

ADRIANO MARTINS DA COSTA REZENDE

**COORDINATION AND CONTROL OF
FIXED-WING AERIAL ROBOTS**

Dissertation presented to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

ADVISOR: LUCIANO CUNHA DE ARAÚJO PIMENTA
CO-ADVISOR: VINICIUS MARIANO GONÇALVES

Belo Horizonte

February 2019

© 2019, Adriano Martins da Costa Rezende.
Todos os direitos reservados.

Martins da Costa Rezende, Adriano

D1234p Coordination and Control of Fixed-Wing Aerial
Robots / Adriano Martins da Costa Rezende. — Belo
Horizonte, 2019
xxi, 97 f. : il. ; 29cm

Dissertação (mestrado) — Federal University of
Minas Gerais

Orientador: Luciano Cunha de Araújo Pimenta

1. Sistemas multi-robôs. 2. Programação linear
inteira mista. 3. Evitamento de colisões. 4. Campos
vetoriais. 5. Controle não linear.

CDU 519.6*82.10

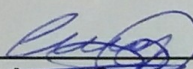
"Coordination And Control Of Fixed-wing Aerial Robots"

Adriano Martins da Costa Rezende

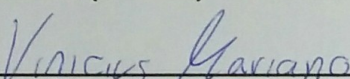
Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 18 de fevereiro de 2019.

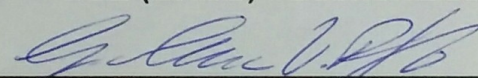
Por:



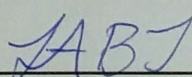
Prof. Dr. Luciano Cunha de Araújo Pimenta
DELT (UFMG) - Orientador



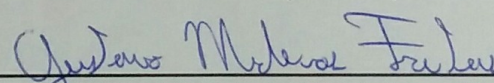
Prof. Dr. Vinicius Mariano Gonçalves
DEE (UFMG) - Coorientador



Prof. Dr. Guilherme Vianna Raffo
DELT (UFMG)



Prof. Dr. Leonardo Antônio Borges Tôres
DELT (UFMG)



Prof. Dr. Gustavo Medeiros Freitas
DEE (UFMG)

Acknowledgments

First, I should thank my entire family for all the support during the master course and for the incentives to the studies since I was a child.

I would like to acknowledge professors Luciano Pimenta and Vinicius Gonçalves for the orientation during the last two years. I also acknowledge the other professors of the program for all the knowledge they have transmitted to me. Besides, I should thank the professors Guilherme Raffo, Leonardo Tôrres and Gustavo Freitas, the other members of the valuation committee, for the valuable suggestions to improve this dissertation.

I acknowledge José Lucas Gomes Olavo for making available the simulator with the aircraft model used to validate the results in this work.

For undergrad students, master and PhD students from the CORO laboratory I should say thanks for the company and for making the laboratory a nice environment.

Finally, I am so thankful for the financial support provided by CNPq and the Electrical Engineering Graduate Program (PPGEE) of the UFMG.

“Teamwork is the ability to work together toward a common vision. The ability to direct individual accomplishments toward organizational objectives. It is the fuel that allows common people to attain uncommon results.”

(Andrew Carnegie)

Resumo

Sistemas multi-agente vem recebendo muita atenção nos últimos anos, uma vez que eles são capazes de realizar tarefas ou melhor, ou mais rápido ou até mesmo que são infactíveis para um único agente. Este trabalho apresenta uma formulação baseada em MILP (programação linear inteira mista) para a coordenação de múltiplos robôs que devem percorrer, persistentemente, curvas fechadas com pontos de interseção. A estratégia consiste em um planejamento *off-line* para as velocidades dos robôs de forma que estes não precisem, em tempo real, realizar manobras para evitar colisões entre eles. O modelo do robô considera restrições de velocidade mínima e máxima, fato que possibilita a aplicação da estratégia a robôs aéreos de asa fixa. Questões que são usualmente negligenciadas na literatura de sistemas multi-robô são consideradas neste trabalho, a saber: separação espacial mínima; limites de aceleração; e incertezas nas velocidades e posições.

Para controlar cada robô no seu caminho específico, uma técnica de campo vetorial guia foi desenvolvida. Nesta parte da dissertação uma representação mais completa da dinâmica de um avião é considerada. O modelo de referência utilizado é não holonômico e admite entradas de controle limitadas. Uma estratégia de campo vetorial já consolidada é utilizada para efetuar o cálculo das entradas de controle. Convergência assintótica é provada através da Teoria de Lyapunov e conjuntos invariantes são encontrados quando distúrbios são levados em consideração. A estratégia de controle desenvolvida também apresenta vantagens sobre trabalhos relacionados. Sua principal característica é a habilidade de lidar com curvas de formato genérico em 3 dimensões.

Simulações com até 48 robôs demonstram a eficiência computacional da estratégia de coordenação baseada em MILP. A eficiência do controlador guiado por campo vetorial proposto é demonstrada em simulações com um modelo de aeronave mais completo, com 6 graus de liberdade e 12 estados. Um experimento real com 3 robôs e-puck é apresentado para validar a estratégia de coordenação e controle de múltiplos robôs em um cenário do mundo real.

Abstract

Multi-agent systems have receiving much attention in the last years, since they are able to perform tasks better, faster of even that are unfeasible for single agents. This work presents a MILP (mixed integer linear programming) based formulation for the coordination of multiple robots that must follow closed intersecting paths persistently. The strategy consists on an off-line planning for the robots' velocities, which prevents the need of online inter-robot collision avoidance maneuvers. The robot model considers minimum and maximum speed constraints, which allows the strategy to be applied to fixed-wing aerial robots. Issues that are often disregarded in the literature of multi-robot systems are addressed in the present work, they are: minimum spatial separation; acceleration limits; and uncertainties on the speeds and positions.

In order to control each robot on its specified path, a guidance vector field approach technique was developed. In this part of the dissertation a more complete representation of the dynamics of an airplane is considered. The used reference model is non holonomic and has constrained input controls. A consolidated vector field strategy is used to guide the computation of the control inputs. Asymptotic stability is proven with Lyapunov Theory and ultimate bounds are found when actuation disturbances are taken into account. The developed control strategy also has advantages over the strategies in related works. Its main characteristic is the ability to deal with generic curve shapes in 3 dimensions.

Simulations with up to 48 robots show the computational efficiency of the MILP based coordination strategy. The efficiency of the proposed guidance vector field controller is demonstrated by simulations with a more complete aircraft model, with 6 degrees of freedom and 12 states. A real experiment with 3 actual e-puck robots is presented in order to validate the multi-robot coordination and control strategy in a real world scenario.

List of Figures

1.1	Multiple robotic manipulators, manufactured by Comau, working on an automobile on the production line at Fiat Chrysler Automobiles. <i>Source: https://www.bloomberg.com.</i>	10
1.2	Multiple robots that transport stacks of products at Amazon warehouse. <i>Source: www.infobae.com</i>	10
1.3	On the left is a quadrotor DJI Matrice 100 (<i>Source: https://store.dji.com</i>). On the right is a fixed-wing UAV developed at UFMG.	12
1.4	Multi-stage structure of the problem solution.	14
2.1	Example of the ORCA algorithm with incorporation of kinematic car-like constraints proposed in [Alonso-Mora et al., 2012].	16
2.2	Paths of multiple robots planned in [Avellar et al., 2015] to cover a polygonal area.	17
2.3	Example of multiple paths to be periodically followed by robots presented in [Gonçalves et al., 2013].	20
2.4	Block diagram representing how the target dynamics is imposed to the UAV. <i>Source: [Olavo et al., 2018]</i>	23
2.5	Low level controllers considered in [Thums, 2012], [Jesus et al., 2013], [Olavo et al., 2018].	24
2.6	Examples of vector fields in \mathbb{R}^2 . In red are the target curves and in blue a normalized vector field.	25
2.7	Representation of the strategy in [Gonçalves et al., 2010]. Two level sets in \mathbb{R}^3 and the intersection between them.	27
2.8	Experiments presented in [Kapitanyuk et al., 2017] with ground robots. . .	28
2.9	Results obtained according to the strategy in [Olavo et al., 2018].	29
3.1	Example of a possible configuration of our problem. Two closed paths and dangerous stretches (dashed black). Dotted circles show the robots' size. . .	32

3.2	Illustration of the target points (green) placement and regions of uncertainty (between black curved traces) in the example of Fig. 3.1.	33
3.3	Illustration of the definitions made in Section 3.3.	40
3.4	Illustration of the safety constraints.	41
3.5	Illustration of the average speed computation.	44
4.1	Definitions for Theorem 4. In blue is the straight line with orientation θ . In red are vectors with orientations β_1 and β_2 . In green are vectors with orientations $\phi(\beta_1)$ and $\phi(\beta_2)$	68
5.1	Scenario used for a simulation of the coordination strategy. Collision stretches are the ones in thick black. Inflated collision zones, represented by the gray polygons, are composed of the segments therein. Target points are in black and the regions of uncertainty are between the black curved traces. Robots are represented by arrows indicating their motion direction and circles indicating their sizes.	72
5.2	Scenario used for the timing analysis.	73
5.3	Execution time of the algorithms. Scenario of Fig. 5.2	74
5.4	Number of variables and constraints. Scenario of Fig. 5.2	74
5.5	Box plot of the total time of the coordination strategy applied to the scenario of Fig. 5.2.	75
5.6	Numerical simulation. The UAV's trajectory, in red, converges to a generic path and remains inside the bounds, represented by the blue curves.	76
5.7	In the top is the Lyapunov function V_θ . The dashed blue/yellow trace is associated with the time the UAV was inside \mathcal{B}_Ω . In the bottom is the field potential function P . The dashed lines are the corresponding ultimate bounds.	77
5.8	Performance of the UAV in the simulation of Figure 5.6. From top to bottom: height z ; ground speed $v(t)$; and bank angle.	77
5.9	Simulation considering a "saddle like" path in \mathbb{R}^3 . In red is the trajectory followed by the UAV, which converges to the blue tube.	78
5.10	Functions V_θ and P associated to the 3 dimensional simulation.	78
5.11	Performance of the UAV in the simulation of Figure 5.9.	79
5.12	Results obtained according to the control strategy presented in this work. The initial conditions are the same as in Figure 2.9.	79
5.13	Scenario of the simulation with UAVs' realistic model. The inflated collision zones, target points and regions of uncertainty are placed as in Fig. 5.1.	80

5.14	Block diagram in Simulink representing the controllers developed in this work.	82
5.15	Histogram of the normalized position errors at the instants $t[q, k]$ for robots 1, 2 and 3, in the Simulink simulation.	82
5.16	In the top are the reference velocity and performed velocity of the blue UAV in the Simulink simulation. In the bottom is the error between the velocities.	83
5.17	Scenario of the actual robot experiment post printed on a frame of the camera system.	84
5.18	Histogram of the normalized position errors at the instants $t[q, k]$ for robots 1, 2 and 3, in the e-puck experiment.	85

List of Tables

5.1	Results for robot 1 in the simulation of Figure 5.1	73
5.2	Results for robot 1 in the experiment of Figure 5.13	81
5.3	Results for robot 1 in the experiment of Figure 5.17	84

Contents

Acknowledgments	v
Resumo	ix
Abstract	xi
List of Figures	xiii
List of Tables	xvii
1 Introduction	9
1.1 Multi-agent systems	9
1.2 Aerial robots	11
1.3 Contributions	13
1.3.1 Publications	14
1.4 Dissertation structure	14
2 Related works	15
2.1 Multi-robot coordination	15
2.1.1 Collision free motion	15
2.1.2 MILP formulations for multi-robot systems	17
2.1.3 Other approaches to the multi-robot coordination	17
2.1.4 Multi-robot systems with cyclic tasks	19
2.1.5 Coordination strategy in this work	20
2.2 Robot control strategies	21
2.2.1 Fixed-wing UAV reference model	22
2.2.2 Vector fields	25
2.2.3 Guidance vector field strategies	27
2.2.4 Control strategy in this work	30

3	Coordination strategy	31
3.1	Multi-robot problem statement	31
3.1.1	Speed model	33
3.1.2	Uncertainty model	34
3.2	Definition of collision zones	35
3.2.1	Practical implementation	37
3.3	MILP formulation	38
3.3.1	Safety and periodicity	40
3.3.2	Average speed reference	43
3.3.3	Uncertainty	44
3.3.4	Acceleration limits	46
3.3.5	Speed limits	47
3.3.6	Final MILP problem	48
3.4	Discussion on the coordination solution	50
3.4.1	Alternative objective functions	50
3.4.2	Absence of solution	50
3.4.3	Alternative approaches	51
3.4.4	Speed profile along the paths	52
4	Control strategy	55
4.1	Control problem statement	55
4.2	Control design	56
4.2.1	Heading control	57
4.2.2	Velocity control	59
4.2.3	Altitude control	60
4.2.4	Disturbance analysis	61
4.3	Field's Singularities	67
5	Results	71
5.1	Coordination strategy	71
5.1.1	Simulated experiments	71
5.1.2	Timing analysis	73
5.2	Vector field control validation	75
5.2.1	Simulated experiments	75
5.3	Coordination and control	80
5.3.1	Multiple airplanes simulation	80
5.3.2	Multiple real robots experiment	82

5.4	Summary of results	84
6	Conclusion	87
6.1	Final considerations	87
6.2	Future works	88
	Bibliography	89

List of abbreviations and acronyms

UFMG	Universidade Federal de Minas Gerais
PPGEE	Programa de Pós-Graduação em Engenharia Elétrica
UAV	Unmanned Aerial Vehicle
MILP	Mixed Integer Linear Programming
GPS	Global Positioning System
IMU	Inertial Measurement Unit
PID	Proportional Integral Derivative
ORCA	Optimal Reciprocal Collision Avoidance
CHOPs	Circular Holding Patterns
MPC	Model Predictive Control
NMPC	Non-linear Model Predictive Control
MAV	Micro Air Vehicle
DMPC	Decentralized Model Predictive Control
DAG	Direct Acyclic Graph
RRT	Rapidly-Exploring Random Tree
PRM	Probabilistic Roadmap
BTT	Bottleneck Tree
LOS	Line-of-sight
NLGL	Nonlinear Guidance Law

GVF	Guidance Vector Field
DOF	Degrees-of-freedom
PI	Proportional Integral
gdc	greatest common divisor
LP	Linear Problem
ROS	Robot Operating System

List of Symbols

\mathbb{R}	Set of real numbers
\mathbb{I}	Set of integer numbers
\mathbb{N}	Set of natural numbers
\mathcal{R}	Set of robots
\mathcal{P}_i	Set of points in \mathbb{R}^3 that belong to the path of the robot $i \in \mathcal{R}$
q	Index of target point
Δs	Enlargement of the collision zones
t	Time
t^*	Time passed since the instant the speed command of a robot is changed
$v(t)$	Forward velocity of a robot
τ_i	Time spent by robot i change its velocity between to setpoint speeds
a	Acceleration used by robot i to change its speed
$u(v)$	Uncertainty in the forward velocity of a robot along its path
u_v	Velocity uncertainty that is independent of the current velocity
$u_{\%}$	Velocity uncertainty that is proportional to the current velocity
U_v^i	Maximum value of u_v associated to robot i
$U_{\%}^i$	Maximum value of $u_{\%}$ associated to robot i
U_p^i	Maximum value of the position uncertainty associated to robot i
\mathcal{Q}	Set of points in \mathbb{R}^3 that belong to at least one path $\mathcal{P}_i, i \in \mathcal{R}$

\mathcal{C}	Set of collision points
ρ_{ij}	Minimum distance between robots $i, j \in \mathcal{R}$ to avoid collision between them
\mathbb{E}_1	Relation that defines neighbor points
\mathbb{E}_2	Relation that defines closed points in distinguishable paths
\mathbb{E}^*	Equivalence relation that defines points in the same collision zone
\mathcal{Z}	Set of collision zones
\mathcal{S}	Set of collision stretches
\mathcal{E}	Set of entrance target points
\mathcal{O}	Set of output target points
\mathcal{H}	Set of all target points
$\mathcal{W}(q)$	Set of target points that are entrances of the region in which q is entrance
$n(q)$	Function next target point
$p(q)$	Function previous target point
$f(i)$	Function first target point
$b(q)$	Function that maps each target point q to the associated robot i
$L(q)$	Length of the stretch from q to $s(q)$
$t[q, k]$	Instant robot i must reach target point q in the k -th cycle
$r[q]$	Half of the length of the region of uncertainty associated to q
$\mathcal{L}[q]$	Length between target point q and $s(q)$
C_0	Fundamental cycle time
$C[i]$	Cycle time of robot $i \in \mathcal{R}$
$T[q, k]$	Time robot i must spend from q to $s(q)$
$\lambda(i)$	Integer multiplier for the cycle time $C[i]$
Δk	Auxiliary variable to prove safety

ξ	Very large positive number
$B[q_1, q_2]$	Binary variable associated to the pair $(q_1, q_2) \in \mathcal{E}^2$
$e[q, k]$	Error of the position of a robot at instant $t[q, k]$
$\hat{e}[q, k]$	Measurement of the error $e[q, k]$
$\bar{V}[q, k]$	Average velocity to be executed between points q and $n(q)$ in the k -th cycle
D_m	Minimum distance that a robot travels when there is uncertainty
D_{id}	Ideal distance that a robot should travel
$V_i[q, k]$	Instantaneous forward velocity of robot i at point q in the k -th cycle
$V_f[q, k]$	Desired forward velocity of robot i at point $n(q)$ in the k -th cycle
v_{min}^i	Minimum forward velocity of robot i
v_{max}^i	Maximum forward velocity of robot i
\mathbf{a}_{min}^i	Minimum linear acceleration of robot i
\mathbf{a}_{max}^i	Maximum linear acceleration of robot i
Δv^i	Difference between the maximum and minimum forward velocities of robot i
$w(q_1, q_2)$	Integer number to relax the safety constraints
\mathbf{d}	Greatest common divisor between $\lambda(i_1)$ and $\lambda(i_2)$
a_q	Auxiliary constant
b_q	Auxiliary constant
\mathbf{A}	Matrix with constants a_q
\mathbf{B}	Vector of constants b_q
\mathbf{V}_i	Vector of ideal velocities at the target points
x	Cartesian coordinate x of the UAV
y	Cartesian coordinate y of the UAV
z	Cartesian coordinate z of the UAV

θ	Heading angle of the UAV
v	Forward velocity of the UAV
z_c	Command for the height z
θ_c	Command for the heading angle θ
v_c	Command for the forward velocity v
τ_z	Time constant of the height dynamics
τ_θ	Time constant of the heading dynamics
τ_v	Time constant of the forward velocity dynamics
u_z	Uncertainty in the height dynamics
u_ω	Uncertainty in the heading dynamics
u_a	Uncertainty in the forward velocity dynamics
U_z	Maximum absolute value of the uncertainty u_z
U_ω	Maximum absolute value of the uncertainty u_ω
U_a	Maximum absolute value of the uncertainty u_a
v_{max}^z	Maximum absolute value of the vertical velocity of the UAV
ω_{max}	Maximum absolute value of the turning rate of the UAV
v_{min}	Minimum forward velocity of the UAV
v_{max}	Minimum forward velocity of the UAV
\mathbf{p}	Vector in \mathbb{R}^3 representing the UAV position
Φ	Vector field in \mathbb{R}^3
α_1	First basic function to generate the vector field
α_2	Second basic function to generate the vector field
G	Gain function for the convergence to the curve
H	Gain function for the curve's circulation

P	Lyapunov function of the vector field
∇	Nabla operator
$\hat{\Phi}$	Normalized and projected field \mathbb{R}^3
$\hat{\Phi}_x$	Component of the normalized and projected field in the x direction
$\hat{\Phi}_y$	Component of the normalized and projected field in the y direction
$\hat{\Phi}_z$	Vertical component associated to the normalized and projected field
$\theta_f(\mathbf{p})$	Heading angle of the vector field at point \mathbf{p}
$\bar{\theta}(\mathbf{p}, \theta)$	Orientation error of the UAV with respect to θ_f
$\omega(t)$	Desired reference turning rate for the UAV
V_θ	Lyapunov function that measures the orientation error
k_p	Gain of the heading angle controller
M	Maximum value of gradients of the field's components
$\ \dot{\mathbf{p}}\ _{max}$	Maximum value of the speed the UAV may reach
v_r	Reference forward velocity to be tracked by the UAV
k_v	Gain of the forward velocity controller
$a(t)$	Desired forward linear acceleration for the UAV
V_v	Lyapunov function that measures the forward velocity error
\tilde{v}_z	Desired vertical velocity for the UAV
v_z	Saturated desired vertical velocity for the UAV
γ	Ultimate bound for the error on $\bar{\theta}$
\mathcal{I}_θ	Invariant set of $\bar{\theta}$
\bar{v}	Error of the forward velocity
μ	Ultimate bound for the error on \bar{v}
\mathcal{I}_v	Invariant set of \bar{v}

\bar{v}_z	Error of the vertical velocity
ζ	Ultimate bound for the error on \bar{v}_z
\mathcal{I}_z	Invariant set of \bar{v}_z
w_1, w_2, w_3	Unit vectors in \mathbb{R}^3
Ψ	Reference vector for the UAV's velocity $\dot{\mathbf{p}}$
\mathcal{I}	Invariant set for the position of the UAV
χ	Auxiliary variable to compute \mathcal{I}
\mathbf{p}_s	Position of a singularity point on the vector field
\mathcal{B}_Ω	Ball of radius Ω around the singularity
Ω	Radius of the circle around the singularity
β	Angular coordinate for a point in the border of \mathcal{B}_Ω
$\Delta\beta$	Variation on β while the UAV traverses the circle \mathcal{B}_Ω
$\phi(\beta)$	Orientation of the projected field at the border of \mathcal{B}_Ω indicated by angle β
$\delta(\beta)$	Orientation of the projected field with respect to the normal vector of \mathcal{B}_Ω at β
ΔV_θ	Variation of V_θ while the UAV traverses the circle \mathcal{B}_Ω

Chapter 1

Introduction

1.1 Multi-agent systems

Cooperative robotics is a field designated to the study of strategies to make a group of robots perform a desired task together. These multi-agent systems have the purpose of executing tasks that are either difficult or impossible to be performed by a single agent. Multi-agent systems may offer important advantages such as shorter execution time and robustness to agent failure.

Multiple robots working together are widely present in several types of assembly lines. An example is in the automobile industry. Figure 1.1 shows a group of robots working in the same car. A strategy to coordinate them is necessary to avoid collisions between the robots and also minimize the working time, as proposed in [Spensieri et al., 2016]. Such strategies can be extremely useful in the industry, since they have direct economic implications.

When using multiple robots moving in the same environment, problems related to collisions and communication need to be carefully addressed. In this work, we focus on the the problem of inter-robot collisions. The communication issue will not be a problem as we propose an offline solution that does not require any communication between the robots.

Another example of system that considers multiple mobile robots moving in the same area, as proposed in this work, is the multi-robot system deployed in the Amazon warehouse. Figure 1.2 shows the robots used to transport the products' stacks. These mobile robots need to move around the warehouse while avoiding collisions between each other. Robots are far away from replacing human workers entirely, however, they are a practicable option to transport products, goods and also commodities.

Besides manufacturing and transporting tasks, multiple robots are also useful in



Figure 1.1. Multiple robotic manipulators, manufactured by Comau, working on an automobile on the production line at Fiat Chrysler Automobiles. *Source:* <https://www.bloomberg.com>.



Figure 1.2. Multiple robots that transport stacks of products at Amazon warehouse. *Source:* www.infobae.com

area coverage missions. As pointed out in [Palacios-Gasós et al., 2016], the problem of covering an environment with a team of robots, commonly called *multi-robot coverage problem*, can fall in three different categories: static, dynamic or persistent. In static coverage, fixed positions for a set of agents are defined in order to statically monitor an environment. Dynamic coverage is related to exploration missions, where an area must be explored until some level of knowledge is reached. Different from the previous scenarios, in persistent coverage a given area or specific points inside the area must be continuously kept under monitoring, which means that robots must keep periodically revisiting determined places in the environment.

Such persistent coverage tasks in which the robots must keep visiting periodically specific points of the environment may be modeled as the problem of controlling the robots to traverse periodically predefined closed paths [Keller et al., 2017]. In general, such paths might have intersections. In this work a planning strategy to safely co-

ordinate a group of robots in such scenario is presented. The objective is to cover closed paths that have intersection points with each other. This problem was earlier addressed in [Gonçalves et al., 2013], in which the objective of planning a collision-free velocity profile for point robots is achieved by solving a MILP (mixed integer linear programming) problem considering minimum and maximum speed limits as physical constraints. In the present work, those results are extended by considering some practical issues which are usually disregarded in multi-robot coverage literature. Real size robots, acceleration limits and uncertainties in the individual robot controllers are taken into account. Minimum and maximum speed limits are also taken into account, which allows for the application of our strategy to fixed-wing aerial vehicles. The objective is to maximize safety. The minimum spatial distance between the robots is maximized by extending the sizes of the so called *collision zones*. This minimum distance is maximized in order to improve safety and coverage quality. Safety is improved because the robots will be more distant apart. Coverage efficiency will be implicitly improved because if the robots are not very close to each other they are not covering the same area at the same time. In addition, in the case of aerial robots this separation might help to minimize aerodynamic interference. In [Tang et al., 2017] the authors use a multi-robot planner to solve a point-to-point problem even when the robots are in different heights. They experimentally show the better performance when the planner is used and justify it with the decrease in aerodynamic interference.

In order to illustrate the efficiency of the approach, several simulations are presented. Scenarios with up to 48 robots are solved within less than 8 seconds with MATLAB. An experiment with 3 actual e-puck robots is also presented. It shows the robustness of the solution and the stability of the online computation of reference speed that keeps the robots synchronized.

Besides coverage tasks, the strategy presented in this work also envisions other important applications for the proposed framework. In fact, the solution might be used in any context where there are multiple robots executing cyclic motion in a shared workspace with the risk of collisions. Direct examples are: planning the motion of multiple robots transporting products in a warehouse periodically; and the planning of the schedule of multiple trains connecting multiple cities.

1.2 Aerial robots

Nowadays UAVs (*Unmanned Aerial Vehicles*) are used in many military and civil applications. In particular, fixed-wing vehicles are used in tasks such as surveillance,

monitoring, convoy protection, and others. As an example, fixed-wing UAVs are used for the communication and also for collecting atmospheric data [Frew et al., 2013]. Also, these small airplanes are used to collect image data of given areas for the posterior creation of 3 dimensional maps. These types of data may also be acquired by a group of UAVs, as proposed by [Avellar et al., 2015]. The advances in electronics, and consequently, the low costs associated with sensors such as GPS (Global Positioning System), IMU (Inertial Measurement Unit), barometers and compass, have collaborated to the increase in the research interest in such aerial vehicles. Fixed-wing UAVs have an important advantage regarding the autonomy in terms of energy in comparison to rotary-wing vehicles. In this context, guidance techniques for such aircraft are receiving a lot of attention in the academy and industry. Figure 1.3 shows a rotary-wing vehicle on the left and a fixed wing on the right.



Figure 1.3. On the left is a quadrotor DJI Matrice 100 (*Source: <https://store.dji.com>*). On the right is a fixed-wing UAV developed at UFMG.

The quadrotor robot has an important advantage regarding its mobility. Informally, at low speeds the quadrotor can be controlled to move along any direction regardless its pose. Besides, quadrotors can hover. Their limitations are only related to the dynamic behavior of the system. Given the constant motion of the wings of the quadrotor and its aerodynamics, they have a lower autonomy, usually between 10 to 25 minutes of flight. The advantageous mobility feature of quadrotors is, unfortunately, not present in the fixed-wing vehicles. Informally, the UAV can not instantaneously move itself sideways. This effect is the same as the one in differentially driven ground robots. In this context, in order to explore the advantages of the fixed-wing UAV regarding energetic efficiency, a more sophisticated high level guidance technique is needed.

In the present work multiple fixed-wing UAVs will be used to cover closed paths. Thus, a control strategy that makes these airplanes follow the given paths will also be presented, as a stage of the whole methodology. The control strategy will be based on artificial vector fields and on an aircraft reference model that incorporates the non-holonomic behavior of the UAV. These fields of study will be reviewed in the next

Chapter.

1.3 Contributions

The contributions of this work can be summarized in the following two topics: (i) development of a coordination strategy for multiple fixed-wing UAVs traverse prespecified closed curves while avoiding collisions between them; and (ii) development of a control strategy able to guide a fixed-wing UAV along a prespecified path with a given velocity profile.

The coordination strategy is a methodology able to coordinate multiple robots, each one along a prespecified path, while avoiding collisions between them. It was done by planning a speed profile for each robot along its associated path. The planning takes into account practical issues associated to velocity and acceleration limits, robots with real size and uncertainties on the velocities and positions of the agents.

The developed control strategy is in a lower level in comparison to the coordination strategy. It is able to make a robot converge to and circulate the desired curve. A reference model of the airplane that considers its main features, such as speed limits and the non-holonomic constraint, was considered to design the controller. Since the reference model does not represent the fixed-wing UAV exactly, uncertainties in the model were taken into account in order to ensure robustness.

The two aforementioned objectives correspond to two of the stages of a multi-stage solution to the area coverage problem with fixed wing UAVs. The first stage is the higher level planner that defines the paths to be followed. This stage is not tackled in this work. The second stage is the motion planning of the UAVs along the predefined paths, which is the coordination problem addressed here. The third stage is the high level controller that considers a simplified, but representative, model of the airplane, with the objective of guiding a UAV towards its path. The fourth stage consists of low level PID controllers that are responsible to impose the reference model assumed in the previous stage to the airplane. The design of these PID controllers is also out of the scope of the present work.

