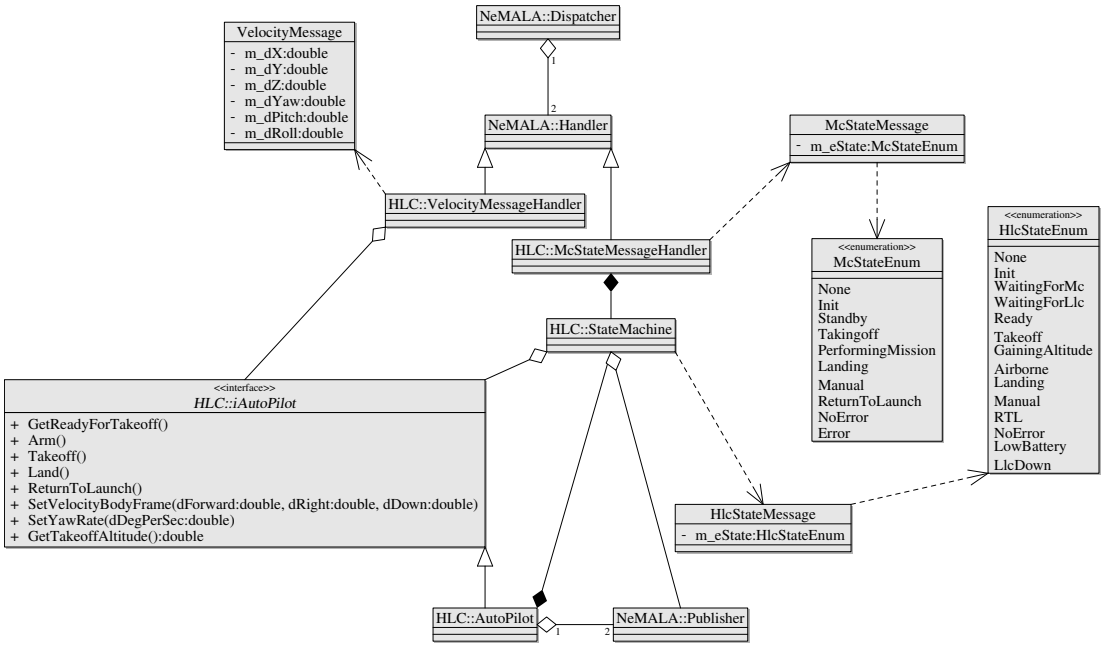


Figure 5.6: AntAlate's High Level Control class diagram. The iAutoPilot interface is a framework contract, the concrete autopilot is a Python-C++ bridge.

(drag-and-dropping) one of them.

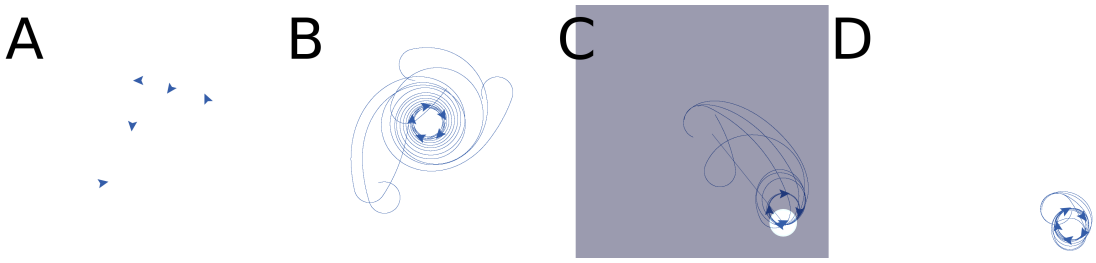


Figure 5.7: Our algorithm implemented with NetLogo. Five agents are initially dispersed at random on the plane **(A)**. The agents gather to a rotating regular pentagon **(B)**. Dragging one of the agents to the bottom left corner of the arena, the rest of the agents follow and ultimately form a rotating square about the "dragged leader" **(C)**. "Releasing" the leader, it returns to the swarm, and again a rotating pentagon forms, at the new location **(D)**.

Once satisfied with the results, we can choose a suitable mobile platform and design our application. Our swarming algorithm is fairly simple: every agent either detects other agents in its field of view and turns gently to one direction, or does not detect other agents and turns sharply to the same direction. In other words, the algorithm's input is a boolean valued true if other agents are detected or false otherwise, while the algorithm's output is a real value representing angular velocity, which switches between two predefined values according to the input.

The TurtleBot2 platform is perfectly suited for handling this algorithm, and the next step is to see which interfaces fit our algorithm's needs. Figure 5.8 shows the ROS graph of *our application*[26]. To detect other agents, we created a counter node which counts the number of magenta colored poles in an image frame and publishes the result. Figure 5.9, taken from *this short video*[27], shows our robots fitted with clearly visible colored rods, following a hand-held rod which acts as a leader, similar to the simulated behavior shown in Figure 5.7. Capitalizing on the TurtleBot2 capabilities, we added a detector node which reports if the agent has bumped into something or if its laser scanner has detected an obstacle nearby. The controller node translates the detected rod count to "false if zero, true otherwise", and executes our algorithm along with an overriding obstacle avoidance procedure if necessary. The controller then publishes a message with the correct forward velocity and turning rate to the relevant topic the robot's velocity command multiplexer subscribes to.

