DECENTRALIZED COORDINATION AND CONTROL IN ROBOTIC SWARMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ANDAÇ TÖRE ŞAMİLOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

DECENTRALIZED COORDINATION AND CONTROL IN ROBOTIC SWARMS

submitted by **ANDAÇ TÖRE ŞAMİLOĞLU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan ÖZGEN
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Süha ORAL
Head of Department, **Mechanical Engineering**

Assist. Prof. Dr. A. Buğra KOKU
Supervisor, **Mechanical Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Y. Samim ÜNLÜSOY
Mechanical Engineering, METU

Asst. Prof. Dr. A. Buğra KOKU
Mechanical Engineering, METU

Prof. Dr. Veysel GAZİ
Electrical and Electronics Engineering, İstanbul Kemerburgaz Univ.

Prof.Dr. Reşit SOYLU
Mechanical Engineering, METU

Prof. Dr. Osman PARLAKTUNA
Electrical and Electronics Engineering, Osman Gazi Univ.

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ANDAÇ TÖRE ŞAMİLOĞLU

Signature :

ABSTRACT

DECENTRALIZED COORDINATION AND CONTROL IN ROBOTIC SWARMS

ŞAMİLOĞLU, ANDAÇ TÖRE
Ph.D., Department of Mechanical Engineering
Supervisor : Assist. Prof. Dr. A. Buğra KOKU

September 2012, 273 pages

In this thesis study the coordination and control strategies for leaderless, decentralized robotic swarms are developed. The mathematical models of the collective motion of agents are derived by mimicry of swarm of organisms like schools of fish, herds of quadrupeds, flocks of flying birds. There are three main parts of this study (i) mathematical modelling, (ii) analytical analysis (iii) experimental and simulation based validations of the results. These works are performed on the (i) Fundamental agreement behaviors of swarms, (ii) The flocking and distribution behaviors of swarms, (iii) The orientation agreements of swarms, (iv) Circling behaviour, (v) Line formation, (vi) The control strategies and stabilization of formations, (vii) Switching between the formations of swarm of robots. The development of suitable control strategies and stability analysis are performed by the utilization of non-linear and discrete time control theories. The developed strategies can be applied on swarm of robots in the applications of terrestrial, space and oceanic exploration, military surveillance and rescue missions, and other automated collaborative operations.

Keywords: Robotic Swarms, Decentralized Control, Autonomous robots, Circling Behavior, Line Formation

ÖZ

ROBOT SÜRÜLERİNDE MERKEZDEN BAĞIMSIZ KOORDİNASYON VE KONTROL

ŞAMİLOĞLU, ANDAÇ TÖRE
Doktora, Makine Mühendisliğ Bölümü
Tez Yöneticisi : Y. Doç Dr. A. Buğra KOKU

Eylül 2012, 273 sayfa

Bu tez çalışmasında merkezden bağımsız robot sürülerinde koordinasyon ve kontrolcü stratejileri geliştirilmiştir. Robot sürülerinin kollektif davranış dinamiklerinin matemetiksel modelleri doğada karşılaşılan robot sürülerindeki (kuş, balık, bizon sürüleri gibi) davranışların taklidiyle oluşturulmuştur. Bu çalışmanın 3 ana evresi vardır. Bunlar, (i) matematiksel modelleme ve analitik analiz, (ii) benzetim tabanlı doğrulamalar ve (iii) deneysel doğrulamalardır. Bu evreler robot sürülerinde (i) temel uzlaşma problemleri, (ii) toplanma ve dağılma dinamikleri (iii) ortak yön belirleme stratejileri, (iv) dairesel hareket davranışları, (v) doğrusal geometri oluşturma, (vi) geometrik oluşımların doğrusal olmayan konrolcüleri ve kararlılık analizi, (vii) bu stratejiler arasındaki geçişler üzerine uygulanmıştır. Bu davranışların kararlılık analizi ve uygun kontrolcülerin tasarımı doğrusal olmayan kontrol, ayrık kontrol vb. teorileri kullanarak yapılmıştır. Geliştirilen tüm kontrol stratejileri alan tarama, askeri gözetleme, arama-kurtarma uygulamaları gibi koordinasyon gerektiren robot uygulamalarına uyarlanabilirdir.

Anahtar Kelimeler: Robot Sürüleri, Merkezden Bağımsız kontrol, Otonom Robotlar, Dairesel Hareket, Doğrusal Geometri Oluşumu

To My Love Fulya, and My Family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Asst. Prof. Dr. A. Buğra Koku and Prof. Dr. Veysel Gazi for their instructive guidance, advice, criticism, encouragements and insight throughout the research. I would also like to thank them for their model academic and ethical attitude, which will guide me through the rest of my carrier and my life.

I am deeply grateful also to my family who supported and encouraged me through all my life.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES xiii

LIST OF FIGURES xiv

CHAPTERS

1 INTRODUCTION 1

  1.1 Motivation of the Thesis Study 3

  1.2 Literature Survey 4

    1.2.1 Mathematical Models 5

    1.2.2 Swarm Coordination and Control Problems 10

    1.2.3 Modeling, Coordination, and Control Approaches 13

  1.3 Scope of the Thesis Study 17

2 Asynchronous Cyclic Pursuit 19

  2.1 Asynchronous High-Level Model 20

  2.2 Convergence Under Total Synchronism 23

  2.3 Convergence Analysis of the Asynchronous Model 26

  2.4 Simulation Examples 28

  2.5 Experimental Examples 30

  2.6 Conclusions 31

3 Orientation Agreement Problem 33

  3.1 Introduction 33

  3.2 High Level Dynamics 35

3.3 Strategies for Orientation Agreement 39
3.3.1 Strategy 1 (Averaging) 39
3.3.2 Strategy 2 (Relative Angles) 40
3.3.3 Strategy 3 (Vector Sum) 41
3.4 Turn Angle Restrictions 42
3.5 Simulation Results 43
3.5.1 Effect of α for Unbounded Region 46
3.5.1.1 Effect of α for Unbounded Region in Synchronous model 46
3.5.1.2 Effect of α for Unbounded Region in Asynchronous model 47
3.5.2 Effect of α for Bounded Region 50
3.5.2.1 Effect of α for Bounded Region in Synchronous model 50
3.5.2.2 Effect of α for Bounded Region in Asynchronous model 52
3.6 Discussions 54
3.7 Conclusions 66
4 Controllers for Tracking, Circling, and Line Following 71
4.1 Introduction 71
4.2 Mathematical Model 71
4.3 Targeting and Circling Around a Target 74
4.3.1 Constant Speed Controllers 75
4.3.1.1 Stability Analysis via Lyapunov Functions 79
4.3.2 Speed and Angular Velocity Controllers 81
4.3.2.1 Stability via Linearization 82
4.3.2.2 Stability Analysis via Lyapunov Functions 84
4.3.3 Angular Velocity Controllers 87
4.3.3.1 Proportional Angular Velocity Controller 87
4.3.3.2 Nonlinear Angular Velocity Controller 88
4.3.4 Speed Controllers 89
4.3.4.1 Constant Speed Controller 89

4.3.4.2 Linear Speed Controller - Type I 90

4.3.4.3 Linear Speed Controller - Type II 92

4.3.4.4 Quadratic Speed Controller 95

4.3.4.5 Stability via Lyapunov Functions 97

4.3.5 Simulation Results 105

4.3.5.1 Target Tracking 106

4.3.5.2 Circling Around the Target 108

4.4 Switching Gradient Method for Circling Around a Target 110

4.4.1 Stability Analysis via Lyapunov Functions 114

4.4.2 Simulation Results 116

4.5 Switching Gradient Method for Line Formation 117

4.5.1 Stability Analysis via Lyapunov Functions 120

4.5.2 Simulation Results 122

4.6 Conclusion 122

5 Circling Controllers Developed by Feedback Linearization 126

5.1 Introduction 126

5.2 Mathematical Model 130

5.2.1 Stationary Target 131

5.2.2 Non-Stationary Target 132

5.3 Circling Around a Target 133

5.3.1 Circling - SISO Case - Stationary Target 134

5.3.2 Circling - SISO - Stationary Target - Simulation Results . 138

5.3.3 Circling - MIMO - Stationary Target Case 140

5.3.4 Circling - MIMO - Stationary Target - Simulation Results . 147

5.3.5 Circling - MIMO - Non-Stationary Target Case 149

5.3.6 Circling - MIMO - Non-Stationary Target - Simulation Results 157

5.3.7 Application of Controllers to Multi-agent Systems 159

5.3.7.1 Multi-Agent System with SISO - Stationary Target Following Controller 161

5.3.7.2 Multi-Agent System with MIMO - Stationary Target Following Controller 162

5.3.7.3 Multi-Agent System with MIMO - Non-Stationary Target Following Controller 167

5.3.7.4 Multi-Agent System with MIMO - Non-Stationary Target Following Controller - Simulations . . 182

5.3.8 Concluding Remarks 182

6 Line Following Controllers Developed by Feedback Linearization 186

6.1 Introduction 186

6.2 Following a line passing through a Target with a specified slope . . . 187

6.2.1 Line Following - SISO - Stationary Target Case 188

6.2.2 Line Following - SISO - Stationary Target - Simulation Results 193

6.2.3 Line Following - MIMO - Stationary Target 197

6.2.4 Line Following - MIMO - Stationary Target - Simulation Results 203

6.2.5 Line Following - MIMO - Non-Stationary Target Case . . 207

6.2.6 Line Following - MIMO - Non-Stationary Target - Simulation Results 214

6.2.7 Application of Controllers to Multi-agent Systems 218

6.2.7.1 Line Formation of Multi-Agent System with SISO - Stationary Target Controller 218

6.2.7.2 Line Formation of Multi-Agent System with MIMO - Stationary Target Controller 219

6.2.7.3 Line Formation of Multi-Agent System with MIMO - Non-Stationary Target Controller . . 222

6.2.7.4 Multi-Agent System with MIMO - Non-Stationary Target Following Controller - Simulations . . 236

6.2.8 Discussion and Future Directions 237

6.2.9 Concluding Remarks 242

7 Conclusion 244

APPENDICES

A Experimental Set-up For Multi-Robot Applications 249

A.1 Introduction 249

A.2 The Set-Up Structure 251

A.3 Image Processing Setup/Methods 253

A.3.1 Determining Robot Locations 253

A.3.2 Determining the Robot Orientations 255

A.3.3 Determining Robot ID’s 256

A.4 Transmitting Position and Orientation Information 258

A.5 Concluding Remarks 259

REFERENCES 260

VITA 271

LIST OF TABLES

TABLES

Table 3.1 Pseudocode for Calculating Θ 41

Table 3.2 Pseudocode of Simulation Steps 45

LIST OF FIGURES

FIGURES

Figure 2.1 Finite State Machine Model 21

Figure 2.2 (a) Gershgorin disc with center at $1 - p$ and radius p. (b) Gershgorin discs for $p = 0.25,\ 0.50,\ 0.75$, and 1.00. 25

Figure 2.3 (a) Simulation results for synchronous convergence. (b) Simulation results for asynchronous convergence. 29

Figure 2.4 Convergence performance of synchronous and asynchronous pursuits. . . . 30