Figure 1.4 depicts the multi-stage solution structure. The “Fixed-wing UAV” block represents the airplane, which can be the physical aircraft or a complex model for simulation. The “State estimator” block represents the sensors and filters used to estimate the states of the aircraft. Among the four aforementioned stages, represented in the figure by the rectangular shapes, the objective is then the development of the stages represented by the red blocks.

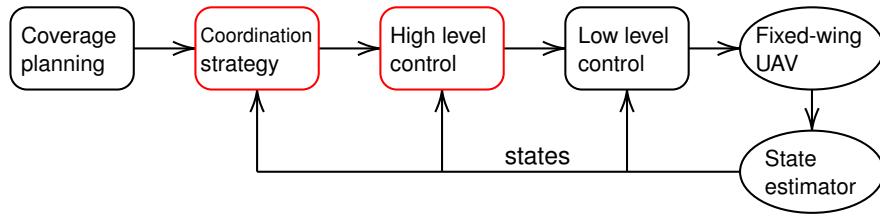


Figure 1.4. Multi-stage structure of the problem solution.

1.3.1 Publications

This work originated two publications: [Rezende et al., 2018a] and [Rezende et al., 2018b]. The first one is entitled *Controle de VANT de asa fixa com campos vetoriais arbitrários*, and was presented in the *Congresso Brasileiro de Automática (CBA)*, in João Pessoa, Paraíba, Brazil. The second one is entitled *Robust Fixed-Wing UAV Guidance with Circulating Artificial Vector Fields* and was published on the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), in Madrid, Spain. Both works are associated to the guidance vector field based control strategy for fixed-wing UAVs presented in this dissertation.

1.4 Dissertation structure

This dissertation is organized as follows: in the next Chapter a literature review is made and the related works are compared to the present work. In Chapter 3 the coordination strategy for multiple agents is developed. In Chapter 4 the focus goes to a single fixed-wing UAV and the control strategy is presented. In Chapter 5 several simulations and experiments are presented to validate the developed strategies. Finally, in Chapter 6 the work is concluded and possible future extensions are proposed.

Chapter 2

Related works

2.1 Multi-robot coordination

This section is dedicated to a revision on works in the literature that deal with the problem of multi-robot systems.

2.1.1 Collision free motion

When dealing with multiple agents, a recurrent problem is the inter robot collision avoidance. In the literature, this issue is commonly addressed with stopping policies [Soltero et al., 2011], [Wallar et al., 2015], or sensing-repelling strategies [Lin and Saripalli, 2015],[Zhang et al., 2010]. The first one is not applicable when the robots are fixed-wing UAVs, since a minimum velocity is needed for aerodynamic lift. The second one is not usually applicable when the robots must be kept on their planned paths, to avoid other fixed obstacles for example. Besides, in general there is no guarantee of absence of collisions and deadlocks, which may occur when reactive methods are used.

In [Borkar et al., 2016] the motion of multiple robots is planned over a single previously defined Lissajous curve. An algorithm is used to define a curve and the number of agents to cover a rectangular area. The motion is then planned by maximizing the size of the agents such that collision free trajectories are still feasible.

A widely used algorithm for collision avoidance is ORCA (Optimal Reciprocal Collision Avoidance), [Van Den Berg et al., 2011]. The algorithm is based on the definition of velocity obstacles. Each robot detects the other agents on its surroundings and considers them as moving obstacles in order to plan its own motion. No communication between the agents is necessary and the strategy is completely decentralized. Several

extensions of the ORCA algorithm are available, for instance, in [Ruffi et al., 2013] smooth piecewise trajectories are generated and in [Alonso-Mora et al., 2012] kinematic constraints of car-like vehicles are incorporated. Figure 2.1 exemplifies the strategy proposed in [Alonso-Mora et al., 2012]. In the left, five cars move between antipodal points of a circle. In the center, the result of a similar simulation is presented for MagneBikes vehicles. On the right a simulation with five cars is presented again, in which one of the cars moves regardless of the others and is naturally perceived as a moving obstacle by its teammates.

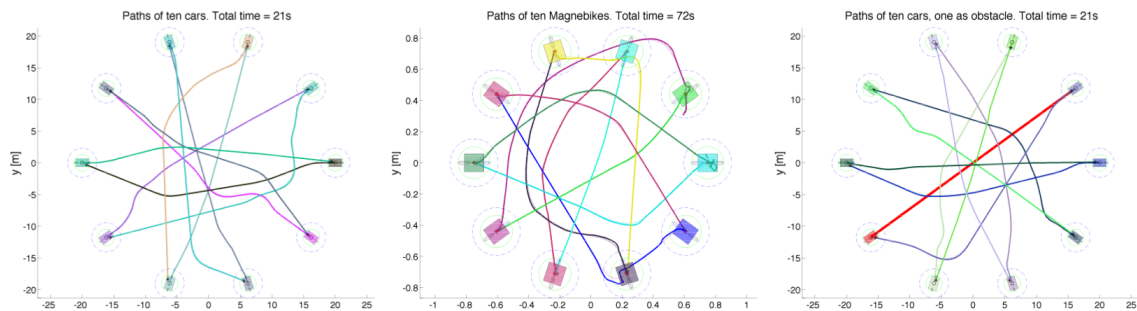


Figure 2.1. Example of the ORCA algorithm with incorporation of kinematic car-like constraints proposed in [Alonso-Mora et al., 2012].

An algorithm that has a similar purpose of ORCA is the CHOPs (Circular Holding Patterns), which was used in [Tang et al., 2017] to control a team of quadrotors that must move, each one, to a prespecified point. Given initial straight line trajectories, the CHOPs algorithm changes the straight lines near the points that they intersect each other and make the UAVs circulate around this point to avoid collision. Afterwards, continuous piecewise polynomial trajectories are generated and given as reference to the robots. The fundamental difference of CHOPs, with respect to ORCA, is that the former is a centralized planner.

Another technique used to deal with the collision avoidance problem is MPC (Model Predictive Control). In [Kamel et al., 2017] the authors use a Non-linear MPC (NMPC) to control multiple MAVs (Micro Air Vehicle). Each agent must track a trajectory while avoiding collision with other members of the team. The strategy considers the dynamics of the vehicles, uncertainties in the state estimators and is decentralized. In [Schwartz et al., 2017] NMPC is used to generate safe trajectories to an autonomous car in environments with other cars.

Several times, MILP (Mixed integer Linear Programming) formulations are also used to deal with the collision avoidance problem and other common issues of multi-robot systems, as in the present work. Next section is dedicated to such approaches.

2.1.2 MILP formulations for multi-robot systems

Several works have used MILP formulations in the coordination of multi-robot systems. In [Avellar et al., 2015] the MILP solution provides the number of UAVs and the paths they need to follow in order to cover an area in minimum time. Figure 2.2 shows an example presented in [Avellar et al., 2015]. The paths for each UAV (green, blue and red paths) are computed in a way that the polygonal area is covered with minimum time. In [Richards and How, 2002] a theory that plans paths through a set

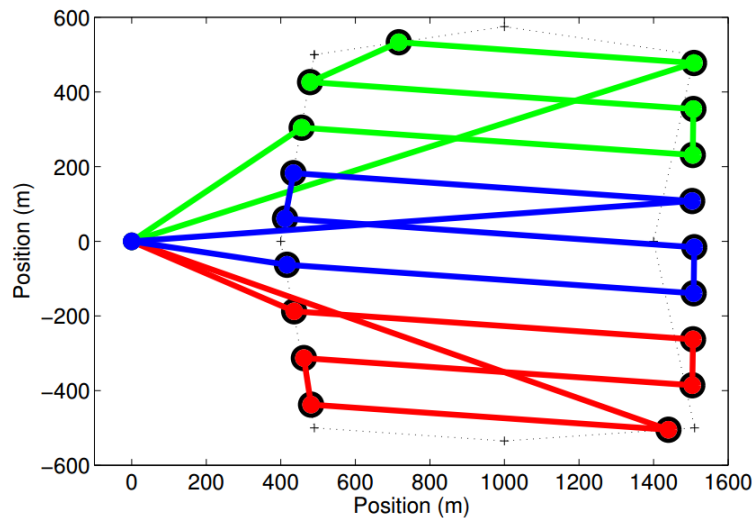


Figure 2.2. Paths of multiple robots planned in [Avellar et al., 2015] to cover a polygonal area.

of waypoints in a sequence that minimizes the elapsed time is proposed. Collision with other agents and with obstacles are avoided. In order to take into account dynamic limits the authors use a conservative strategy writing the constraints in a linear form. In [Richards and How, 2004] the authors use Decentralized Model Predictive Control (DMPC) together with a MILP formulation to generate collision free trajectories for multiple UAVs with linearized dynamics. The problem is solved online at each time step. In [Altché et al., 2016] an offline MILP formulation is used to plan the motion of multiple actual size robots in a cross section, which has its main application in the autonomous driving through intersections. In [Peng and Akella, 2003] and [Jufeng and Srinivas, 2005] MILP formulations are used to coordinate a set of robots from initial to final points along predefined paths without collisions.

2.1.3 Other approaches to the multi-robot coordination

In contrast to the MILP formulation, in [Abichandani et al., 2013] and [Abichandani et al., 2015] the motion planning of multiple robots between sets

of initial and final points is performed by a nonlinear programming formulation, and assurance of communication between robots is also incorporated. In fact, communication limit is an issue that have been also considered in the literature of multi-robot systems. This problem is critical when the strategy is computed online [Clark et al., 2003], [Otte and Correll, 2014]. In these works, the robots build local communication networks and execute, each one, a multi-robot planner, usually probabilistic, in order to find feasible trajectories to a goal position. The best solution is then spread through the network. In this context, the implementation of solutions provided by centralized and offline computed strategies has an important advantage: it does not require any communication between the robots. In addition, an offline strategy has also the advantages of providing a predictable behavior for the entire system and also does not require the performance of complex real time computations during the navigation.

2.1.3.1 Energy supply issues

Another issue that is also of concern in persistent multi-agent coverage tasks is the energy supply limitation [Mitchell et al., 2015], [Kamra and Ayanian, 2015]. In [Mathew et al., 2013], the motion of a set of charging robots is planned to refuel a set of task robots. The problem is tackled with a DAG (*direct acyclic graph*) formulation and solved with MILP. In [Scherer and Rinner, 2016], the environment is decomposed in cells and an algorithm is proposed to compute the path for each robot of a set. They move between base stations and points of interest, while communication with the basis and enough energy to return are ensured.

2.1.3.2 Computational complexity issues

Centralized strategies consider the planning of all robots in a single stage. In general, this type of approach leads to an exponential growth in the computation time when the number of robots increases, fact that is also refereed to as the *curse of dimensionality*. Some strategies to deal with this problem use sequential prioritized planners. In [den Berg and Overmars, 2005] the movement of higher priority robots are planned first and afterwards they are considered as moving obstacles in the planning of lower priority ones. These planners have the advantage of scalability with the number of robots, since the multi-robot problem is dealt by solving multiple single robot problems. In [Čáp et al., 2015] the problem is dealt with a revised method of prioritized planning and a novel asynchronous decentralized implementation. The algorithm is proven to be complete and deadlock free. In fact, incompleteness properties and deadlocks are

common problems of many sequential approaches. In [Van Den Berg et al., 2010] an algorithm is presented in order to decouple a multi-robot path planning problem. The algorithm divides the group of robots into subgroups, which are as small as possible, in a way that these subgroups are decoupled. Then, a multi-robot planner is used to plan the motion of each subgroup. Problems with many robots are solved in reasonable time since solving many small subproblems is faster than solving a single large problem. Another strategy to deal with the dimensionality problem is subdimensional expansion [Wagner et al., 2012]. It consists on the planning in the dimension of single robots when they are far away and in the extended dimension only when they are close to each other. With RRT and PRM based planners, scenarios with up to 32 robots were solved in a matter of some minutes.

The majority of works discussed until now considers the point-to-point problem, when multiple robots must follow, each one, a route (predefined or not) from an initial to a final point avoiding collisions. The point-to-point multi-robot motion problem is sometimes tackled with probabilistic planners as in [Solovey and Halperin, 2017]. The presented algorithm, bottleneck tree (BTT), is able to solve the problem of safe coordination of multiple robots moving between predefined sets of points following predefined paths. The point-to-point problem is also seen as a Pebble Motion Planning on a Graph [Alotaibi and Al-Rawi, 2016], [Surynek, 2009], [Yu and Rus, 2015], [Surynek, 2014]. These methods are capable of solving problems with a few thousands of robots/nodes in a matter of seconds. However, physical properties such as dynamics and uncertainties are not considered. Besides, the literature focus on the point-to-point problem, not covering the persistent movements in cycles.

2.1.4 Multi-robot systems with cyclic tasks

When optimization methods are used, the coordination of multiple robots moving persistently in cyclic paths is not as frequently addressed as the point-to-point problem. A difficulty that appears is the mathematical formulation of a model that satisfies the constraints for all instants of time. The work in [Gonçalves et al., 2013] introduces a strategy to solve this infinite horizon cyclic problem with a finite number of variables and constraints. As the authors show, this is possible due to the assumed periodicity and commensurability properties of the cycles. In fact, this is one of the main assumptions of their formulation. The problem is formulated as a MILP, whose constraints directly ensures the satisfaction of the problem's requirements. These are related to safety, periodicity, commensurability and speed limitation. Figure 2.3 shows an example considered in [Gonçalves et al., 2013], each of the paths must be followed by a

robot and the black dots represent intersection points between paths.

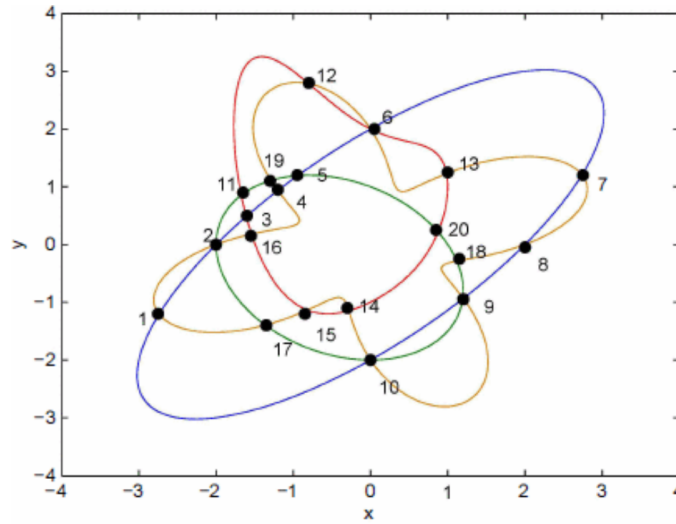


Figure 2.3. Example of multiple paths to be periodically followed by robots presented in [Gonçalves et al., 2013].

2.1.5 Coordination strategy in this work

In the present work, the results in [Gonçalves et al., 2013] for planning cyclic trajectories are extended by considering constraints related to finite size robots, speed and acceleration limits and uncertainties, which are most of the time disregarded in previous related works. The consideration of finite size robots makes the guarantee of collision free trajectories more realistic. The maintenance of minimum and maximum speed limits allows the method to be applied to fixed wing UAVs. Acceleration limits ensure the achievement of feasible trajectories. Taking into account uncertainties in realistic scenarios is also essential. The method is based on the definition of an online computation of reference speeds that keeps the cycles of the robots synchronized, which is necessary to the veracity of the safety constraints. In fact, the coordination strategy can be divided in two stages. The first one is a discrete high level strategy solved by the MILP. The second one is a continuous lower level strategy which, according to the dynamics of the robot, ensures the performance planned by the MILP. The latter is completely decentralized and consists on the online feedback based computation of average speeds for each robot at each segment of its path.

In contrast to [Borkar et al., 2016], which maximizes the sizes of the agents while preventing collisions, the approach presented in the current work considers robots with known physical dimensions and then maximizes the size of the enlargement, further defined as Δs , of the so called collision zones, where the presence of multiple robots

is forbidden at the same instant of time. It was possible to deal with the collision problem by avoiding multiple robots simultaneously in the same collision zone. In the present work, “collision zones” are defined by entrance and exit points in the robots’ predefined paths similarly to what was done in [Peng and Akella, 2003] and [Jufeng and Srinivas, 2005]. In fact, their MILP formulation is very similar to the one in this work in the way that they consider collision zones, and in the way they use a conservative strategy to consider dynamic constraints. The main difference is that here cyclic paths are considered while they consider fixed finite paths – the point-to-point problem.

Thus, the present method required a strategy to solve this infinite horizon problem with a finite number of variables and constraints, the one presented in [Gonçalves et al., 2013]. An important advantage of the collision avoidance treatment presented here is that the robots do not need to execute unpredictable online computed maneuvers. Their trajectories are defined in a way that multiple agents will never traverse a given intersection point very close to each other on time. In addition, limited uncertainties on the velocities executed by the physical robots with respect to the ones computed from the result of the solution of the optimization problem were considered.

Energy issues are out of the scope of this work. However, the techniques previously discussed, such as [Mitchell et al., 2015] and [Mathew et al., 2013], may be directly or indirectly incorporated in the present framework. The predictability of the presented solution may be an important feature to ease the application of a refueling strategy.

Despite the centralization of the presented method, in the results section it is shown to be scalable as well, since it solves an instance with 48 robots within a few seconds. In fact, the consideration of fixed predefined paths and their division into long unidimensional stretches (not small segments associated to small time steps as in [Richards and How, 2004]) can be seen as the justification for the computational efficiency of the method.

2.2 Robot control strategies

In the previous Section, strategies to plan the motion of multiple robots were discussed. In general, those strategies do not tackle the problem of controlling a robot to execute the planned path. This section focus on discussing the state of the art on techniques to control fixed-wing UAVs along desired paths.

An extensive survey about several approaches to the guidance problem is pre-

sented in [Sujit et al., 2014]. The work is destined to practitioners, which shows the importance of such methods in modern technology. For instance, line-of-sight (LOS) [Ambrosino et al., 2009] and nonlinear guidance laws (NLGL) [Park et al., 2007] are based on a virtual target point, which guides the UAV towards the path. A different approach, which is simple and generates good trajectories, considers a guidance vector field (GVF) [Nelson et al., 2007] together with a control law to make the UAV converge to a desired path.

Different approaches to the fixed-wing UAV control include the use of Model Predictive Control (MPC) for path tracking. In [Alessandretti and Aguiar, 2017] a MPC is developed to enforce a UAV to track a path in 2D. In fact, the limitation of fixed height is very common in the literature. In [Oettershagen et al., 2014] a MPC combined with \mathcal{L}_1 -navigation strategy is used to make a fixed-wing UAV track a desired path in 3D. In [Andersen and Kristiansen, 2017] quaternion algebra is applied to perform the path tracking in 3 dimensions, but straight lines and planar circles are considered. A comparison of the performance of control strategies in 3 dimensions is presented in [Pelizer et al., 2017].

Another strategy that is also receiving attention is based on artificial vector fields. They will be better discussed in Sections 2.2.2 and 2.2.3. The use of vector fields to control UAVs along target paths requires the consideration of simplified dynamical models of the airplane. Next section discusses about these models.

2.2.1 Fixed-wing UAV reference model

The modeling of physical systems has a fundamental importance in the development of control strategies. The problem of modeling fixed-wings aircrafts, with their proper characteristics, is tackled in [Stevens et al., 2015]. In Chapter 2 of this book a detailed model with 6 degrees of freedom (DOF) and 12 states is presented. This representative model is suitable for the development of low level controllers and for simulating the UAV dynamics, however, it is too complex to be used in the development of path following strategies.

In order to design guidance laws for UAVs, reference models are used [Jesus et al., 2013], [Olavo et al., 2018]. These models are simple representations of the airplane dynamics that incorporate its basic behavior, such as the presence of a

non holonomic constraint. A model considered in the literature is

$$\dot{x} = v \cos(\theta), \quad (2.1a)$$

$$\dot{y} = v \sin(\theta), \quad (2.1b)$$

$$\dot{z} = \frac{z_c - z}{\tau_z} + u_z, \quad \frac{|z_c - z|}{\tau_z} \leq v_{max}^z \quad (2.1c)$$

$$\dot{\theta} = \frac{\theta_c - \theta}{\tau_\theta} + u_\omega, \quad \frac{|\theta_c - \theta|}{\tau_\theta} \leq \omega_{max} \quad (2.1d)$$

$$\dot{v} = \frac{v_c - v}{\tau_v} + u_a, \quad v_{min} \leq v_c \leq v_{max} \quad (2.1e)$$

in which x , y and z are the Cartesian coordinates of the center of mass of the UAV, θ is the yaw angle, and v is the forward speed in the xy plane with respect to the ground. The control inputs of altitude, yaw angle and forward velocity are z_c , θ_c and v_c , respectively. The time constants, all strictly positive, associated to z , θ , and v are τ_z , τ_θ and τ_v , respectively. The terms u_z , u_θ and u_v are additive uncertainties. The actuation limits are v_{max}^z for the absolute value of the vertical speed, ω_{max} for the absolute value of the turning ratio, and $v_{max} > v_{min} > 0$ for the minimum and maximum forward velocities.

The use of model in (2.1) is suitable given that low level controllers are used to impose its dynamics to the UAV, as it is in shown [Olavo et al., 2018]. Figure 2.4 depicts the scheme to perform this imposition. The target dynamics block is the system in (2.1) and acts as a filter that ensures smooth input control signals to the low level controllers.

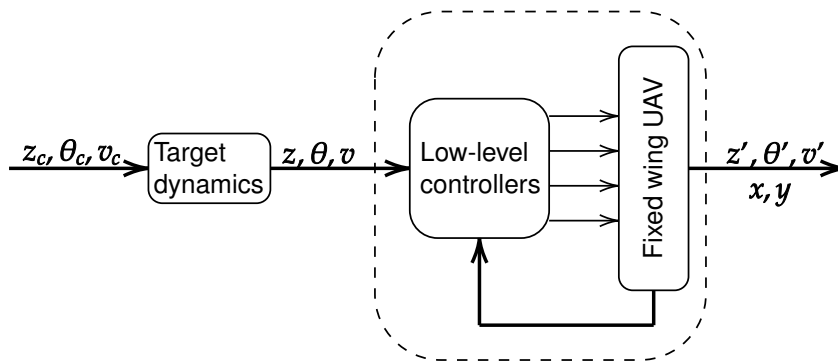


Figure 2.4. Block diagram representing how the target dynamics is imposed to the UAV. *Source:* [Olavo et al., 2018].

In [Thums, 2012] a structure for a low level controller is also presented to control the complete model of an aircraft. The structure consists of 5 PID controllers, 1 PI and 3 cross-coupling gains. A tuning method for the gains of these controllers

is proposed based on evolutionary algorithms [Deb et al., 2002] and Linear Matrix Inequalities (LMI) [Gonçalves et al., 2006].

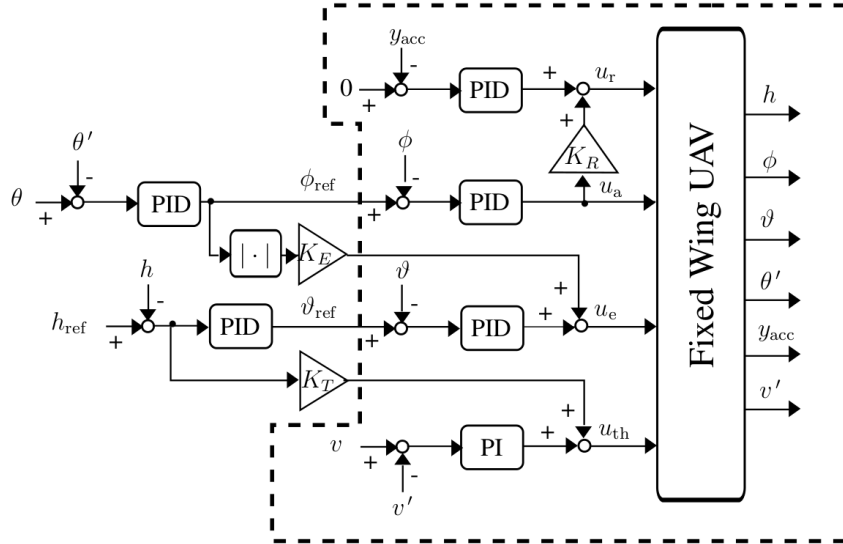


Figure 2.5. Low level controllers considered in [Thums, 2012], [Jesus et al., 2013], [Olavo et al., 2018].

The uncertainties in model (2.1) may account for differences in the reference model and the closed loop system composed of the complete aircraft dynamics, where atmospheric disturbances such as wind gusts and turbulence are present. This is the reason why the low-level PID controllers cannot follow perfectly the reference model (2.1). In other words, the uncertainties u_z , u_ω and u_a may account for natural imperfections in the low-level control of the fixed-wing aircraft. If a guidance strategy is robust to these uncertainties in model (2.1), it may be successfully applied to the closed loop system composed by the complete UAV model plus the PID controllers. These uncertainties are considered to be limited in norm.

Model (2.1) considers a first order dynamics for the height, yaw angle and forward velocity. However, different approaches are also presented in the literature. For instance, in [Quintero et al., 2015] a simpler model is considered, in which the first order dynamics for θ is disregarded. Some works consider an extra step in the integration of the yaw angle command, or yet, as considered in [Ren and Atkins, 2005], the bank angle is considered to have a first order dynamics and the yaw angle rate depends on the bank angle. In [Hull, 2007] a model that includes the pitch angle, in the x , y and z dynamics is considered. In [Jesus et al., 2013] and [Olavo et al., 2018] the model in (2.1) is used and the authors explore the advantages of using a representation of it in polar coordinates.

2.2.2 Vector fields

Among the control strategies to guide robots through specific paths, the artificial vector field based approach is receiving much attention in recent years. It consists on the definition of a vector function Φ that associates an n dimensional vector to each position in the n dimensional workspace. In general, these strategies are easy to implement, have low computational costs, and are accompanied by formal convergence proofs.

The vector field is defined in a way that the integral lines of the vector field converges to the target curve. Thus, if a robot follows the velocity indicated by the vector field it will converge to the curve. Figure 2.6 exemplifies two vector fields in 2 dimensions.

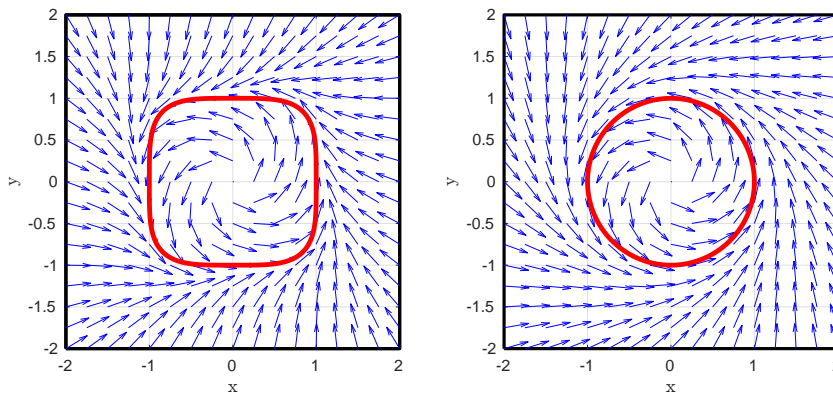


Figure 2.6. Examples of vector fields in \mathbb{R}^2 . In red are the target curves and in blue a normalized vector field.

In recent years, several works have tackled the problem of constructing vector fields for robot control. In [Lawrence et al., 2007] vector fields that converge to circular loiters are developed and convergence proofs are presented based on Lyapunov Theory. Fields that converge to different curve shapes are obtained via diffeomorphisms. In [Lawrence et al., 2007] only planar curves are considered, in other words, fields in 2D or fields in 3D converging to curves with fixed height. In [Frew and Lawrence, 2012] a vector field strategy is developed for three dimensional star shaped curves. A radial Fourier basis set is used to represent the curves and convergence proofs are presented. In [Ceccarelli et al., 2008] a method is developed to make a group of robots converge and circulate a given beacon, which is possibly moving. However, no specific curve is defined to be circulated. In [Liang et al., 2015] the authors propose two static 3D vector field strategies. The proposed field is based on the tangent vector associated to a given point in the curve and in a distance measurement to the curve. Convergence proofs are also presented.

Finally, in [Gonçalves et al., 2010] a strategy is developed for generating n -dimensional time varying vector fields. The authors propose a strategy for generating vector fields with integral curves that converge to and circulate a closed time varying curve \mathcal{P} in n -dimensional spaces.

Here we aim to rapidly recall the vector field strategy. Let n be the dimension of the workspace and $\mathbf{p} \in \mathbb{R}^n$ a coordinate for a point in this space. The strategy in [Gonçalves et al., 2010] relies on the ability of finding $n - 1$ well-behaved scalar functions $\alpha_1(\mathbf{p}, t)$, $\alpha_2(\mathbf{p}, t)$, ... $\alpha_{n-1}(\mathbf{p}, t)$ such that the intersection of the level sets $\alpha_i(\mathbf{p}, t) = 0$, $i = 1, 2, \dots, n - 1$, implicitly defines the target curve \mathcal{P} . Next, the structures of the vector fields for 2 and 3 dimensions are presented without considering the time variation.