Our process using AntAlate was similar, and we included the swarm algorithm, as well as a video capture behavior, as a template application in the AntAlate code repository. Figure 5.10 shows the template application's deployment diagram, where each of the two behaviors gets its own BMA node. The swarm algorithm's BMA

[26]https://gitlab.com/dave.dovrat/turtle_bale
[27]https://www.youtube.com/watch?v=OA4ri3X4izw

subscribes to the direction command, telemetry, and operator payload topics from which it derives the direction the operator wants the swarm to move towards, the azimuth the agent is moving towards, and the existence of peer agents in a sector in front of the agent, respectively. The algorithm BMA's output is a velocity command which the HLC subscribes to, and which incorporates the operator's direction command with the swarm algorithm's output. The video capture behavior uses the command line

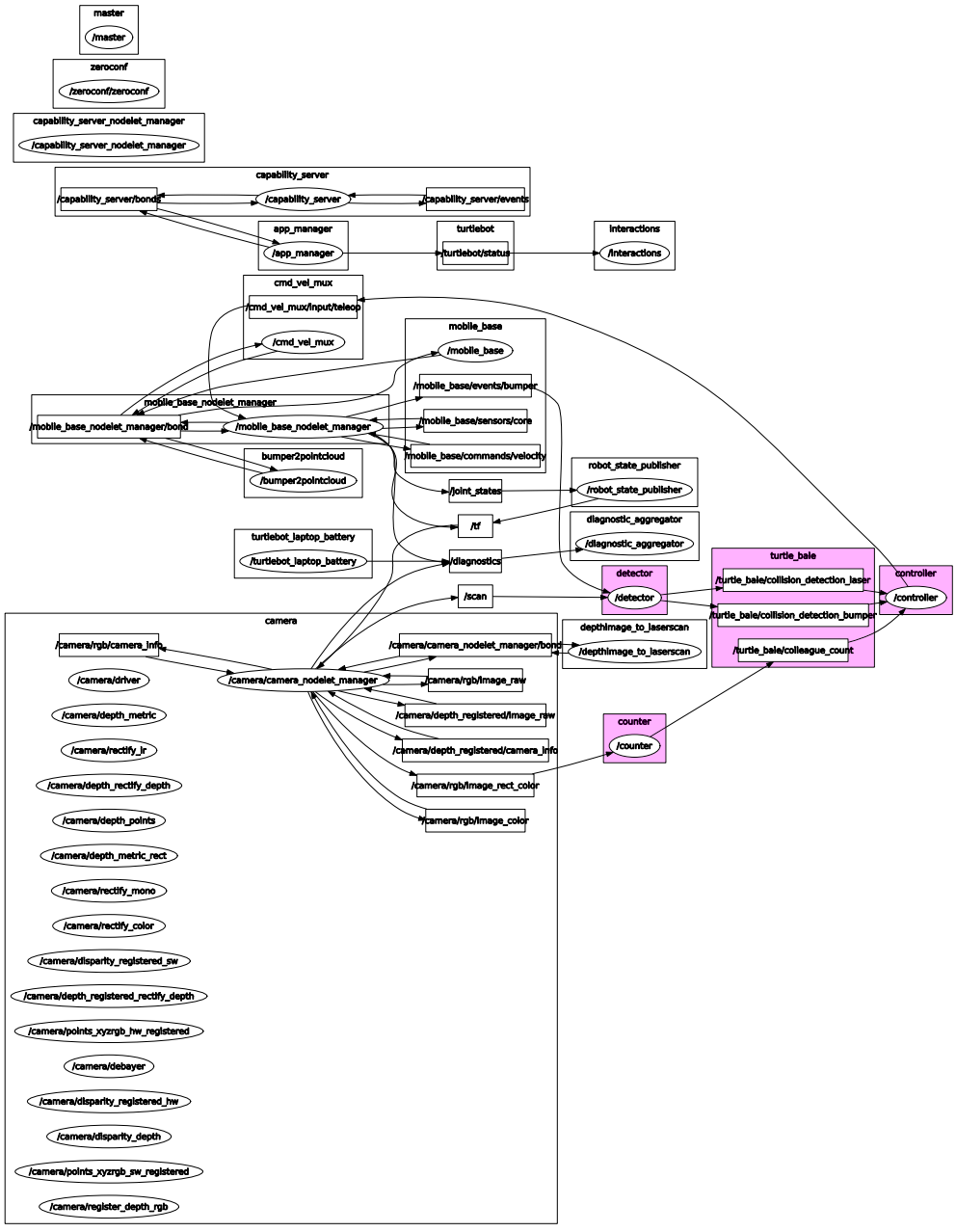


Figure 5.8: Our Turtlebot application's ROS graph. The application topics and nodes written by us are highlighted, while the rest of the graph was made available to us by the ROS community.

tool *ffmpeg*[28], which requires separate installation on the target machine, to capture video from a device specified in the application's configuration file; it neither subscribes nor publishes to any topic. Both BMAs subscribe to the MC state topic in order to activate their behaviors when appropriate.

AntAlate's modular and configurable design makes it fit for deployment using containers; we created *docker*[29] images for each AntAlate component type (MC, HLC, OSC, BMA), for linux/arm and linux/amd64 architectures, as well as two BMA images with pre-built behaviors, one for the swarm algorithm and one for the video capture behavior. We made these images publicly available on *Docker Hub*[30]. With these docker images we deployed the same code to three different configurations: A simulation that

[28]https://ffmpeg.org/
[29]https://www.docker.com/
[30]https://hub.docker.com/u/nemala

Figure 5.9: Our algorithm implemented with TurtleBots and ROS. The agents gather to a formation resembling a rotating square **(A)**. The agents follow a human leader **(B)**. When the leader goes away, the agents eventually return to their previous formation at a new location **(C)**.