Figure 2.5 Path of 5 E-puck robots in cyclic pursuit obtained in the set-up. 31

Figure 2.6 Cyclic Pursuit of 5 robots. 31

Figure 3.1 Finite State Machine Model 36

Figure 3.2 Possible arrangements of orientation of an agent and its neighbor's orientation 40

Figure 3.3 Orientation rule of Strategy 3 for only one neighbor (j) of agent i. 42

Figure 3.4 $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and unbounded region. 47

Figure 3.5 $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and unbounded region. 48

Figure 3.6 The number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and unbounded region. 49

Figure 3.7 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Synchronous case - Unbounded region). 50

Figure 3.8 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Synchronous case - Unbounded region). 51

Figure 3.9 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Synchronous case - Unbounded region). 52

Figure 3.10 $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and unbounded region. 53

Figure 3.11 $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and unbounded region. 54

Figure 3.12 the number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and unbounded region. 55

Figure 3.13 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Asynchronous case - Unbounded region). 56

Figure 3.14 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Asynchronous case - Unbounded region). 57

Figure 3.15 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Asynchronous case - Unbounded region). 58

Figure 3.16 $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and bounded region. 59

Figure 3.17 $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and bounded region. 60

Figure 3.18 the number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and bounded region. 61

Figure 3.19 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Synchronous case - Bounded region). 62

Figure 3.20 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Synchronous case - Bounded region). 63

Figure 3.21 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Synchronous case - Bounded region). 64

Figure 3.22 $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and bounded region. 65

Figure 3.23 $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and bounded region. 66

Figure 3.24 the number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and bounded region. 67

Figure 3.25 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Asynchronous case - Bounded region). 68

Figure 3.26 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Asynchronous case - Bounded region). 69

Figure 3.27 Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Asynchronous case - Bounded region). 70

Figure 4.1 System model in polar coordinates. 73

Figure 4.2 System model for relative coordinates. 74

Figure 4.3 $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ for several v values. 78

Figure 4.4 $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ and the distance d for constant $v = r\alpha\pi/2$ where r is the radius of the circle on which the robot travels. 80

Figure 4.5 $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ and the distance d for constant $v = r\alpha\pi/2$ where r is the radius of the circle on which the robot travels. 85

Figure 4.6 $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ for several v values. 86

Figure 4.7 Phase Portrait for Constant Speed and Proportional Angular Velocity Controllers. 91

Figure 4.8 Simulation result for Constant Speed and Proportional Angular Velocity Controllers. 91

Figure 4.9 Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers, $A < \alpha\pi/2$. 93

Figure 4.10 Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers, $A > \alpha\pi/2$. 93

Figure 4.11 Simulation result for Linear Speed and Proportional Angular Velocity Controllers, $A < \alpha\pi/2$. 94

Figure 4.12 Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers. 95

Figure 4.13 Simulation result for Linear Speed and Proportional Angular Velocity Controllers. 96

Figure 4.14 Quadratic Speed Function and its derivative with respect to distance. . . . 96

Figure 4.15 Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers. 98

Figure 4.16 Simulation result for Quadratic Speed and Proportional Angular Velocity Controllers. 98

Figure 4.17 Simulation result for Quadratic Speed and Proportional Angular Velocity Controllers. 99

Figure 4.18 Lie Derivative for Speed function $v : constant$ 101

Figure 4.19 Lie Derivative at $d = d_{eq}$ for Speed function $v : constant$ 101

Figure 4.20 Lie Derivative at $\gamma = 90^o$ for Speed function $v : constant$ 102

Figure 4.21 Lie Derivative for Speed function $v(d) = Ad + B$ 103

Figure 4.22 Lie Derivative at $d = d_{eq}$ for Speed function $v(d) = Ad + B$ 103

Figure 4.23 Lie Derivative at $\gamma = 90^o$ for Speed function $v(d) = Ad + B$ 104

Figure 4.24 Lie Derivative for Speed function $v(d) = Ad - B$ 104

Figure 4.25 Lie Derivative at $d = d_{eq}$ for Speed function $v(d) = Ad - B$ 105

Figure 4.26 Lie Derivative at $\gamma = 90^o$ for Speed function $v(d) = Ad - B$ 105

Figure 4.27 Lie Derivative for Speed function $v(d) = (d - C)^2$ 106

Figure 4.28 Lie Derivative at $d = d_{eq}$ for Speed function $v(d) = (d - C)^2$ 107

Figure 4.29 Lie Derivative at $\gamma = 90^o$ for Speed function $v(d) = (d - C)^2$ 107

Figure 4.30 Trajectories of the robot for different initial positions and orientations. . . . 108

Figure 4.31 Distance between robot and target for different initial positions and orientations. 109

Figure 4.32 Trajectories of the robot for different initial orientations. 109

Figure 4.33 Distance between robot and target for different initial orientations. 110

Figure 4.34 Path of the Target. 110

Figure 4.35 Robot trajectory tracking a dynamic target. 111

Figure 4.36 Distance of the robot from the dynamic target. 111

Figure 4.37 Robot trajectories circling around a static target at $[20, 20]$. 112

Figure 4.38 Distance between robot and static target at $[20, 20]$ for circling around the target strategy. 112

Figure 4.39 Robot trajectory circling around a dynamic target. 112

Figure 4.40 Distance between target and robot circling around a dynamic target. 113

Figure 4.41 $\dot{\theta} - \dot{psi}$ versus $\theta - \psi$ for several v values. 115

Figure 4.42 $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ and the distance d for constant $v = r\alpha\pi/2$ where r is the radius of the circle on which the robot travels. 115

Figure 4.43 Robot trajectories circling around a static target at $[20, 20]$. 117

Figure 4.44 Distance between robot and static target at $[20, 20]$ for circling around the target strategy. 118

Figure 4.45 Robot trajectories circling around a static target at $[20, 20]$ for different initial angles. 118

Figure 4.46 Robot trajectory circling around a dynamic target. 119

Figure 4.47 Distance between target and robot circling around a dynamic target. 119

Figure 4.48 Robot trajectories converging to line passing through a static target at $[20, 20]$.123

Figure 4.49 Distance between robot and line for different initial positions. 123

Figure 4.50 Robot trajectories converging to line passing through a static target at $[20, 20]$ for different initial orientations. 124

Figure 4.51 Distance between robot and line for different initial orientations. 124

Figure 5.1 System model in polar coordinates. 131

Figure 5.2 System model for relative coordinates. 132

Figure 5.3 System model with non-stationary target in relative coordinates. 133

Figure 5.4 A sample phase plane of Circling-SISO system. 138

Figure 5.5 The change of the distance between robot and target, and the angle difference γ with respect to time. Overdamped response for circling behavior. 139

Figure 5.6 The path of the robot. Overdamped response for circling behavior. 139

Figure 5.7 The change of the distance between robot and target, and the angle difference γ with respect to time. Underdamped response for circling behavior. 140

Figure 5.8 The path of the robot. Underdamped response for circling behavior. 141

Figure 5.9 The change of the distance between robot and target, and the angle difference γ with respect to time. Overdamped response for circling behavior. 148

Figure 5.10 The path of the robot. Overdamped response for circling behavior. 148

Figure 5.11 The change of the distance between robot and target, and the angle difference γ with respect to time. Underdamped response for circling behavior. 149

Figure 5.12 The path of the robot. Underdamped response for circling behavior. 150

Figure 5.13 The distance between agent and target d with respect to time. Overdamped response for circling behavior with non-stationary target follower controller. . . . 158

Figure 5.14 Velocities of agent and target. Overdamped response for circling behavior with non-stationary target follower controller. 158

Figure 5.15 Path of the agent and target. Agent path: solid blue, target path: dashed black. 159

Figure 5.16 The distance between agents and target d. The responses of non-stationary and stationary target follower controllers. 160

Figure 5.17 Velocities of agents with non-stationary and stationary target follower controllers. 160

Figure 5.18 Path of the agents and target. Non-stationary controller path: solid blue, stationary controller path: dash-dotted red, target path: dashed black. 161

Figure 5.19 Distance (a) and path (b) of parallel motion of 2 agents with Circling SISO-Stationary Targeting controllers. 162

Figure 5.20 Distance (a) and path (b) of coinciding motion of 2 agents with Circling SISO-Stationary Targeting controllers. 163

Figure 5.21 Distance (a) and path (b) of regular circling motion of 2 agents with Circling SISO-Stationary Targeting controllers. 163

Figure 5.22 Distance (a) and path (b) of regular circling motion of 3 agents with Circling SISO-Stationary Targeting controllers. 164

Figure 5.23 Axis of rotation 165

Figure 5.24 Distance (a) and velocities (b) of orbital circling motion of 2 agents with Circling MIMO-Stationary Targeting controllers. 165

Figure 5.25 Path of the agents and center for orbital circling with MIMO-Stationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black. 166

Figure 5.26 Distance (a) and velocities (b) of regular circling motion of 2 agents with Circling MIMO-Stationary Targeting controllers. 166

Figure 5.27 Path of the 2 agents and center for regular circling with MIMO-Stationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black. 167

Figure 5.28 Distance (a) and velocities (b) of regular circling motion of 3 agents with Circling MIMO-Stationary Targeting controllers. 167

Figure 5.29 Path of the 3 agents and center for regular circling with MIMO-Stationary Targeting controllers. Agent paths: dashed colored, center path: dashed black. . . 168

Figure 5.30 Distance (a) and velocities (b) of regular circling motion of 10 agents with Circling MIMO-Stationary Targeting controllers. 168

Figure 5.31 Path of the 10 agents and center for regular circling with MIMO-Stationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black. 169

Figure 5.32 Two agents moving with the equilibrium of controller when $K = 1$. 170

Figure 5.33 Agent positions and velocity vectors at equilibrium. Samples for 2 and 3 number of agents. 179

Figure 5.34 Distance (a) and velocities (b) of regular circling motion of 2 agents with Circling MIMO-NonStationary Targeting controllers. 182

Figure 5.35 Path of the 2 agents and center for regular circling with MIMO-NonStationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black. 183

Figure 5.36 Distance (a) and velocities (b) of regular circling motion of 3 agents with Circling MIMO-NonStationary Targeting controllers. 183

Figure 5.37 Path of the 3 agents and center for regular circling with MIMO-NonStationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black. 184

Figure 5.38 Distance (a) and velocities (b) of regular circling motion of 10 agents with Circling MIMO-NonStationary Targeting controllers. 184

Figure 5.39 Path of the 10 agents and center for regular circling with MIMO-NonStationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black. 185

Figure 6.1 Relative Coordinate Frame. Origin is target. Circle represents the vehicle. The line to be followed is the one with the slope β. 189

Figure 6.2 The change of the distance between robot and target (d), and robot and line (h) with respect to time. Overdamped response for line following behavior. 194

Figure 6.3 The change of the angles θ and ψ with respect to time. Overdamped response for line following behavior. 194

Figure 6.4 The path of the robot. Overdamped response for line following behavior. . 195

Figure 6.5 The change of the distance between robot and target (d), and robot and line (h) with respect to time. Underdamped response for line following behavior. . . . 195

Figure 6.6 The change of the angles θ and ψ with respect to time. Underdamped response for line following behavior. 196

Figure 6.7 The path of the robot. Underdamped response for line following behavior. . 196