Let $\Phi(\mathbf{p}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be the vector field. It will be assumed that n may be 2 or 3, depending on the context. For $n = 2$, we have

$$\Phi(x, y) = G(P) \frac{\nabla P}{\|\nabla P\|} + H(P) \frac{R\nabla\alpha_1}{\|R\nabla\alpha_1\|}, \quad (2.2)$$

where $R \in SO(2)$ is a rotation matrix corresponding to a ± 90 degrees rotation (i.e. clockwise or counter-clockwise) and the operator ∇ performs the gradient with respect to the spatial coordinates. The scalar function $P \equiv P(\alpha_1, \dots, \alpha_{n-1}) : \mathbb{R}^{n-1} \rightarrow \mathbb{R}^+$ is zero at \mathcal{P} and positive elsewhere. The function P represents a distance to \mathcal{P} , and $P(\mathbf{p}) = 0 \iff \mathbf{p} \in \mathcal{P}$. It is also considered a scalar function $G(P) : \mathbb{R}^+ \rightarrow \mathbb{R}^-$, that is $G(P) \leq 0$, such that $G(P) = 0 \iff P = 0$, $|G(P)| \leq 1$. Also, $G(P)$ is decreasing, therefore invertible. The function $H(P)$ is defined as $H(P) = \sqrt{1 - G(P)^2}$. Alternatively, in \mathbb{R}^3 we have

$$\Phi(x, y, z) = G(P) \frac{\nabla P}{\|\nabla P\|} + H(P) \frac{\nabla\alpha_1 \times \nabla\alpha_2}{\|\nabla\alpha_1 \times \nabla\alpha_2\|}, \quad (2.3)$$

where the operator \times corresponds to the cross product.

Note that if $\|\nabla P\| = 0$ or $\|\nabla\alpha_1 \times \nabla\alpha_2\| = 0$ there is a problem with the normalization. We have $\nabla P = 0$ when $\mathbf{p} \in \mathcal{P}$ or in some isolated equilibrium points outside \mathcal{P} . The former case is not a problem since $G(P) \rightarrow 0$ as well. Besides, it is assumed that in the latter case the isolated equilibrium points are repulsive points. The same repulsiveness is assumed when $\|\nabla\alpha_1 \times \nabla\alpha_2\| = 0$.

Figure 2.7 illustrates the level sets used in the construction of the vector field given by (2.3) The zero level sets of two scalar functions α_1 and α_2 are highlighted in blue and red. The intersection between them (black line) defines the closed curve to which the integral lines of the field converge.

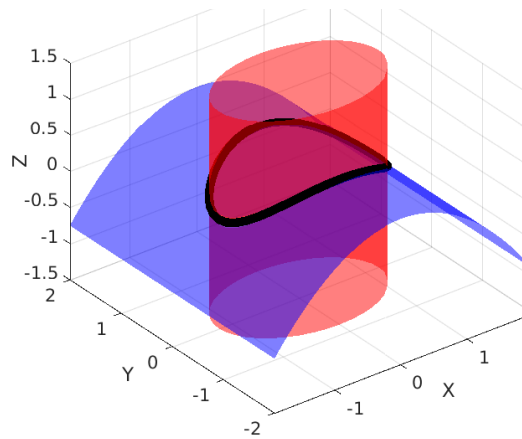


Figure 2.7. Representation of the strategy in [Gonçalves et al., 2010]. Two level sets in \mathbb{R}^3 and the intersection between them.

The terms in (2.2) and (2.3) can be easily interpreted as a convergent component and a circulation component. Let Φ_v be the term that multiplies $G(P)$ and Φ_t the one that multiplies $H(P)$. The components Φ_v and Φ_t are responsible for the convergence to the curve and for the curve tracking, respectively. In [Gonçalves et al., 2010] some important properties of the field are proved, from which two of them are recalled.

Property 1 *Considering the simple integrator model, that is, $\dot{\mathbf{p}} = \Phi$, the function $P(\mathbf{p})$ is a Lyapunov function and $\dot{P} = \nabla P^T \Phi = G(P) \|\nabla P\| \leq 0$.*

Property 2 *For every non fixed point, the convergent component (Φ_v) and the circulating one (Φ_t) are perpendicular to each other, i.e., $\Phi_v \perp \Phi_t$.*

The recent work in [Wu et al., 2018] has presented a strategy to generate vector fields based on contraction analysis theory. The authors consider curves in generic manifolds and use generic metrics to compute the distance to the curve. An advantage of their work is that the curve is considered to be represented by a parametric equation, which is simpler than the representation with implicit functions of [Gonçalves et al., 2010].

2.2.3 Guidance vector field strategies

The velocities indicated by a vector field Φ can be directly applied to a robot if it is represented by the simple integrator model $\dot{\mathbf{p}} = \mathbf{u}$. In general, holonomic robots, those that do not have velocity constraints, can be controlled such that the closed loop system is well represented by the simple integrator. A good example is the aerial robot of the quadrotor type in low velocities. Since fixed-wing UAVs have a non holonomic constraint, it is not possible to develop low level controllers such that the closed loop

system is well represented by the simple integrator. Models such as the ones described in Section 2.2.1 are the sort of models that can be obtained. Thus, if it is desired to control a fixed-wing vehicle to follow a vector field, additional control laws are needed. These controllers are usually referred to as *Guidance Vector Field* (GVF) laws. In general, these laws consists mainly on the computation of yaw angle rates.

Guidance vector field laws are often investigated using straight lines and circles [Frew et al., 2008]. In [Lawrence et al., 2007], GVF laws initially developed for circles are then applied to different shapes obtained by covariant transformations. In [Kapitanyuk et al., 2017] a controller is developed for generic curve shapes. The method develops a controller that is able to align the UAV yaw angle with the orientation of the vector field. Asymptotic convergence of the orientation is obtained with a controller with a proportional and a feed-forward term. The authors of [Kapitanyuk et al., 2017] point out the difficulties in finding a formal proof establishing that getting oriented asymptotically with the field leads to convergence to the target path, despite this convergence happens in all practical examples. Figure 2.8 shows the result obtained when the strategy developed in [Kapitanyuk et al., 2017] was applied to make ground robots follow a Cassini oval curve. In [de Marina et al., 2017] a very

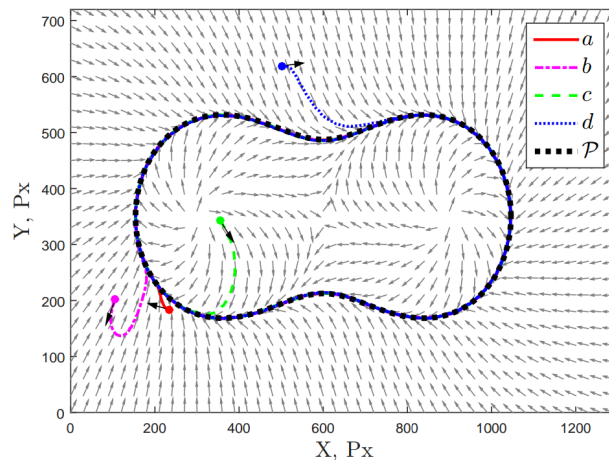


Figure 2.8. Experiments presented in [Kapitanyuk et al., 2017] with ground robots.

similar strategy, which considers the effect of constant wind disturbance, is used to make an actual small size fixed-wing UAV follow an ellipse.

Vector fields in 2 dimensions that converge to closed curves and are continuous always admit at least one fixed point, i.e. a point in which the vector field is null. This is a consequence of the Brouwer fixed-point Theorem [Khamsi and Kirk, 2011]. These fixed points become discontinuity points when the vector field is normalized, which is a usual step in the guidance strategies. See for instance Figures 2.6 and 2.8.

The work in [Kapitanyuk et al., 2017] also presents a formal discussion related to these singularities. There they establish a set of initial conditions for the UAV that lead it to convergence to the path without passing through the singularities. In general, such discussions are disregarded in the literature of vector field guidance strategies.

In [Liang et al., 2015], controllers to guide a UAV in vector fields defined in 3 dimensions are developed. Convergence of the two proposed fields are presented, but no proof regarding the guidance controllers is performed. In fact, obtaining convergence proofs for the proposed controller is an important objective of many related works. In [Jesus et al., 2013] a simplified non-holonomic model is considered in order to prove the convergence to a circular path with a hybrid controller. Despite the use of the simplified model, the authors use a set of low level controllers that impose the simplified dynamics to the actual UAV. In [Olavo et al., 2018] the work is extended with uncertainties in the model, which may account for modeling errors and external disturbances. In the presence of uncertainties the UAV converges to a band around the target circle. In Fig. 2.9 the trajectories are shown, for different initial conditions, for a UAV following a circle with radius 500m according to the strategy in [Olavo et al., 2018]. No additional disturbance was added to the complete model. Note that, due to the switches in the control law, sometimes the UAV follows a straight line or a circle. The UAV is guided by the vector field only when the vehicle is approaching the circle from outside with a proper orientation. In other configurations the airplane follows or a straight line or a circle with fixed radius of curvature.

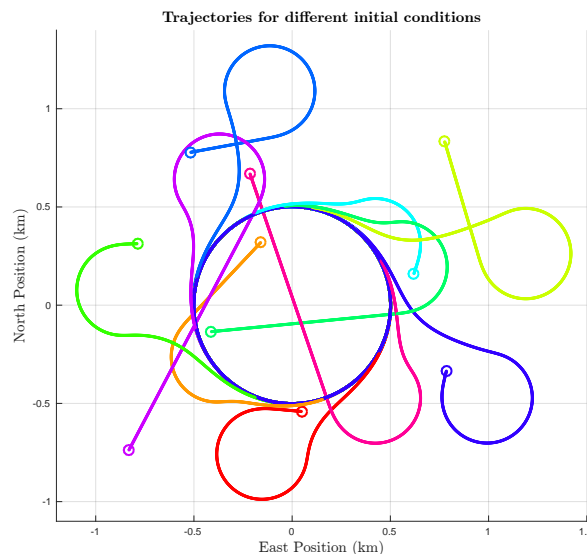


Figure 2.9. Results obtained according to the strategy in [Olavo et al., 2018].

2.2.4 Control strategy in this work

The control strategy presented in this work is based on vector fields and is able to make a UAV converge to and circulate a non planar closed curve with generic shape in 3 dimensions. The controllers are designed upon the reference model in (2.1). The convergence proofs are presented based on the vector field structure developed in [Gonçalves et al., 2010], since it has a formal formulation and is able to generate fields whose integral lines converge to generic curve shapes.

The guidance law is similar to the one developed in [Kapitanyuk et al., 2017], in the sense that it is composed of a proportional term and a feedforward term. However, non planar curves are also considered here. In addition, model uncertainties are considered. When the uncertainties are taken into account it is proved that the UAV does not converge exactly to the curve but it remains inside a tube around it. In other words, the existence of an invariant set is proved. This result is analogous to the one in [Olavo et al., 2018], in which a similar ultimate bound is found for circular curve shapes only.

The present work also tackles the problem of singularities in the vector field. These singularities are points in which the field admits fixed points, which become discontinuity points when the field is normalized. The work in [Kapitanyuk et al., 2017] also tackle this problem, where the authors compute a set of initial conditions for the UAV that will not lead it to the singularities, given the developed control laws. A different strategy is adopted here. We define a circle around the singularity point and the UAV is commanded to fly in a straight line when inside this circle. The strategy is proven to converge.

In summary, the control strategy developed in this work is able to deal with the majority of the details commonly addressed separately in different works that consider GVF, namely: 3 dimensional workspace as in [Liang et al., 2015]; generic curve shapes as in [Kapitanyuk et al., 2017] and [Liang et al., 2015]; uncertainties in the reference model as in [Olavo et al., 2018]; and treatment of singularities in the vector field as in [Kapitanyuk et al., 2017]. Thus, the contribution is a guidance vector field technique that considers generic curve shapes in 3D while dealing with uncertainties in the aircraft reference model and singularities in the field.

Chapter 3

Coordination strategy

3.1 Multi-robot problem statement

The multi-robot problem addressed in this work consists on the coordination of a group of robots with the task of following curves, i.e. each agent must follow a pre-specified closed path periodically with given orientation. Formally consider the following definition

Definition 1 *Let \mathcal{R} be the set of robots indexed by i and \mathcal{P}_i a subset of \mathbb{R}^3 defined by the points that compose the path to be cyclically traveled by the robot $i \in \mathcal{R}$.*

These paths are assumed to have a geometry that is feasible to be followed by the robots. It is admitted that a robot's path have intersecting points with other paths. The objective is to elaborate a speed planning for each robot in such a way that collisions between robots are avoided. This problem is essentially the same as the one proposed in [Gonçalves et al., 2013]. However, issues related to practical implementations are now also tackled. In fact, four important features are incorporated:

1. At all time the robots must remain separated by a minimal spatial distance considering their finite sizes.
2. The robots' speeds must be in the intervals defined by minimum and maximum limits.
3. The robots might be unable of changing their velocities instantaneously, i.e. a velocity dynamics shall be considered.
4. The robots' speeds and positions might be uncertain.

Fig. 3.1 shows a simple possible scenario, where two robots must follow two closed curves in counterclockwise direction, the red and blue ellipses. The stretches dotted in black are the collision stretches. The union of collision stretches with associated collision points forms a collision zone. The *limit points* are defined as the boundaries of the collision stretches, represented as black dots in Fig. 3.1. Note that a collision zone may contain more than one intersection point. A collision zone may also contain stretches of paths of more than two robots (it is not the case of Fig. 3.1). The definition

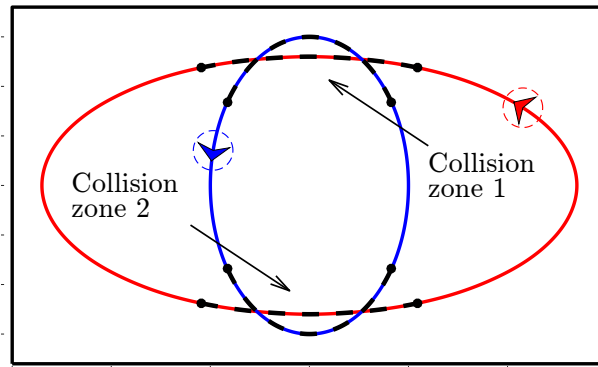


Figure 3.1. Example of a possible configuration of our problem. Two closed paths and dangerous stretches (dashed black). Dotted circles show the robots' size.

of the collision zones must be made in such a way that for a collision to occur between robots i_1 and i_2 , they must be in the same collision zone. Therefore it is established that: *the presence of more than one robot in the same collision zone at the same time should be forbidden.*

For now, an important assumption in the coordination problem is that the robots are always at their pre-specified paths. The problem of controlling the robots to remain in their paths will be addressed in the next chapter. It is evident that, if we want to control a robot to follow a specified path, a position measurement is necessary. In addition, if we want it to travel with a given velocity, a velocity measurement should also be available. These data are assumed to be available within an uncertainty margin, as will be discussed still in this section.

On the path of each robot we define points named *target points* that will be either the robots' entrance to or exit from a collision zone. They divide the path into *segments*. Their positions are not defined *a priori*, i.e. it is considered that they exist but their exact location will be determined as part of the solution to the problem. Desired time instants to reach each target point of a path must also be found. In the surroundings of each target point we define an *uncertainty stretch* or *region of uncertainty*, where it will be possible to guarantee the presence of the robot in the specified instants. The target points must be placed somewhere outside the collision zones considering also

the size of the regions of uncertainty. This ensures that the robot will never be inside the collision zone at the wrong time even in the presence of uncertainties. The size of these uncertainty regions are not known *a priori* either.

Fig. 3.2 shows how the target points could be placed in the paths and possible regions of uncertainty around them. Note that the target points are outside the collision zones and their uncertainty regions do not overlap the collision stretches. The formulation must accomplish that. In the figure, the red robot must travel through the points $q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_1$ and the blue one through $q_5 \rightarrow q_6 \rightarrow q_7 \rightarrow q_8 \rightarrow q_5$. Now, since the instants for a robot to reach each point q must be found, it is evident that the time to traverse each segment from one target point to the other will also be found.

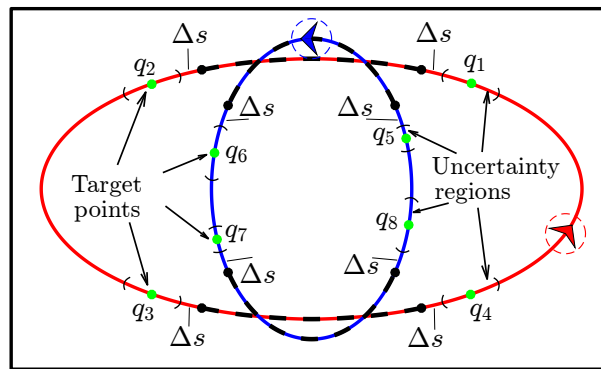


Figure 3.2. Illustration of the target points (green) placement and regions of uncertainty (between black curved traces) in the example of Fig. 3.1.

The length Δs is defined as the geodesic distance between the regions of uncertainty and the collision stretches, see Fig. 3.2. This distance will be a decision variable in our formulation.

3.1.1 Speed model

In order to ensure that the agents will not be in the wrong place at the wrong time it is necessary to control the velocities of the robots as they travel along their paths. It is considered that, first, the speed references for the robots will be allowed to change at specific instants of time. These are the instants in which the robots are required to reach the target points. Second, it is considered, for now, the existence of a low-level controller that will make the change of speed evolve as a ramp that connects the initial velocity to the final one in a predefined time interval τ . If t^* is the time that has passed since the instant the speed command is changed and $v(0)$, $v(T)$ are the speed at the beginning and at the final time $t^* = T$, respectively, the reference velocity of the robot

will be given by

$$v(t^*) = \begin{cases} v(0) + \mathbf{a}t^*, & \text{if } 0 < t^* \leq \tau, \\ v(T), & \text{if } \tau < t^* \leq T, \end{cases} \quad (3.1)$$

in which $\mathbf{a} = (v(T) - v(0))/\tau$ is the acceleration within the interval $0 \leq t^* \leq \tau$. Acceleration and velocity limits are also incorporated to model (3.1). For this purpose, \mathbf{a}_{min} and \mathbf{a}_{max} are defined as the minimum and maximum values of the linear acceleration and v_{min} and v_{max} as the lower and upper limits of the instantaneous commanded speed. The choice of the model (3.1) is interesting for two reasons. First, its simplicity will allow to write velocity constraints in a linear form. Second, it is possible to choose τ as small as possible, according to the acceleration limits, in order to explore the capacity of a robot to change its speed. In fact, note that the inferior limitation of τ implies in an acceleration limit if the linear speed is limited.

3.1.2 Uncertainty model

It is also considered that the velocities of real robots are subject to uncertainty, i.e, the profile given by (3.1) may not be perfectly tracked by the low level motion controller. This can happen, for example, due to measurement or actuation errors. This controller is in a level even lower than the controller responsible to perform (3.1) and, for now, will not be discussed in detail. The deviation of the actual velocity from that profile is modeled as an additive disturbance given by

$$u(v) = u_v + u_{\%}v, \quad (3.2)$$

where $u(v)$ is the disturbance, v is the commanded velocity and u_v and $u_{\%}$ are bounded random variables limited by $[-U_v^i, U_v^i]$ and $[-U_{\%}^i, U_{\%}^i]$, respectively, where i indicates the robot. The portion u_v is an error that is independent of the commanded speed while the portion $u_{\%}v$ is proportional to it. Uncertainty in the measurement of position is also considered as a random variable limited by $[-U_p^i, U_p^i]$. This implies that the exact position errors that will be used in the online speed reference computation are not perfectly measured.

The general problem of coordinating the speeds of the robots to maximize safety while they traverse their associated curves is translated in this work as the following problems:

Problem 1 Find the instants in time each robot must reach each uncertainty region associated with a target point such that Δs , the size of the enlargement of the collision zones, is maximized without violation of speed and acceleration constraints. In parallel, find the position of the target points and the sizes of the uncertainty regions given the bounds $[-U_v^i, U_v^i]$, $[-U_{\%}^i, U_{\%}^i]$ and $[-U_p^i, U_p^i]$.

Problem 2 In order to execute the solution of Problem 1, find the control law to generate appropriate reference velocities and accelerations for the robots' low-level controllers.

3.2 Definition of collision zones

In this section the collision zones will be mathematically defined. The definition will be made with basis on an equivalence relation that will partition a set of collision points on collision zones. For details on equivalence relations see Chapter I, Section 7 of [Kunen, 2009].

Definition 2 Let $\mathcal{Q} = \bigcup_{i \in \mathcal{R}} \mathcal{P}_i$, i.e., the union of the points of all paths.

Definition 3 The set of collision points is a collection \mathcal{C} of all points $c \in \mathcal{Q}$ such that there exists another $c' \in \mathcal{Q}$ such that a robot placed in c collides with a different robot placed in c' . Formally

$$\mathcal{C} = \{c \in \mathcal{Q} \mid \exists i, j \neq i, c \in \mathcal{P}_i, c' \in \mathcal{P}_j : \text{dist}(c, c') < \rho_{ij}\}, \quad (3.3)$$

in which ρ_{ij} is a predefined minimum separation between robots i and j and $\text{dist}(c, c')$ is the Euclidean distance between c and c' .

In order to deal with the issue of the minimum spatial separation the set \mathcal{C} will be partitioned. Each partition will represent a region where the presence of more than one robot may cause a collision. A convenient and elegant way to partition a set is by creating an equivalence relation on \mathcal{C} . According to some criterion, this relation describes if two points should be in the same partition or not. In the current problem, it is reasonable to require that two points on the same path \mathcal{P}_i that are in the same connected component, considering connectivity in \mathcal{C} , should be in the same partition. Furthermore, two points in \mathcal{C} of different paths should be in the same partition if they are separated by a distance that causes a collision. Now consider the following two relations on \mathcal{C} that will be used to represent the two aforementioned partition criteria.

Definition 4 Let the relation of neighboring points in the same path be

$$\mathbb{E}_1 = \{(c, c') \in \mathcal{C}^2 \mid c, c' \in \mathcal{P}_i : \text{dist}(c, c') < \epsilon\}, \quad (3.4)$$

in which ϵ is a fixed small number. And the relation of proximity of points in different paths be

$$\mathbb{E}_2 = \{(c, c') \in \mathcal{C}^2 \mid c \in \mathcal{P}_i, c' \in \mathcal{P}_j, i \neq j : \text{dist}(c, c') < \rho_{ij}\}. \quad (3.5)$$

Furthermore, the two requirements are joined as

$$\mathbb{E} = \mathbb{E}_1 \cup \mathbb{E}_2. \quad (3.6)$$

Note that \mathbb{E} is symmetric and reflexive but it is not an equivalence relation, since it is not transitive, see [Kunen, 2009]. In order to transform it in an equivalence relation it is necessary to recall some concepts.

Definition 5 Given relations \mathbb{A} and \mathbb{B} on the same set \mathcal{C} , the composition of them is

$$\mathbb{A} \circ \mathbb{B} = \{(a, b) \in \mathcal{C}^2 \mid \exists c \in \mathcal{C}, (a, c) \in \mathbb{A} \wedge (c, b) \in \mathbb{B}\}. \quad (3.7)$$

Furthermore, let \mathbb{I} be the identity relation, i.e., all the pairs $\{(a, a) \in \mathcal{C}^2\}$. Let also $\mathbb{A}^k = \mathbb{A}^{k-1} \circ \mathbb{A}$, $\mathbb{A}^0 = \mathbb{I}$. Finally, the transitive reflexive closure of \mathbb{A} is

$$\mathbb{A}^* = \bigcup_{k=0}^{\infty} \mathbb{A}^k = \mathbb{I} \cup \mathbb{A} \cup \mathbb{A}^2 \cup \mathbb{A}^3 \dots \quad (3.8)$$

The transitive reflexive closure \mathbb{A}^* is the minimal equivalence relation that contains \mathbb{A} .

As an example, consider \mathbb{E}_1^* , which is the relation that is true if and only if the pair (c, c') is in the same connected component of \mathcal{C} according to ϵ , in other words, if p and p' are in the same *collision stretch*, that will be mathematically defined ahead.

The equivalence relation can be used to induce a partition by considering two points c and c' in the same set if and only if $(c, c') \in \mathbb{E}^*$. Due to the properties of equivalence relations, it is guaranteed that every point will be placed in one and only one set, thus creating a partition. This procedure, represented by \mathcal{C}/\mathbb{E}^* , is called the quotient of \mathcal{C} by \mathbb{E}^* .

Definition 6 *The set \mathcal{Z} of collision zones is defined as:*

$$\mathcal{Z} = \mathcal{C}/\mathbb{E}^*. \quad (3.9)$$

In Section 3.2.1 it is shown how this quotient is implemented.

In addition, a *collision stretch* is considered as the collection of points in the path of a given robot i_1 in which if robot i_1 is placed in one of those points, there exists a configuration of a different robot i_2 , on its respective path, which might cause a collision between i_1 and i_2 . A collision stretch is the collection of points of a given path \mathcal{P}_i that are connected in \mathcal{C} .

Definition 7 *The set \mathcal{S} of collision stretches is defined as:*

$$\mathcal{S} = \mathcal{C}/\mathbb{E}_1^*. \quad (3.10)$$

Take Figure 3.1 as an example. The set \mathcal{Z} has two elements, the collision zone in the top and the one in the bottom. The set \mathcal{S} has four elements, two stretches in the red path and two more in the blue path, they are the dashed stretches.

Finally, given the definition of the equivalence relation \mathbb{E}^* , if the presence of more than one robot in an element of \mathcal{Z} is avoided, the multi-robot system is collision free. The formulation in Section 3.3 will accomplish that.

3.2.1 Practical implementation

Now, the algorithms able to compute the collision zones in practice will be described. First, it is considered that the curves \mathcal{P}_i are represented by a collection of m points ($m = 1000$ for example).

In order to compute the set \mathcal{C} the impossibility of collisions between a pair of paths is checked first. This can be done by comparing the extreme points in the x and y directions (points with minimum and maximum values of the x and y coordinates) of both paths. For each pair of paths for which the possibility of existence of intersection points is verified, the algorithm checks the distance for each pair of points. If the distance between these points is smaller than a distance ρ_{ij} the pair of points is considered to belong to the collision set \mathcal{C} . The distance ρ_{ij} is defined as the sum of the radius of the two robots. The complexity of the algorithm is then $O(n^2m^2)$, in which n is the number of paths and m is the number of points representing a path ($m = 1000$ in all of the scenarios). This strategy allows the identification of all collision stretches. For that, the small distance ϵ is defined such that it is bigger than the distance between

every pair of consecutive points in a path \mathcal{P}_i , $i \in \mathcal{R}$. In addition, it is possible to define connections between stretches with points that are close to each other. That is, a collision stretch A is connected to a collision stretch B if there is a point in A that is close to a point in B . With this information it is possible to define a graph whose nodes are the collision stretches and the edges establish the connections between pairs of stretches which are close to each other. Finally, Tarjan's algorithm [Tarjan, 1972], which is linear with the number of nodes (number of collision stretches), is then used to find the connected components of this graph. Each component is equivalent to a collision zone.

3.3 MILP formulation

The approach to solve Problem 1 is based on the formulation of a MILP problem, in which constraints are used to model the features previously described. Each boundary of the collision stretches will be extended by a length of Δs . The objective function will be the maximization of this enlargement Δs . To the extended collision stretches and corresponding extended collision zones will be given the names *inflated collision stretches* and *inflated collision zones*, respectively.

In order to ensure a sustained synchronism for the strategy, the aimed average speed to be performed at each segment is computed in a way that the robots are always inside the next regions of uncertainty in the specified instants. This will be the solution to the Problem 2.

Recall that \mathcal{R} is the set of robots indexed by i and \mathcal{P}_i is the path of robot $i \in \mathcal{R}$. Let \mathcal{E} be the set of target points that are entrances of an inflated collision zone and \mathcal{O} the set of target points that are exits of an inflated collision zone. The set \mathcal{H} is the union of \mathcal{E} and \mathcal{O} , $\mathcal{H} = \mathcal{E} \cup \mathcal{O}$. Let $\mathcal{W}(q)$, in which $q \in \mathcal{E}$, be the set of the other target points that are entrances of the collision zone in which q is entrance. Now, consider the following functions:

- $n(q) : \mathcal{H} \mapsto \mathcal{H}$ is the “next point” function, which maps the target point q to the next one on the path that contains q .
- $p(q) : \mathcal{H} \mapsto \mathcal{H}$ is the “previous point” function, which maps each target point to the previous one on its path.
- $f(i) : \mathcal{R} \mapsto \mathcal{H}$ is the “first point” function, which maps each robot i to the defined initial target point on its path.
- $b(q) : \mathcal{H} \mapsto \mathcal{R}$ is the function that maps each target point q to the associated robot i .

- $L(q) : \mathcal{H} \mapsto \mathbb{R}^+$ is the function that returns the length of the stretch of \mathcal{P}_i , in which $i = b(q)$, from the boundary of the collision stretch near to q to the boundary of the collision stretch near to $n(q)$.

Now consider the following decision variables:

- $r[q] \in \mathbb{R}^+$: a half of the length of the region of uncertainty associated to the target point q . It is also the distance from q to the boundary of its associated inflated collision stretch.

- $\mathcal{L}[q] \in \mathbb{R}^+$: the length of the segment of \mathcal{P}_i , in which $i = b(q)$, from the target point q to $n(q)$.

- $\Delta s \in \mathbb{R}^+$: extra length to be added to each boundary of the collision stretches.

- $C[i] \in \mathbb{R}^+$: the desired time to be spent by a robot i to fulfill its periodic path once.

- $t[q, k] \in \mathbb{R}$: the desired time instant in which point q must be reached by the associated robot $i = b(q)$ in the k -th cycle of i . The index $k \in \mathbb{Z}$ is incremented every cycle.