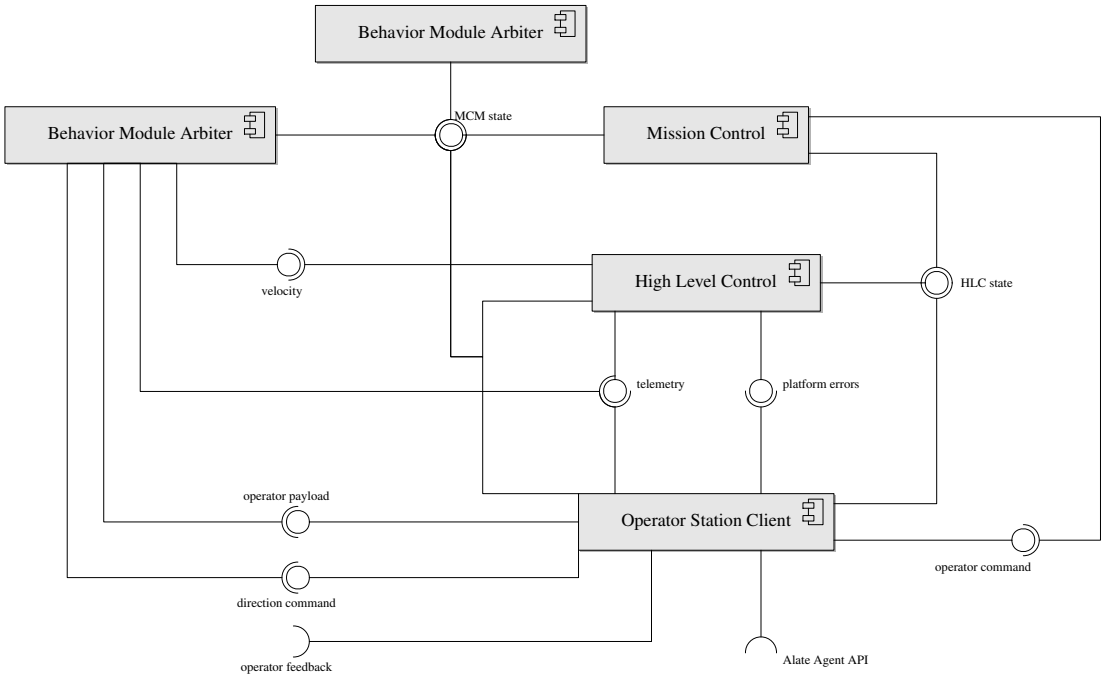


Figure 5.10: AntAlate template application deployment.

uses an external *SITL ArduCopter*[31] simulator as an autopilot and communicates with it via tcp; a 470mm UAV frame with a *pixhawk*[32] autopilot and a *Raspberry Pi*[33] on-board that runs AntAlate and communicates with the pixhawk through a *mavlink*[34] serial connection; and a *Tello*[35] with a companion Raspberry Pi that communicates with the autopilot via wifi using the Tello API. Figures 5.11, 5.12, and 5.13 are taken from a short video[36] featuring these configurations. From a design-space point of view, the first two configurations use the same *Dronekit*[37] autopilot, though on a different computer architecture, and the last two use the same computer architecture but with two different autopilots. The simulation, having no video devices, doesn't run a video capture BMA. We chose to run the template application's behaviors on separate nodes, but one BMA node would have sufficed.

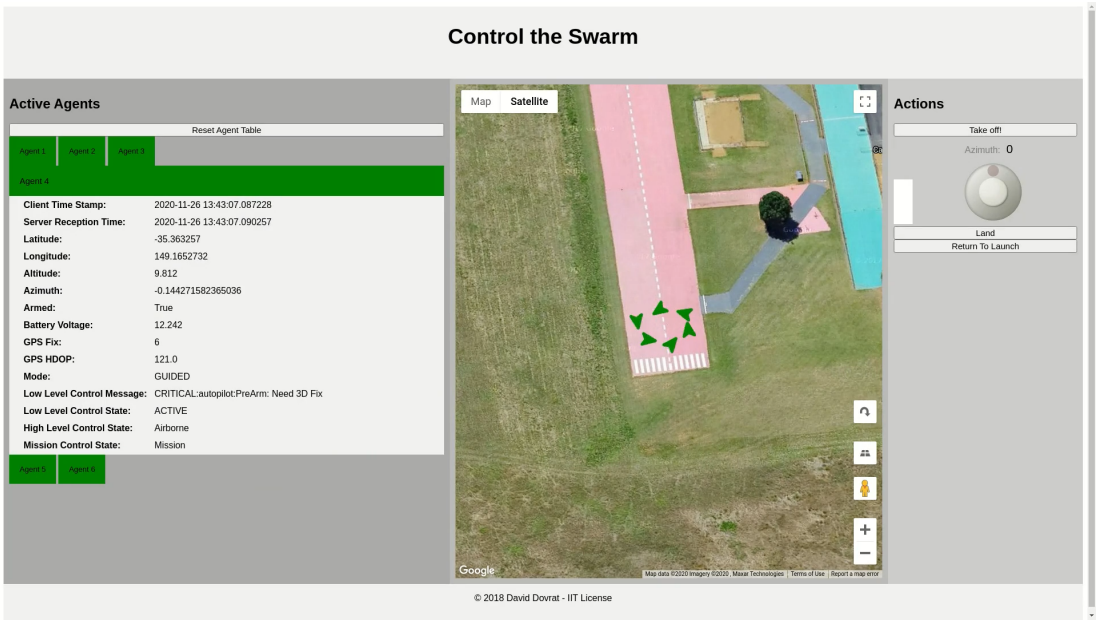


Figure 5.11: AntAlate template application demonstrated using ArduCopter's Software in The Loop (SITL) simulator. Screen capture from the AntAlate Server Graphical User Interface (GUI).

The template application's algorithm requires the detection of other agents as an input, yet the simulated agents have no camera or sensing device capable of detecting other agents, so we compensated for the missing ability by using a tailored server along with the OSC. The OSC's HTTP post request includes the HLC state which in turn includes the agent's location and orientation. When a new agent posts its

[31] https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html
[32] https://pixhawk.org/
[33] https://www.raspberrypi.org/
[34] https://mavlink.io/
[35] https://www.ryzerobotics.com/tello
[36] https://youtu.be/i9kctIlLkTg
[37] https://dronekit.io/

state for the first time, *our server*[38] assigns an index to it for further updates. The server keeps a data base, and each time an agent updates the server with its state, the