Figure 6.8 The path of the robot following the slope $\beta + \pi$. 197

Figure 6.9 The change of the distance between robot and target (d), and robot and line (h) with respect to time. Overdamped response for line following behavior. 204

Figure 6.10 The change of the angles θ and ψ with respect to time. Overdamped response for line following behavior. 204

Figure 6.11 The path of the robot. Overdamped response for line following behavior. . 205

Figure 6.12 The change of the distance between robot and target (d), and robot and line (h) with respect to time. Underdamped response for line following behavior. . . . 205

Figure 6.13 The change of the angles θ and ψ with respect to time. Underdamped response for line following behavior. 206

Figure 6.14 The path of the robot. Underdamped response for line following behavior. . 206

Figure 6.15 The path of the robot following the slope $\beta + \pi$. 207

Figure 6.16 The distance between agents and target d. The responses of non-stationary and stationary line following controllers. 216

Figure 6.17 Error of ψ ($e_{\psi} = \beta - \psi$) for non-stationary and stationary line following controllers. 216

Figure 6.18 Velocities of target and agents with non-stationary and stationary line following controllers. 217

Figure 6.19 Path of the agents and target. Non-stationary controller path: dash-dotted blue, stationary controller path: dashed red, target path: solid black. 217

Figure 6.20 Distance h (a) and error of e_{ψ} (b) of 5 agents with line following SISO controller 219

Figure 6.21 Path of 5 agents with line following SISO controller. Arrow heads: agents, Cross: center of swarm. 220

Figure 6.22 The relation between target and center of swarm. 220

Figure 6.23 Distance h (a) and error e_{ψ} (b) of 5 agents with line following MIMO controller for stationary target. 221

Figure 6.24 Velocities of agents and center of swarm (a), and path of 5 agents (b) with line following MIMO controller for stationary target. 222

Figure 6.25 Line formation of agents at equilibrium. 225

Figure 6.26 Distance h (a) and error e_{ψ} (b) of 5 agents with line following MIMO controller for stationary target. 237

Figure 6.27 Velocities of agents and center of swarm (a), and path of 5 agents (b) with line following MIMO controller for non-stationary target. 237

Figure 6.28 Distance h (a) and error e_ψ (b) of 5 agents traveling on a hexagon with line following controller. 238

Figure 6.29 Velocities of agents and center of swarm (a), and path of 5 agents (b) 5 agents traveling on a hexagon with line following controller. 238

Figure 6.30 Distance d (a) and distance h (b) of 5 agents switching between line following and circling behaviors. 239

Figure 6.31 Error e_ψ (a) and velocities of agents and center of swarm (b), of 5 agents switching between line following and circling behaviors. 240

Figure 6.32 Path of 5 agents switching between line following and circling behaviors. . 240

Figure 6.33 Distance h (a) and velocities (b) of 5 agents traveling on parallel paths. . . 242

Figure 6.34 Error e_ψ (a) and orientation θ (b), of 5 agents traveling on parallel paths.. . 242

Figure 6.35 Path of 5 agents traveling on parallel paths.. 243

Figure A.1 Experimental setup consisting of an arena, robots, PC and overhead camera. 251

Figure A.2 (a)A sample robot hat used to find the position, orientation and identity of robot. (Dimensions are in mm). (b) Three robots in the arena. The colored dots and the boundaries that other robots should stay out of are shown. 254

Figure A.3 (a) The colored dots that are used to find the position and orientation of a robot. (b)Speeds of a differential drive robot. 256

Figure A.4 (a) Robot Identities. (b) Vectors from P_1 to P_2 and P_1 to P_3 for two different cases: P_1 is on the left for the left figure and P_1 is on the right for the right figure. 257

CHAPTER 1

INTRODUCTION

The collective motion of organisms like schools of fish, herds of quadrupeds, flocks of flying birds, and groups of migrating bacteria, molds, ants, or pedestrians is an interesting area studied by many biologists, physicists, and even engineers in recent years. The coordinated behavior of such animal groups results in complex and meaningful emergent or self-organizing behavior with only local interactions of relatively simple or “dumb” individuals (or agents as we call them in this study). Life sciences like theoretical biology and animal ethology can benefit from the ideas or principles derived from the operation of natural multi-agent systems. The developed ideas and principles may also be utilized in many engineering fields including swarm robotics [1, 2], optimization [3, 4, 5, 6, 7], self-organizing distributed sensor networks [8], decentralized/distributed coordination and control of groups of unmanned air, space, land and underwater vehicles, or even problems of social sciences including organization theory, economics, and cognitive psychology. Hence, for several decades many scientists from different fields have been trying to understand, model, and mimic/reproduce the behavior seen in natural swarms.

In general, a multi-agent dynamic system can be defined as a network of a number of agents. The agents are loosely coupled dynamics units like robots, vehicles, dynamic sensors, etc.

The main purpose of using multi-agent systems is to collectively achieve goals that are difficult to reach by an individual agent. If the main dynamic action of interest is motion, sometimes, the term swarm or formation are used in place of multi-agent system. Recent robotics research has been focusing on multi-agent systems or basically groups of autonomous mobile agents. Such systems are of interest for several reasons: (i) Tasks may be too complex or sometimes impossible for a single agent to achieve; (ii) Performance of the system may be improved by using multiple agents; (iii) The agents of a multi-agent system may be easier to build, cheaper, more flexible, and more fault tolerant than a single agent designed for each separate complex task; (iv) The constructive, synthetic logic developed for cooperative mobile robotics can also be beneficial in the problems of other sciences; especially for social sciences including organization theory, economics, cognitive psychology or life sciences like theoretical biology and animal ethology.

The amount of researches about multi-robot or swarm-robot systems have increased importantly. Ever-developing technology made these kinds of systems realizable and this is the main reason of increasing researches in this area. The main advantages of these kinds of systems that are formed by autonomous robots are their robust, cheap and flexible properties. Their flexibility comes from the possibility of re-distributing the tasks among them in order to adapt different situations and/or missions or negate the absence of some individuals. Similarly robustness comes from the possibility of re-organization of the individuals to negate the absence of lost individual and to complete the mission. The robots used in this kind of systems are not so complicated. They are simpler and cheaper than the robots which do all the job individually since the tasks are distributed between these simple robots. Also the failure probability of these simple robots is less than the more complex robots. This characteristic strengthens the robustness property of multi-robot systems. Also the production costs of these simple robots are lower with respect to the complex multitasking robots.

Another important property of swarm robotic systems is *emergence*. The collective behavior of these systems emerges through the local interactions between the agents and the environment. Usually, the modeling and the design process of local interactions to result in a desired emergent behavior is difficult and not has been studied too much in the literature. It remains an important open problem.

The information (i.e. sensing, control, and communication) exchange between agents is an important element of a swarm system. The dynamic interactions among the agents need to be considered for formations where the individual agent dynamics are coupled.

1.1 Motivation of the Thesis Study

The thesis study here focus on the decentralized geometric formation of the multi-agent systems. The geometric formation of the agents is mostly required in area exploration and surveillance missions. During these missions the agents are supposed to form special geometric shapes, i.e. distribution evenly over an area, spreading over a line and move in parallel directions, circling around a target. These formations may be static (agents are stationary at a geometric form) or dynamic (agents are moving preserving a special geometry). The main problem in these formations is the emergence property of multi-agent systems. Since the initial conditions, number of agents in the swarm, uncertainties, under modeled dynamics etc. may not be predictable, the resulting formations are emergent. Therefore, the controllers of each individual agent should be independent of pre-specified or re-specified paths and the identity of the other agents in the swarm. The global or local interaction strategies between agents should not change with respect to the type, identity or number of the agents to preserve the robustness and flexibility properties of multi-agent systems mentioned above. Therefore, the controllers developed in this study do not utilize pre-generated paths of agents, they only use absolute or relative position, orientation and the time derivatives of these variables in the manipulation input calculations. Using these agent motion parameters, the controllers designed achieved

- Orientation agreement behavior,
- Circling around static and dynamic target behaviors,
- Following line passing through specified point and with a specified orientation behavior,
- Area surveillance with parallel motion behavior,
- Switching between the above controller strategies

In the development of these controllers we utilized linear and nonlinear control theories. The controller designs mostly utilize PID, Lyapunov functions, and feedback linearization techniques. There are no controllers even for single agents in the literature similar to the ones developed in this thesis study. Therefore, especially for circling and line following behaviors we had to first develop single agent controllers to circle around a target and follow a line. Then, we apply these controllers to multi-agent systems with some minor modifications.

Furthermore, the multi-agent systems are usually asynchronous systems. There are time delays in the sensing, computing and motion states and the time delays may differ in between agents. Therefore, the controllers should be robust to this asynchronism. The first part of this thesis study focus on the asynchronism property of multi-agent systems. One of the most simple geometric formation problems, cyclic pursuit is examined with an asynchronous model.

The controllers developed in the study are mostly applicable to the formation problems of unmanned air, space, land and underwater vehicles. The agents are considered to travel in 2-D space and only the kinematic models are utilized in the derivations. The kinetics of the agents are not considered since all the controllers developed for the proof of concept. All controllers are validated by computer simulations. Additionally, an experimental set-up is utilized in the validation of some of the controllers running on mobile ground robots.

In the following sections we first investigate the studies in the literature. Then, we will present the scope of the thesis in the last section.

1.2 Literature Survey

The output of multi-agent systems research has implications on many fields of (engineering) applications such as terrestrial, space and oceanic exploration, military surveillance and rescue missions, and other automated collaborative operations. The desired approach in solving such engineering problems is achieving the global objective or emergent behavior by simple local rules/interactions. However, determination of agent level simple interaction rules that yields the desired global behavior is a challenging problem that has not been solved yet. On this subject one of the earliest famous study was performed by Reynolds [9]. He introduced a model and wrote a program called *boids* (or bird-oids) that simulates a flock of birds in flight. He showed that, if followed by the simulated agents, three simple rules can result in realistic behavior similar to the one observed in bird flocks. The behavior-based techniques used by Reynolds were also studied by Balch and Arkin [10]. They designed reactive behaviors to implement multivehicle formations in combination with rules for collision avoidance and other navigational goals.

Among the first relevant works by biologists are the studies by Breder [11], Warburton and

Lazarus [12], Okubo and Grunbaum [13, 14, 15], and Parrish [16] for swarm aggregations and coordination. Inspired by these works, a recent series of studies [17, 18, 19, 20, 21, 22, 23, 24] has provided rigorous stability and convergence analysis of swarm aggregations. One of the early literature surveys on the topic of multi-agent (multi-robot) systems is the study of Mataric [2] in 1995. There are also some recent books [3, 25] that may be useful references about the swarms in nature and engineering applications that inspire from these swarms. The references in [26, 27, 28, 29] are some relevant books and special issues of journals that include the recent studies on the swarm-robotics. In [30] the advantages and some applications of swarm-robotics are presented and principal definitions of some properties of these systems are stated. The studies [1, 2] provide comprehensive reviews of multi-agent systems studies.

One very recent survey that considers multi-agent systems from the perspective of control engineering can be found in [31]. In the following sections the mathematical models, the swarm coordination and control problems, and some approaches to modeling and control of swarm issues in the literature are investigated based on the work in [31].