- $T[q, k] \in \mathbb{R}^+$: the desired time to be spent by a robot $i = b(q)$ traveling from q to $n(q)$ in the k -th cycle.

Note that parenthesis are used on values that are known *a priori* and brackets on decision variables that will be known only after the MILP solution.

In Fig. 3.3 the above definitions are represented by showing a scenario where the path \mathcal{P}_1 (blue) intersects \mathcal{P}_2 (red). The target points are in green, and q_1, q_3, q_5 and q_7 , belong to \mathcal{E} , while q_2, q_4, q_6 and q_8 , belong to \mathcal{O} . The collision stretches are in dashed black and the collision zones are the stretches inside each gray polygon. The regions of uncertainty are represented by the curved traces around the target points. The orange arrows indicate the length Δs . The lengths $L(q)$ are exemplified for the points q_4 and q_5 , while the lengths $\mathcal{L}[q]$ are shown for the points q_2 and q_7 . The lengths $r[q]$ are indicated for the points q_5 and q_6 . Other examples are $q_2 = n(q_1)$ and $q_8 = p(q_5)$.

In the MILP formulation, the consideration of fixed paths and collision zones allows a complete abstraction of the curves geometry and one has to consider only their one dimensional lengths between the limit points and between the target points. In fact, this is a very important feature of the formulation, since it leads to a very reasonable execution time for the MILP solver.

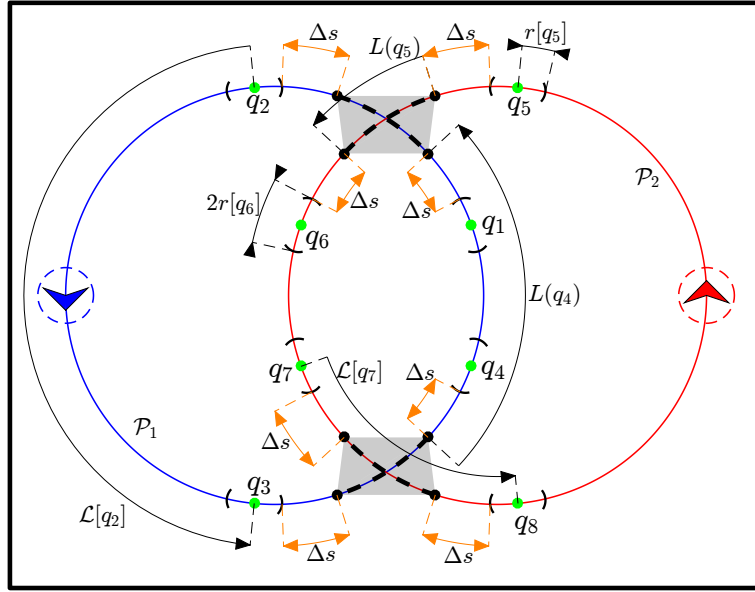


Figure 3.3. Illustration of the definitions made in Section 3.3.

3.3.1 Safety and periodicity

According to the definitions made, the time interval $T[q, k]$ is

$$T[q, k] = \begin{cases} t[n(q), k] - t[q, k], & \text{if } n(q) \neq f(b(q)), \\ t[n(q), k + 1] - t[q, k], & \text{if } n(q) = f(b(q)). \end{cases} \quad (3.11)$$

In order to ensure that a robot enters an inflated collision zone only after all the other robots have already left, the following constraints are defined

$$\begin{aligned} t[q_1, k_1] - t[q_2, k_2] &\geq T[q_2, k_2] & \text{if } t[q_1, k_1] \geq t[q_2, k_2], \\ t[q_2, k_2] - t[q_1, k_1] &\geq T[q_1, k_1] & \text{if } t[q_2, k_2] \geq t[q_1, k_1], \\ \forall q_1 \in \mathcal{E}; \quad \forall q_2 \in \mathcal{W}(q_1); \quad k_1, k_2 \in \mathbb{Z}. \end{aligned} \quad (3.12)$$

Figure 3.4 illustrates these constraints. The gray polygon is the collision zone, and in order to keep it without the presence of two robots at the same time, one of the two equations in (3.12) must be satisfied. The first equation means that robot 2 (red) reaches point $n(q_2)$ (leaves the region) before robot 1 reaches point q_1 (enters in the region). The second one is symmetric, meaning that that robot 1 reaches point $n(q_1)$ before robot 2 reaches point q_2 . Note that the definition in (3.11) was used to write the instants the points $n(q_1)$ and $n(q_2)$ are reached.

Note that (3.12) is an infinite set of constraints. The periodicity and commensurability of the cycles will allow the limitation of the number of constraints in (3.12).

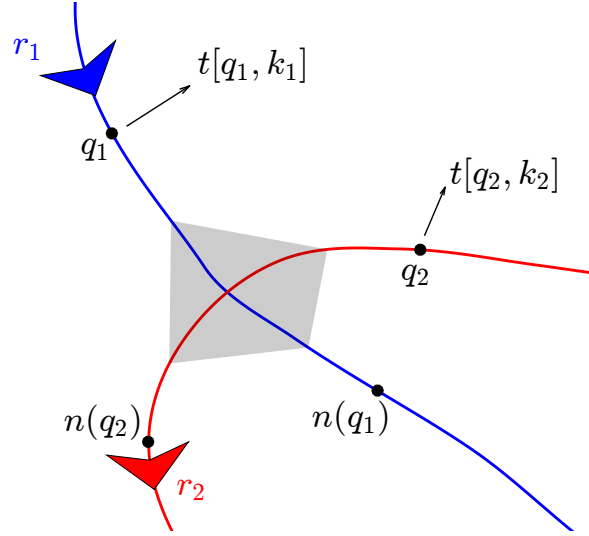


Figure 3.4. Illustration of the safety constraints.

In order to make the cycles commensurable it is enough to force the cycle time of each robot to be an integer multiple of a fundamental cycle time. Let C_0 be the fundamental cycle time and $\lambda(i) \in \mathbb{N}^+$ the integer multiplier for the robot i . Thus, the cycle time $C[i]$, of robot i , is given by

$$C[i] = \lambda(i)C_0, \quad \forall i \in \mathcal{R}. \quad (3.13)$$

In order to make the cycles periodic it is necessary to force every robot i reach every target point periodically with a period equal to its cycle time. For that the following equality constraint is considered, in which $i = b(q)$

$$t[q, k] = t[q, 0] + kC[i], \quad \forall q \in \mathcal{H}; \quad k \in \mathbb{Z}. \quad (3.14)$$

The constraints in (3.13) and (3.14) imply

$$T[q, k] = T[q, 0], \quad \forall q \in \mathcal{H}; \quad k \in \mathbb{Z}. \quad (3.15)$$

Based on these ideas of periodicity and commensurability, the next Lemma provides an important result which will allow the limitation of the number of constraints.

Lemma 1 *Consider the following complementary constraints:*

$$\begin{aligned} t[q_1, 0] - t[q_2, 0] + C_0 &\geq T[q_2, 0], \\ t[q_2, 0] - t[q_1, 0] + C_0 &\geq T[q_1, 0], \\ \forall q_1 \in \mathcal{E}; \quad \forall q_2 \in \mathcal{W}(q_1). \end{aligned} \quad (3.16)$$

If the complementary conditions in (3.16) are satisfied then it is enough to consider the constraints in (3.12) with $k_1 = k_2 = 0$ to guarantee safety for all cycles.

Proof: Using (3.14) and (3.15) to rewrite the inequality in (3.12) the following constraints are obtained

$$\begin{aligned} t[q_1, 0] - t[q_2, 0] + \Delta k C_0 &\geq T[q_2, 0] & \text{if } t[q_1, 0] - t[q_2, 0] + \Delta k C_0 &\geq 0, \\ t[q_2, 0] - t[q_1, 0] - \Delta k C_0 &\geq T[q_1, 0] & \text{if } t[q_2, 0] - t[q_1, 0] - \Delta k C_0 &\geq 0, \\ \forall q_1 \in \mathcal{E}; \quad \forall q_2 \in \mathcal{W}(q_1); \quad \Delta k &\in \mathbb{Z}. \end{aligned} \tag{3.17}$$

in which $\Delta k = k_1 \lambda(i_1) - k_2 \lambda(i_2)$, $i_1 = b(q_1)$ and $i_2 = b(q_2)$. Note that since $k_1, k_2 \in \mathbb{Z}$ and $\lambda(i_1), \lambda(i_2) \in \mathbb{N}^+$ we can infer that $\Delta k \in \mathbb{Z}$. Considering the case in which $\Delta k \geq 1$ the first inequality in (3.16) implies that the first inequality in (3.17) is active. Then, the first inequality in (3.16) implies that the active inequality in (3.17) is satisfied. The symmetric effect happens when $\Delta k \leq -1$. In this case the second inequality in (3.16) implies that the second inequality in (3.17) is both active and satisfied.

The only remaining case is then when $\Delta k = 0$. Thus, if the conditions in (3.16) are satisfied, safety can be guaranteed by satisfying (3.17) with $\Delta k = 0$ which is the same as (3.12) with $k_1 = k_2 = 0$. This concludes the proof. \blacksquare

It is now necessary to rewrite the safety conditions in (3.12) in a format that can be used in a MILP formulation. Let ξ be a very large positive number and $B[q_1, q_2] \in \mathbb{N}$ be a binary variable (can only assume the values 0 or 1) for each pair of points $q_1, q_2 \in \mathcal{E}$ associated to the same collision zone. Details on the usage of this large number when dealing with this type of constraint can be found in Section 4.3 of [Bazaraa et al., 2011]. The safety constraints can be written as

$$\begin{aligned} t[q_1, 0] - t[q_2, 0] &\geq T[q_2, 0] - \xi B[q_1, q_2], \\ t[q_2, 0] - t[q_1, 0] &\geq T[q_1, 0] + \xi(B[q_1, q_2] - 1), \\ \forall q_1 \in \mathcal{E}; \quad \forall q_2 \in \mathcal{W}(q_1). \end{aligned} \tag{3.18}$$

The binary variable is used to define which robot is going to be the first one to enter in the region. Finally, it is also necessary to consider the variables $t[q, k]$ for $k = 0$ and for $k = 1$ if $q = f(i)$, in which $i = b(q)$. The unit value of k is necessary in this case to allow the transitions between cycles, in other words, to account for the completion of one cycle. The periodicity and commensurability can then be guaranteed

by considering the following constraint

$$t[f(i), 1] - t[f(i), 0] = \lambda(i)C_0, \quad \forall i \in \mathcal{R}. \quad (3.19)$$

To conclude, the constraints that will be included in the MILP formulation to guarantee safety, periodicity and commensurability are (3.16), (3.18) and (3.19), respectively.

3.3.2 Average speed reference

In order to avoid the loss of synchronism due to the presence of uncertainty in the speeds, reference values of the average speeds for the coming segments are computed online at every instant $t[q, k]$. A speed value is given as reference to the robot so that it can control itself in order to attempt to reach the next target point on time. The computation is based on the estimate of the current position error. The idea behind the strategy is that even in the presence of uncertainty, by computing these reference values the robots will always be able to reach the region of uncertainty of the next target point after the time interval $T[q, k]$. It is considered that the robots know the time perfectly, which allows them to know the instants $t[q, k]$. At these instants, a reference speed is computed with basis on the current position error and applied until the next instant $t[q, k]$.

Let $e[q, k]$ be the position error of robot $i = b(q)$ in the k^{th} cycle at the instant $t[q, k]$. It is the length of the sub-path of \mathcal{P}_i that goes from the target point q to the position of robot i at the instant $t[q, k]$. The error $e[q, k]$ is positive if at the instant $t[q, k]$ the robot i has already traversed q and negative if i has not reached q yet. Let $\hat{e}[q, k]$ be a measurement of $e[q, k]$ that has a measurement error in the range $[-U_p^i, U_p^i]$. It is considered that only the access to $\hat{e}[q, k] = e[q, k] \pm U_p^i$ is available, with the superscript i indicating the robot. Let $\bar{V}[q, k]$ be the average speed to be performed by i in the stretch from q to $n(q)$ on its k^{th} cycle. The time i must spend to go from q to $n(q)$ is $T[q, 0]$. Thus, the average velocity can be simply defined by the ratio between the distance i should travel, $\mathcal{L}[q] - e[q, k]$, and the time it should spend. Since the exact error $e[q, k]$ is not known, but only $\hat{e}[q, k]$ is, the reference average velocity is given by

$$\bar{V}[q, k] = \frac{\mathcal{L}[q] - \hat{e}[q, k]}{T[q, 0]} = \frac{\mathcal{L}[q] - e[q, k] \mp U_p^i}{T[q, 0]}, \quad (3.20)$$

with U_p^i representing an error of position measurement of i .

Figure 3.5 illustrates the use of equation (3.20) at the instant $t[q_1, k]$, when a robot must to compute a reference speed for the stretch between target points q_1 and q_2 . The error $e[q_1, k]$, represented in the figure, is negative. In the absence of uncertainties in the time interval $[t[q_1, k], t[q_2, k]]$, the robot would reach q_2 exactly at instant $t[q_2, k]$. In the presence of uncertainty, the robot will be, at instant $t[q_2, k]$, in the region of uncertainty around point q_2 .

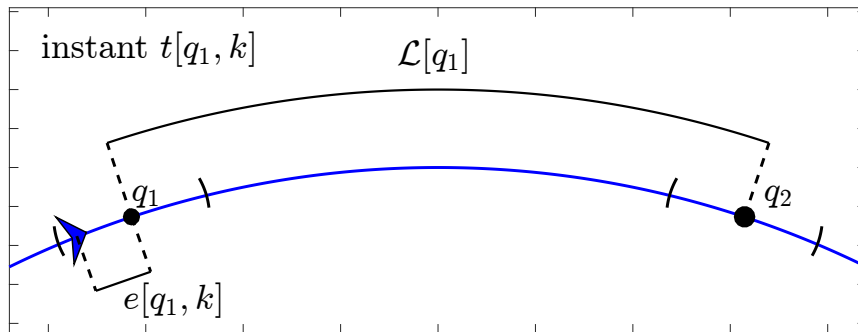


Figure 3.5. Illustration of the average speed computation.

Later, in Section 3.3.4, when the limits of acceleration are considered, the way this average speed is reached will be defined given the robot model (3.1).

3.3.3 Uncertainty

If the reference value defined in (3.20) is used it is clear that in the absence of position measurement error and uncertainty in the velocity in the interval from $t[q, k]$ to $t[n(q), k]$ the robot i reaches $n(q)$ exactly after an interval $T[q, 0]$. According to the definitions made, the position error $e[n(q), k]$ around the point $n(q)$ must be in the interval $[-r[n(q)], r[n(q)]]$. Therefore, the maximum distance (let's say D_m) that i may travel is defined by the sum of the ideal distance (let's say $D_{id} = \mathcal{L}[q] - e[q, k]$) and $r[n(q)]$:

$$D_m = D_{id} + r[n(q)]. \quad (3.21)$$

The maximum traveled distance will happen when the velocity error is at the highest possible value during the whole segment. Using the uncertainty model in (3.2) it is possible to compute

$$D_m = (\bar{V}[q, k] + U_v^i + U_{\%}^i \bar{V}[q, k]) T[q, 0]. \quad (3.22)$$

Note that the cumulative position error due to $U_{\%}^i v$ can always be written as $U_{\%}^i \bar{V}[q, k] T[q, 0]$. Using (3.20) into (3.22) and recalling that $D_{id} = \mathcal{L} - e[q, k]$ it is

possible to write

$$D_m = D_{id} \mp U_p^i + U_v^i T[q, 0] + U_{\%}^i (\mathcal{L}[q] - e[q, k] \mp U_p^i). \quad (3.23)$$

From (3.21) and (3.23), $r[n(q)]$ is given by

$$r[n(q)] = \mp U_p^i + U_v^i T[q, 0] + U_{\%}^i (\mathcal{L}[q] - e[q, k] \mp U_p^i). \quad (3.24)$$

Note that the worst possible case, that is, when the robot will reach the border of the uncertainty region, is when $e[q, k] = -r[q]$ and the measurement error is $-U_p^i$. Considering this case, which may occur in one of the cycles, $r[n(q)]$ becomes

$$r[n(q)] = U_p^i + U_v^i T[q, 0] + U_{\%}^i (\mathcal{L}[q] + r[q] + U_p^i). \quad (3.25)$$

Shifting q back to $p(q)$, in order to formally define $r[q]$ instead of $r[n(q)]$, the definition of the length of half of the uncertainty region is finally defined as

$$\begin{aligned} r[q] &= U_p^i + U_v^i T[p(q), 0] + U_{\%}^i (\mathcal{L}[p(q)] + r[p(q)] + U_p^i), \\ \forall q &\in \mathcal{H}. \end{aligned} \quad (3.26)$$

The set of equality constraints in (3.26) shows that the region of uncertainty around q is defined by the position uncertainty, by the time spent in the segment that precedes q (when there is a constant velocity error) and by the length of the segment that precedes q (when there is a velocity proportional error). Note that we could repeat the discussion considering the minimum possible distance that the robot i may travel. It would give us the same value of $r[q]$, since the uncertainties are considered to be symmetric. Note also that, theoretically, $|e[q, k]| \leq r[q]$ but $|\hat{e}[q, k]|$ may be bigger than $r[q]$, more precisely $|\hat{e}[q, k]| \leq r[q] + U_p^i$.

Now, in order to avoid that a robot, due to the uncertainty, is inside a collision zone when it should not be, a target point q must be placed at a distance $r[q]$ away from the boundary of the inflated collision zone (see Fig. 3.3). The region of uncertainty will then be a sub-path of \mathcal{P}_i with length $2r[q]$ centered at the target point q . With these definitions, variables $\mathcal{L}[q]$ can be written as

$$\mathcal{L}[q] = \begin{cases} L(q) + r[q] + r[n(q)] + 2\Delta s, & \text{if } q \in \mathcal{E}, \\ L(q) - r[q] - r[n(q)] - 2\Delta s, & \text{if } q \in \mathcal{O}. \end{cases} \quad (3.27)$$

The constraint $\Delta s \geq 0$ is also added in order to avoid solutions with a decrease

in the length of the collision stretches.

3.3.4 Acceleration limits

The approach that we are developing here relies on the fact that at each instant $t[q, k]$ the robot $i = b(q)$ will be inside the region of uncertainty related to q and have its speed command changed in order to fulfill the speed requirement for the next segment. In order to take into account acceleration limits in the speed variation, this change is assumed to evolve as described in (3.1) with time interval τ_i . Let $V_i[q, k]$ and $V_f[q, k]$ be the initial and final velocities, respectively, of a robot in a segment from q to $n(q)$ and t^* be the time that has passed since the instant $t[q, k]$. The velocity profile on the segment (let's say $v(t^*)$) is

$$v(t^*) = \begin{cases} V_i[q, k] + \mathbf{a}[q, k]t^*, & \text{if } 0 < t^* \leq \tau_i, \\ V_f[q, k], & \text{if } \tau_i < t^* \leq T[q, 0], \end{cases} \quad (3.28)$$

in which $\mathbf{a}[q, k]$ is the used constant acceleration computed by the low-level controller at the instant $t[q, k]$ according to

$$\mathbf{a}[q, k] = \frac{V_f[q, k] - V_i[q, k]}{\tau_i}. \quad (3.29)$$

Note that the acceleration may vary according to how drastically the speed is supposed to change. The acceleration is limited by choosing high enough values for τ_i . This acceleration time can then be defined *a priori* from the minimum and maximum allowed speeds of the robot i , v_{min}^i and v_{max}^i , and minimum and maximum allowed accelerations, $\mathbf{a}_{min}^i < 0 < \mathbf{a}_{max}^i$, also for the robot i

$$\tau_i = \frac{v_{max}^i - v_{min}^i}{\min(-\mathbf{a}_{min}^i, \mathbf{a}_{max}^i)}. \quad (3.30)$$

The velocity $V_f[q, k]$ has also to be computed by the low-level controller to be used as a reference in order to make the robot move according to the desired average speed defined in (3.20). This is simply done by evaluating the following integral

$$\int_0^{T[q, 0]} v(t^*) dt^* = \bar{V}[q, k]T[q, 0]. \quad (3.31)$$

Using (3.28) in (3.31) and solving for $V_f[q, k]$

$$V_f[q, k] = \frac{2\bar{V}[q, k]T[q, 0] - V_i[q, k]\tau_i}{2T[q, 0] - \tau_i}, \quad (3.32)$$

in which $V_i[q, k]$ is a measurement of the current velocity of the robot. Note that any measurement error in $V_i[q, k]$ can be accounted for in U_v^i and $U_{\%}^i$.

3.3.5 Speed limits

Note that the acceleration limits, as described in the previous section, are satisfied as long as $V_f[q, k]$ is in the interval $[v_{min}^i, v_{max}^i]$. Thus, it is necessary to find a set of constraints that implicitly limits the velocities $V_f[q, k]$. This will also limit $V_i[q, k]$, since this is the final velocity of the previous stretch. Being $i = b(q)$, the objective is to guarantee that:

$$v_{min}^i \leq V_f[q, k] \leq v_{max}^i, \quad \forall q \in \mathcal{H}, \quad \forall k \in \mathbb{I}. \quad (3.33)$$

Using the result in (3.32) and the definition (3.20) into (3.33), and solving for $T[q, 0]$, the following constraints are obtained

$$\begin{aligned} T[q, k] &\geq \frac{\mathcal{L}[q] + r[q] + U_p^i}{v_{max}^i} + \frac{\tau_i(v_{max}^i - V_i[q, k])}{2v_{max}^i}, \\ T[q, k] &\leq \frac{\mathcal{L}[q] - r[q] - U_p^i}{v_{min}^i} - \frac{\tau_i(v_{min}^i - V_i[q, k])}{2v_{min}^i}. \end{aligned} \quad (3.34)$$

$\forall q \in \mathcal{H}, \quad \forall k \in \mathbb{I}.$

Since the values of $\hat{e}[q, k]$ used in (3.20) may vary each cycle, the worst cases are considered, which correspond to $r[q] + U_p^i$ in the lower limit of $V_f[q, k]$ and to $-r[q] - U_p^i$ in the upper limit. The values of $V_i[q, k]$ are not MILP decision variables because it would lead to nonlinear equations. This fact can be checked by using (3.32) into the speed constraint (3.34). It prevents the use of (3.34) directly as one of the constraints. To deal with that, (3.34) is rewritten in a conservative form. For the lower limit of $T[q, k] = T[q, 0]$ the speed $V_i[q, k]$ is replaced by v_{min}^i and for the upper limit v_{max}^i is used. Finally, the final speed constraints are

$$\frac{\mathcal{L}[q] + r[q] + U_p^i}{v_{max}^i} + \frac{\tau_i \Delta v^i}{2v_{max}^i} \leq T[q, 0] \leq \frac{\mathcal{L}[q] - r[q] - U_p^i}{v_{min}^i} - \frac{\tau_i \Delta v^i}{2v_{min}^i}, \quad (3.35)$$

$\forall q \in \mathcal{H}.$

in which $\Delta v^i = v_{max}^i - v_{min}^i$.

Note that the limitation is on the velocity setpoint. It means, for example, that a robot may travel with a speed higher than v_{max}^i if the setpoint has a very high value and the speed uncertainty is such that the robot crosses this limit. A possible solution for this is to decrease v_{max}^i by $U_v^i + U_{\%}^i v_{max}^i$. The same can be done for the lower speed limit, that is, v_{min}^i becomes $v_{min}^i + U_v^i + U_{\%}^i v_{min}^i$. In the present work these correction were not considered.

3.3.6 Final MILP problem

Model 1 below completely defines the MILP formulation that maximizes safety, i.e., Δs .

The convention constraint defines a time for the first robot to be on the first point of its path. The symbol $f(1)$ was used to represent the first objective point of the first robot. It is used because the solution is invariant to shifts in the variables $t[q, k]$.

Note that the conditional statements on the last two sets of constraints depend on previously known values. Thus, binary variables are not necessary as in the safety conditions.

The solution of the obtained final MILP formulation, more specifically the variables $t[q, k]$, $r[q]$, Δs and C_0 , is the solution of Problem 1. Incorporated in the constraints, more precisely in the speed constraints, is the lower level control strategy defined in Sections 3.3.2 and 3.3.4. In those sections, equations (3.20) and (3.32) are the solution of Problem 2, that is, the reference speed for the robots, the lower level controller. Problem 1 is then solved offline, while Problem 2 is solved online. In Section 3.4.4 a method to solve Problem 2 offline is presented in the idealized case of null uncertainties.

Model 1 (MILP formulation)**max:** Δs **subject to:***Safety conditions:*

$$t[q_1, 0] - t[q_2, 0] \geq T[q_2, 0] - \xi B[q_1, q_2],$$

$$t[q_2, 0] - t[q_1, 0] \geq T[q_1, 0] + \xi(B[q_1, q_2] - 1),$$

$$\forall q_1 \in \mathcal{E}; \forall q_2 \in \mathcal{W}(q_1).$$

Periodicity and commensurability of cycles:

$$t[f(i), 1] - t[f(i), 0] = \lambda(i)C_0,$$

$$\forall i \in \mathcal{R}.$$

Speed constraints:

$$\frac{\mathcal{L}[q] + r[q] + U_p^i}{v_{max}^i} + \frac{\tau_i (v_{max}^i - v_{min}^i)}{2v_{max}^i} \leq T[q, 0] \leq \frac{\mathcal{L}[q] - r[q] - U_p^i}{v_{min}^i} - \frac{\tau_i (v_{max}^i - v_{min}^i)}{2v_{min}^i},$$

$$\forall q \in \mathcal{H}; i = b(q).$$

Regions of uncertainty:

$$r[q] = U_p^i + U_v^i T[p(q), 0] + U_{\%}^i (\mathcal{L}[p(q)] + r[p(q)] + U_p^i),$$

$$\forall q \in \mathcal{H}; i = b(q).$$

Complementary conditions:

$$t[q_1, 0] - t[q_2, 0] + C_0 \geq T[q_2, 0],$$

$$t[q_2, 0] - t[q_1, 0] + C_0 \geq T[q_1, 0],$$

$$\forall q_1 \in \mathcal{E}; \forall q_2 \in \mathcal{W}(q_1).$$

Positivity of Δs :

$$\Delta s \geq 0.$$

Convention:

$$t[f(1), 0] = 0.$$

Time intervals:

$$T[q, 0] = \begin{cases} t[n(q), 0] - t[q, 0], & \text{if } n(q) \neq f(i), \\ t[n(q), 1] - t[q, 0], & \text{if } n(q) = f(i). \end{cases}$$

$$\forall q \in \mathcal{H}; i = b(q).$$

Segment lengths:

$$\mathcal{L}[q] = \begin{cases} L(q) + r[q] + r[n(q)] + 2\Delta s, & \text{if } q \in \mathcal{E}, \\ L(q) - r[q] - r[n(q)] - 2\Delta s, & \text{if } q \in \mathcal{O}. \end{cases}$$

$$\forall q \in \mathcal{H}.$$

3.4 Discussion on the coordination solution

3.4.1 Alternative objective functions

In the formulation of previous section, any feasible solution with a positive Δs is enough, since the minimum distances, defined by the size of the collision zones, are ensured by the safety constraints. This fact allows one to think about different objective functions, for example, the minimization of the base cycle time C_0 , as in [Peng and Akella, 2003] and [Jufeng and Srinivas, 2005], in order to obtain a more frequent data acquisition.

Another possibility is to maximize the time interval (let's say Δt) in which a collision zone remains without any robot after the departure of the last one. This would require simple modifications in the safety conditions, complementary conditions and stretch lengths. In this case the formulation will be closer to the one in [Gonçalves et al., 2013].

3.4.2 Absence of solution

Note that in constraint (3.35) it is not possible to ensure that the lower limit of $T[q, 0]$ is smaller than the upper limit. This means that it is not possible to ensure that

$$\frac{\mathcal{L}[q] + r[q] + U_p^i}{v_{max}^i} + \frac{\tau_i \Delta v^i}{2v_{max}^i} \leq \frac{\mathcal{L}[q] - r[q] - U_p^i}{v_{min}^i} - \frac{\tau_i \Delta v^i}{2v_{min}^i}. \quad (3.36)$$

By rearranging (3.36) the following is obtained

$$\mathcal{L}[q] \geq \frac{(r[q] + U_p^i)(v_{max}^i + v_{min}^i)}{v_{max}^i - v_{min}^i} + \frac{\tau_i(v_{max}^i + v_{min}^i)}{2}. \quad (3.37)$$

The relation in (3.37) requires that the distance between the target points be greater than a minimum value. Since neither $\mathcal{L}[q]$ nor $r[q]$ are known before the solution of the MILP problem, the feasibility of the problem can not be ensured. In practice the stretches, including the collision ones, must be long enough in order to enable a solution. The worst case consideration in (3.35) requires a minimum time for speed changes from one limit to the other in every stretch. A very small length may turn this scheme unfeasible. As the robots may not need to make this big change of speed every time, the non feasibility in this case is then caused by the worst case consideration. The variable size of the segments given by the variables $r[q]$ and Δs might help to deal with this source of unfeasibility, but it may not be enough. Thus, given the conservative

simplification, it can be stated that the algorithm is incomplete, in the sense it can fail to find a solution even when it exists.

In addition, even if a limit on acceleration is not considered ($\tau_i \rightarrow 0$), the problem may still be unfeasible. When the scenario has many robots in a relatively small area and the majority of the points on the paths are in collision, the set of constraints in (3.18) may never be completely satisfied.