[38] https://gitlab.com/nemala/operator-station

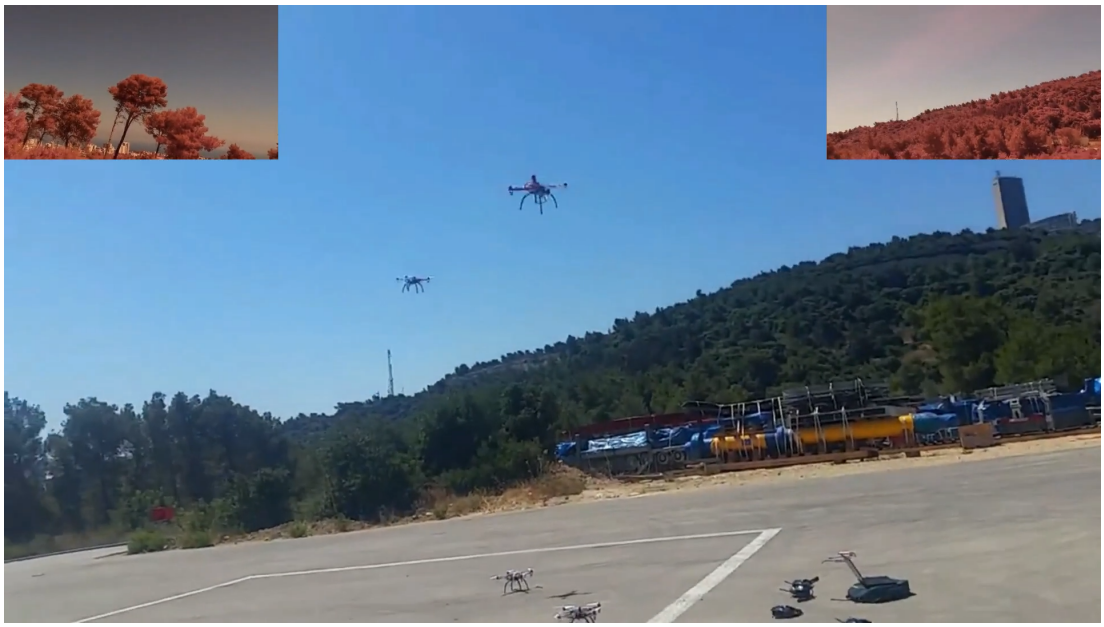


Figure 5.12: AntAlate template application demonstrated with 470mm quadcopters. Frames captured by the video capture behavior are presented in the upper corners.

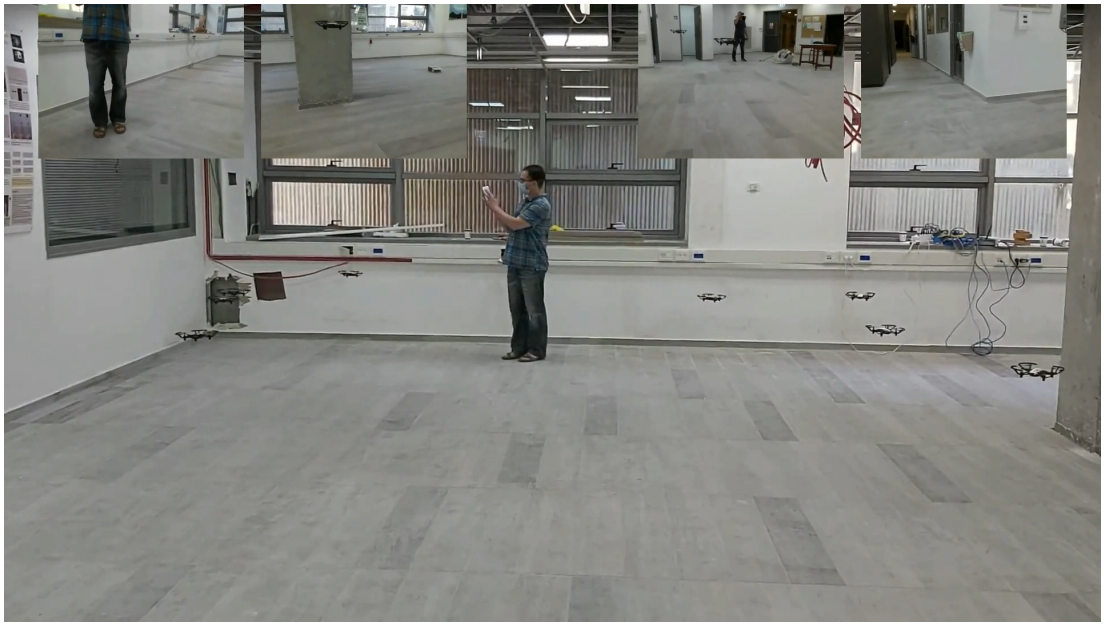


Figure 5.13: AntAlate template application demonstrated with Tello platforms. The upper part of the figure shows frames from the video capture behavior of four of the agents.

server records the state and solves the inverse geodesic problem using *geographiclib*[39] to answer whether another agent is in the sector in front of the updating agent in the HTTP reply's payload field. The detection range and field of view are server parameters. The OSC parses the HTTP reply to publish to the AntAlate operator command and direction command topics if necessary, and forwards the payload to the AntAlate operator payload topic. The BMA running the swarm algorithm subscribes to the payload topic and receives the server-calculated response as its required detection input. Had there been a peer detecting sensor, a BMA encapsulating that sensor would have published to some peer-detection topic, the swarming algorithm would have subscribed to that topic instead of the operator payload topic, and the server's database would have been unessential to the application. Which topic the component subscribes to is detailed in the configuration file given as input to AntAlate components.

Though the resulting applications are very different, the first being a ground robot that can avoid and handle bumping into obstacles, and the second an aerial robot that accepts broadcast signals, the workflow was almost identical: come up with an algorithm, build a process that uses available interfaces to encapsulate the algorithm, refine the resulting process to take advantage of available assets and compensate for missing assets, simulate, test, and deploy.