1.2.1 Mathematical Models

Some of the mathematical models for agent/vehicle dynamics considered in the swarm systems literature are presented here. A swarm consisting of N individuals/agents moving in an n-dimensional Euclidean space are considered in the following. $x_i \in \mathbb{R}^n$ denotes the state vector and $u_i \in \mathbb{R}^m, m \leq n$ denotes the control input of agent i. The state vector x_i may denote (a collection of) the position, orientation, synchronization frequency, information to be agreed upon.

Higher-Level Model

This model is the simplest mathematical model in the literature considering just the kinematics of the agents and does not deal with the lower level dynamics of agents. Thefore, it may be called also as higher-level or single integrator model. The agent motions in this model are given by

$$\dot{x}_i = u_i, i = 1, \dots, N, \tag{1.1}$$

where x_i is the state of agent A_i, and u_i is its control input. The *dot* represents the time derivative of states. The state x_i may be composed of the position p_i, and/or the orientation angle and/or synchronization frequency θ_i, and/or other variables.

This model is useful in studying the higher level control strategies and algorithms to obtain “proof of concept” type results for swarm behaviors. Usually, in the path planning control problems the higher level agent models are preferred; the trajectories generated using this model can be used as reference trajectories for the actual agents to track. The tracking controllers are considered in the lower level dealing with agent dynamics. However, still the effects of the actual agent dynamics should also be investigated in the models obtained by eEquation (1.1). In this thesis study, all the controllers are designed regarding the higher level agent models. Therefore, the results are “proof of concept” type results for swarm behavior. The lower level dynamics are left as future works of this study.

The model in (1.1) is a realistic simplified kinematic model for a class of *omni-directional* mobile robots with so-called *universal* (or *Swedish*) *wheels* [32, 33, 34]. As example works using this agent model the reader may refer to [35, 36, 37, 17, 18, 19].

Point Mass Model

Point mass model is another dynamic model commonly used in the multi-agent coordination and control literature and also called as*double integrator* model. The agent dynamics are given by

$$\begin{aligned} \dot{p}_i &= v_i, \\ \dot{v}_i &= \frac{1}{m_i} u_i, i = 1, \ldots, N, \end{aligned} \tag{1.2}$$

where p_i is the position, v_i is the velocity, m_i is the mass of the agent, and u_i is the force input. The state of the systems can be defined as $x_i^\top = [p_i^\top, v_i^\top]$. This model is relevant with many biological and engineering systems and has been considered in [38, 39, 40, 41, 20, 42, 19].

Fully Actuated Model

Compared to the higher-level and the point mass models, a more realistic model for agent dynamics is the following model

$$M_i(p_i)\ddot{p}_i + f_i(p_i, \dot{p}_i) = u_i, 1 \leq i \leq N, \tag{1.3}$$

where p_i represents the position or configuration, $M_i(p_i) \in \mathbb{R}^{n \times n}$ is the mass or inertia matrix, $f_i(p_i, \dot{p}_i) \in \mathbb{R}^n$ represents the centripetal, Coriolis, gravitational effects and additive parameters. In [32, 33, 34], this model is utilized for omni-directional mobile robots and for some fully actuated manipulators. There are some more realistic studies considering the uncertainties and disturbances in the force input and unknown mass/inertia matrix as well. These uncertainties provide an opportunity for developing robust control strategies as considered, e.g., in [24, 43].

Non-Holonomic Unicycle Model

The motion of agents in 2-dimensional space is also described by the unicycle model in the literature. Many mobile robots used for experiments in the laboratories (e.g., robots with differential drive) obey this model. The equations for this model are

$$\begin{aligned}
\dot{p}_{ix} &= v_i \cos(\theta_i), \\
\dot{p}_{iy} &= v_i \sin(\theta_i), \\
\dot{\theta}_i &= \omega_i, \\
\dot{v}_i &= \frac{1}{m_i} F_i, \\
\dot{\omega}_i &= \frac{1}{J_i} \tau_i, 1 \leq i \leq N,
\end{aligned} \tag{1.4}$$

where p_{ix} and p_{iy} are position components in the cartesian coordinates, θ_i is the orientation, v_i is the translational speed, and ω_i is the angular speed, m_i and J_i are mass and the moment of inertia of each agent, respectively. F_i and τ_i are the control inputs (force and torque inputs, respectively). In [44, 45, 36, 37, 46, 47, 48, 49], the kinematic part and the remaining parts of the equations in (1.4) are utilized in multi-agent models. The studies considering the feedback linearization techniques by adding one more integrator to the force input are [49, 50].

Dubins' Vehicle Model

Some of the studies on connection between oscillator synchronization and collective motions use multi-agent systems with constant translational speed [51]. This model may be written as

$$\begin{aligned} \dot{p}_i &= e^{j\theta_i} \\ \dot{\theta}_i &= u_i \end{aligned} \qquad (1.5)$$

where p_i is the position vector in complex coordinate frame (here $j = \sqrt{-1}$) and θ_i is the orientation of this vector. The speed is normalized with magnitude $v = 1$. The slightly general form of this model also called as "Dubins' vehicle model" is in utilized in [52, 53, 47] and given by

$$\begin{aligned} \dot{p}_{ix} &= v\cos(\theta_i), \\ \dot{p}_{iy} &= v\sin(\theta_i), \\ \dot{\theta}_i &= \omega_i, 1 \leq i \leq N, \end{aligned} \qquad (1.6)$$

where p_{ix} and p_{iy} are the position components in cartesian coordinates. This model is useful in studies considering the constant translational speed like unmanned aerial vehicles (UAV) at a specified altitude [53, 54, 53, 55]. In this thesis study the line following and circling behaviors are also investigated with constant speed agents and are applicable to UAV's.

Self-Propelled Particle Model

The model of self-propelled particles considered by Vicsek [56] is similar in nature to the model of Reynolds [9], except that the particles in the Vicsek's model have constant speed. In that work, they considered a self-propelled particle system with dynamics based on the simple rule "at each time step a given particle driven with a constant absolute velocity assumes the average direction of motion of the particles in its neighborhood of radius r with some random perturbation added" [56], and investigated clustering, transport, and phase transition in non-equilibrium systems. They showed that their model results in a rich, realistic dynamics despite the simplicity of the model. The model is described as

$$\begin{aligned} p_{ix}(t+1) &= p_{ix}(t) + v\cos(\theta_i(t+1)), \\ p_{iy}(t+1) &= p_{iy}(t) + v\sin(\theta_i(t+1)), i = 1, ..., N, \end{aligned} \tag{1.7}$$

where p_{ix} and p_{iy} are the position components in Cartesian coordinates, v is the translational speed, and θ_i is the steering angle of a particle with index i. The directions of motion are updated at each step based on

$$\theta_i(t+1) = \omega_i(t), i = 1, ..., N, \tag{1.8}$$

where the control input $\omega_i(t)$ is calculated based on the current direction of the agent and the direction of its neighbors with some additive noise. Note that the self-propelled particle model in (1.7)-(1.8) is the discrete time equivalent of the Dubins' vehicle model in (1.6).

In [57] and [58] Czirók et al. study biologically inspired, inherently non-equilibrium models consisting of self-propelled particles. Similar to [56] the particles move on a plane with constant speed and interact with their neighbors by choosing at each time step a heading equal to the average direction of their neighbors. In [57], they showed that the far-from-equilibrium system of self-propelled particles can be described using the framework of classical critical phenomena and the analysis show new features when compared with the analogous equilibrium systems. In [58] the authors summarize some of the results of large-scale simulations and theoretical approaches about the effects of noise and dimensionality on the scaling behavior of such systems. In [59] the authors introduce a generic phenomenological model for the collective motion of bacteria on a solid agar surface taking into account nutrient diffusion, reproduction, and sporulation of bacteria, extracellular slime deposition, chemo-regulation, and inhomogeneous population. The model is based on a ferromagnetic-like coupling of the velocities of self-propelled particles and is capable of describing the hydrodynamics on the intermediate level. In [60] the authors demonstrate that a system of self-propelled particles exhibits spontaneous symmetry breaking and self-organization in one dimension. They derived a new continuum theory that can account for the development of the symmetry broken state. The collective motion of organisms in the presence of fluctuations is discussed in [61]. In this study Vicsek utilized the simple rule of motion of particles as in [56]. The author demonstrated that there is a transition from disordered to ordered motion at the finite noise level and particles segregate into lanes or jam into a crystalline structure in a model of pedestrians.

In [62], Savkin gives a qualitative analysis of the dynamics of a system of several mobile robots coordinating their motion using simple local nearest neighbor rules referring to Vicsek's model in [56]. The author states that under some assumptions the headings of all robots will be eventually the same. Similar analysis was performed by Jadbabaie et. al. in [63], where they consider both discrete and continuous models as well as leaderless and leader-based situations and show that under certain connectivity conditions the heading of all the agents will converge to the same value, thus providing in a sense a theoretical explanation to the results obtained by Vicsek et. al (i.e. they considered the model by Vicsek without the additive noise and the position dynamics). Later these results were extended by Moreau [64] and independently by Ren and Beard [65] to more general classes of systems. A discrete model consisting of self-propelled particles that obey simple interaction rules is studied in [66]. The authors showed that the model can self-organize and exhibit coherent localized solutions in one-dimensional and in two-dimensional space. Furthermore, they develop a continuum version of their discrete model and show the agreement between these models.

1.2.2 Swarm Coordination and Control Problems

There exist a number of different swarm coordination and control tasks investigated in the systems and control literature. The most common ones of these studies focus on aggregation and foraging, flocking, rendezvous, formation stabilization, formation acquisition, formation reconfiguration, formation maintenance, agreement, cohesive motion and cooperation. For the brief explanations of these studies please refer to [31].

In this part of the literature survey we will mention some studies on aggregation and flocking problems. *Aggregation* (or gathering together) is a basic behavior that many swarms in nature exhibit. Moreover, many of the collective behaviors seen in biological swarms and some behaviors to be possibly implemented in engineering multi-agent dynamic systems emerge in aggregated swarms. Therefore, developing mathematical models for swarm aggregations and studying the dynamics and properties of these models have been an important study subject. Aggregation in biological swarms were initially modelled and simulated by biologists [11, 13, 12, 14]. Inspired by these works, a recent series of studies [17, 18, 19, 20, 21, 22, 23, 24] has provided rigorous stability and convergence analysis of swarm aggregations based on artificial potential functions both with continuous-time and discrete-time formulations. Particularly, in [17, 18] a biologically inspired n-dimensional (where n is arbitrary) continuous time synchronous swarm model based on artificial potentials is considered and some results on cohesive swarm aggregation have been obtained. Similar results based on artificial potentials and virtual leaders have been independently obtained by Leonard and coworkers in [38, 39] for agents with point mass dynamics. In [24], which has more emphasis on design than analysis a particular control strategy for aggregation in swarms has been developed based on artificial potential functions and sliding mode control, assuming simple integrator agent dynamics with model uncertainties and disturbances.

Later in [67] the results in [24] were extended to a significantly more realistic and more difficult setting with non-holonomic unicycle agent dynamics models, again using the tools of artificial potential functions and sliding mode control, but in a slightly different way than [24]. Furthermore, in [68] the results were further exteded to include the foraging and formation control problems (in addition to the aggregation problem considered in [67]).