Another source of unfeasibility is related to the predefined values of $\lambda(i)$. The values of the $\lambda(i)$ must be selected according to the lengths of the paths \mathcal{P}_i and the velocity of the robots. For example, suppose that robot i_1 and robot i_2 have the same speed limits and the length of \mathcal{P}_1 is twice the length of \mathcal{P}_2 . If we select $\lambda(i_1) = 1$ and $\lambda(i_2) = 2$, we are requiring i_1 to have an average speed along \mathcal{P}_1 four times the one i_2 has along \mathcal{P}_2 . Then, if $v_{max}^i < 4v_{min}^i$, the problem is unfeasible. Thus, the values of the λ 's must be properly chosen.

3.4.3 Alternative approaches

The constraints in (3.16), as discussed in [Gonçalves et al., 2013], may be too conservative, which might also lead to unfeasibility. The problem can be relaxed if these constraints are replaced by

$$\begin{aligned} t[q_1, 0] - t[q_2, 0] + w(q_1, q_2)C_0 &\geq T[q_2, 0], \\ t[q_2, 0] - t[q_1, 0] + w(q_1, q_2)C_0 &\geq T[q_1, 0], \\ \forall q_1 \in \mathcal{E}; \forall q_2 \in \mathcal{W}(q_1), \end{aligned} \tag{3.38}$$

in which $w(q_1, q_2) \in \mathbb{N}^+$ is defined *a priori*. The next Lemma extends Lemma 1 for the case $w(q_1, q_2) \geq 1$.

Lemma 2 *If the complementary conditions in (3.38) are satisfied then it is enough to consider the constraints in (3.17) with $|\Delta k| < w(q_1, q_2)$ to guarantee safety for all cycles.*

Proof: Remember that $\Delta k = k_1\lambda(i_1) - k_2\lambda(i_2)$ and since $k_1, k_2 \in \mathbb{Z}$ and $\lambda(i_1), \lambda(i_2) \in \mathbb{N}^+$, we have $\Delta k \in \mathbb{Z}$. Then, considering the case in which $\Delta k \geq w(q_1, q_2)$ the first inequality in (3.38) implies that the first inequality in (3.17) is active. Then the first in (3.38) implies that the active inequality in (3.17) is satisfied. The symmetric effect happens when $\Delta k \leq -w(q_1, q_2)$. In this case the second inequality in (3.38) implies that the second inequality in (3.17) is both active and satisfied. The only case left is the one in which $|\Delta k| < w(q_1, q_2)$. This concludes the proof. ■

In summary, in order to relax Model 1, the complementary conditions therein should be replaced by the ones in (3.38). As a consequence, the safety conditions in Model 1 should be replaced by a subset of constraints in (3.17) such that $|\Delta k| < w(q_1, q_2)$. Lemma 2 provides a sufficient condition on the values of $|\Delta k|$. The following Lemma shows that not necessarily all Δk such that $|\Delta k| < w(q_1, q_2)$ should be considered in the safety constraints.

Lemma 3 *If the constraints in (3.38) are used, the values of Δk to be considered in (3.17) must be such that $|\Delta k|/\mathbf{d} \in \mathbb{N}$, in which $\mathbf{d} = \text{gcd}(\lambda(i_1), \lambda(i_2))$, gcd denoting the greatest common divisor.*

Proof: If $\mathbf{d} = \text{gcd}(\lambda(i_1), \lambda(i_2))$, $|\Delta k|$ can be written as:

$$|\Delta k| = |k_1 \lambda(i_1) - k_2 \lambda(i_2)| = \mathbf{d} \left| k_1 \frac{\lambda(i_1)}{\mathbf{d}} - k_2 \frac{\lambda(i_2)}{\mathbf{d}} \right|. \quad (3.39)$$

The term that multiplies \mathbf{d} belongs to \mathbb{N} . Thus all possible values of $|\Delta k|$ are integer multiples of \mathbf{d} . This concludes the proof. ■

This approach is useful when there is a set of $\lambda(i)$ with discrepant values. In this same scope, another strategy that may be used is to consider that the shortest paths, the ones that require the selection of smaller $\lambda(i)$, are transformed in multiple paths. For example, if there are two robots, i_1 and i_2 , initially with $\lambda(i_1) = 2$ and $\lambda(i_2) = 1$, it is possible to consider that the path of i_2 (\mathcal{P}_2) is composed by two laps. Each target point of its path will be transformed in two, one for each lap. In this new situation it is possible to consider $\lambda(i_1) = \lambda(i_2) = 1$. This approach can not give a solution worse than the previous one, since now the robot i_2 will be allowed to execute two laps with different speed profiles, having more freedom to find a better solution.

3.4.4 Speed profile along the paths

In principle it is not possible to define a robot's speed (average, initial and final) in the segments of its path because it is always varying in order to preserve the synchronism. However, it is possible to do it if the uncertainty is disregarded. Suppose that a solution for the MILP problem is already available and the uncertainty is null. The average speed for a segment that starts in a target point q of the path of a given robot is $\bar{V}[q] = \mathcal{L}[q]/T[q, 0]$, since the position error is null. Now, consider (3.32) rewritten in

the following way

$$a_q V_i[q, 0] + V_i[n(q), 0] = b_q, \text{ in which:}$$

$$a_q = \frac{\tau_i}{2T[q, 0] - \tau_i}, \quad b_q = \frac{2\bar{V}[q, 0]T[q, 0]}{2T[q, 0] - \tau_i}, \quad i = b(q). \quad (3.40)$$

Note that in (3.40) the fact that $V_f[q, 0] = V_i[n(q), 0]$ was used and the cycle counter k was replaced by 0, since in the null error case these velocities do not vary between cycles. From the set of equations in (3.40) the following linear system is obtained

$$\begin{bmatrix} 1 & 0 & \dots & a_n \\ a_1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & a_{n-1} & 1 \end{bmatrix} \begin{bmatrix} V_i[q_1, 0] \\ V_i[q_2, 0] \\ \vdots \\ V_i[q_n, 0] \end{bmatrix} = \begin{bmatrix} b_n \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}, \quad (3.41)$$

in which n is the number of objective points in a single path. The system in (3.41) can be written as

$$\mathbf{A} \mathbf{V}_i = \mathbf{B}. \quad (3.42)$$

Theorem 1 *Let \mathbf{A} be a matrix with the form of the first matrix in the left of (3.41). If $0 < \tau_i < T[q, 0] < \infty$ for any q , then \mathbf{A} is nonsingular.*

Proof: The determinant of \mathbf{A} , represented by $|\mathbf{A}|$, defined by using Laplace's formula in the first line is

$$|\mathbf{A}| = 1 + (-1)^{n+1} a_n \cdot |\mathbf{A}_{1n}|. \quad (3.43)$$

The matrix \mathbf{A}_{1n} is upper triangular, thus its determinant is given by

$$|\mathbf{A}_{1n}| = \prod_{q=1}^{n-1} a_q. \quad (3.44)$$

Replacing (3.44) into (3.43), $|\mathbf{A}|$ becomes

$$|\mathbf{A}| = 1 + (-1)^{n+1} \cdot \prod_{q=1}^n a_q. \quad (3.45)$$

Since $0 < \tau_i < T[q, 0] < \infty$ it is clear that according to (3.40), the values of the elements a_q in \mathbf{A} are such that $0 < a_q < 1$. Thus, from (3.45), $|\mathbf{A}|$ is always different

from 0 and \mathbf{A} is nonsingular. ■

Since \mathbf{A} is nonsingular, a solution for \mathbf{V}_i can be easily found by $\mathbf{V}_i = \mathbf{A}^{-1}\mathbf{B}$. With that the result \mathbf{V}_i and (3.1) Problem 2 can be solved offline by generating a cyclic trajectory for each robot. A method to implement the solution of Problem 1 would be the use of a controller to track the obtained trajectory, which will be parametrized on time. Considering $\delta_v = 0$ and $\delta_{\%} = 0$, this controller would have to keep a tracking error smaller than $r[q] = U_p^i$ in order to ensure safety.

Chapter 4

Control strategy

This chapter is dedicated to the development of a control strategy able to guide a fixed wing UAV along a prespecified path. It will allow the implementation of the coordination strategy developed in Chapter 3.

4.1 Control problem statement

The control problem addressed in this chapter aims to establish a control law for a fixed-wing UAV such that it converges to and circulates a closed curve in a given direction. The objective is to apply this control strategy to control each robot $i \in \mathcal{R}$ along the curve \mathcal{P}_i . The application of the control strategy to different robots has absolutely no relation. Thus, in this chapter the sub-index i , indicating the robot, will be suppressed. It will be assumed that a closed curve \mathcal{P} is the limit-cycle of an artificial vector field Φ defined in \mathbb{R}^2 or \mathbb{R}^3 space. This vector field should determine the reference to the controller.

Recall that, from Section 2.2.1, a reference model can be considered for the fixed-wing UAV if low level controllers are used to impose this dynamics. Thus, as done in [Jesus et al., 2013] and [Olavo et al., 2018], the following reference model is considered

$$\dot{x} = v \cos(\theta), \quad (4.1a)$$

$$\dot{y} = v \sin(\theta), \quad (4.1b)$$

$$\dot{z} = \tau_z^{-1}(-z + z_c) + u_z, \quad (4.1c)$$

$$\dot{\theta} = \tau_\theta^{-1}(-\theta + \theta_c) + u_\omega, \quad (4.1d)$$

$$\dot{v} = \tau_v^{-1}(-v + v_c) + u_a, \quad (4.1e)$$

where x , y and z are the Cartesian coordinates of the center of mass of the UAV, θ is the yaw angle, and v is the forward speed in the xy plane with respect to the ground. The control inputs of altitude, yaw angle and forward velocity are z_c , θ_c and v_c , respectively. The time constants, all strictly positive, associated to z , θ , and v are τ_z , τ_θ and τ_v , respectively. The terms u_z , u_ω and u_a are additive uncertainties with absolute value limited by U_z , U_ω and U_a , respectively, i.e.

$$|u_z| \leq U_z, \quad |u_\omega| \leq U_\omega, \quad |u_a| \leq U_a. \quad (4.2)$$

Uncertainties may account for differences between the simplified model (4.1) and the real system and also for measurement and actuation errors. Besides, the following constraints are added to the model in (4.1)

$$|z_c - z| \leq \tau_z v_{max}^z, \quad (4.3a)$$

$$|\theta_c - \theta| \leq \tau_\theta \omega_{max}, \quad (4.3b)$$

$$v_{min} \leq v_c \leq v_{max}, \quad (4.3c)$$

in which v_{max}^z is the maximum absolute value of the velocity in z , ω_{max} is the maximum turning rate, and $v_{max} > v_{min} > 0$ are the minimum and maximum forward speeds. Note that $v_{min} > 0$, which represents the necessity of the airplane to maintain its aerodynamic lift.

The control problem addressed in this work can be stated as follows:

Problem 3 *Find a set of control laws, for z_c , θ_c and v_c , which obey the actuation constraints (4.3) and make the fixed-wing UAV, subject to bounded uncertainties, described by (4.1) and (4.2), converge to a neighborhood of a stable limit-cycle associated with a given guidance vector field.*

4.2 Control design

The key idea to control the model (4.1) is to orient the UAV heading angle with the vector field. Intuitively, by orienting the UAV its non-holonomic constraint is respected, and it will be able to follow the field's integral lines. Thus, a control law for the turning ratio $\omega(t) \equiv \dot{\theta}$, when $u_\omega = 0$, is first developed. It will enable the orientation of the UAV with the field. In the sequence, control laws for forward speed and vertical motion are designed. Next, an analysis when the disturbances are present is performed.

Through the development of the controllers, the interest is only on the orientation of the vector fields. Then the normalized field is considered. Note that convergence properties are not affected by any positive normalization, as proved in [Wu et al., 2018]. Let $\hat{\Phi} \equiv [\hat{\Phi}_x \ \hat{\Phi}_y \ 0]^T \equiv \hat{\Phi}(x, y, z) \in \mathbb{R}^3$ be a horizontal field (no component in the z direction) normalized with the xy components of Φ . Let also $\hat{\Phi}_z \equiv \hat{\Phi}_z(x, y, z) \in \mathbb{R}$ be the vertical component of Φ scaled accordingly. Formally, the definition is

$$\hat{\Phi} = \frac{1}{\sqrt{\Phi_x^2 + \Phi_y^2}} \begin{bmatrix} \Phi_x \\ \Phi_y \\ 0 \end{bmatrix}, \quad \hat{\Phi}_z = \frac{\Phi_z}{\sqrt{\Phi_x^2 + \Phi_y^2}} \quad (4.4)$$

The definition of the *normalized and projected field* $\hat{\Phi}$ will be very important to control the angle θ . Its necessity is intuitive since θ is a rotation around z world axis, being independent of the z component of Φ .

4.2.1 Heading control

Let $\theta_f(\mathbf{p})$ be the orientation of the vector field at point \mathbf{p} , i.e., $\theta_f \equiv \text{atan2}(\Phi_y, \Phi_x)$, where atan2 is the four quadrant arc tangent function. This definition allows to compute the robot's orientation error $\bar{\theta} \equiv \bar{\theta}(\mathbf{p}, \theta)$ as $\bar{\theta} = \theta_f - \theta$ with $\bar{\theta} \in (-\pi, \pi]$. Disregard for now the uncertainty u_ω and consider the time-derivative of the error $\bar{\theta}$, which is

$$\dot{\bar{\theta}}(t) = \dot{\theta}_f(x, y, z) - \omega(t). \quad (4.5)$$

If $\bar{\theta}$ is stabilized at the origin, the UAV will be oriented with the field and will follow its integral lines. The proof that the stabilization of $\bar{\theta}$ leads to convergence of \mathbf{p} to \mathcal{P} will be performed in Subsection 4.2.4. Now, consider the following Lyapunov candidate function:

$$V_\theta(\bar{\theta}) = 1 - \cos(\bar{\theta}), \quad -\pi < \bar{\theta} < \pi. \quad (4.6)$$

Note that $V_\theta(0) = 0$ and $V_\theta(\bar{\theta}) > 0 \ \forall \ \bar{\theta} \neq 0$. The time-derivative of (4.6) is given by

$$\dot{V}_\theta(\bar{\theta}) = \sin(\bar{\theta})\dot{\bar{\theta}} = \sin(\bar{\theta}) \left[\dot{\theta}_f(x, y, z) - \omega \right]. \quad (4.7)$$

In order to define a stabilizing control law ω , it is necessary to compute $\dot{\theta}_f(\mathbf{p})$. Then, consider the following theorem.

Theorem 2 Let $\mathbf{p}(t) = [x(t) \ y(t) \ z(t)]^T \in \mathbb{R}^3$ be an arbitrary trajectory, and $\mathbf{\Phi}(\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be a vector field such that $\mathbf{\Phi}(\mathbf{p}) = [\Phi_x(\mathbf{p}) \ \Phi_y(\mathbf{p}) \ \Phi_z(\mathbf{p})]^T$. Consider Φ_x , Φ_y and Φ_z continuous and differentiable functions with respect to x , y and z . Assume that $\Phi_x(t) \equiv \Phi_x(\mathbf{p}(t))$, $\Phi_y(t) \equiv \Phi_y(\mathbf{p}(t))$, and that the function θ_f is defined as $\theta_f(t) \equiv \text{atan2}(\Phi_y(t), \Phi_x(t))$. Then, the time-derivative $\dot{\theta}_f(t)$ throughout the trajectory $\mathbf{p}(t)$ is given by

$$\dot{\theta}_f(t) = \left(\hat{\Phi}_x \nabla \hat{\Phi}_y^T - \hat{\Phi}_y \nabla \hat{\Phi}_x^T \right) \dot{\mathbf{p}}, \quad (4.8)$$

in which $\hat{\Phi} = [\hat{\Phi}_x \ \hat{\Phi}_y \ 0]^T = [\Phi_x \ \Phi_y \ 0]^T (\Phi_x^2 + \Phi_y^2)^{-0.5}$ is the field projected onto the local horizontal plane and normalized. The vectors $\nabla \hat{\Phi}_x$ and $\nabla \hat{\Phi}_y$ are the gradients of $\hat{\Phi}_x$ and $\hat{\Phi}_y$, respectively, with respect to x , y and z .

Proof: First, note that the function θ_f can be computed from $\hat{\Phi}$ in (4.4). In order to verify this property it is enough to observe that any normalization factor cancels inside the atan function. Then the time-derivative of θ_f can be computed based on $\hat{\Phi}$ as

$$\dot{\theta}_f(t) = \frac{d}{dt} \text{atan} \left(\frac{\hat{\Phi}_y}{\hat{\Phi}_x} \right) = \frac{\dot{\hat{\Phi}}_y \hat{\Phi}_x - \hat{\Phi}_x \dot{\hat{\Phi}}_y}{\hat{\Phi}_x^2 + \hat{\Phi}_y^2}. \quad (4.9)$$

Note that the derivatives of the functions atan and atan2 are equivalent. Note also that $\hat{\Phi}_x^2 + \hat{\Phi}_y^2 = 1$. Replacing it in (4.9) and applying the chain rule to compute the time-derivatives, (4.9) becomes

$$\dot{\theta}_f(t) = \left(\nabla \hat{\Phi}_y^T \dot{\mathbf{p}} \right) \hat{\Phi}_x - \left(\nabla \hat{\Phi}_x^T \dot{\mathbf{p}} \right) \hat{\Phi}_y. \quad (4.10)$$

By isolating $\dot{\mathbf{p}}$, equation (4.10) can be written as in (4.8). This concludes the proof. ■

Note that Theorem 2 provides the value of $\dot{\theta}_f$ for any smooth vector field in \mathbb{R}^3 and for any direction the UAV is moving. This result enables the strategy to be applicable to 3 dimensional paths. Now, replacing (4.8) into (4.7), a stabilizing control law is given by

$$\omega(t) = \left(\hat{\Phi}_x \nabla \hat{\Phi}_y^T - \hat{\Phi}_y \nabla \hat{\Phi}_x^T \right) \dot{\mathbf{p}} + k_p \sin(\bar{\theta}), \quad (4.11)$$

in which $k_p > 0$. Given $u_\omega = 0$, the control law in (4.11) yields,

$$\dot{V}_\theta(\bar{\theta}) = -k_p \sin(\bar{\theta})^2. \quad (4.12)$$

Since for $-\pi < \bar{\theta} < \pi$ the function $\dot{V}_\theta(\bar{\theta}) < 0 \ \forall \ \bar{\theta} \neq 0$, (4.6) is indeed a Lyapunov

function, and the control law in (4.11) ensures asymptotically stability of the robot's orientation error at the origin.

It is important to emphasize that for $\bar{\theta} \sim \pi$ the function $\dot{V}_\theta(\bar{\theta}) = 0$. In fact, the system admits an equilibrium point at $\bar{\theta} \sim \pi$. However, this equilibrium point is unstable, since $V_\theta(\bar{\theta})$ has its maximum value at $\bar{\theta} = \pi$ and $\dot{V}_\theta(\pi - \epsilon) < 0$ for any ϵ arbitrarily small. Since $\bar{\theta} \in \mathbb{S}^1$, it is impossible to establish a continuous control law that stabilizes $\bar{\theta}$ at 0, for which there is no other equilibrium point at any $\bar{\theta} \neq 0$.

Now, limits for $\omega(t)$ will be found given maximum values of the field derivatives. Consider a positive scalar M such that $\|\nabla\hat{\Phi}_x\| \leq M$ and $\|\nabla\hat{\Phi}_y\| \leq M$. Note that in (4.11), $\hat{\Phi}_x = \cos(\theta_f)$ and $\hat{\Phi}_y = \sin(\theta_f)$. Then, by using triangular inequality in (4.11) the following is obtained

$$|\omega(t)| \leq |\cos(\theta_f) - \sin(\theta_f)|M\|\dot{\mathbf{p}}\| + |k_p \sin(\bar{\theta})|. \quad (4.13)$$

The maximum norm of the difference of trigonometric functions is $\sqrt{2}$, thus

$$|\omega(t)| \leq \sqrt{2}M\|\dot{\mathbf{p}}\| + k_p. \quad (4.14)$$

Note that $\sqrt{2}M\|\dot{\mathbf{p}}\| + k_p > \sqrt{2}M\|\dot{\mathbf{p}}\| > 0$. It is now necessary to impose that the right side of (4.14) to be less than ω_{max} , thus, a sufficient condition that ensures constraint (4.3b) is

$$0 \leq \frac{\sqrt{2}M\|\dot{\mathbf{p}}\|_{max}}{\omega_{max} - k_p} \leq 1, \quad (4.15)$$

where $\|\dot{\mathbf{p}}\|_{max}$ is the maximum speed the UAV may reach, that is, $\|\dot{\mathbf{p}}\|_{max} = \sqrt{(v_{max})^2 + (v_{max}^z)^2}$. Note that to control θ , it is required that the vector field itself obeys condition (4.15) in order to (4.3b) be satisfied.

Finally, given equations (4.1d) and (4.11) it is possible to define the control θ_c as

$$\theta_c = \tau_\theta \left(\hat{\Phi}_x \nabla \hat{\Phi}_y^T - \hat{\Phi}_y \nabla \hat{\Phi}_x^T \right) \dot{\mathbf{p}} + \tau_\theta k_p \sin(\bar{\theta}) + \theta. \quad (4.16)$$

4.2.2 Velocity control

Note that the control law ω , defined in (4.11), considers the velocity $v(t)$ in the vector $\dot{\mathbf{p}}$. This allows a decay of the Lyapunov function V_θ that is not dependent on v , (see (4.12)). This decoupling between $v(t)$ and $\omega(t)$ allows the definition of independent references for $v(t)$. Thus, consider a velocity profile $v_r(t)$, such that $v_{min} \leq v_r(t) \leq v_{max}$, to be followed by the UAV. Consider for now a null uncertainty, $u_a = 0$, and let $a = \dot{v}$ be

the acceleration in the horizontal plane. Let $\bar{v} = v_r - v$ be defined as the error in the forward velocity. The following control law is proposed

$$a(t) = \dot{v}_r + k_v \bar{v}, \quad (4.17)$$

in which $k_v > 0$. Consider a Lyapunov candidate function given by $V_v = \frac{1}{2} \bar{v}^2 = \frac{1}{2} (v_r - v)^2$. Its derivative with respect to time is

$$\dot{V}_v = (v_r - v) (\dot{v}_r - \dot{v}). \quad (4.18)$$

Assuming for now $u_a = 0$ it is possible to impose $\dot{v} = a$. Given that, the replacement of (4.17) into (4.18) yields

$$\dot{V}_v = -k_v (v_r - v)^2 \leq 0. \quad (4.19)$$

Since $\dot{V}_v \leq 0$, V_v is indeed a Lyapunov function and $v = v_r$ is an equilibrium point. Finally, with $a(t)$ defined in (4.17) and the dynamics in (4.1e), the control law for v_c is defined as

$$v_c = \tau_v \dot{v}_r + \tau_v k_v \bar{v} + v. \quad (4.20)$$

4.2.3 Altitude control

The dynamics of the height, z , in model (4.1) is completely decoupled from others. In (4.4) $\hat{\Phi}_z$ was defined according to the normalization factor $\|[\hat{\Phi}_x \ \hat{\Phi}_y]\|$. Then, in order to make the UAV follow the field's integral lines, it is necessary to scale a desired velocity \tilde{v}_z to $\hat{\Phi}_z$ with the same factor that $\|[\dot{x} \ \dot{y}]\|$ is scaled to $\|[\hat{\Phi}_x \ \hat{\Phi}_y]\|$. Since $\|[\hat{\Phi}_x \ \hat{\Phi}_y]\| = 1$, \tilde{v}_z is defined as

$$\tilde{v}_z = \|[\dot{x} \ \dot{y}]\| \hat{\Phi}_z = v \hat{\Phi}_z. \quad (4.21)$$

The computation in (4.21) may still result in $|\tilde{v}_z| > v_{max}^z$. Then, the following saturation function is defined

$$v_z = \begin{cases} \tilde{v}_z, & \text{if } |\tilde{v}_z| \leq v_{max}^z, \\ v_{max}^z \tilde{v}_z |\tilde{v}_z|^{-1}, & \text{if } |\tilde{v}_z| > v_{max}^z. \end{cases} \quad (4.22)$$

Note that, assuming feasibility for the path \mathcal{P} , it is ensured that the saturation in (4.22) is inactive for $\mathbf{p} \in \mathcal{P}$. Also, a formal convergence proof is still necessary when the

saturation is considered. Nonetheless, in practice, the convergence is always reached. Finally, according to (4.1c) the height control is defined as

$$z_c = \tau_z v_z + z. \quad (4.23)$$

4.2.4 Disturbance analysis

If the uncertainty u_ω is considered and the control law (4.11) is assumed, the function \dot{V}_θ in (4.12) becomes

$$\dot{V}_\theta = -k_p \sin(\bar{\theta})^2 - \sin(\bar{\theta})u_\omega. \quad (4.24)$$

Assuming the maximum value $|u_\omega| = U_\omega$ and applying the triangle inequality in (4.24), the following is true

$$\dot{V}_\theta \leq |\sin(\bar{\theta})| (-k_p |\sin(\bar{\theta})| + U_\omega). \quad (4.25)$$

Consider $\gamma = \text{asin}(U_\omega k_p^{-1}) < \pi/2$. From (4.25) it is clear that, for an infinitesimal $\epsilon > 0$, we have $|\bar{\theta}| = \gamma + \epsilon \implies \dot{V}_\theta < 0$. Then, the following invariant set for $\bar{\theta}$ can be defined

$$\mathcal{I}_\theta = \{\bar{\theta} \in \mathbb{S}^1 \mid |\bar{\theta}| \leq \gamma\}. \quad (4.26)$$

Note that for \mathcal{I}_θ to be bounded it is necessary that $k_p > U_\omega$. Otherwise γ is not defined as a real number and $\mathcal{I}_\theta = \mathbb{S}^1$. Note also that if $\pi - |\bar{\theta}| < \gamma$, \dot{V}_θ can be positive, however, this condition is unstable. Formally, convergence can be ensured for every initial condition $|\bar{\theta}(0)| < \pi - \gamma$. Nonetheless, in practice convergence happens for all initial conditions.

Now, the influence of the uncertainty u_a in the velocity control will be considered. Assuming the control law (4.17), the time derivative of the function V_v in (4.19) becomes

$$\dot{V}_v = -k_v (v_r - v)^2 - (v_r - v) u_a. \quad (4.27)$$

Given the limit $|u_a| \leq U_a$, an upper bound for \dot{V}_v is

$$\dot{V}_v \leq |v_r - v| (-k_v |v_r - v| + U_a). \quad (4.28)$$

Consider $\mu = U_a k_v^{-1}$. From (4.28) it is clear that if $\bar{v} \geq \mu$ then $\dot{V}_v \leq 0$. Thus, in the presence of the uncertainty u_a , the following invariant set for the error \bar{v} can be

established

$$\mathcal{I}_v = \{\bar{v} \in \mathbb{R} \mid |\bar{v}| \leq \mu\}. \quad (4.29)$$

It is important to emphasize that it is interesting to have μ significantly smaller than v_r , that is $\mu \ll v_r$. From the definition of μ , the property $\mu \ll v_r$ can be obtained by selecting k_v such that $k_v \gg U_a v_r^{-1}$.

Now, let $\zeta \equiv U_z$ and $\bar{v}_z = v_z - \dot{z}$. Given (4.23) and (4.1c) the following invariant set can be directly inferred

$$\mathcal{I}_z = \{\bar{v}_z \in \mathbb{R} \mid |\bar{v}_z| \leq \zeta\}. \quad (4.30)$$

The sets \mathcal{I}_ψ , \mathcal{I}_v and \mathcal{I}_z define maximum absolute values, γ , μ and ζ , for the orientation error $\bar{\theta}$, the forward velocity error \bar{v} and the vertical velocity error \bar{v}_z , respectively. It is now necessary to define how these bounds reflect in the value of the field's potential function P . In other words, the objective is to verify if ultimate boundedness in $\bar{\theta}$, \bar{v} and \bar{v}_z implies in ultimate boundedness in the function P . For that, consider the following Lemma.

Lemma 4 *Let $w_1, w_2, w_3 \in \mathbb{R}^3$ vectors such that*

$$w_1^T w_3 = A, \quad (4.31a)$$

$$w_2^T w_3 \geq B, \quad (4.31b)$$

$$\|w_1\| = \|w_2\| = \|w_3\| = 1, \quad (4.31c)$$

in which $-1 \leq A \leq 0$, $0 \leq B \leq 1$ are constants. Then

$$w_1^T w_2 \leq AB + \sqrt{1 - A^2} \sqrt{1 - B^2}. \quad (4.32)$$

Proof: Let $r = w_1^T w_2$ and consider a matrix Q defined as $Q = w_1 w_1^T - w_2 w_2^T$. Regarding the Euclidean norm of $Q w_3$ the following is true

$$\|Q w_3\|^2 \leq \|Q\|^2 \|w_3\|^2 = \|Q\|^2. \quad (4.33)$$

Since $\|w_1\| = \|w_2\| = 1$, one can show that the eigenvalues of Q are 0, $\sqrt{1 - r^2}$ and $-\sqrt{1 - r^2}$. And since Q is symmetric its Euclidean norm is the greatest absolute value of its eigenvalues, thus, $\|Q\| = \sqrt{1 - r^2}$. The norm of $Q w_3$ can also be written as $\|Q w_3\|^2 = \|(w_1 w_1^T - w_2 w_2^T) w_3\|^2 = (w_1^T w_3)^2 + (w_2^T w_3)^2 - 2(w_1^T w_3)(w_2^T w_3)r$, and

from (4.33) one can write

$$(w_1^T w_3)^2 + (w_2^T w_3)^2 - 2(w_1^T w_3)(w_2^T w_3)r \leq 1 - r^2. \quad (4.34)$$

Using $A = w_1^T w_3$ and defining $C = w_2^T w_3$, (4.34) can be written as

$$r^2 - 2ACr + A^2 + C^2 - 1 \leq 0. \quad (4.35)$$

The solution of inequality (4.35) is

$$r \leq AC + \sqrt{1 - A^2}\sqrt{1 - C^2}. \quad (4.36)$$

Since $A \leq 0$ and $C \geq B \geq 0$ we conclude that $AC \leq AB$ and $\sqrt{1 - C^2} \leq \sqrt{1 - B^2}$. By applying these facts to (4.36) and recalling that $r = w_1^T w_2$ the result in (4.32) is finally proven. \blacksquare

Consider also the following Remark.