Looking at the amount of bytes in manually written files as an indicator of human effort, both workflows are comparable. The ROS application weighs 13,191 bytes not including the counter and collision detection nodes, and 23,638 bytes including them. the AntAlate application's behavior plugin code and configuration files weigh 18,118 bytes. This example showcases the power of a good framework - a few hundred lines of custom application code utilize thousands of lines of framework code. With AntAlate, we use the same nodes over and over; we code a new behavior once and configure it, as well as combine it with other behaviors, to form new applications without going over the entire design process for each added behavior. The application design is mostly implemented in the framework up until the point of concrete behaviors, which are left for the custom application developers to program and deploy according to their application's requirements and constraints.

5.4 Discussion

This paper introduces and describes AntAlate, a software framework we created for the future development of UAV MARS applications. AntAlate is a manifestation of our multi-agent systems paradigm; we chose a system of systems approach and focused on the single agent, while avoiding constraints on the swarming mechanism, expressing our preference for local over global sensing and communication, anonymity and obliviousness over distributed and shared memory, decentralized autonomy over centralized

[39] https://geographiclib.sourceforge.io/html/python/index.html

control. We incorporated an operator entity as a means of inter-agent and agent-human communication in an effort to keep the framework useful for MARS applications outside the scope of our paradigm notwithstanding.

We identified the UAV-MARS design-space and enforced framework contracts that promote maintainable future extensions. We employed proxies with configurable endpoints to enable the physical distribution of AntAlate nodes, and created a modular, interoperable, architecture which allows future developers to code once and deploy the same code on many different platforms and simulators, as we demonstrated with an example application.

Future framework enhancements include general types of operators that bridge between underlying frameworks and communication modals, in addition to the existing HTTP client. An example of such an operator could be a ROS node operator that exposes and forwards the AntAlate topics to ROS topics and vice versa, allowing the AntAlate agent to integrate into ROS applications. In addition, the development of some useful behaviors, such as peer recognition, obstacle avoidance, and simultaneous localization and mapping, could prove useful for developers interested in using these behaviors as building blocks in their own application without having to re-implement the wheel. Integrating a wider range of autopilots into the framework is another development priority, with the *Crazyflie API*[40] at the top of the autopilot backlog.

We hope the multi-agent robotics community will find AntAlate useful, and that AntAlate becomes the framework of choice for easy implementation of many interesting swarming behaviors, as well as an instrument for future collaborations and discussions.

[40] https://github.com/bitcraze/crazyflie-lib-python

Chapter 6

Discussion and Conclusion

This thesis discusses two frameworks; we solved the PCBOUP and UCSLSV problems, and flew a swarm of UAVs running applications of the multi-agent version of UCSLSV, but this thesis is more about how we solved these problems than about the actual solutions.

The fact that the automaton generating algorithm from Chapter 2 is the underlying process that abstracted away the otherwise unsolved, perhaps even unsolvable, nonlinear differential equations that describe the agent's trajectories and behavior in both PCBOUP and UCSLSV, is evidence to the main contribution of this work being the underlying algorithm rather than the solutions generated with it.

The same goes for the AntAlate application demonstrated in Section 5.3. Writing AntAlate required a great amount of effort, an effort we could have directed into creating perhaps dozens of stand-alone UAV applications. Yet now that AntAlate exists, the reduced overhead allows for the same amount of effort, previously required to create dozens of stand-alone UAV applications, to create hundreds of AntAlate applications. Will the investment pay off? That depends on the amount of applications that will be written with AntAlate in the future, on the collaborations that will be established due to using AntAlate as a medium, and to the general adoption of AntAlate by the robotics community.

As for the automata theory method for the analysis of unicycle pursuit problems, the investment in infrastructure has already payed off; the PCBOUP problem came up while we were trying to solve the multi-agent UCSLSV problem, and wanted to use PCBOUP as a stepping stone towards the multi-agent goal, only to find that this specific variation of the pursuit problem was yet unsolved. Now, having not only solved PCBOUP, but also creating Algorithm 2.1 in the process, we have a clear road-map towards solving the multi-agent UCSLSV, as well as tracking and capture of differently moving targets with agents using different control strategies and utilizing various sensing capabilities.

Bibliography

[AB11] Yaniv Altshuler and Alfred M. Bruckstein. Static and expanding grid coverage with ant robots: Complexity results. *Theoretical Computer Science*, 412(35):4661–4674, 2011.

[AB19] Michael Amir and Alfred M. Bruckstein. Probabilistic pursuits on graphs. *Theoretical Computer Science*, 795:459–477, 2019.

[APB18] Yaniv Altshuler, Alex Pentland, and Alfred M. Bruckstein. *Cooperative Swarm Cleaning of Stationary Domains*, chapter 2, pages 15–49. Springer International Publishing, 2018.

[BB04] F. Belkhouche and B. Belkhouche. On the tracking and interception of a moving object by a wheeled mobile robot. In *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, volume 1, pages 130–135 vol.1, 2004.

[BB17] Levi Itzhak Bellaiche and Alfred Bruckstein. Continuous time gathering of agents with limited visibility and bearing-only sensing. *Swarm Intelligence*, 11(3):271–293, 2017.

[BBR06] F. Belkhouche, B. Belkhouche, and P. Rastgoufard. Line of sight robot navigation toward a moving goal. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(2):255–267, 2006.

[BCE91] AM Bruckstein, N Cohen, and A Efrat. Ants, crickets, and frogs in cyclic pursuit. techreport CIS9105, Computer Science Department, Technion, http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/1991/CIS/CIS9105, May 1991.

[BDMB21] Ariel Barel, Thomas Dagès, Rotem Manor, and Alfred M Bruckstein. Probabilistic gathering of agents with simple sensors. *SIAM Journal on Applied Mathematics*, 81(2):620–640, 2021.

[Bee95] Randall D. Beer. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1):173–215, 1995.

[BMB16] Ariel Barel, Rotem Manor, and Alfred M. Bruckstein. Come together: Multi-agent geometric consensus (gathering, rendezvous, clustering, aggregation). techreport CIS-2016-03, Technion, http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2016/CIS/CIS-2016-03, March 2016.