The social foraging behavior in biological swarms usually require the aggregation of the swarm. This increases the probability of success for the individuals of success for the individuals [69, 15]. The *ant colony optimization* method [3], and *particle swarm optimization* methods are some of the resulting methods of studies on bio-mimicry of foraging.

The collective motion behavior of large number of interacting agents with a common group objective is defined as the *Flocking* behavior. The work by Reynolds [9] is the first study in the literature on flocking. This work has proposed (i) separation, (ii) alignment, and (iii) cohesion rules to implement a flocking behavior. These rules have been used in the simulations of the flocking behavior of animal swarms.

Initial studies on flocking from control theoretic perspective were performed by Tanner and coworkers in [40, 41] using point mass and in [70] considering non-holonomic agents with continuous time dynamics. On the other hand, in a recent study [42] Olfati-Saber developed a theoretical framework for analysis and design of flocking systems with agents with point mass dynamics. He considered two different types of flocking algorithms (which incorporate Reynolds rules): free flocking, in which the agents try to move to a particular distance from its nearest neighbors and also to stay aligned to them, and constrained flocking in which the agents are following virtual agents while performing free flocking. The second algorithm is in principle centralized although it can also be implemented in a decentralized fashion if all the virtual agents for all the individuals have the same dynamics and exchange information initially.

The flocking behavior of multi-agent systems are modeled to work synchronously in many of the studies. On the other hand there have been some studies [71, 21, 22, 23, 72] on the asynchronous modeling of multi-agent systems. The work in [71] considers the asynchronous convergence of a linear swarm to a synchronously achievable configuration in the reconfiguration of patterns problems. In this study a sufficient condition for the asynchronous convergence of a linear swarm to a synchronously achievable configuration is proven to exist. In [21, 22, 23] the stability of one-dimensional and M-dimensional asynchronous swarms are studied. They focus on asynchronous swarm models with time delays for swarm aggregation in discrete-time settings. In [72] Beni shows that asynchronous swarms may converge in cases in which synchronous swarms may not and that achieving an order from disordered actions is a basic characteristic of swarms and states that ”swarms may undergo a transition from non-convergence to convergence as their degree of partial synchronicity diminishes”. A study on the aggregation problem is performed in [73] with agents that are anonymous, homogeneous, memoryless, and lack communication capabilities. In a similar study in [74] the authors showed that asynchronous autonomous agents which have limited visibility and no memory, would gather at the same location in finite time although they are totally asynchronous in their actions, computations, and movements, provided that they have a compass. A systematic analysis of probabilistic aggregation strategies in swarm robotic systems is presented in [75], which considers four basic behaviors of the agents -*obstacle avoidance*, *approach*, *repel*, and *wait*- for aggregation. Similarly, in [76], the effects of different evolutionary parameters on the performance and scalability of system are studied.

The *formation stabilization* is the task of convergence of a group of agents that are initially at random positions to a particular geometrical configuration. Note that, the aim is not necessarily achieving a match with a pre-defined geometric pattern, instead it is the construction of a structured formation. In the literature the studies [77, 78, 79, 46, 80, 47] deals with the formation of geometric shapes. The *Formation maintenance* and *cohesive motion control* problems deals with the maintenance of an achieved formation structure of a swarm during any continuous motion of the swarm [78, 43, 37].

Some of the swarm coordination and control studies focus on formation reconfiguration and switching behaviors. The studies [81, 82, 83, 84] includes the maintenance of rigidity and persistence during certain changes or operations on the formation structure. These operations are *merging*, *splitting*, and *agent loss*. Merging is combining of two formations via some information links in between to form a single post-merged rigid formation. Splitting is the "reverse" of merging, i.e. division of a pre-split formation into two post-split smaller formations via breaking some of the information links. And the agent loss, is the break down of information link of one or more agents. In *formation switching*, the swarm changes from one shape to another, i.e. as a reaction to environmental changes. In [85] and [86] the formation switching problems are considered. The line following and circling behaviors of agents in this study are also investigated in the means of formation switching. The controllers switch at predefined time instances in order and the resulting behaviors are examined.

The development of distributed or decentralized control strategies for agreement on some information (agent position, velocity, oscillation phase, decision variable, etc.) are called as *distributed agreement problems* or *distributed consensus seeking*. The study [87], consists of survey on distributed agreement problems. *Agreement* or *consensus* is achieved if the corresponding states of all agents converge to the same value. The studies [88, 89, 90, 91, 92] deals with the consensus seeking problem of multi-agent systems.

1.2.3 Modeling, Coordination, and Control Approaches

Most of the swarm application controllers are considered to be decentralized. The centralized controllers (e.g., a central commander) are complex and have high computational costs, they are sensitive to loss of agents, and communication delays. Therefore, controllers running individually on each agent has advantages with respect to the central control schemes considering the mentioned issues. In the literature there is a large variety of approaches and techniques used to develop decentralized coordination and control strategies. For example, based on the higher level model (1.1) using potential functions [17, 18, 19], the point mass dynamics (1.2) using potential functions [80], the non-holonomic dynamics in (1.4) using Lyapunov analysis [45, 44] and feedback linearization [49], the fully actuated uncertain dynamics in (1.3) using potential functions and sliding mode control [24], etc. are some of these studies.

The approaches for swarm coordination and control considered in the literature based on *artificial potential functions*, *Lyapunov analysis* and other nonlinear control techniques, *sliding mode control* and *feedback linearization*, *neuro-fuzzy* techniques, *behavior modelling*, *probabilistic* and *evolutionary* methods, etc. as well as hybrid approaches combining two or more of these techniques are examined extensively in [31]. Here, we will mention some leading studies on these techniques.

Some of the robot navigation and control studies had utilized the artificial potential functions [93, 94]. One of the first application of these functions for multi-robot systems is done by Reif and Wang [95]. Artificial potential functions are being used for swarm aggregations, formation stabilization and acquisition, and some other multi-agent coordination and control tasks. For example, in [17, 18, 19] attraction/repulsion functions have been used for swarm aggregations while in [38, 39] similar potentials have been used for control of a group of point-mass agents. Potential functions are used for formation stabilization in [44, 80], while they are used for generating hunting behavior in fully actuated robot troops in [35]. The potential functions may represent only the inter-agent interactions as in [17, 18] or may include also environmental effects as in [19, 39, 20, 35] or may be defined for some other purpose. Some of the works address directly the issues of collisions between the agents, while some do not. One approach to avoid collisions using artificial potentials may be to use unbounded repulsion functions to guarantee collision avoidance [38, 18]. The controlled dynamics of swarms with artificial potential functions are usually analyzed with Lyapunov methods.

Spears and Gordon [96] have addressed the problems of formation stabilization, acquisition, maintenance, formation and cohesion during motion by introducing the *artificial physics* based approach which is a subclass of potential function based methods. It is a method based on the fundamental laws of physics, particularly mechanics, such as the Newton's laws of motion. Furthermore, hey have analyzed the problem of chemical plume tracing [97, 98] with simulation and implementation on real robots. In cases in which analytical analysis is intractable they apply evolutionary methods to learn parameter settings of the system. Moreover, they've developed an online learning algorithm that adjusts the system parameters in real-time in dynamic environments.

Another controller method in swarm systems is the sliding mode controller especially utilized in the lower level dynamics to obey the higher level controller commands. The sliding mode control method [99] is a method in which a switching controller with high enough gain is applied to suppress the effects of modelling uncertainties and disturbances, and the agent dynamics are forced to move along a stabilizing manifold called *sliding manifold*. The value of the gain is computed using the known bounds on the uncertainties and disturbances. Given the agent dynamics, using the sliding mode control technique, it is possible to design each of the control inputs u_i to enforce satisfaction of the trajectories generated by the *higher-level* model [24]. The main advantage of the sliding mode approach is that it works despite the existence of uncertainties and disturbances in the agent dynamics. This is mainly because of the suppression and robustness properties of the sliding mode control method. On the other hand the shortcomings of the method are the so-called chattering effect and possible generation of high-magnitude control signals. These shortcomings may possibly be avoided or relaxed via integration and some other filtering techniques. Application of the sliding mode approach with complex agent dynamics models is currently being investigated by researchers.

The nonlinear dynamics of agents are also studied to be linearized by feedback linearization techniques in the literature [49, 100]. The linearization of the system dynamics may allow design of linear or non-linear type of controllers for e.g. agent position regulation, formation stabilization. In this thesis study, the line following and circling behaviors are also investigated considering the feedback linearization techniques.

The asymptotic stability and convergence properties in coordination and control of swarms are usually analyzed with *Lyapunov* or *Lyapunov-like* functions ([63, 101, 42, 36, 17]). The Lyapunov functions may also be utilized in the controller design stage, e.g. [102]. Beside the Lyapunov-based ones, there exist some other nonlinear control and mathematical tools i.e. *separation - separation* and *separation - bearing* controllers [45, 78, 79], the concept of *virtual leader* [38, 103, 37], *continuous switching* [37] employed in the swarm coordination and control literature. Furthermore, there exist a number of other studies in the literature on applications of various nonlinear control frameworks, such as *neural networks*, *dynamic inversion*, *backstepping*, *adaptive control*, *output regulation* etc. ([104, 105, 106, 27, 107]).

Another common approach for coordinating groups of robots is the *behavior based* approach. One of the first studies using behavior based approach is the work by Reynolds [9]. In a more recent study in [10] Balch and Arkin present and evaluate a reactive behavior based approach for formation stabilization of *line* (robots traveling in a line parallel to each other), *column* (robots traveling behind each other), *diamond* (robots traveling in a diamond shaped formation), and *wedge* (robots traveling in a ”V“ shaped formation). They also integrate the formation behaviors with other navigational behaviors such as avoiding collisions with obstacles and other robots, reaching goals/targets, etc. The algorithms developed by authors includes three main techniques: *Unit Center Referencing*, *Leader Referencing*, and *Neighbor Referencing*. The line formation (robots traveling in a line parallel to each other) is one of the behaviors achieved with feedback linearization techniques in polar coordinate frame dynamics of agents, in this thesis study.

Behavior based approaches have also been used for studying aggregation strategies in swarm robotic systems in [75] and Bahçeci and Şahin in [76]. In [75] three basic behaviors namely, *approaching, repelling*, and *waiting* together with *obstacle avoidance* are defined and a finite state machine with different probabilities for the switching between these behaviors is utilized. The cyclic pursuit and orientation agreement behaviors in this thesis study also include the finite state machine models for switching between similar behaviors.