Remark 1 For functions $f_1(s)$, $f_2(s)$, $g_1(s)$ and $g_2(s) > 0$ and a set \mathcal{S} we have that:

$$\min_{s \in \mathcal{S}} (f_1(s) + f_2(s)) \geq \min_{s \in \mathcal{S}} f_1(s) + \min_{s \in \mathcal{S}} f_2(s). \quad (4.37)$$

$$\min_{s \in \mathcal{S}} \frac{g_1(s)}{g_2(s)} \geq \frac{\min_{s \in \mathcal{S}} g_1(s)}{\max_{s \in \mathcal{S}} g_2(s)}. \quad (4.38)$$

Let the vector Ψ be the desired velocity for the UAV in the ideal case. In this case $v = \|\dot{x} \ \dot{y}\| = v_r$, $\theta = \theta_f$ and the velocity in z is computed according to (4.21). This vector can be written as

$$\Psi = \begin{bmatrix} v_r \cos(\theta_f) \\ v_r \sin(\theta_f) \\ v_r \hat{\Phi}_z \end{bmatrix} = v_r \begin{bmatrix} \cos(\theta_f) \\ \sin(\theta_f) \\ \Phi_z (1 - \Phi_z^2)^{-0.5} \end{bmatrix} = \frac{v_r}{\sqrt{1 - \Phi_z^2}} \Phi. \quad (4.39)$$

Note that the scaling factor $\sqrt{\Phi_x^2 + \Phi_y^2} = \sqrt{1 - \Phi_z^2}$.

The vector $\dot{\mathbf{p}}$, which is the actual velocity of the UAV, can be written as the vector Ψ disturbed by the errors $\bar{\theta}$, \bar{v} and \bar{v}_z as below

$$\dot{\mathbf{p}} = \begin{bmatrix} (v_r + \bar{v}) \cos(\theta_f - \bar{\theta}) \\ (v_r + \bar{v}) \sin(\theta_f - \bar{\theta}) \\ (v_r + \bar{v}) \Phi_z (1 - \Phi_z^2)^{-0.5} + \bar{v}_z \end{bmatrix} = (v_r + \bar{v}) \begin{bmatrix} \cos(\theta_f - \bar{\theta}) \\ \sin(\theta_f - \bar{\theta}) \\ \Phi_z (1 - \Phi_z^2)^{-0.5} + \frac{\bar{v}_z}{v_r + \bar{v}} \end{bmatrix}. \quad (4.40)$$

The norms of the vectors Ψ and $\dot{\mathbf{p}}$ are

$$\begin{aligned}\|\Psi\| &= \frac{v_r}{\sqrt{1-\Phi_z^2}}, \\ \|\dot{\mathbf{p}}\| &= (v_r + \bar{v}) \sqrt{1 + \frac{\Phi_z^2}{1-\Phi_z^2} + 2 \frac{\Phi_z}{\sqrt{1-\Phi_z^2}} \frac{\bar{v}_z}{(v_r + \bar{v})} + \frac{\bar{v}_z^2}{(v_r + \bar{v})^2}} \\ &= (v_r + \bar{v}) \sqrt{\frac{1}{1-\Phi_z^2} + 2 \frac{\Phi_z}{\sqrt{1-\Phi_z^2}} \frac{\bar{v}_z}{(v_r + \bar{v})} + \frac{\bar{v}_z^2}{(v_r + \bar{v})^2}}.\end{aligned}\tag{4.41}$$

Now consider the following definitions

$$w_1 = \frac{\nabla P}{\|\nabla P\|}, \quad w_2 = \frac{\dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|}, \quad w_3 = \frac{\Psi}{\|\Psi\|} = \Phi\tag{4.42}$$

The function \dot{P} can be written as

$$\dot{P} = \nabla P^T \dot{\mathbf{p}} = \|\nabla P\| \|\dot{\mathbf{p}}\| \left(\frac{\nabla P}{\|\nabla P\|} \right)^T \frac{\dot{\mathbf{p}}}{\|\dot{\mathbf{p}}\|} = \|\nabla P\| \|\dot{\mathbf{p}}\| w_1^T w_2\tag{4.43}$$

The quantity $w_1^T w_3$ can be computed as

$$w_1^T w_3 = \frac{\nabla P^T \Psi}{\|\nabla P\| \|\Psi\|} = \frac{\nabla P^T \Phi}{\|\nabla P\|}.\tag{4.44}$$

Given Property 1, in Section 2.2.2, $w_1^T w_3$ in (4.44) can be written as

$$w_1^T w_3 = \frac{G(P) \|\nabla P\|}{\|\nabla P\|} = G(P) = A.\tag{4.45}$$

Note that since $G(P) \leq 0$, $A \leq 0$, and since $\|w_1\| = \|w_2\| = 1$, $-1 \leq A \leq 0$.

Now, given (4.39) and (4.40), the quantity $w_2^T w_3$ can be written as

$$\begin{aligned}w_2^T w_3 &= \frac{\dot{\mathbf{p}}^T \Psi}{\|\dot{\mathbf{p}}\| \|\Psi\|} \\ &= (v_r^2 + v_r \bar{v}) \frac{\cos(\bar{\theta}) + \Phi_z^2 (1 - \Phi_z^2)^{-1} + \Phi_z (1 - \Phi_z^2)^{-0.5} \frac{\bar{v}_z}{v_r + \bar{v}}}{\frac{v_r^2 + v_r \bar{v}}{\sqrt{1-\Phi_z^2}} \sqrt{\frac{1}{1-\Phi_z^2} + 2 \frac{\Phi_z}{\sqrt{1-\Phi_z^2}} \frac{\bar{v}_z}{(v_r + \bar{v})} + \frac{\bar{v}_z^2}{(v_r + \bar{v})^2}}} \\ &= \frac{\cos(\bar{\theta}) + \Phi_z^2 (1 - \Phi_z^2)^{-1} + \Phi_z (1 - \Phi_z^2)^{-0.5} \frac{\bar{v}_z}{v_r + \bar{v}}}{\frac{1}{\sqrt{1-\Phi_z^2}} \sqrt{\frac{1}{1-\Phi_z^2} + 2 \frac{\Phi_z}{\sqrt{1-\Phi_z^2}} \frac{\bar{v}_z}{(v_r + \bar{v})} + \frac{\bar{v}_z^2}{(v_r + \bar{v})^2}}}.\end{aligned}\tag{4.46}$$

Multiplying the numerator and the denominator of (4.46) by $(1 - \Phi_z^2)$ it is rewritten as

$$w_2^T w_3 = \frac{(1 - \Phi_z^2) \cos(\bar{\theta}) + \Phi_z^2 + \Phi_z \sqrt{1 - \Phi_z^2} \frac{\bar{v}_z}{v_r + \bar{v}}}{\sqrt{1 + 2\Phi_z \sqrt{1 - \Phi_z^2} \frac{\bar{v}_z}{(v_r + \bar{v})} + (1 - \Phi_z^2) \frac{\bar{v}_z^2}{(v_r + \bar{v})^2}}}. \quad (4.47)$$

In order to use Lemma 4 it is necessary to find a lower bound B for $w_2^T w_3$ in (4.47). For that, we will use the limits $|\bar{\theta}| \leq \gamma$, $|\bar{v}_z| \leq \zeta$, $\bar{v} \leq \mu$ and $|\Phi_z| \leq 1$ will be used to successively lower bound (4.46). Starting with the bound $|\bar{\theta}| \leq \gamma$ one can write

$$w_2^T w_3 \geq \frac{(1 - \Phi_z^2) \cos(\gamma) + \Phi_z^2 + \Phi_z \sqrt{1 - \Phi_z^2} \frac{\bar{v}_z}{v_r + \bar{v}}}{\sqrt{1 + 2\Phi_z \sqrt{1 - \Phi_z^2} \frac{\bar{v}_z}{(v_r + \bar{v})} + (1 - \Phi_z^2) \frac{\bar{v}_z^2}{(v_r + \bar{v})^2}}}. \quad (4.48)$$

In order to use the bound $|\Phi_z| \leq 1$, consider $s = \Phi_z$, $\mathcal{S} = \{s \mid |s| \leq 1\}$, $g_1(s)$ as the numerator of (4.48) and $g_2(s)$ as the denominator of (4.48). We will then use (4.38) to find a lower bound for (4.48). The minimum value of $g_1(s)$ is found by using (4.37) with $f_1(s) = (1 - s^2) \cos(\gamma) + s^2$ and $f_2(s) = s \sqrt{1 - s^2} \frac{\bar{v}_z}{v_r + \bar{v}}$. Then, the following is obtained

$$w_2^T w_3 \geq \frac{\cos(\gamma) - \frac{1}{2} \left| \frac{\bar{v}_z}{v_r + \bar{v}} \right|}{\sqrt{1 + \left| \frac{\bar{v}_z}{v_r + \bar{v}} \right| + \frac{\bar{v}_z^2}{(v_r + \bar{v})^2}}}. \quad (4.49)$$

Note that the extreme points of $s \sqrt{1 - s^2}$ occur at $s = \pm \frac{1}{2}$.

Now, considering the bound $|\bar{v}_z| \leq U_z = \zeta$, let $s = \bar{v}_z$ and $\mathcal{S} = \{s \mid |s| \leq \zeta\}$. Applying (4.38) with $g_1(s)$ and $g_2(s)$ as the numerator and denominator of (4.49), respectively, the following bound is obtained

$$w_2^T w_3 \geq \frac{\cos(\gamma) - \frac{1}{2} \frac{\zeta}{|v_r + \bar{v}|}}{\sqrt{1 + \frac{\zeta}{|v_r + \bar{v}|} + \frac{\zeta^2}{(v_r + \bar{v})^2}}}. \quad (4.50)$$

Finally, considering the bound $|\bar{v}| \leq \mu \leq v_r$, let $s = \bar{v}$ and $\mathcal{S} = \{s \mid |s| \leq \mu\}$. Once more, applying (4.38) with $g_1(s)$ and $g_2(s)$ as the numerator and denominator of (4.50), respectively, the following new bound is obtained

$$w_2^T w_3 \geq \frac{\cos(\gamma) - \frac{1}{2} \frac{\zeta}{v_r - \mu}}{\sqrt{1 + \frac{\zeta}{v_r - \mu} + \frac{\zeta^2}{(v_r - \mu)^2}}}. \quad (4.51)$$

The lower bound B is finally defined as the right hand side of (4.51) in the following way

$$B = \frac{(v_r - \mu) \cos(\gamma) - \frac{1}{2}\zeta}{\sqrt{(v_r - \mu)^2 + \zeta(v_r - \mu) + \zeta^2}}. \quad (4.52)$$

Now, the following theorem can be stated.

Theorem 3 *Let $\gamma \leq \pi/2$ and $\zeta \leq 2(v_r - \mu) \cos(\gamma)$. Then, there exists $\chi \geq 0$ such that $\cos(\chi) = B$ and the set*

$$\mathcal{I} = \{\mathbf{p} \in \mathbb{R}^3 \mid P(\mathbf{p}) \leq G^{-1}(-\sin(\chi))\} \quad (4.53)$$

is positively invariant.

Proof: First, from (4.45) $-1 \leq A \leq 0$. Note also that since $\zeta \leq 2(v_r - \mu) \cos(\gamma)$, which is a very gentle constraint in practical cases where $\zeta, \mu \ll v_r$, then $0 \leq B \leq 1$. Thus, there exists $\chi \geq 0$ such that $\cos(\chi) = B$. Now, let $D = \|\nabla P\| \|\dot{\mathbf{p}}\| > 0$. Then, using equations (4.43), (4.42), (4.45) and (4.52) it is possible to apply Lemma 4 and write

$$\dot{P} \leq D \left(AB + \sqrt{1 - A^2} \sqrt{1 - B^2} \right). \quad (4.54)$$

Having in mind that $A \leq 0$, the condition to have $\dot{P} \leq 0$ can be written as

$$-AB \geq \sqrt{1 - A^2} \sqrt{1 - B^2}. \quad (4.55)$$

Using $B = \cos(\chi)$ inequality in (4.55) can be written as

$$\frac{-A}{\sqrt{1 - A^2}} \geq \frac{\sqrt{1 - \cos(\chi)^2}}{\cos(\chi)} = \frac{\sin(\chi)}{\sqrt{1 - \sin(\chi)^2}}. \quad (4.56)$$

From (4.56) it is clear that $-A \geq \sin(\chi)$ implies $\dot{P} \leq 0$. Using the definition of A in (4.45) the condition is

$$-G(P) \geq \sin(\chi) \equiv G(P) \leq -\sin(\chi). \quad (4.57)$$

Now, since the function $G(P)$ is decreasing, G^{-1} exists, and applying G^{-1} to both sides of the second inequality of (4.57) the following sufficient condition on P is

obtained in order to have $\dot{P} \leq 0$

$$P \geq G^{-1}(-\sin(\chi)). \quad (4.58)$$

Finally, from (4.58) it is clear that if $\mathbf{p} \notin \mathcal{I}$ then $\dot{P} \leq 0$. Thus the set \mathcal{I} is positively invariant. This concludes the proof. ■

In conclusion, Theorem 3 states that when bounded uncertainties u_ω , u_a and u_z are present, the function P , which encodes the distance from the curve \mathcal{P} , admits an ultimate bound. Now, next Corollary formalizes the statement made in Subsection 4.2.1, that is, getting oriented with the field, in the absence of uncertainties, implies in convergence to the target curve.

Corollary 1 *In the absence of uncertainties, that is: $U_\omega=0$, $U_a=0$, $U_z=0$, the UAV converges asymptotically to the curve \mathcal{P} .*

Proof: Given the definitions of γ , μ and ζ , it occurs that $U_\omega = 0$, $U_a = 0$ and $U_z = 0$ imply $\gamma = 0$, $\mu = 0$ and $\zeta = 0$, respectively. Using these facts to compute B in (4.52) it is easy to obtain $B = 1$. Since $B = 1$, $\chi = 0$, $\sin(\chi) = 0$ and \mathcal{I} becomes

$$\mathcal{I} = \{\mathbf{p} \in \mathbb{R}^3 \mid P \leq G^{-1}(0)\} = \{\mathbf{p} \in \mathbb{R}^3 \mid P = 0\} = \mathcal{P}. \quad (4.59)$$

Since, in the absence of uncertainties, the invariant set \mathcal{I} collapses to the target curve \mathcal{P} , asymptotic convergence in the absence of uncertainties has been proven. ■

4.3 Field's Singularities

In order to compute the control law (4.11), it is necessary to perform operations, to compute gradients of field's components, over a normalized vector field. This normalization necessarily creates a discontinuity point in the interior of a planar closed curve, fact that is a direct consequence of the Brouwer fixed-point Theorem [Khamsi and Kirk, 2011]. The values of the field derivatives go to infinity as we approach the singularity point, which generates extremely high $\omega(t)$ commands, making it impossible to respect (4.15). In this section a strategy to handle this problem is presented. It is formally defined in \mathbb{R}^2 , but will be extended to \mathbb{R}^3 . Without loss of generality, let $\mathbf{p}_s = \mathbf{0}$ be the singularity point. Consider a ball of radius Ω around the singularity, that is, $\mathcal{B}_\Omega = \{\mathbf{p} \in \mathbb{R}^2 : \|\mathbf{p} - \mathbf{p}_s\| \leq \Omega\}$. The strategy consists in the establishment of $\omega(t) = 0$ inside the ball. In this way, while the UAV is in \mathcal{B}_Ω , it will continue on a rectilinear trajectory.

Let us define some variables based on Figure 4.1. The circle represents \mathcal{B}_Ω . Let $\beta \in \mathbb{S}^1$ be an angular coordinate to the border of the ball \mathcal{B}_Ω defined as $\bar{\mathcal{B}}_\Omega$. Consider also the function $\phi(\beta) : \mathbb{S}^1 \rightarrow \mathbb{S}^1$ that associates to each local β in $\bar{\mathcal{B}}_\Omega$ a value $\phi(\beta) \in \mathbb{S}^1$. This function is equivalent to the orientation θ_f at $\bar{\mathcal{B}}_\Omega$. Defining $\mathbf{p}_\beta \in \bar{\mathcal{B}}_\Omega$ such that $\mathbf{p}_\beta = [\Omega \cos(\beta), \Omega \sin(\beta)]^T$, the function $\phi(\beta)$ is formally defined as:

$$\phi(\beta) = \text{atan2}(\Phi_y(\mathbf{p}_\beta), \Phi_x(\mathbf{p}_\beta)). \quad (4.60)$$

A function $\delta(\beta) : \mathbb{S}^1 \rightarrow \mathbb{S}^1$, in which $\delta(\beta) = \phi(\beta) - \beta$, is also defined. Let $\beta_1, \beta_2 \in \mathbb{S}^1$ be coordinates relative to the intersections of a straight line path of the UAV with $\bar{\mathcal{B}}_\Omega$. This path is parallel to the vector $[\cos(\theta) \ \sin(\theta)]^T$, and the intersections occur at instants t_1 and $t_2 > t_1$.

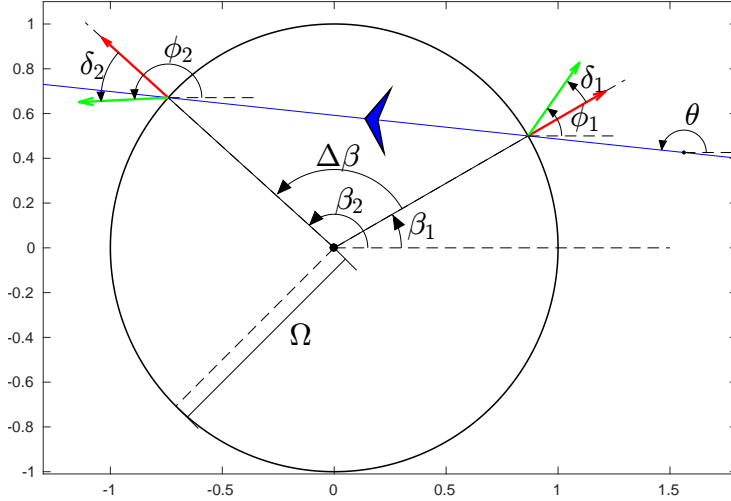


Figure 4.1. Definitions for Theorem 4. In blue is the straight line with orientation θ . In red are vectors with orientations β_1 and β_2 . In green are vectors with orientations $\phi(\beta_1)$ and $\phi(\beta_2)$.

Evidently, it is not possible to ensure that the Lyapunov function V_θ in (4.6) will decrease while the UAV is inside \mathcal{B}_Ω . However, Theorem 4 establishes sufficient conditions to ensure that the value of the Lyapunov function in the instant t_2 is smaller than its value in the instant t_1 .

Theorem 4 Consider that $V_\theta(t_i)$ is the evaluation of the function $1 - \cos(\phi(\beta_i) - \theta)$ at the intersections of the straight line with $\bar{\mathcal{B}}_\Omega$ at β_i , $i = 1, 2$. If

$$\frac{d^* \phi}{d\beta}(\beta) > 0, \quad \forall \beta \in \mathbb{S}^1, \quad (4.61a)$$

$$\cos(\delta(\beta)) > 0, \quad \forall \beta \in \mathbb{S}^1, \quad (4.61b)$$

then

$$V_\theta(t_2) < V_\theta(t_1), \quad \forall \beta_1, \beta_2 \in \mathbb{S}^1, \quad (4.62)$$

in which the derivative in (4.61a) is defined as

$$\frac{d^*\phi}{d\beta}(\beta) = \lim_{\Delta\beta \rightarrow 0} \frac{\sin(\phi(\beta + \Delta\beta) - \phi(\beta))}{\Delta\beta}. \quad (4.63)$$

Proof: First, it is important to mention that the definition in (4.63) is only a formalization of the derivative of $\phi(\beta)$ with respect to β . Remember that $\sin(x) \rightarrow x$ as $x \rightarrow 0$.

Now, let ΔV_θ be the variation function defined as $\Delta V_\theta = V_\theta(t_2) - V_\theta(t_1)$, in other words, how much is the increase of V_θ . Thus $\Delta V_\theta = \cos(\phi(\beta_1) - \theta) - \cos(\phi(\beta_2) - \theta)$. Without loss of generality consider that the angle β_2 is written as

$$\beta_2 = \beta_1 + \Delta\beta = \beta_1 + \int_{\beta_1}^{\beta_1 + \Delta\beta} d\beta, \quad (4.64)$$

in which $\Delta\beta \in [0, 2\pi]$ is the angle that goes from β_1 to β_2 in the counter clockwise direction. The integral definition in (4.64) is made to avoid displacements of more than 2π between β_1 and β_2 . To simplify the notation consider $\phi_i = \phi(\beta_i)$ and $\delta_i = \delta(\beta_i)$, $i = 1, 2$. Using the trigonometric property $\cos(A) - \cos(B) = -2\sin((A-B)/2)\sin((A+B)/2)$, the condition $\Delta V_\theta < 0$ can be written as

$$\Delta V_\theta = -2\sin\left(\frac{\phi_1 - \phi_2}{2}\right)\sin\left(\frac{\phi_1 + \phi_2 - 2\theta}{2}\right) < 0, \quad (4.65)$$

which after using $\phi_1 = \delta_1 + \beta_1$ and $\phi_2 = \delta_2 + \beta_2 = \delta_2 + \beta_1 + \Delta\beta$ becomes

$$\Delta V_\theta = 2\sin\left(\frac{\phi_2 - \phi_1}{2}\right)\sin\left(\frac{\delta_1 + \delta_2 + 2\beta_1 + \Delta\beta - 2\theta}{2}\right) < 0, \quad (4.66)$$

Using the fact that $\Delta\beta \in [0, 2\pi]$, according to (4.64), “ $\beta_2 > \beta_1$ informally”, a geometric analysis in Figure 4.1 allows the inference of

$$2\beta_1 + \Delta\beta - 2\theta = -\pi. \quad (4.67)$$

Using this relation in (4.66) and using $\sin(A - \pi/2) = -\cos(A)$, the condition for

having $\Delta V_\theta < 0$ can be written as

$$\sin\left(\frac{\phi_2 - \phi_1}{2}\right) \cos\left(\frac{\delta_1 + \delta_2}{2}\right) > 0, \quad (4.68)$$

Now it will be shown that (4.61) implies the veracity of (4.68). The condition (4.61b) implies that $-\pi/2 < \delta_1, \delta_2 < \pi/2$. Thus the cosine in (4.68) is positive. Now it is necessary to show that the sine is also positive. Rewriting ϕ_2 as an integral from β_1 to β_2 , the sine in (4.68) can be written as

$$\sin\left(\frac{1}{2} \int_{\beta_1}^{\beta_1 + \Delta\beta} \frac{d^*\phi}{d\beta}(\beta') d\beta'\right) > 0, \quad (4.69)$$

If $\Delta\beta = 0$, the integral in 4.69 is 0. Since the function $\phi(\beta)$ is continuous, the value of the integral at $\Delta\beta = 2\pi$ must be an integer multiple of 2π . However, if the value of the integral is any multiple different from 2π itself, condition (4.61b) would be violated. Thus, if $\Delta\beta = 2\pi$ the value of the integral is also 2π . The condition (4.61a) ensures that the integral is a crescent function of $\Delta\beta$. Thus, the integral only assume values in between 0 and 2π and, consequently, the argument of the sine function lies between 0 and π . Evidently, the sine is positive, which concludes the proof. \blacksquare

Theorem 4 is directly applicable only for fields in \mathbb{R}^2 space. For fields in \mathbb{R}^3 , it is common that the normalized and projected fields admit singularities at the z axis. In this case, $\omega = 0$ and $v_z = 0$ can be set inside a cylinder of radius Ω . In this case, (4.62) can be ensured if (4.61), computed from the normalized and projected field, is respected for all z .

A more generic formal strategy for dealing with singularities in \mathbb{R}^3 that are not only on the z axis, or in \mathbb{R}^2 that are not a single point, is still needed.

This treatment of field's singularities closes this chapter. The control strategy developed here can be directly used to control multiple UAVs, each one in a specific prespecified path, as in the scenario addressed in Chapter 3.

Chapter 5

Results

In this chapter the results of the developed coordination and control strategies will be presented. The results that allow the analysis of the multi-robot coordination strategy will be shown first, and the results associated to the fixed-wing UAV control strategy are presented in the sequence. At the end, results that combine the two strategies will be presented.

An additional video with more results and some animations is available at <https://www.youtube.com/watch?v=UNPZ6HSMwqI&feature=youtu.be>.

5.1 Coordination strategy

This section focus on the presentation of results relative to the coordination strategy developed in Chapter 3.

5.1.1 Simulated experiments

In order to illustrate the coordination approach an example with eight robots was solved. At this point, instead of considered fixed-wing UAVs, differential drive robots are going to be used in the simulation. The considered scenario is illustrated in Fig. 5.1, where the position of the target points derived from the solution are already shown together with the corresponding regions of uncertainty. The robots 1, 2, 3, 4, 5, 6, 7 and 8 travel, in the counterclockwise direction, along the curves in blue, red, green, magenta, orange, dark green, clear blue and brown, respectively. The map is $16\text{m} \times 8\text{m}$.

Note that there are 14 collision zones. The 2 regions in the right part of the image were actually 4 according to mathematical definition in Section 3.2. However,

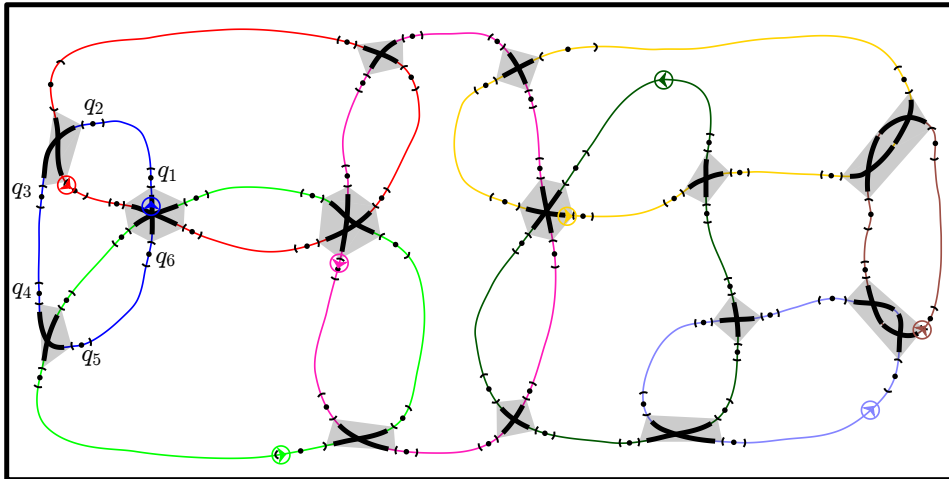


Figure 5.1. Scenario used for a simulation of the coordination strategy. Collision stretches are the ones in thick black. Inflated collision zones, represented by the gray polygons, are composed of the segments therein. Target points are in black and the regions of uncertainty are between the black curved traces. Robots are represented by arrows indicating their motion direction and circles indicating their sizes.

in order to avoid very short stretches, which could lead to unfeasibility according to Section 3.4.2, they were merged.

The cycle time multipliers were defined as $\lambda(1) = 1$, $\lambda(2) = 2$, $\lambda(3) = 2$, $\lambda(4) = 2$, $\lambda(5) = 2$, $\lambda(6) = 2$, $\lambda(7) = 1$, $\lambda(8) = 1$, according to the lengths of the paths, and the approach in Section 3.4.3 was used with $w(q_1, q_2) = 2 \forall q_1, q_2$. The uncertainties considered were the velocity proportional uncertainty of $U_{\%}^i = 0.07 \forall i$ and the position uncertainty of $U_p^i = 0.05m \forall i$. All the robots were assumed to be circles with 0.30m of diameter, thus $\rho_{ij} = 0.3$ was set for all pairs $(i, j) \in \mathcal{R}^2$. The velocities of the robots were limited between 0.08m/s and 0.30m/s and $\tau_i = 1.5s, \forall i$ was chosen. The objective function was $\mathbf{max} : \Delta s$. Using the solver `lp_solve` [Berkelaar et al., 2004], the MILP problem itself was solved in an Intel CORE™i7 processor with a 2.6GHz clock in 28.2ms average. The solution provided a basic cycle time of $C_0 = 45.90s$ and an enlargement $\Delta s = 0.129m$. Tab. 5.1 shows the results found for robot 1 with the average speeds computed based on a null position error. The units are s, m/s and m. The points q are indicated in Fig. 5.1.