[BMB18] Ariel Barel, Rotem Manor, and Alfred M. Bruckstein. On steering swarms. In Marco Dorigo, Mauro Birattari, Christian Blum, Anders L. Christensen, Andreagiovanni Reina, and Vito Trianni, editors, *Swarm Intelligence*, pages 403–410, Cham, 2018. Springer International Publishing.

[Bru93] Alfred M Bruckstein. Why the ant trails look so straight and nice. *The Mathematical Intelligencer*, 15(2):59–62, 1993.

[CMZ+17] Mohammadreza Chamanbaz, David Mateo, Brandon M. Zoss, Grgur Tokić, Erik Wilhelm, Roland Bouffanais, and Dick K. P. Yue. Swarm-enabling technology for multi-robot systems. *Frontiers in Robotics and AI*, 4:12, 2017.

[DB17] D. Dovrat and A. M. Bruckstein. On gathering and control of unicycle a(ge)nts with crude sensing capabilities. *IEEE Intelligent Systems*, 32(6):40–46, November 2017.

[DLN21] Rollen S D'Souza, Robbert Louwers, and Christopher Nielsen. Piecewise linear path following for a unicycle using transverse feedback linearization. *IEEE Transactions on Control Systems Technology*, 2021.

[DMNS97] Serge Demeyer, Theo Dirk Meijler, Oscar Nierstrasz, and Patrick Steyaert. Design guidelines for 'tailorable frameworks'. *Communications of the ACM*, 40(10):60–64, 1997.

[EB12a] Yotam Elor and Alfred M Bruckstein. Multi-a (ge) nt graph patrolling and partitioning. In *Science: Image In Action*, pages 18–33. World Scientific, 2012.

[EB12b] Yotam Elor and Alfred M. Bruckstein. A "thermodynamic" approach to multi-robot cooperative localization. *Theoretical Computer Science*, 457:59–75, 2012.

[EB14] Yotam Elor and Alfred M. Bruckstein. "Robot Cloud" gradient climbing with point measurements. *Theoretical Computer Science*, 547:90–103, 2014.

[EB16] Gidi Elazar and Alfred M. Bruckstein. Antpap: Patrolling and fair partitioning of graphs by a(ge)nts leaving pheromone

traces. techreport CIS-2016-04, Computer Science Department, Technion, http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2016/CIS/CIS-2016-04, April 2016.

[ET 16] Koray S. Erer, Raziye Tekin, and M. Kemal Özgören. Biased proportional navigation with exponentially decaying error for impact angle control and path following. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 238–243, June 2016.

[FB21] Roee M. Francos and Alfred M. Bruckstein. Search for smart evaders with sweeping agents. *Robotica*, FirstView:1–36, April 2021.

[FSAB06] Ariel Felner, Yaron Shoshani, Yaniv Altshuler, and Alfred M Bruckstein. Multi-agent physical a* with large pheromones. *Autonomous Agents and Multi-Agent Systems*, 12(1):3–34, 2006.

[GEB08] Noam Gordon, Yotam Elor, and Alfred M. Bruckstein. Gathering multiple robotic agents with crude distance sensing capabilities. In Marco Dorigo, Mauro Birattari, Christian Blum, Maurice Clerc, Thomas Stützle, and Alan F. T. Winfield, editors, *Ant Colony Optimization and Swarm Intelligence*, pages 72–83, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[GJH+20] R. Ghosh, J. P. Jansch-Porto, C. Hsieh, A. Gosse, M. Jiang, H. Taylor, P. Du, S. Mitra, and G. Dullerud. Cyphyhouse: A programming, simulation, and deployment toolchain for heterogeneous distributed coordination. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6654–6660, 2020.

[Gla75] Leon Glass. Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology*, 54(1):85–107, 1975.

[JTWS19] Bhargav Jha, Ronny Tsalik, Martin Weiss, and Tal Shima. Cooperative guidance and collision avoidance for multiple pursuers. *Journal of Guidance, Control, and Dynamics*, 42(7):1506–1518, 2019.

[LaS60] J. LaSalle. Some extensions of Liapunov's second method. *IRE Transactions on Circuit Theory*, 7(4):520–527, 1960.

[Lib03] Daniel Liberzon. *Switching in systems and control*. Springer Science & Business Media, 2003.

[LPLK12] H. Lim, J. Park, D. Lee, and H. J. Kim. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *IEEE Robotics Automation Magazine*, 19(3):33–45, Sep. 2012.

[MB18] Rotem Manor and Alfred M. Bruckstein. *Chase Your Farthest Neighbour*, volume 6, chapter 2, pages 103–116. Springer International Publishing, Cham, 2018.

[MG10] Eran D. B. Medagoda and Peter W. Gibbens. Synthetic-waypoint guidance algorithm for following a desired flight trajectory. *Journal of Guidance, Control, and Dynamics*, 33(2):601–606, 2010.

[MMWG14] Luca Mottola, Mattia Moretta, Kamin Whitehouse, and Carlo Ghezzi. Team-level programming of drone sensor networks. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 177–190, 2014.

[Moo90] Cristopher Moore. Unpredictability and undecidability in dynamical systems. *Phys. Rev. Lett.*, 64:2354–2357, May 1990.

[Nah07] P.J. Nahin. *Chases and escapes: The mathematics of pursuit and evasion.* Princeton University Press, Princeton, New Jersey, 2007.

[OAE16] Tiago Oliveira, A. Pedro Aguiar, and Pedro Encarnação. Moving path following for unmanned aerial vehicles with applications to single and multiple target tracking problems. *IEEE Transactions on Robotics*, 32(5):1062–1078, 2016.

[OB12] Frédérique Oggier and Alfred Bruckstein. *On Cyclic and Nearly Cyclic Multiagent Interactions in the Plane*, volume 218, chapter 1, pages 513–539. Springer Basel, Basel, 2012.