Multi-agent dynamic systems in general act in asynchronous manner since they are naturally distributed systems. It is difficult to implement synchronous motion in multi-agent systems. In fact analysis of asynchronous systems is more difficult with respect to the synchronous ones. Therefore, many of the models and approaches in the literature consists synchronous models and controllers. For example the flocking behavior of multi-agent systems are modeled to work synchronously in many of the studies. On the other hand there have been some studies [71, 21, 22, 23, 72] on the asynchronous modeling of multi-agent systems as well. The work in [71] considers the asynchronous convergence of a linear swarm to a synchronously achievable configuration in the reconfiguration of patterns problems. In this study a sufficient condition for the asynchronous convergence of a linear swarm to a synchronously achievable configuration is proven to exist. In [21, 22, 23] the stability of one-dimensional and M-dimensional asynchronous swarms are studied. In [72] Beni shows that asynchronous swarms may converge in cases in which synchronous swarms may not and that achieving an order from disordered actions is a basic characteristic of swarms and states that ”swarms may undergo a transition from non-convergence to convergence as their degree of partial synchronicity diminishes”. A study on the aggregation problem is performed in [73] with agents that are anonymous, homogeneous, memoryless, and lack communication capabilities. In a similar study in [74] the authors showed that asynchronous autonomous agents which have limited visibility and no memory, would gather at the same location in finite time provided that they have a compass. In [23], authors consider asynchronous swarms in one-dimensional space with different rules for inter-individual interactions, and using results on contractive mappings developed for parallel and distributed computation in computer networks in [92], show that swarm stability or convergence to a *comfortable position* will be obtained under assumptions of the sector boundedness of the attraction/repulsion function and *total asynchronism* in the motion of the agents. Similar approach is taken also in [108] for showing convergence of asynchronous cyclic pursuit. Some part of this study is also presented in this thesis. Some other recent empirical studies on the flocking behavior of asynchronous multi-agent systems are the works in [109, 110]. In [109], the effects of the level of asynchronism and size of neighborhood on the clustering performance of a swarm of self-propelled particles are studied. In [110], on the other hand, rotation angle restrictions (a type of non-holonomic constraint) are imposed on the self-propelled particles and their effect on the performance of the system is investigated. In the chapters considering the cyclic pursuit and orientation agreement problems, the asynchronism in the multi-agent dynamic models are also investigated with analytical and simulation based methods.

Lastly, there are probabilistic approaches in the swarm behavior modeling literature. An example work on this approach from the biological literature is in [111] where a general continuous model for animal group size distribution is presented. Also an interesting comparative study is in [112], where authors compare four different approaches to modeling the dynamics of spatially distributed systems by using three different examples, each with different realistic biological assumptions. They show that the solutions of all the models do not always agree, and argue in favor of the discrete models. The studies [113, 114, 115] from the swarm robotics literature considers probabilistic models in aggregation behaviors of swarms.

1.3 Scope of the Thesis Study

In this thesis study, the agent dynamics are modeled as higher level dynamics (kinematics) simply described in (1.4). The controllers are developed for the proof of concept and they do not include the agent kinetic models. Agents are traveling in 2-D space in all of the models in the study. This is preferred for their application on mobile ground vehicles.

The study starts with one of the most simple swarm behavior, cyclic pursuit (Chapter 2). The agent i follows the next indexed agent $i + 1$, and the last agent (n'th agent), follows the first one (i'th agent). The contribution of the study is the examination of asynchronism between agents. The study resulted in the proof of convergence of the swarm behavior under finite asynchronism. Furthermore, some simulation and experimental results are presented.

In the next Chapter (3), the orientation agreement problem is studied. Three different agreement strategies are developed and compared in this Chapter. The analytical and simulation based results are presented and discussed.

The proportional controllers for the speed and angular velocity of the agents for achieving approaching and circling behavior around a static target and approaching to the line passing through the target with a pre-specified orientation are developed in Chapter 4. The Lyapunov stability analysis are performed for the proposed controllers and the results are validated by simulations.

The circling and line following behaviors are investigated; furthermore, in Chapters 5 and 6. The controllers are first developed for single agents with feedback linearization techniques. Then, they are redesigned for multi-agent systems. The chapters include SISO, MIMO and Multi-agent system controllers and their validations via simulations.

The discussions and future directions belonging to each chapter are given at the end of chapters.

In the appendix, we present the experimental set-up developed for the practical application of controller strategies. The set-up structure (E-puck robots, camera, software etc.) is presented in this part of the study.

CHAPTER 2

Asynchronous Cyclic Pursuit

In this part of the thesis study, we focus on the problem of a multi-agent system performing cyclic pursuit with asynchrony in motion and sensing. Cyclic pursuit behavior had been studied in detail in the literature. However, these studies mostly focus on the assumption that the agents are somehow synchronized.

The very first scientist worked on the mathematics of *pursuit curves* was the French scientist Bouguer (c. 1732) [116]. In 1877, Lucas asked what trajectories would be generated if three dogs, initially placed at the vertices of an equilateral triangle, were to run one after the other? Brocard showed that each dog's pursuit curve would be that of a logarithmic spiral and that the dogs would meet at a common point (*Brocard point*) [116]. Klamkin and Newman [117] showed that, three bugs in cyclic pursuit which are not initially collinear, will meet at a point and this meeting will be mutual. Behroozi and Gagnon [118] later on proved that if all the bugs have the same speed and a nonmutual capture occurs, then this capture should be a head on collision. Richardson [119] showed that for the n-bugs problem, the head on collision is possible even for non-collinear initial positions but the probability of this collision is zero if the initial positions of the bugs are determined due to a smooth probability distribution. Similarly, Bruckstein, Cohen, and Efrat [120] considered a deterministic continuous pursuit in cyclic order and with preassigned varying speeds.

A study on the aggregation problem is performed in [73] with agents that are anonymous, homogeneous, memoryless, and lack communication capabilities. In a similar study in [74] the authors showed that asynchronous autonomous agents which have limited visibility and no memory, would gather at the same location in finite time although they are totally asynchronous in their actions, computations, and movements, provided that they have a compass. A systematic analysis of probabilistic aggregation strategies in swarm robotic systems is presented in [75], which considers four basic behaviors of the agents -*obstacle avoidance*, *approach*, *repel*, and *wait*- for aggregation. Similarly, in [76], the effects of different evolutionary parameters on the performance and scalability of system are studied.

A particular version of pursuit problem is studied in [121] for a system of n wheeled vehicles which are subject to a single nonholonomic constraint. The study provides a full stability analysis for the special case when $n = 2$ and how the global behavior of the system can be shaped through appropriate controller gain assignments. The same authors showed in [122] that the equilibrium formations of the system are generalized regular polygons and studied the local stability of these equilibrium polygons. The authors extend their work by studying the stability of equilibrium formations for multiple unicycle systems in cyclic pursuit in [123] and provide a complete local stability analysis for the general case $n \geq 2$. The study of Lin, Broucke, and Francis [124] is similar to these in means of the convergence of agents under certain conditions. The problem of gathering to a point is studied by several researchers within different contexts and under different names such as synchronization, consensus seeking, rendezvous, and others [64, 63, 125, 126, 127].

In our study we use a finite state machine to describe the sequence of behaviors of each agent and a discrete asynchronous mathematical model on a higher-level. After presenting the proof of convergence for synchronous cyclic pursuit model, we analyze the properties of the asynchronous model. Finally, we provide some numerical simulations to illustrate the results of the study.

2.1 Asynchronous High-Level Model

Consider the architecture shown in Figure 2.1 which consists of three behaviors: *wait*, *sense and compute*, and *move*. During the *sense and compute* behavior the i^{th} agent gets (measures or receives by other means) the relative position of the $i + 1^{th}$ agent and computes its own next desired position or way-point. During the *move* behavior the i^{th} agent moves towards the computed way-point. During the *wait* behavior, the agent doesn't move or basically stays in place. These behaviors are arbitrated by using a finite state machine (FSM) in an infinite loop and in the sequence shown in Figure 2.1.

Since here we are concerned with cyclic pursuit the computed next positon of the i^{th} agent is always towards the sensed position of the $i + 1^{th}$ agent and during the move state the agent moves towards this way-point. We assume that each agent has a low level control which guarantees that the agent reaches the computed way-point in a finite time. We are not concerned with the low level dynamics and how the low-level control is implemented. Therefore, the analysis below is applicable for many systems with variety of different low-level vehicle dynamics including heterogenous swarms/systems (i.e. swarms consisting of more than one type of agents). Moreover, we ignore the issue of collisions between the agents. The resulting sequence of behaviors can be summarized as: Move towards the pursued agent. Wait for a predetermined time interval. Then sense the location of the next agent and move again towards that agent. In this system we assume that the agents are ordered from 1 to n. Agent i pursues agent $i + 1$ modulo n. In other words, the last (n^{th}) agent pursues the first one. The agents are assumed to move in 2-D space and the position of the agents is given by

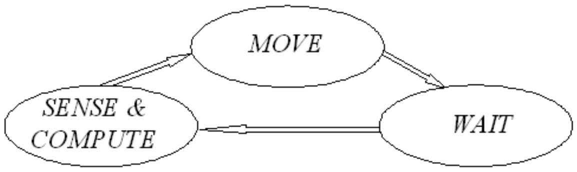


Figure 2.1: Finite State Machine Model

$$z_i(t) = [x_i(t), y_i(t)]^T \in R^2, i = 1, 2, ..., n \tag{2.1}$$

Note, however, that this is not a critical assumption and the results developed will be valid also for $z_i(t) \in R^m$ ($m = 1, 2, ...$) for a finite positive integer m.

Recall that during the *sense and compute* behavior the i^{th} agent gets the position of the $i + 1^{th}$ agent and then computes its own next desired position or way point. However, during these sensing and computing processes of the i^{th} agent the $i + 1^{th}$ agent may be in its *move* state and therefore, the measured position of the next agent may be outdated. Moreover, the measurement of the position of the next agent may itself incure some delay. Whether ultrasonic, infrared or other type of sensors are used the propagation delay of the signals may lead to measurement of old (outdated) positions. Similarly delay will be also present even if the positions are obtained by inter-agent communication or by other means such as global positioning system. Therefore, the modeling of the dynamics of agents in cyclic pursuit should be designed including the position sensing delays. Referring to this phenomena we introduce the variables $\tau_{i+1}(t)$ which satisfy $0 \leq \tau_{i+1}(t) \leq t$ in order to represent the delay in the position measurements. In other words, we assume that at time t agent i knows $z_{i+1}(\tau_{i+1}(t))$ instead of the actual $z_{i+1}(t)$ about the position of agent $i + 1$. In other words, $z_{i+1}(\tau_{i+1}(t))$ is the *perceived position* of agent $i + 1$ by agent i at time t. Also since each agent operates on its own local clock following the state machine cycle on Figure 2.1 without a need for synchronization with the other agents, we introduce a set of time indices T^i, $i = 1, 2, ..., n$, at which the agent i updates its way-point z_i. It is assumed that at the other instances the agent i does not perform way-point calculation (it might be in one of other states/behaviors at these time instants). With these in mind the "high-level" dynamics of each agent can be represented as

$$\begin{aligned} z_i(t+1) &= (1-p)z_i(t) + p\, z_{i+1}(\tau_{i+1}(t)), \qquad t \in T^i \qquad (2.2) \\ z_i(t+1) &= z_i(t), \qquad t \notin T^i \end{aligned}$$

where p is the gain satisfying $0 \leq p \leq 1$ and as mentioned above the variables $\tau_{i+1}(t)$, $i = 1, \ldots, n$, are used to represent the time index of the position information of the $i + 1^{th}$ agent. These variables satisfy $0 \leq \tau_{i+1}(t) \leq t$ for all $t \in T^i$ and for all i. If agent i has not yet obtained any information about the $i + 1^{th}$ agent's position and still has the initial position information, then $\tau_{i+1}(t) = 0$ whereas $\tau_{i+1}(t) = t$ means that agent i has the current position information of the $i + 1^{th}$ agent. The difference between the current time t and the value of the variable $\tau_{i+1}(t)$ is the delay occurring due to the sensory, computing and/or communication processes or other reasons.