The Robot Operating System, ROS [Quigley et al., 2009], was used to command the robots in a simulation using Gazebo [Koenig and Howard, 2004] with eight iCreate differential drive robot models. In order to make the robots follow the desired curves the vector field based technique [Gonçalves et al., 2010] was used together with feedback linearization [Yun and Yamamoto, 1992]. In order to simulate uncertainty, a random velocity error of 4% was introduced every instant $t[q, k]$. The experiment lasted 240

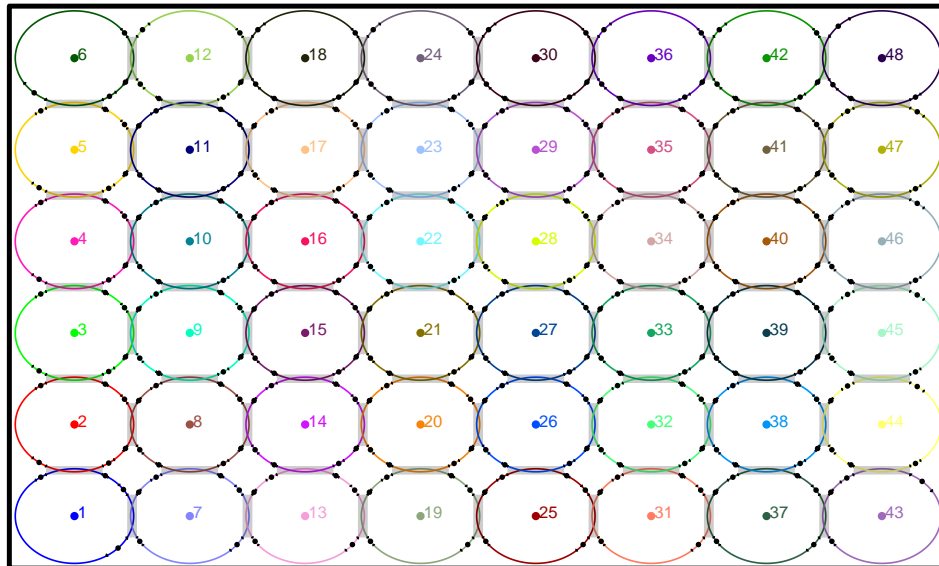
Table 5.1. Results for robot 1 in the simulation of Figure 5.1

Point	$t[q, 0]$	$T[q, 0]$	$\mathcal{L}[q]$	$\bar{V}[q, 0]$	$r[q]$
q_1	0.00	6.78	1.66	0.245	0.16
q_2	6.78	12.50	1.70	0.136	0.18
q_3	19.28	6.77	1.63	0.241	0.19
q_4	26.05	6.11	1.44	0.235	0.18
q_5	32.16	8.03	2.03	0.252	0.17
q_6	40.19	5.71	1.29	0.226	0.21

minutes and the robots were kept synchronized. This time is equivalent to around 312 laps for the robots in the shorter paths, $\lambda(i)=1$, and 156 laps for the robots in the longer paths, $\lambda(i)=2$.

5.1.2 Timing analysis

In order to analyze the performance of the coordination strategy, the scenario in Fig. 5.2 was considered. In this figure there are 48 paths filling a rectangular area. The number of each path was placed near its center. The algorithms were applied to different numbers of robots in this scenario, varying from 2 up to 48, considering the order indicated by the numbers.

**Figure 5.2.** Scenario used for the timing analysis.

The following four different parts of the strategy were analyzed:

- Scenario time: consider the time spent for computing a polynomial interpolation with 16 points; finding a representation of the curve with 1000 points; computing the

points in the collision set \mathcal{C} ; partitioning \mathcal{C} in the collision zones; and computing the lengths $L(q)$.

- Matrices time: consider the time for creating the matrices representing the constraints for the MILP formulation.
- MILP time: time spent by the functions *lp_maker* and *mxlpsolve*, from *lp_solve* library [Berkelaar et al., 2004].
- Placement time: time spent to find the position of the target points given the solution of the MILP.

In Fig. 5.3 the execution times described above are presented for different numbers of robots, from 2 to 48. The graph corresponds to an average of 50 executions for each number of robots.

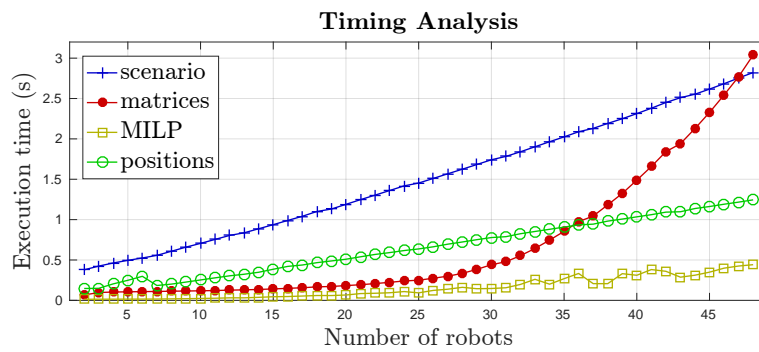


Figure 5.3. Execution time of the algorithms. Scenario of Fig. 5.2

In Fig. 5.4 the size of the problem, regarding number of constraints and number of variables, is presented. The variables were separated in real and binary ones, while the constraints in equality and inequality ones. Note that the number of binary variables is 246 for 48 robots. It is clear that the evolution of the number of constraints and variables has a linear pattern with respect to the number of robots.

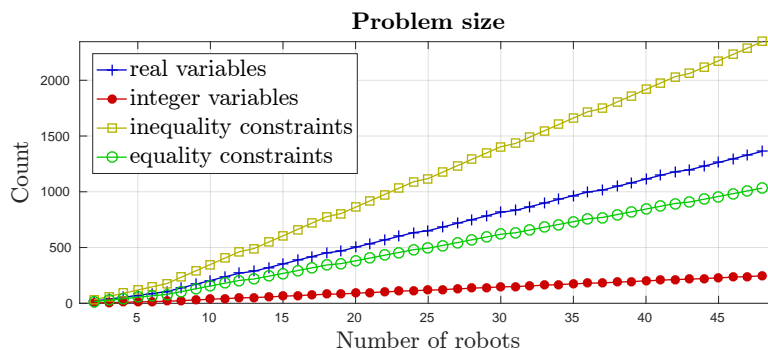


Figure 5.4. Number of variables and constraints. Scenario of Fig. 5.2

In Fig. 5.5 a box plot of the total time of the coordination strategy is presented. For each number of robots, 50 executions were performed. Note that the variance of the data is very low, which is expected, since the method is deterministic.

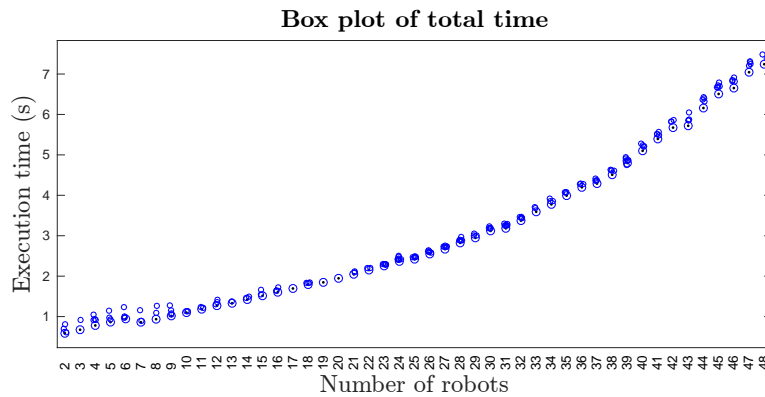


Figure 5.5. Box plot of the total time of the coordination strategy applied to the scenario of Fig. 5.2.

It is important to mention that, specifically for the simple scenario of Figure 5.2, there is a straightforward solution, which is all robots executing the same trajectory with respect to their respective ellipses. However, the objective of the scenario is to perform a timing analysis only.

5.2 Vector field control validation

This section focuses on the presentation of results relative to the control strategy developed in Chapter 4.

5.2.1 Simulated experiments

To illustrate the operation of the developed controllers, simulations were performed. The complete model of a UAV presented in Chapter 2 of [Stevens et al., 2015] with the parameters of the Aerosonde UAV in Appendix E.2 of [Beard and McLain, 2012] were considered. The aircraft model has 6 degrees of freedom, 12 states and atmospheric disturbances are incorporated. Random additive disturbances (measurement and actuation) were added in order to validate the robustness of the approach. The low level controllers from [Olavo et al., 2018] were used. The results will be shown separately for 2 and 3 dimensions.

Regarding the model in (4.1), based on [Olavo et al., 2018], the following parameters were considered for the UAV: $v_{min} = 18m/s$, $v_{max} = 28m/s$, $v_{max}^z = 3m/s$, $\tau_z = 20s$, $\tau_\theta = 28s$ and $\tau_v = 20s$.

The developed controller was used to make a UAV traverse a curve defined by $\alpha(x, y) = ax^4 + bx^2y^2 + cy^4 - 1 = 0$. The values of the parameters were $a = 6$, $b = 0$,

$c = 18$, for x and y in km. The following functions were also defined: $P = (1/2)\alpha^2$, and $G = -(2/\pi)\text{atan}(2\sqrt{P})$. The height was set to 200m . The disturbances on the measured θ and actuated θ_c were band limited values up to 0.56rad . In order to account for these disturbances and the model error, $U_\theta = 0.06\text{rad/s}$ was set. Then, $k_p = 0.18$ and $k_v = 0.15$ were also set. U_v was set to 0.3m/s . The radius of the ball \mathcal{B}_Ω was set to $\Omega = 0.2\text{km}$ and was enough to validate the condition of Theorem 4 and to validate the restriction on (4.15).

In Figure 5.6 it is possible to see the normalized vector field and also the trajectory executed by the UAV with initial condition $[x \ y]^T = [-0.3 \ 0]^T\text{km}$ and $\theta = \pi/6$. Note that the aircraft passed through \mathcal{B}_Ω and then converged to the path in black. The invariant set \mathcal{I} , defined in Section 4.2.4 is the region between the blue curves.

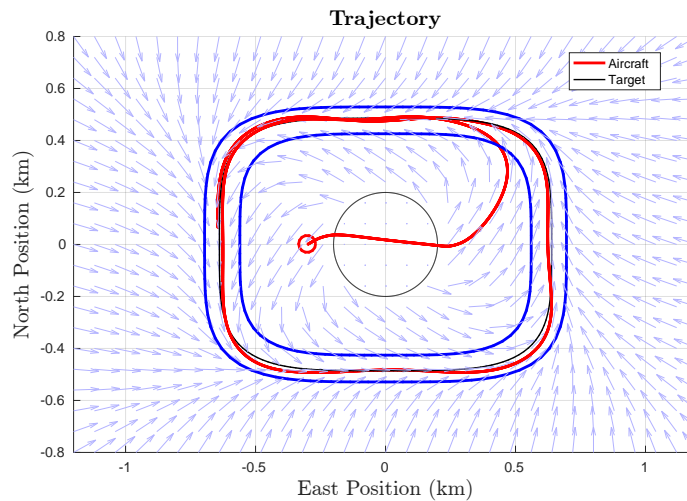


Figure 5.6. Numerical simulation. The UAV's trajectory, in red, converges to a generic path and remains inside the bounds, represented by the blue curves.

In Figure 5.7 it is possible to see the decay of the Lyapunov function V_θ . The blue/yellow dashed trace is related to the time the UAV was traversing \mathcal{B}_Ω . At this time V_θ increases, but the overall effect is $\Delta V_\theta < 0$. In the bottom is the field's original potential function P , which increases at the beginning when the UAV is not oriented, but, as V_θ decays, P also decays, in agreement with the analysis in Subsection 4.2.4. The dashed red lines in Figure 5.7 are the corresponding ultimate bounds.

In Figure 5.8 the evolution of, from top to bottom, height, forward speed and bank angle can be seen. The height stabilized at 200m . The speed followed a trapezoidal reference respecting the ultimate bounds shown in red. Despite not considered here, the low level controllers were able to track the desired $\omega(t)$ without reaching the safe limits for the bank angle.

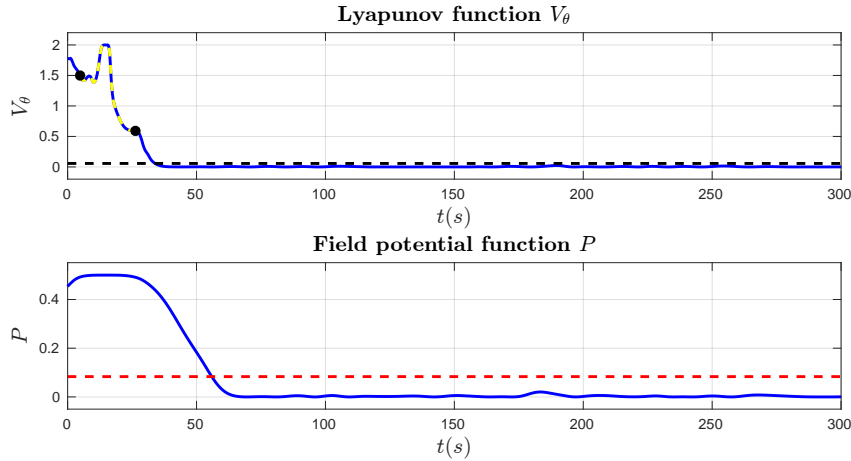


Figure 5.7. In the top is the Lyapunov function V_θ . The dashed blue/yellow trace is associated with the time the UAV was inside \mathcal{B}_Ω . In the bottom is the field potential function P . The dashed lines are the corresponding ultimate bounds.

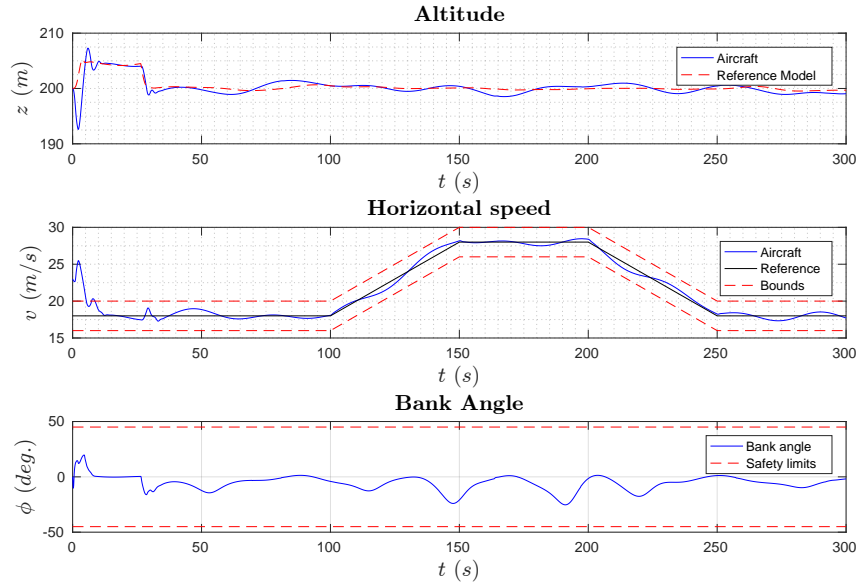


Figure 5.8. Performance of the UAV in the simulation of Figure 5.6. From top to bottom: height z ; ground speed $v(t)$; and bank angle.

For the 3 dimensional simulations, the basic functions $\alpha_1 = z + c(a^{-2}x^2 - 1)$ and $\alpha_2 = a^{-2}x^2 + b^{-2}y^2 - 1$ were considered. The values of the parameters were $a = 600\text{m}$, $b = 400\text{m}$ and $c = 50\text{m}$. Note that the value of c is limited given v_{max}^z . Then $P = (1/2)10^{-5}\alpha_1^2 + (1/2)\alpha_2^2$, and $G = -(2/\pi) \text{atan}(3\sqrt{P})$ were chosen. The curve defined by α_1 and α_2 was shifted to an altitude of 250m . The same disturbances on θ and θ_c were applied. The limits $U_\theta = 0.06\text{rad/s}$, $U_z = 0.3\text{m/s}$ and $U_v = 0.3\text{m/s}$ were considered. The controller gains were $k_p = 0.2$ and $k_v = 0.1$. The singularity was treated by considering a cylinder with radius $\Omega = 0.2\text{km}$. Figure 5.9 shows the trajectory executed by the simulated UAV with initial condition $[x \ y \ z]^T = [0.1 \ 0.1 \ 0.05]^T\text{km}$ and $\theta = 0$.

The set point speed was $v_r = 23\text{m/s}$, constant. The invariant set \mathcal{I} corresponds to the region inside the tube in transparent blue.

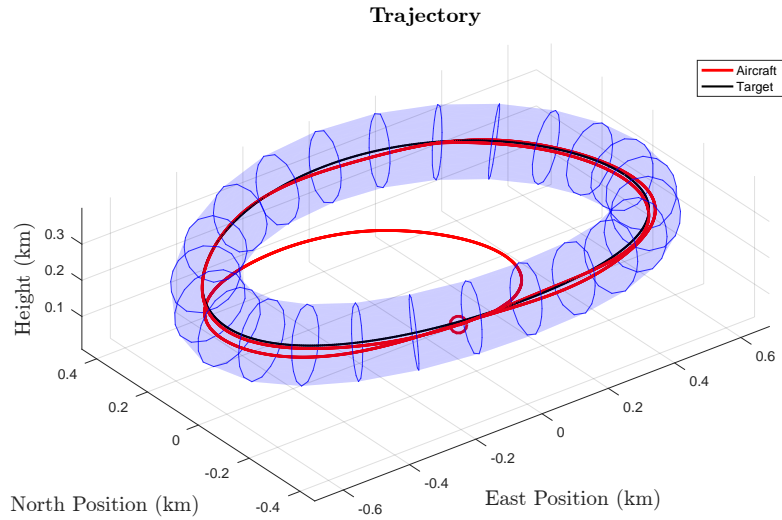


Figure 5.9. Simulation considering a “saddle like” path in \mathbb{R}^3 . In red is the trajectory followed by the UAV, which converges to the blue tube.

Figure 5.10 shows the decay of V_θ in the top and of P in the bottom. Note again that, when the UAV was inside a cylinder of radius Ω , the function V_θ also decreased.

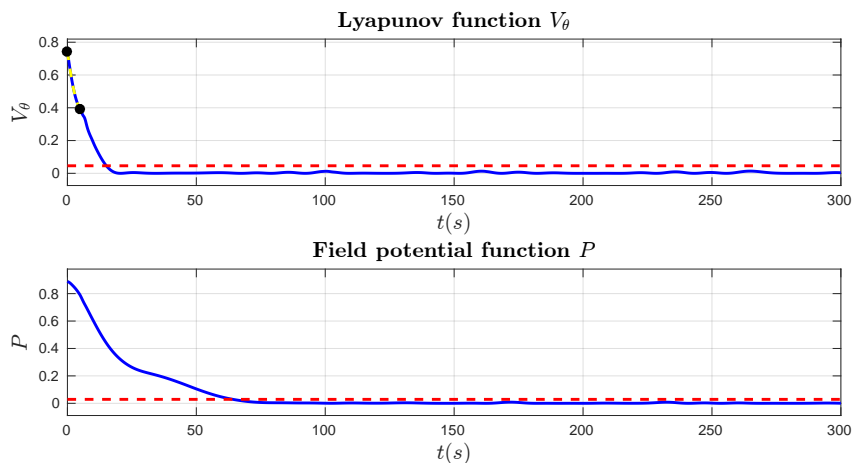


Figure 5.10. Functions V_θ and P associated to the 3 dimensional simulation.

Figure 5.11 shows the evolution of z and v and bank angle. The height z is varying since the curve is not planar. Note the actuation of the saturation function in the signal for v_z that happens when the UAV is way below the curve, making it climb with a constant rate. It is possible to observe that the velocity v increases when the UAV is losing height due to the coupling in the model used for simulation. The bank angle was kept in the safety limits.

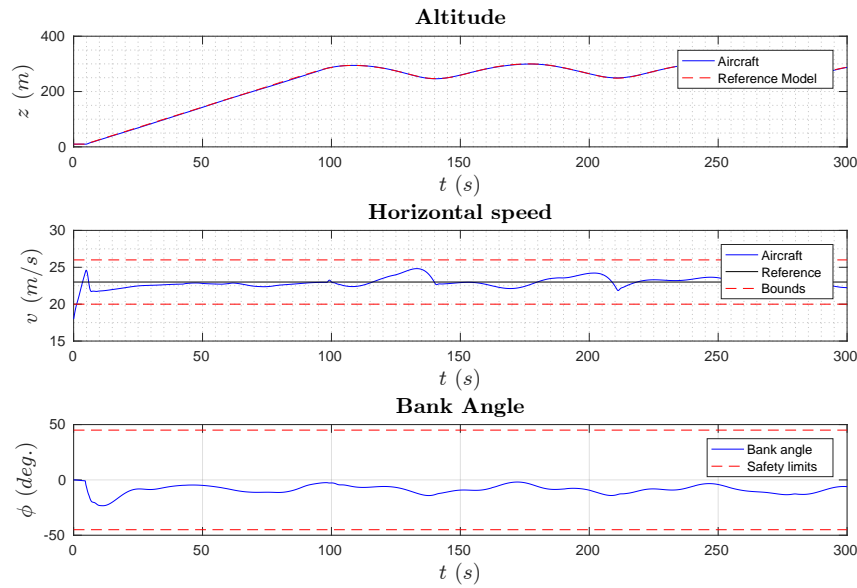


Figure 5.11. Performance of the UAV in the simulation of Figure 5.9.

In Figure 5.12 the trajectories of a UAV following a circle with radius 500m are shown for different initial conditions. In this case, no additional disturbance was added in the simulation. The initial conditions were the same as the ones in Figure 2.9, when the control strategy in [Olavo et al., 2018] was used. It is possible to see that, by using the current approach, the UAV reaches the target circle more rapidly. In other words, the distance traveled until UAV reaches the curve and circulates it is shorter.

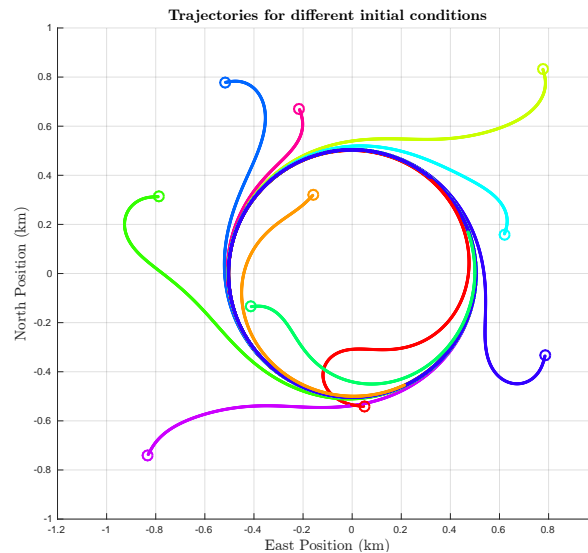


Figure 5.12. Results obtained according to the control strategy presented in this work. The initial conditions are the same as in Figure 2.9.

5.3 Coordination and control

This section presents results of the multi-robot coordination strategy of Chapter 3, in which the control strategy of Chapter 4 was used to control the robots.

5.3.1 Multiple airplanes simulation

The scenario illustrated in Figure 5.13 was considered in a simulation, on Simulink, with the whole closed loop system, i.e. the complete model of the UAV and the low level PID controllers. The UAVs 1, 2 and 3 are in blue, red and green, respectively. Each elliptical path was described by the following equations

$$\alpha_1 = z - 200,$$

$$\alpha_2 = \frac{[\cos(d)(x - c_x) + \sin(d)(y - c_y)]^2}{a^2} + \frac{[\sin(d)(x - c_x) - \cos(d)(y - c_y)]^2}{b^2} - 1.$$

The surface $\alpha_1 = 0$ is a horizontal plane at the height $z = 200\text{m}$ and the surface $\alpha_2 = 0$ is an elliptical cylinder with center at $[c_x, c_y]^T$ with semi axis a and b rotated by an angle d .

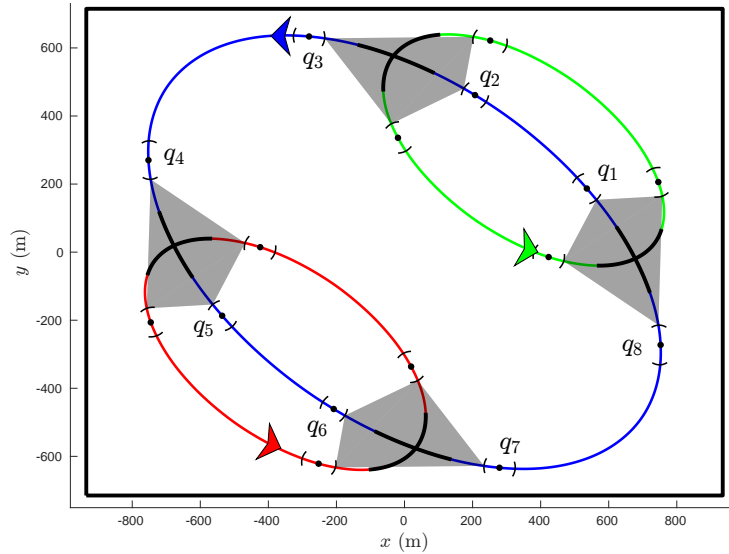


Figure 5.13. Scenario of the simulation with UAVs' realistic model. The inflated collision zones, target points and regions of uncertainty are placed as in Fig. 5.1.

The controller developed in Chapter 4 was used with the functions $P = (1/2)10^{-5}\alpha_1^2 + (1/2)\alpha_2^2$, and $G = -(2/\pi)\text{atan}(4\sqrt{P})$. The considered gains were $k_p = 0.2$ and $k_v = 0.1$. Given the sizes of the ultimate bounds found numerically according to equation (4.53), and pictured in Figure 5.9, the safe distance ρ_{ij} was defined according to $\rho_{ij} = 100\text{m} \forall (i, j) \in \mathcal{R}^2$.

We also considered $\lambda(1) = 2$, $\lambda(2) = 1$, $\lambda(3) = 1$ and, again, $w(q_1, q_2) = 2 \forall q_1, q_2$. The choices of the λ 's were due to the length of the curves, which are 4.31km, 2.32km and 2.32km for the blue, red and green paths, respectively. The uncertainties considered were the velocity uncertainty of $U_v^i = 1.5\text{m/s} \forall i$ and the position uncertainty of $U_p^i = 10\text{m} \forall i$. The velocities of the robots were limited between 18m/s and 28m/s. Given the limits of acceleration $\mathbf{a}_{max} = -\mathbf{a}_{min} = 1.25\text{m/s}^2$ the constant τ_i was defined according to equation (3.30) such that $\tau_i = 8s \forall i$. The objective function was $\mathbf{max} : \Delta s$. The solution provided a basic cycle time of $C_0 = 96.71s$ and an extra enlargement of the collision zones of $\Delta s = 98.02\text{m}$. Tab. 5.2 shows the results found for robot 1 with the average speeds computed based on a null position error. The points q are indicated in Fig. 5.13.

Table 5.2. Results for robot 1 in the experiment of Figure 5.13

Point	$t[q, 0]$ (s)	$T[q, 0]$ (s)	$\mathcal{L}[q]$ (m)	$\bar{V}[q, 0]$ (m/s)	$r[q]$ (m)
q_1	0.00	18.70	430.1	23.0	43.5
q_2	18.70	24.06	521.1	21.7	38.1
q_3	42.76	31.12	690.7	22.2	46.1
q_4	73.88	22.19	514.5	23.2	56.7
q_5	96.07	18.66	429.1	23.0	43.3
q_6	114.72	24.04	520.8	21.7	38.0
q_7	138.77	32.32	689.6	21.3	46.1
q_8	171.09	22.34	516.9	23.1	58.5

Figure 5.14 represents the block diagram of the developed controllers. In the top right is the guidance vector field controller developed in Chapter 4. It receives a reference velocity given by the block in the bottom center. This block receives the plan given by the MILP solution and generates reference velocities to the UAV in order to keep it synchronized. The block in the left inserts uncertainties in the measurements.

The experiment lasted 15 minutes, or 900 seconds, and the airplanes were kept synchronized all the time. This time corresponds to more than 9 laps for the UAVs in the shorter paths (red and green) and more than 4 laps for the UAV in the longer path (blue). The normalized position errors ($e[q, k]/r[q]$) of the three robots are in the histograms of Fig. 5.15. Note that there is no occurrence of a normalized error with absolute value bigger than the unit, meaning $|e[q, k]| < r[q]$. It shows that the UAVs were always inside the correct regions of uncertainty in the required instants.

Figure 5.16 shows in the top the reference velocity computed for the UAV 1 (blue one) and the velocity that it executed. In the bottom the velocity error is shown.