[OYWB08] Eliyahu Osherovich, Vladimir Yanovki, Israel A. Wagner, and Alfred M. Bruckstein. Robust and efficient covering of unknown continuous domains with simple, ant-like a(ge)nts. *The International Journal of Robotics Research*, 27(7):815–831, 2008.

[PB16] C. Pinciroli and G. Beltrame. Swarm-oriented programming of distributed robot networks. *Computer*, 49(12):32–41, 2016.

[PHSA17] James A. Preiss, Wolfgang Hönig, Gaurav S. Sukhatme, and Nora Ayanian. Crazyswarm: A large nano-quadcopter swarm. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3299–3304. IEEE, 2017. Software available at `https://github.com/USC-ACTLab/crazyswarm`.

[RHGS15] Ashwini Ratnoo, Shmuel Y. Hayoun, Asaf Granot, and Tal Shima. Path following using trajectory shaping guidance. *Journal of Guidance, Control, and Dynamics*, 38(1):106–116, 2015.

[SB16] Ilana Segall and Alfred Bruckstein. On stochastic broadcast control of swarms. In Marco Dorigo, Mauro Birattari, Xiaodong Li, Manuel López-Ibáñez, Kazuhiro Ohkura, Carlo Pinciroli, and Thomas Stützle, editors, *Swarm Intelligence*, pages 257–264, Cham, 2016. Springer International Publishing.

[SG07] Tal Shima and Oded M Golan. Head pursuit guidance. *Journal of Guidance, Control, and Dynamics*, 30(5):1437–1444, 2007.

[Shi07] Tal Shima. Deviated velocity pursuit. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6782, 2007.

[Shn98] Neryahu A Shneydor. *Missile guidance and pursuit: kinematics, dynamics and control*. Elsevier, 1998.

[SKHP97] Olaf Stursberg, Stefan Kowalewski, Ingo Hoffmann, and Jörg Preußig. Comparing timed and hybrid automata as approximations of continuous systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems IV*, pages 361–377, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[SLMB+17] Jose Luis Sanchez-Lopez, Martin Molina, Hriday Bavle, Carlos Sampedro, Ramón A. Suárez Fernández, and Pascual Campoy. A multi-layered component-based approach for the development of aerial robotic systems: The aerostack framework. *Journal of Intelligent & Robotic Systems*, 88(2):683–709, Dec 2017.

[ST10] A. V. Savkin and H. Teimoori. Bearings-only guidance of a unicycle-like vehicle following a moving target with a smaller minimum turning radius. *IEEE Transactions on Automatic Control*, 55(10):2390–2395, 2010.

[TS17] Twinkle Tripathy and Arpita Sinha. Unicycle with only range input: An array of patterns. *IEEE Transactions on Automatic Control*, 63(5):1300–1312, 2017.

[WB97] Israel A Wagner and Alfred M Bruckstein. Row straightening via local interactions. *Circuits, Systems and Signal Processing*, 16(3):287–305, 1997.

[Wil99] U. Wilensky. Netlogo. software, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL., 1999. http://ccl.northwestern.edu/netlogo/.

[WLB96] Israel A. Wagner, Michael Lindenbaum, and Alfred M. Bruckstein. Smell as a computational resource - A lesson we can learn from the ant. In *Fourth Israel Symposium on Theory of Computing and Systems, ISTCS*, pages 219–230. IEEE Computer Society, June 1996.

[WLB99] I.A. Wagner, M. Lindenbaum, and A.M. Bruckstein. Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15(5):918–933, 1999.

[YWB03] Vladimir Yanovski, Israel A Wagner, and Alfred M Bruckstein. A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37(3):165–186, 2003.

בעיית הרדיפה הראשונה אותה נפתור בחיבור זה הינה בעיית הרדיפה הקלאסית אותה הציג פייר בוגה, אך בשני שינויים שהופכים אותה למאתגרת, עד כדי כך שפתרונה פורסם לראשונה (על ידנו) שבועות מספר לפני הגשת חיבור זה. השינוי הראשון הוא שהסוכן הרודף מאולץ לנוע כחד-אופן, והשינוי השני הוא שהסוכן והמטרה נעים באותה מהירות קבועה. הבעיה השנייה אותה נפתור היא בעיה בה על הסוכן הרודף להגיע למשואה נייחת, כאשר חוק התנועה של הסוכן הוא שכאשר הסוכן מגלה באמצעות חיישן בעל זוית מרכזית מוגבלת הפונה קדימה את המשואה, עליו להסתובב בסיבוב עם רדיוס קבוע, וכאשר הסוכן אינו רואה את המטרה, עליו להסתובב ברדיוס קבוע קטן מהרדיוס המקורי. את הבעיה השניה פתרנו במקור לפני שנים אחדות, על ידי חישובים גיאומטריים מפרכים, ואנו פותרים אותה עתה שוב תוך שימוש בשיטה המוצגת כדי להדגים איך השיטה המוצגת מפשטת במידה רבה את ההוכחה.

בנוסף, מוקדש פרק בחיבור זה להצגת עבודת מסגרת תוכנה בשם AntAlate (נמלה מכונפת) שכתבנו על מנת לאפשר פיתוח מהיר של יישומים (אפליקציות) לכלי טיס אוטונומיים, כגון רחפנים. בעבודת המסגרת הפשטנו (אבסטרקציה) את כלי הטייס ובכך אנו מאפשרים למפתחי היישומים להתרכז בפן האלגוריתמי של ההתנהגות העצמאית (אוטונומיה) של הכלים, וכן הפחתנו את הצימוד בין כלי הטיס (הפלטפורמה) לשימוש בו (האפליקציה). השימוש בנמלה מאפשר למפתחי היישומים לבחור בעצמם ברמת האוטונומיה הרצויה לכלי הטיס שלהם בקלות ובמהירות, בזכות ממשקים גמישים המאפשרים בחירה בין ניהוג מרחוק, הנחיה מרחוק, פיקוח מרחוק, ועד אוטונומיה מלאה ללא כל התערבות אנושית. כתבנו את הנמלה המכונפת כמענה לצורך שעלה במסגרת עבודתנו במעבדה לחקר מערכות מרובות סוכנים בפקולטה למדעי המחשב בטכניון. רצינו מערכת שתאפשר לנו לבחון אלגוריתמים נחיליים בסימולציה, בניסוי במעבדה, ולבסוף גם בניסוי שדה, ורצינו להשתמש בדיוק באותו הקוד מבלי להזדקק להתאמות ושינויים עבור כל שלב ניסוי. כדי לתמוך במחקר שלנו, אך גם בגישות יותר ריכוזיות בניהול קבוצות של כלי טייס, הכנסנו מנגנונים שמאפשרים תמיכה בגישות ביזור וריכוז שונות.