In equation (2.2), the elements of the set $T^i \subset \{0, 1, 2, ...\}$ are the indices of the sequence of ordered physical times $\mathcal{T} = \{t_0, t_1, t_2, \ldots\}$ similar to the times of events in discrete-event systems where $t_i < t_{i+1}$ are the time instants at which the events in the system occur. The times t_i do not need to be equally spaced, i.e., the intervals $t_{i+1} - t_i$ do not have to be equal. Referring to the FSM model in Figure 2.1 during the time interval between two subsequent indices of T^i the agent performs its *move*, *wait*, and *sense and compute* behaviors. As expected the completion of the sequence of the behaviors may take different time intervals for different agents and for the same agent at different steps. For instance the distance of the way-point of the agents may change at each step and so the *move* states may last for different amounts of times. Since behaviors of agents last for different time intervals, each agent has its own time set, T^i and these time sets are independent. However, it is possible to have $T^i \cap T^j \neq \emptyset$ for $i \neq j$ which means that sometimes two or more agents may update their state simultaneously. Note that the set T^i is needed only for analysis purposes and in order to implement the iteration in (2.2) it is not required for the agents to know it. Similarly, the agents do not need to know neither the sets T^i nor the set of physical times $\mathcal{T}$. Therefore, there is no need for a global clock or means for synchronization for implementing equation (2.2) and each agent can operate based on its internal logic and using only its local clock without a need for synchronization. Before analyzing the convergence performance of this proposed asynchronous model, we will focus on the synchronous case in the following section, after which the asynchronous case will be analyzed in detail.

2.2 Convergence Under Total Synchronism

In this section we assume that the agents are synchronized and analyze the systems behavior based on this assumption. From practical point of view synchronism is hard to implement in swarm of individual agents with decentralized control since each agent has different duration of states. Still we analyze the convergence of the synchronous case because later in the following section we will use the results from this section to establish the stability of the asynchronous case. We start with the assumption of no delay in the position information. In particular we assume that $\tau_{i+1}(t) = t$ for all i and that $T^i = T = \{0, 1, ...\}$ for all i. In other words, all of the agents will move at the same time instants and each one knows the current position information of the agent it pursues. With respect to this assumption the dynamics of the model become

$$z_i(t+1) = (1-p)z_i(t) + p\, z_{i+1}(t) \tag{2.3}$$

Writing these equations in matrix form we obtain.

$$z(t+1) = Az(t) \tag{2.4}$$

where $z(t) = [z_1(t)\; z_2(t)\; \dots\; z_n(t)]^T \;\; \in \mathbb{R}^{n\times 2}$ and

$$A = \begin{bmatrix} 1-p & p & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & \ddots & p \\ p & 0 & \dots & 0 & 1-p \end{bmatrix}$$

The stability of equation (2.4) depends on the eigenvalues of A. All eigenvalues of the state matrix A should lie within the unit circle. To show that this is the case we will use a result from matrix theory. In particular, we will use Gerchgorin's Theorem [128] which we present below for the convenience of the reader.

Gershgorin's Theorem Let $A_{n\times n} = [a_{ij}]$, and let

$$R_i(A) \equiv \sum_{j=1, j\neq i}^{n} |a_{ij}|, \quad 1 \leq i \leq n \tag{2.5}$$

denote the *deleted absolute row sums* of A. Then, all the eigenvalues of A are located in the union of n discs

$$\bigcup_{i=1}^{n} \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i(A)\} \equiv G(A)$$

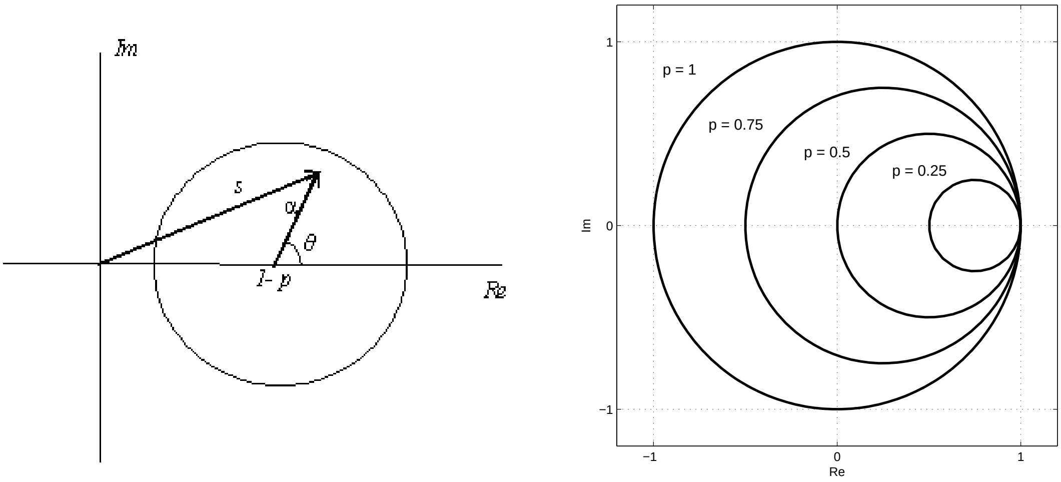
where $\mathbb{C}$ denotes the complex plane.

Therefore, for an $n \times n$ square matrix A, n circles can be drawn with centers at the diagonal elements of A, i.e., a_{ii}, $i = 1; 2; ...; n$ and with radius of each of the circles equal to the sum of the absolute values of the other elements in the same row, that is, $\sum_{j\neq i} |a_{ij}|$. Such circles are called Gershgorin's discs. Then all the eigenvalues of A lie in the region formed by the union of all the n discs. From Gershgorin's Theorem we know that all the eigenvalues of the matrix A in (2.4) are located within discs centered at $(1 - p)$ and having radius p. Then as seen in Figure 2.2a the vector of points, s in the smaller circle can be formed as $s = (1 - p) + \alpha e^{j\theta}$ where $\alpha \leq p$ is the distance of the eigenvalue to the Gershgorin's disc center. Then it can be shown that $|s| = (1 - p)^2 + \alpha^2 + 2(1 - p)\alpha\cos(\theta)$. Moreover, since $\alpha \leq p$ we have

$$(1 - p)^2 + \alpha^2 + 2(1 - p)\alpha\cos(\theta) \leq (1 - p)^2 + p^2 + 2(1 - p)p\cos(\theta)$$

and if $p \leq 1$ then $(1 - p)^2 + p^2 + 2(1 - p)p\cos(\theta) \leq 1$ and $(1 - p)^2 + \alpha^2 + 2(1 - p)\alpha\cos(\theta) \leq 1$ or basically $|s| \leq 1$ is satisfied. Therefore, the eigenvalues of matrix A are within the unit circle if $p \leq 1$. The circles that enclose the location of eigenvalues for the values of $p = 0.25,\ 0.50,\ 0.75,$ and 1.00 are plotted in Figure 2.2b. Note that the circle for $p = 1$ is indeed the unit circle.

Another issue to note here is that one of the eigenvalues of the matrix A is always on the unit circle at $\lambda = 1$ and the convergence point of the system depends on that eigenvalue. We can simply show that for the $n \times n$ state matrix A in (2.4) the characteristic polynomial is



(a) (b)

Figure 2.2: (a) Gershgorin disc with center at $1 - p$ and radius p. (b) Gershgorin discs for $p = 0.25,\ 0.50,\ 0.75$, and 1.00.

$P = (1 - p - \lambda)^n + p^n(-1)^{n-1}$. Note that one of the roots of this characteristic polynomial is always $\lambda = (1 - p) + p = 1$ (as stated above) while all the other eigenvalues are within the unit circle. The eigenvector corresponding to this eigenvalue is $\alpha = [1\ 1\ \ldots\ 1]^T$. Now, the solution of (2.4) can be written as

$$z(t) = (\lambda_1)^t \alpha_1 c_1 + (\lambda_2)^t \alpha_2 c_2 + \ldots + (\lambda_n)^t \alpha_n c_n \tag{2.6}$$

where λ_i are the eigenvalues of A and α_i are the corresponding eigenvectors and c_i are arbitrary constants which depend on the initial conditions. (Actually, since $z_i(t) \in \mathbb{R}^2$, $c_i = [c_{1i}, c_{2i}] \in \mathbb{R}^2$ are constant row vectors). Let $\lambda_1 = 1$ be the eigenvalue on the unit circle while $|\lambda_i| < 1$, $\forall i = 2, \ldots, n$ are the other eigenvalues. Then the solution in (2.6) will converge to:

$$\lim_{t \to \infty} z(t) = \alpha_1 c_1 = [c_1^T\ c_1^T\ \ldots\ c_1^T]^T$$

which means that all agents will reach to the same point, $c_1 \in \mathbb{R}^2$ as $t \to \infty$.

Now based on the above convergence result of the synchronous system we will define a sequence of (contracting) sets which will be useful later on in the proof of the asynchronous case. Let us define

$$Y(t) = \{y \in \mathbb{R}^2 | m(t) \leq y \leq M(t)\} \subset \mathbb{R}^2 \tag{2.7}$$

where $m(t) = \min_{i=1,\ldots,n}\{z_i(t)\}$ and $M(t) = \max_{i=1,\ldots,n}\{z_i(t)\}$ where the inequality sign and the *minimum* and *maximum* operators are operated elementwise. Note that the sequence $m(t)$ is non-decreasing and the sequence $M(t)$ is non-increasing. In other words, we have $m(t+1) \geq m(t)$ and $M(t+1) \leq M(t)$ for all t. Moreover, one can show that there exists a finite $\mu > 0$ such that $m(t+\mu) > m(t)$ and $M(t+\mu) < M(t)$ for all t. In fact it is guaranteed that a decrease in $M(t)$ and an increase in $m(t)$ occurs in a few time steps. We will do the below analysis as if $m(t) \in \mathbb{R}$ and $M(t) \in \mathbb{R}$ but similar analysis will hold also for the $m(t) \in \mathbb{R}^2$ and $M(t) \in \mathbb{R}^2$ case.

If $M(t)$ and $m(t)$ are not equal, (in other words, agents have not converged yet to a common point) then $M(t) > m(t)$. Let $I_M(t) = \{i | z_i(t) = M(t)\}$ and $I_m(t) = \{i | z_i(t) = m(t)\}$ denote the sets of agents located at time t at the maximum and the minimum, respectively. Also, denote with $\#(M) = |I_M(t)|$ and $\#(m) = |I_m(t)|$ the number of agents in these sets. Note that at least one of the agents in $I_M(t)$ and $I_m(t)$ is pursuing an agent outside of its corresponding set. Therefore, $\#(M)$ and $\#(m)$ both decrease at each step. This guarantees that $M(t)$ will decrease in at most $\#(M) - 1$ steps and $m(t)$ will increase in at most $\#(m) - 1$ steps from time t. The worst case occurs when half of the agents are at the maximum and the remaining half are at the minimum. Then both $M(t)$ and $m(t)$ do not change for $n/2 - 1$ steps. This implies that the interval between $M(t)$ and $m(t)$ contracts in at most $\mu = n/2$ time units. Then it is clear that $Y(t+\mu) \subset Y(t)$. Defining $Z(k) = Y(k\mu)$ as the sequence of contracting sets and in the light of the preceding convergence analysis, we may write

$$c_1 = Z \subset \ldots Z(k+1) \subset Z(k) \subset \ldots \subset Z(0)$$

Now for every k let us define $\bar{Z}(k)$ such that

$$\bar{Z}(k) = \underbrace{Z(k) \times Z(k) \times \ldots \times Z(k)}_{n} \subset \mathbb{R}^{n \times 2} \tag{2.8}$$

and note that $\alpha_1 c_1 = \bar{Z} \subset \ldots \bar{Z}(k+1) \subset \bar{Z}(k) \subset \ldots \subset \bar{Z}(0)$ is satisfied. We will use these definitions in the next section.