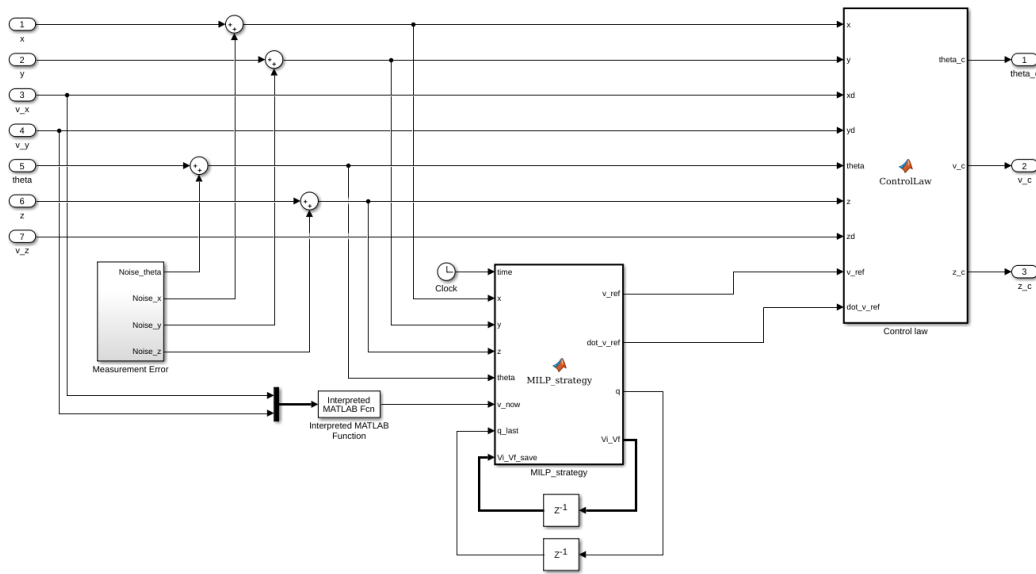


Figure 5.14. Block diagram in Simulink representing the controllers developed in this work.

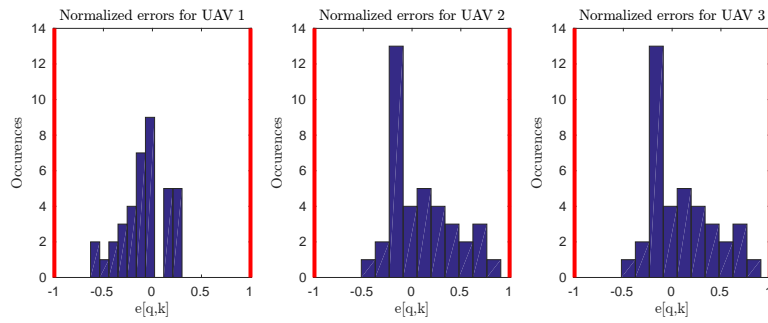


Figure 5.15. Histogram of the normalized position errors at the instants $t[q, k]$ for robots 1, 2 and 3, in the Simulink simulation.

Note that the speed profile is not perfectly periodic due to the online computation of reference speeds in order to preserve synchronism.

5.3.2 Multiple real robots experiment

A problem for a scenario with three robots was also solved and implemented by using real e-puck robots [Mondada et al., 2009]. These differentially driven robots are not, of course, fixed-wing UAVs, however they can be represented by a very similar model, given below

$$\dot{x} = v_c \cos(\theta), \quad (5.1a)$$

$$\dot{y} = v_c \sin(\theta), \quad (5.1b)$$

$$\dot{\theta} = \omega, \quad (5.1c)$$

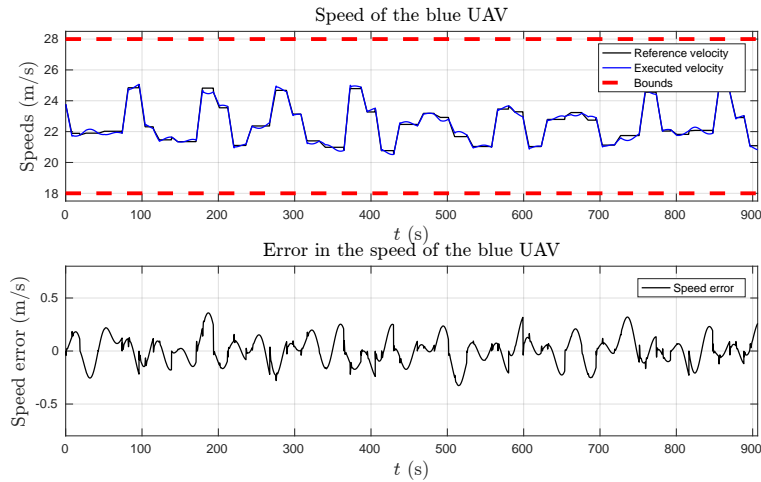


Figure 5.16. In the top are the reference velocity and performed velocity of the blue UAV in the Simulink simulation. In the bottom is the error between the velocities.

in which x , y and θ are the already known Cartesian coordinates and yaw angle. For these robots it is reasonable to assume that the forward velocity and the turning rate are input commands. Thus, in model (5.1), v_c and ω represents the inputs of the model. The control law for the forward velocity becomes $v_c = v_r$ and for the turning ratio is given by equation (4.11) directly.

The Robot Operating System was used to command the robots. A computer vision system was used to detect the position and orientation of the robots, which had markers on their top. The position, orientation and velocity of the robots used in the computations were obtained by using these data into an Extended Kalman Filter [Thrun et al., 2005]. The heading controller gain was set to $k_p = 1$. Fig. 5.17 is a frame of the camera used in the vision system with the drawing of overlaid path curves. The inflated collision zones, the target points and the regions of uncertainty are drawn according to the solution of the MILP. The robots 1, 2 and 3 travel, in counterclockwise direction, the curves in blue, red and green, respectively.

We considered $\lambda(1) = 2$, $\lambda(2) = 1$, $\lambda(3) = 1$ and, again, $w(q_1, q_2) = 2 \forall q_1, q_2$. The uncertainties considered were the velocity proportional uncertainty of $U_{v\%}^i = 0.10 \forall i$ and the position uncertainty of $U_p^i = 0.03m \forall i$. The robots were assumed to be squares with 0.075m of side length and the safe distance ρ_{ij} was defined as their diagonal, i.e., $\rho = 0.106 \forall (i, j) \in \mathcal{R}^2$. The velocities of the robots were limited between 0.05m/s and 0.13m/s, and τ_i was defined as $\tau_i = 1.2s, \forall i$. The objective function was $\mathbf{max} : \Delta s$. Using the same solver, the problem was solved in 17ms average. The solution provided a basic cycle time of $C_0 = 24.31s$ and an extra enlargement of the collision zones of $\Delta s = 0.040m$. Tab. 5.3 shows the results found for robot 1 with the average speeds computed based on a null position error. Units in s, m/s and m. The points q are

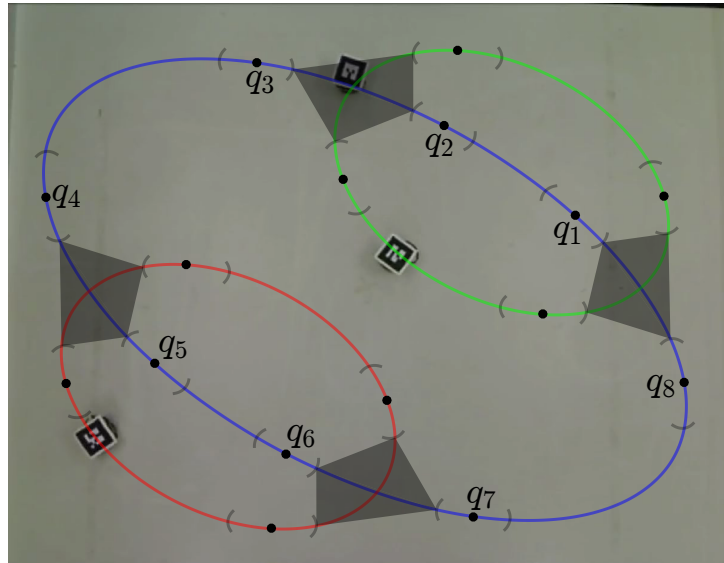


Figure 5.17. Scenario of the actual robot experiment post printed on a frame of the camera system.

indicated in Fig. 5.17.

Table 5.3. Results for robot 1 in the experiment of Figure 5.17

Point	$t[q, 0]$ (s)	$T[q, 0]$ (s)	$\mathcal{L}[q]$ (m)	$\bar{V}[q, 0]$ (m/s)	$r[q]$ (m)
q_1	0.00	4.33	0.39	0.090	0.09
q_2	4.33	4.89	0.48	0.097	0.08
q_3	9.22	9.77	0.73	0.074	0.09
q_4	18.99	5.32	0.50	0.094	0.11
q_5	24.31	4.31	0.39	0.090	0.09
q_6	28.62	6.35	0.48	0.075	0.08
q_7	34.97	8.33	0.73	0.087	0.09
q_8	43.30	5.32	0.50	0.094	0.11

The experiment lasted 25 minutes and the robots were kept synchronized all the time. This time corresponds to 62 laps for the robots in the shorter paths (red and green) and 31 laps for the robot in the longer path (blue). The normalized position errors ($e[q, k]/r[q]$) of the three robots are in the histograms of Fig. 5.18.

5.4 Summary of results

The results presented in this chapter show the efficiency of both the coordination and the control strategies developed in this dissertation. The coordination of several robots was computed in very reasonable time and the computation of average speeds was able to keep the robots synchronized in simulated and real world scenarios. The control

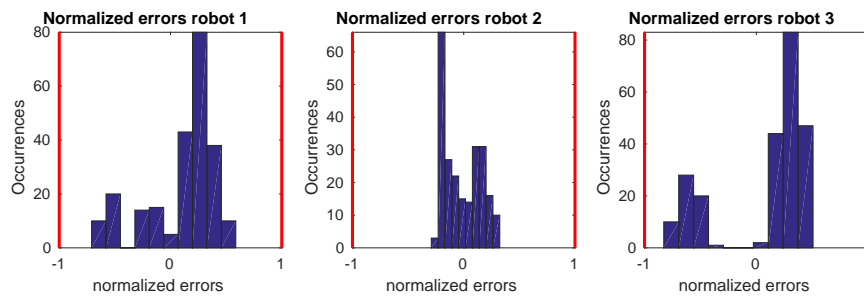


Figure 5.18. Histogram of the normalized position errors at the instants $t[q, k]$ for robots 1, 2 and 3, in the e-puck experiment.

strategy showed to be efficient and robust. It was also shown how the coordination and the control can be successfully used together to control multiple UAVs according to a coordinated plan of velocity profiles.

Chapter 6

Conclusion

6.1 Final considerations

This work presented a methodology to coordinate and control a group of fixed-wing UAVs along predefined paths that have intersection points which each other. The proposed solution is divided in two stages. The coordination stage is responsible for planning the speed of the robots along their respective paths and the control stage is responsible for computing input signals so that a UAV is stabilized on its respective path. The methodology and all of its stages were validated with several simulations and a real robot experiment.

The coordination strategy is developed by structuring a MILP formulation of the problem. Physical properties of the fixed-wing UAVs are taken into account in order to ease the task of the next stage. Those are regarded to minimum and maximum forward velocities and accelerations, robots with finite size and uncertainties in the positions and velocities. The centralized approach shows to be scalable, since scenarios with up to 48 robots were solved within 8 seconds.

The control strategy responsible for guiding each UAV along its predefined path is developed upon a vector field based strategy. Control laws were computed in order to make a UAV converge to the integral lines of a vector field, which in turns, converge to the desired curve. The control laws are designed, using Lyapunov Theory, upon a reference model, whose dynamics is imposed to the aircraft by lower level PID controllers. The controller is robust, since uncertainties in the reference model are considered to account for imperfections of the lower level controllers. Issues regarding singularity points of the vector fields were also formally treated. The control strategy showed to be efficient to guide the UAVs to the paths and perform the coordinated solution provided by the previous stage. Simulations with a complex aircraft model

were performed in order to prove the theoretical results.

The coordination strategy and the control strategy were used together to perform a simulation with 3 fixed wing UAVs following, each one, an ellipse. The complete model of the airplane was used, testifying the robustness of the whole system. In addition, an experiment with 3 actual robots was performed to demonstrate the robustness of the method in a real scenario.

Despite the efficiency of the coordination-control strategy presented here, it has some issues that must be recalled. In order to consider acceleration limits, conservative constraints were included in the MILP formulation. This showed to be a source of unfeasibility when the lengths of the segments are small.

An initialization strategy for the multiple UAVs start following their paths with synchronism was not developed in this work. It would be essential to implement the strategy in a real world scenario with fixed-wing UAVs. A strategy to recharge the batteries of the UAVs was also not developed, and it would also be important to a real world implementation, since we are dealing with persistent motion. Regarding the control strategy, a formal convergence proof when the saturation in the vertical velocity happens was not presented. Also, the strategy lacks a method to tackle singularities in the vector field when they are not on the z axis only.

6.2 Future works

Regarding the coordination strategy it could be extended in order to provide a planning for a group of robotic manipulators that have to execute independent tasks periodically and in the same environment. An idea would be to represent the paths in the joint space, to avoid multiple configurations, and compute the collision zones in the 3D workspace. Such improvement extends even more the applicability of the results, since manipulators executing periodic tasks are widely used in several assembly lines. Another extension would be the development of a closed loop coordination strategy that can be applied when the cycles of the robots are neither periodic nor commensurable. Such strategy would require an optimization problem to be constantly solved online, since collisions that may happen in the future would not be predictable given the absence of periodicity.

Regarding the control strategy, possible extensions would be the consideration of time varying vector fields and more complex reference models, for example, those who consider the dynamics of the roll angle.

Bibliography

- [Abichandani et al., 2013] Abichandani, P., Benson, H. Y., and Kam, M. (2013). Robust communication connectivity for multi-robot path coordination using mixed integer nonlinear programming: Formulation and feasibility analysis. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3600–3605. ISSN 1050-4729.
- [Abichandani et al., 2015] Abichandani, P., Torabi, S., Basu, S., and Benson, H. (2015). Mixed integer nonlinear programming framework for fixed path coordination of multiple underwater vehicles under acoustic communication constraints. *IEEE Journal of Oceanic Engineering*, 40(4):864–873. ISSN 0364-9059.
- [Alessandretti and Aguiar, 2017] Alessandretti, A. and Aguiar, A. P. (2017). A planar path-following model predictive controller for fixed-wing unmanned aerial vehicles. In *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*, pages 59–64. ISSN .
- [Alonso-Mora et al., 2012] Alonso-Mora, J., Breitenmoser, A., Beardsley, P., and Siegwart, R. (2012). Reciprocal collision avoidance for multiple car-like robots. In *2012 IEEE International Conference on Robotics and Automation*, pages 360–366. ISSN 1050-4729.
- [Alotaibi and Al-Rawi, 2016] Alotaibi, E. T. S. and Al-Rawi, H. (2016). Push and spin: A complete multi-robot path planning algorithm. In *2016 14th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–8. ISSN .
- [Altché et al., 2016] Altché, F., Qian, X., and de La Fortelle, A. (2016). Time-optimal coordination of mobile robots along specified paths. In *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5020–5026. ISSN .

- [Ambrosino et al., 2009] Ambrosino, G., Ariola, M., Ciniglio, U., Corraro, F., Lellis, E. D., and Pironti, A. (2009). Path generation and tracking in 3-D for UAVs. *IEEE Transactions on Control Systems Technology*, 17(4):980–988. ISSN 1063-6536.
- [Andersen and Kristiansen, 2017] Andersen, T. S. and Kristiansen, R. (2017). Path-following in three dimensions using quaternions for a fixed-wing UAV. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pages 1117–1122. ISSN .
- [Avellar et al., 2015] Avellar, G. S. C., Pereira, G. A. S., Pimenta, L. C. A., and Iscold, P. (2015). Multi-UAV routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783. ISSN 1424-8220.
- [Bazaraa et al., 2011] Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. (2011). *Linear programming and network flows*. John Wiley & Sons, 4th edition.
- [Beard and McLain, 2012] Beard, R. W. and McLain, T. W. (2012). *Small unmanned aircraft: Theory and practice*. Princeton University Press.
- [Berkelaar et al., 2004] Berkelaar, M., Eikland, K., and Notebaert, P. (2004). lp_solve 5.5, Open source (Mixed-Integer) Linear Programming system. <http://lpsolve.sourceforge.net/5.5/>.
- [Borkar et al., 2016] Borkar, A., Sinha, A., Vachhani, L., and Arya, H. (2016). Collision-free trajectory planning on lissajous curves for repeated multi-agent coverage and target detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1417–1422.
- [Ceccarelli et al., 2008] Ceccarelli, N., Di Marco, M., Garulli, A., and Giannitrapani, A. (2008). Collective circular motion of multi-vehicle systems. *Automatica*, 44(12):3025--3035.
- [Clark et al., 2003] Clark, C. M., Rock, S. M., and Latombe, J. C. (2003). Motion planning for multiple mobile robots using dynamic networks. In *2003 IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 3, pages 4222–4227. ISSN 1050-4729.
- [de Marina et al., 2017] de Marina, H. G., Kapitanyuk, Y. A., Bronz, M., Hattenberger, G., and Cao, M. (2017). Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5740–5745. ISSN .

- [Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197. ISSN 1089-778X.
- [den Berg and Overmars, 2005] den Berg, J. P. V. and Overmars, M. H. (2005). Prioritized motion planning for multiple robots. In *2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 430–435. ISSN 2153-0858.
- [Frew et al., 2013] Frew, E. W., Argrow, B., Lawrence, D., Elston, J., and Stachura, M. (2013). Unmanned aircraft systems for communication and atmospheric sensing missions. In *2013 American Control Conference*, pages 1482–1487. ISSN 0743-1619.
- [Frew and Lawrence, 2012] Frew, E. W. and Lawrence, D. (2012). Tracking expanding star curves using guidance vector fields. In *2012 American Control Conference (ACC)*, pages 1749–1754. ISSN 0743-1619.
- [Frew et al., 2008] Frew, E. W., Lawrence, D. A., and Morris, S. (2008). Coordinated standoff tracking of moving targets using Lyapunov guidance vector fields. *Journal of guidance, control, and dynamics*, 31(2):290–306.
- [Gonçalves et al., 2006] Gonçalves, E. N., Palhares, R. M., Takahashi, R. H. C., and Mesquita, R. C. (2006). New approach to robust \mathcal{D} -stability analysis of linear time-invariant systems with polytope-bounded uncertainty. *IEEE Transactions on Automatic Control*, 51(10):1709–1714. ISSN 0018-9286.
- [Gonçalves et al., 2010] Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A., Dutra, B. C. O., and Pereira, G. A. S. (2010). Vector fields for robot navigation along time-varying curves in n-dimensions. *IEEE Transactions on Robotics*, 26(4):647–659. ISSN 1552-3098.
- [Gonçalves et al., 2013] Gonçalves, V. M., Pimenta, L. C. A., Maia, C. A., and Pereira, G. A. S. (2013). Coordination of multiple fixed-wing UAVs traversing intersecting periodic paths. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 849–854. ISSN 1050-4729.
- [Hull, 2007] Hull, D. G. (2007). *Fundamentals of airplane flight mechanics*. Springer.
- [Jesus et al., 2013] Jesus, T. A., Pimenta, L. C. d. A., Tôrres, L. A. B., and Mendes, E. M. A. M. (2013). On the coordination of constrained fixed-wing unmanned aerial vehicles. *Journal of Control, Automation and Electrical Systems*, 24(5):585–600. ISSN 2195-3899.

- [Jufeng and Srinivas, 2005] Jufeng, P. and Srinivas, A. (2005). Coordinating multiple robots with kinodynamic constraints along specified paths. *The International Journal of Robotics Research*, 24(4):295–310.
- [Kamel et al., 2017] Kamel, M., Alonso-Mora, J., Siegwart, R., and Nieto, J. (2017). Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 236–243. ISSN 2153-0866.
- [Kamra and Ayanian, 2015] Kamra, N. and Ayanian, N. (2015). A mixed integer programming model for timed deliveries in multirobot systems. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 612–617. ISSN 2161-8070.
- [Kapitanyuk et al., 2017] Kapitanyuk, Y. A., Proskurnikov, A. V., and Cao, M. (2017). A guiding vector-field algorithm for path-following control of nonholonomic mobile robots. *IEEE Transactions on Control Systems Technology*, PP(99):1–14. ISSN 1063-6536.
- [Keller et al., 2017] Keller, J., Thakur, D., Likhachev, M., Gallier, J., and Kumar, V. (2017). Coordinated path planning for fixed-wing uas conducting persistent surveillance missions. *IEEE Transactions on Automation Science and Engineering*, 14(1):17–24. ISSN 1545-5955.
- [Khamsi and Kirk, 2011] Khamsi, M. A. and Kirk, W. A. (2011). *An introduction to metric spaces and fixed point theory*, volume 53. John Wiley & Sons.
- [Koenig and Howard, 2004] Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3. ISSN .
- [Kunen, 2009] Kunen, K. (2009). *The foundations of mathematics*. College Publications London.
- [Lawrence et al., 2007] Lawrence, D., Frew, E., and Pisano, W. (2007). Lyapunov vector fields for autonomous uav flight control. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6317.
- [Liang et al., 2015] Liang, Y., Jia, Y., Du, J., and Zhang, J. (2015). Vector field guidance for three-dimensional curved path following with fixed-wing UAVs. In *2015 American Control Conference (ACC)*, pages 1187–1192. ISSN 0743-1619.

- [Lin and Saripalli, 2015] Lin, Y. and Saripalli, S. (2015). Sense and avoid for unmanned aerial vehicles using ADS-B. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6402–6407. ISSN 1050-4729.
- [Mathew et al., 2013] Mathew, N., Smith, S. L., and Waslander, S. L. (2013). A graph-based approach to multi-robot rendezvous for recharging in persistent tasks. In *2013 IEEE International Conference on Robotics and Automation*, pages 3497–3502. ISSN 1050-4729.
- [Mitchell et al., 2015] Mitchell, D., Corah, M., Chakraborty, N., Sycara, K., and Michael, N. (2015). Multi-robot long-term persistent coverage with fuel constrained robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1093–1099. ISSN 1050-4729.
- [Mondada et al., 2009] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, pages 59–65. IPCB: Instituto Politécnico de Castelo Branco.
- [Nelson et al., 2007] Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W. (2007). Vector field path following for miniature air vehicles. *IEEE Transactions on Robotics*, 23(3):519–529. ISSN 1552-3098.
- [Oettershagen et al., 2014] Oettershagen, P., Melzer, A., Leutenegger, S., Alexis, K., and Siegwart, R. (2014). Explicit model predictive control and L1-navigation strategies for fixed-wing UAV path tracking. In *22nd Mediterranean Conference on Control and Automation*, pages 1159–1165. ISSN .
- [Olavo et al., 2018] Olavo, J. L. G., Jesus, T. A., Pimenta, L. C. A., Thums, G. D., Torres, L. A. B., and Palhares, R. M. (2018). Robust guidance strategy for target circulation by controlled UAV. *IEEE Transactions on Aerospace and Electronic Systems*, PP(99):1–1. ISSN 0018-9251.
- [Otte and Correll, 2014] Otte, M. and Correll, N. (2014). Any-com multi-robot path-planning with dynamic teams: Multi-robot coordination under communication constraints. In *Experimental Robotics*, pages 743–757. Springer.
- [Palacios-Gasós et al., 2016] Palacios-Gasós, J. M., Montijano, E., Sagues, C., and Llorente, S. (2016). Multi-robot persistent coverage using branch and bound. In *2016 American Control Conference (ACC)*, pages 5697–5702.

- [Park et al., 2007] Park, S., Deyst, J., and How, J. P. (2007). Performance and Lyapunov stability of a nonlinear path following guidance method. *Journal of Guidance, Control, and Dynamics*, 30(6):1718--1728.
- [Pelizer et al., 2017] Pelizer, G. V., da Silva, N. B. F., and Branco, K. R. L. J. (2017). Comparison of 3d path-following algorithms for unmanned aerial vehicles. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 498–505. ISSN .
- [Peng and Akella, 2003] Peng, J. and Akella, S. (2003). Coordinating the motions of multiple robots with kinodynamic constraints. In *Robotics and Automation (ICRA), 2003 IEEE International Conference on*, pages 4066–4073 vol.3. ISSN 1050-4729.
- [Quigley et al., 2009] Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.
- [Quintero et al., 2015] Quintero, S. A. P., Copp, D. A., and Hespanha, J. P. (2015). Robust uav coordination for target tracking using output-feedback model predictive control with moving horizon estimation. In *2015 American Control Conference (ACC)*, pages 3758–3764. ISSN 0743-1619.
- [Ren and Atkins, 2005] Ren, W. and Atkins, E. (2005). Nonlinear trajectory tracking for fixed wing uavs via backstepping and parameter adaptation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6196.
- [Rezende et al., 2018a] Rezende, A. M. C., Gonçalves, V. M., Raffo, G. V., and Pimenta, L. C. A. (2018a). Controle de VANT de asa fixa com campos vetoriais arbitrários. In *2018 SBA Congresso Brasileiro de Automática (CBA2018)*, pages 1–8.
- [Rezende et al., 2018b] Rezende, A. M. C., Gonçalves, V. M., Raffo, G. V., and Pimenta, L. C. A. (2018b). Robust fixed-wing UAV guidance with circulating artificial vector fields. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5892–5899. ISSN 2153-0866.
- [Richards and How, 2004] Richards, A. and How, J. (2004). Decentralized model predictive control of cooperating UAVs. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pages 4286–4291 Vol.4. ISSN 0191-2216.

- [Richards and How, 2002] Richards, A. and How, J. P. (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, volume 3, pages 1936–1941 vol.3. ISSN 0743-1619.
- [Rufli et al., 2013] Rufli, M., Alonso-Mora, J., and Siegwart, R. (2013). Reciprocal collision avoidance with motion continuity constraints. *IEEE Transactions on Robotics*, 29(4):899–912. ISSN 1552-3098.
- [Scherer and Rinner, 2016] Scherer, J. and Rinner, B. (2016). Persistent multi-uav surveillance with energy and communication constraints. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1225–1230.
- [Schwarting et al., 2017] Schwarting, W., Alonso-Mora, J., Pauli, L., Karaman, S., and Rus, D. (2017). Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1928–1935. ISSN .
- [Solovey and Halperin, 2017] Solovey, K. and Halperin, D. (2017). Efficient sampling-based bottleneck pathfinding over cost maps. In *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2003–2009. ISSN .
- [Soltero et al., 2011] Soltero, D. E., Smith, S. L., and Rus, D. (2011). Collision avoidance for persistent monitoring in multi-robot systems with intersecting trajectories. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3645–3652. ISSN 2153-0858.
- [Spensieri et al., 2016] Spensieri, D., Carlson, J. S., Ekstedt, F., and Bohlin, R. (2016). An iterative approach for collision free routing and scheduling in multirobot stations. *IEEE Transactions on Automation Science and Engineering*, 13(2):950–962. ISSN 1545-5955.
- [Stevens et al., 2015] Stevens, B. L., Lewis, F. L., and Johnson, E. N. (2015). *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons.
- [Sujit et al., 2014] Sujit, P. B., Saripalli, S., and Sousa, J. B. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. *IEEE Control Systems*, 34(1):42–59. ISSN 1066-033X.

- [Surynek, 2009] Surynek, P. (2009). An application of pebble motion on graphs to abstract multi-robot path planning. In *2009 21st IEEE Int. Conf. on Tools with Artificial Intelligence*, pages 151–158. ISSN 1082-3409.
- [Surynek, 2014] Surynek, P. (2014). Solving abstract cooperative path-finding in densely populated environments. *Computational Intelligence*, 30(2):402–450.
- [Tang et al., 2017] Tang, S., Thomas, J., and Kumar, V. (2017). Safe navigation of quadrotor teams to labeled goals in limited workspaces. In Kulić, D., Nakamura, Y., Khatib, O., and Venture, G., editors, *2016 International Symposium on Experimental Robotics*, pages 586–598, Cham. Springer International Publishing.
- [Tarjan, 1972] Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.
- [Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.
- [Thums, 2012] Thums, G. D. (2012). *Sintonia PID Robusta Multi-malha para Veiculos Aéreos Nao Tripulados*. PhD thesis, Universidade Federal de Minas Gerais.
- [Van Den Berg et al., 2011] Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer.
- [Van Den Berg et al., 2010] Van Den Berg, J., Snoeyink, J., Lin, M., and Manocha, D. (2010). Centralized path planning for multiple robots: Optimal decoupling into sequential plans. In *2010 Robotics: Science and systems*, volume 5, pages 137–144.
- [Wagner et al., 2012] Wagner, G., Kang, M., and Choset, H. (2012). Probabilistic path planning for multiple robots with subdimensional expansion. In *2012 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2886–2892. ISSN 1050-4729.
- [Wallar et al., 2015] Wallar, A., Plaku, E., and Sofge, D. A. (2015). Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas. *IEEE Transactions on Automation Science and Engineering*, 12(3):969–980. ISSN 1545-5955.
- [Wu et al., 2018] Wu, C., Chen, J., Jeltsema, D., and a. C. Dai (2018). Guidance vector field encoding based on contraction analysis. In *2018 European Control Conference (ECC)*, pages 282–287. ISSN .

- [Yu and Rus, 2015] Yu, J. and Rus, D. (2015). Pebble motion on graphs with rotations: Efficient feasibility tests and planning algorithms. In *Algorithmic Foundations of Robotics XI*, pages 729--746. Springer.
- [Yun and Yamamoto, 1992] Yun, X. and Yamamoto, Y. (1992). On feedback linearization of mobile robots. *University of Pennsylvania Department of Computer and Information Science Technical Reports No. MS-CIS-92-45*.
- [Zhang et al., 2010] Zhang, M., Shen, Y., Wang, Q., and Wang, Y. (2010). Dynamic artificial potential field based multi-robot formation control. In *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, pages 1530–1534. ISSN 1091-5281.
- [Čáp et al., 2015] Čáp, M., Novák, P., Kleiner, A., and Selecký, M. (2015). Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Transactions on Automation Science and Engineering*, 12(3):835–849. ISSN 1545-5955.