בפרק החותם את החיבור, נפנה את מבטנו אל העתיד, ונצפה אל עבר בעיות שונות אותן ניתן לפתור על ידי השיטה המתוארת כאן.

תקציר

בעיות מרדף עוסקות בתרחישים בהן ישנו סוכן שמפעיל מדיניות מסויימת על מנת להשיג מטרה. למרות שהמדיניות שהסוכן מפעיל על מנת להשיג את מטרתו יכולה להיות פשוטה למדי, למשל תמיד לנוע לכיוון המטרה, אותה מדיניות פשוטה יכולה להוביל להתנהגות שאיננה קלה כלל לתיאור בשפה המתימטית או בכל שפה אחרת. בנוסף, למרות שהמדיניות יכולה להיות פשוטה לכאורה, לא ברור האם אותה מדיניות אכן תסיע לסוכן להשיג את מטרתו, באיזו מידה של וודאות, תוך כמה זמן, ובאילו תנאים.

בעיות מרדף מעניינות מדענים זה מאות שנים. אחד החוקרים שניסחו את בעיית המרדף לראשונה היה פייר בוגה (Pierre Bouguer), כששאל במאה השמונה עשרה איזו צורה תהיה למסלולה של ספינת שודדי-ים המנסים לתפוס אוניית סוחר. המתמטיקאים של דורו רק החלו לייצר את הכלים באמצעותם אפשר לענות על השאלה הזו, כפתרון למשוואה דיפרנציאלית, והבעיה נותרה ללא מענה שנים רבות. גם בכלים הזמינים למתמטיקאים בני זמננו, ישנן הרבה בעיות מרדף שנותרו ללא פתרון, וחיבור זה עוסק בשיטה לבניית אוטומטים המתארים בצורה מופשטת (אבסרטקציה) את יחסי הגומלין בין הסוכן והמטרה.

הפשטה זו מאפשרת לנו לבחון את התנהגות המערכת כתהליך שחל על מודל דטרמיניסטי. אין אנו מעוניינים עוד בהתנהגות הסוכן, אלא אנו ממקדים את מבטנו על החיבור הבלתי נראה שבין הסוכן למטרתו. במונחים של תורת הגרפים, אנו מסתכלים על הקשת המחברת את שני הצמתים במקום להסתכל על הצומת שמייצגת את הסוכן. במונחים מעולם הרובוטיקה, אנו מתייחסים לקשת זו כאל זרוע רובוטית בעלת שלושה מפרקים: כיוון הסוכן שמשתנה לפי מדיניות הרדיפה, המרחק בין הסוכן למטרתו, המשתנה כתוצאה ממשוואות התנועה של הסוכן והמטרה, וכיוון תנועת המטרה, שמשתנה לפי מדיניות הבריחה אם כזו קיימת. את מרחב התצורות (מרחב הקונפיגורציה) של הזרוע הרובוטית אנו מחלקים למחלקות שקילות לפי אופי ההתנהגות, או השינוי לאורך זמן, של הזרוע כאשר תצורת הזרוע נמצאת בתת-המרחב אותו המחלקה מייצגת. לבסוף, אנו מתבוננים במעברים האפשריים בין המחלקות, ומשרטטים גרף בו הצמתים הם המחלקות, והקשתות מייצגות את המעברים האפשריים בין המחלקות השונות. הגרף המתקבל הוא למעשה מכונת מצבים דטרמיניסטית, בו המעבר ממצב למצב נובע מהשינוי במצב המערכת.

בחיבור זה אנו נשתמש בשיטה המתוארת כדי לפתור שתי בעיות רדיפה, כאשר על משוואות התנועה של הסוכנים הרודפים מתקיים אילוץ חד-אופן. אילוצי חד-אופן נהוגים ביישומים רובוטיים רבים, מכיוון שהם מתארים בצורה פשוטה יחסית התנהגות של רכב גלגלי או כל כלי שמסוגל לנוע רק בכיוון אותו נכנה "קדימה", ושיכול לשנות את כיוונו רק באמצעות פנייה לסיבוב ברדיוס שיכול להשתנות אך איננו אפס.

המחקר נעשה בהנחייתו של פרופסור אלפרד ברוקשטיין, בפקולטה למדעי המחשב.

חלק מן התוצאות בחיבור זה פורסמו כמאמרים מאת המחבר ושותפיו למחקר בכנסים ובכתבי-עת במהלך תקופת מחקר הדוקטורט של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

David Dovrat and Alfred M. Bruckstein. On gathering and control of unicycle a(ge)nts with crude sensing capabilities. *IEEE Intelligent Systems*, 32(6):40–46, 2017.

David Dovrat and Alfred M. Bruckstein. Antalate—a multi-agent autonomy framework. *Frontiers in Robotics and AI*, 8:264, 2021.

David Dovrat, Twinkle Tripathy, and Alfred M. Bruckstein. On tracking and capture in proportional-control bearing-only unicycle pursuit. *IEEE Control Systems Letters*, 6:2132–2137, 2022.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

שיטה בתורת האוטומטים לפתרון בעיות מרדף

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

דוד דברת

הוגש לסנט הטכניון — מכון טכנולוגי לישראל
שבט התשפ"ב חיפה ינואר 2022

שיטה בתורת האוטומטים לפתרון בעיות מרדף

דוד דברת