2.3 Convergence Analysis of the Asynchronous Model

In this section, we analyze the convergence properties of the asynchronous system. As mentioned before here each agent performs the behaviors at totally different time instants. Formally z_i's are updated at $t \in T^i$ where T^i for each agent are independent. Moreover, the sensing/measurement process may incur delays. We start with an assumption which establishes a bound on the maximum possible time delay as well as guarantees uniformity in the updates of the agents. The analysis here is based the results on parallel and distributed computation in [92].

Assumption 1 *There exists a positive integer B such that*
(a) For every i and every $t \geq 0$, at least one of the elements of the set $\{t, t+1, \ldots, t+B-1\}$ belongs to T^i.
(b) There holds $t - B < \tau_{i+1}(t) \leq t \ \ \forall i \ \ t \geq 0, t \in T^i$

Assumption 1 is a fairly realistic assumption since in any practical system the measurement/communication delays must be bounded. If an agent is unable to receive information for an unbounded amount of time from its neighbor which it tries to pursue, then it may not be able to follow/pursue it and the pursuit behavior looses its meaning. Similarly, in order for the system to work properly every agent should be able to move to its next way-point and complete the cycle in Figure 2.1 in a finite amount of time. Note, however, that the agents do not need to know the value of B.

Theorem 2.3.1 *For the multi-agent system in cyclic pursuit described by the equation in (2.2) under Assumption 1 as $t \to \infty$ the positions of all the agents will converge to a common point or basically*

$$\lim_{t \to \infty} z_i(t) = c \quad \forall i = 1 \ldots n \tag{2.9}$$

where c is some constant.

Proof. Given time $t_k \in T$ such that $z_i(t) \in Z(k)$ for all $i = 1, 2, ..., n$ and $t \geq t_k$, we will show that there exists a time t_{k+1} such that $z_i(t) \in Z(k+1)$ for all $i = 1, 2, ..., n$ and $t \geq t_{k+1}$. Therefore, let us assume that there exists a time $t_k \in T$ such that $z_i(t) \in Z(k)$ for all $i = 1, 2, ..., n$ and $t \geq t_k$. Consider agent i; from the asynchrony we know that there may be time delay in sensing the position of agent $i+1$ by agent i. Therefore, even though $z_{i+1}(t_k) \in Z(k)$, it might be the case that, $z_{i+1}(\tau_{i+1}(t_k)) \notin Z(k)$. However, by Assumption 1, the delay in the position information update is bounded by B steps. Therefore, at time t_k we have

$$t_k - B < \tau_{i+1}(t_k) \leq t_k \ \ \forall i \ \ t_k \geq 0, t_k \in T$$

Furthermore, for all $t \geq t_1 = t_k + B$ and for each agent $i = 1, 2, ..., n$ it is guaranteed that

$$t_k < \tau_{i+1}(t) \leq t \quad \forall\, t \geq t_1,$$

implying that

$$z_{i+1}(\tau_{i+1}(t)) \in Z(k) \quad \forall\, t \geq t_1$$

Recall also from Assumption 1 that the update of the positions of each agent is subject to the delay which is at most B steps. Then at time $t_2 = t_1 + B = t_k + 2B$, all the agents will have updated their position information. If the agent is at *maximum* and not pursuing an agent at *maximum*, then $z_i(t_2)$ will decrease and if the agent is at *minimum* and not pursuing an agent at *minimum*, then $z_i(t_2)$ will increase. Therefore, if there are at most one agent at each *maximum* and *minimum*, then from the result for the synchronous case in the preceding section, the position set will contract, implying $Z(t_2) \subset Z(k)$. However, recall the worst condition of agent topology in the synchronous convergence problem; the position sets were to converge in at most $\mu = n/2$ amount of steps. Applying this worst condition for the synchronous case together with the discussion above, we find that the position sets are guaranteed to contract in at most $2\mu B$ steps. Let us define $t_{k+1} = t_k + 2\mu B = t_k + nB$. Then it is guaranteed that $z_i(t) \in Z(k+1) \subset Z(k)$ for all $t \geq t_{k+1} = t_k + 2\mu B$. Since at the initial state we have $z_i(0) \in Z(0) \;\; \forall i = 1, 2, ..., n$ the induction is complete. Then using the result above we have

$$c = Z \subset ... \subset Z(k+1) \subset Z(k) \subset ... \subset Z(0)$$

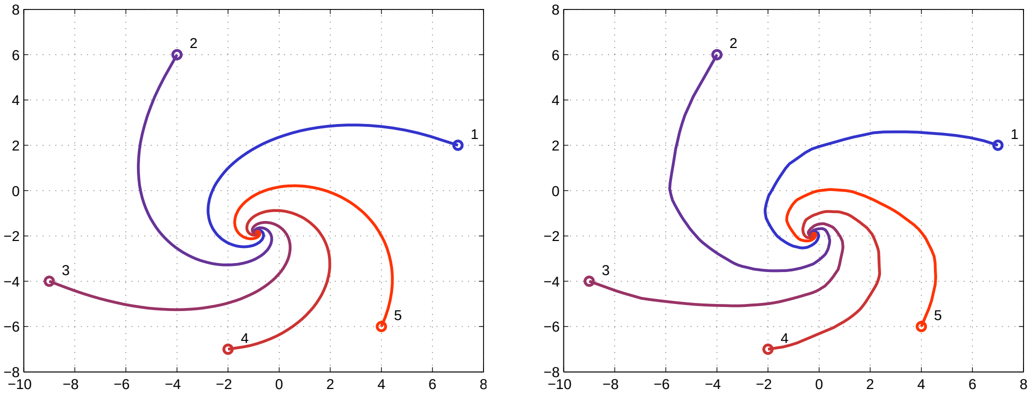
which implies the convergence of agents to a common point, $c \in \mathbb{R}^2$. ■

2.4 Simulation Examples

We simulated the cyclic pursuit for 5 agents. We performed simulation for both the synchronous and asynchronous cases in order to see the differences between the two cases and in particular the effects of asynchronism. The initial positions of agents are

$$S = \{(7,2), (-4,6), (-9,-4), (-2,-7), (4,-6)\}$$

The gain p for the updates is selected to be $p = 0.05$. In the synchronous case the agents converge to $Z_f = [-0.8750, -1.8365]$ after sufficiently long simulation interval. The trajectories of the agents are shown on Figure 2.3a. For the asynchronous case we used the same initial positions of agents and gain (p) value. In order to achieve asynchronism in simulation and also to simulate the delays in sensing and processing we integrated a probability mechanism that decides whether to update the position information of the $i+1^{th}$ agent in the system. In the following simulation sample the probability of update is chosen to be % 20. The result of the simulation for this case is on Figure 2.3b. The agents converge to $Z_f = [-0.2565, -1.9317]$. The convergence point in this case is different from the synchronous case, since the asynchronism in the actions of agents leads to pursuing of next agent with old position information and take the *move* action with some delay. This results in, a different sequence of contradicting sets and therefore, different final position.



(a) (b)

Figure 2.3: (a) Simulation results for synchronous convergence. (b) Simulation results for asynchronous convergence.

Moreover, in order to measure or compare the performance of these two systems we plotted the sum of the distances between the agent positions

$$e(t) = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \|z_i(t) - z_j(t)\|^2$$

in Figure 2.4 for both synchronous and asynchronous cyclic pursuits. It is seen that the synchronous cyclic pursuit converges faster than the asynchronous one. This is an expected result when we consider the delays in actions and position sensing of agents during asynchronous pursuit. Although not shown in Figure 2.4 $e(t)$ converges to zero in the asynchronous case as well.

Note that in the implementation here the value of the probability of update and sensing dictates the value of B. As this probability decreases the value of B increases and this decreases the speed of convergence.

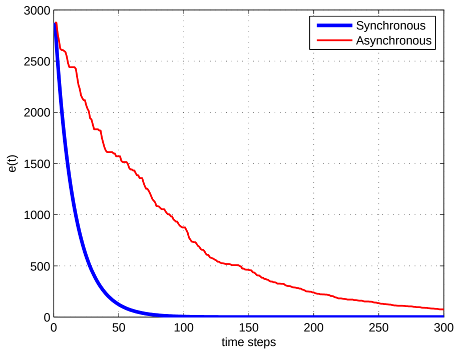


Figure 2.4: Convergence performance of synchronous and asynchronous pursuits.

2.5 Experimental Examples

Here we present the experimental result obtained using the set-up described in Appendix A. The experiment is performed for testing and verification of the problem of cyclic pursuit of a swarm of agents. We assumed that the agent dynamics are arbitrated by a finite state machine (FSM) with a sequence of the behaviors: Move towards the pursued agent; Wait for a predetermined time interval; then sense the location of the next agent and move again towards that agent. As a difference from this procedure, in the experimental application the agents were programmed so that they do not stop at the *wait* state, they continue to travel at the last velocity and orientation. In the simulations we assumed that each agent has a low-level control which guarantees that the agent reaches the computed way-point in a finite time. However, for the experimental application we had to implement such a low-level controller which will guarantee that the robot moves between two subsequent way points and had opportunity to observe low-level dynamics in the resulting behaviors as well.

In Figures 2.5 and 2.6 the results obtained for the cyclic pursuit of 5 E-puck robots in our set-up are shown. Comparing the Figures 2.3b and 2.5, we observe that the analytical and simulation based results are also verified by the experimental results. The frames at 1, 100, 150, and 195 time steps are shown in Figure 2.6. The robots are spread away at the beginning of the simulation. Each robot follows its leader and travels on spiral like path shown in Figure 2.6. At the end they converge to each other. Note that in these video frames there are additional virtual geometries drawn on and between the robots virtually. The lines show the connection between the follower and the leader. The colored dots on the robots show the left, right, and heading points of the robot hats as mentioned in Section A.3.

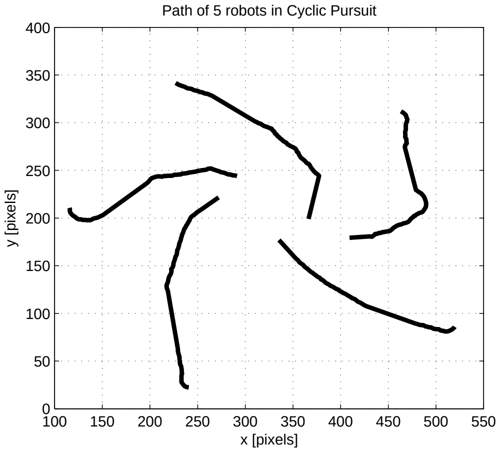


Figure 2.5: Path of 5 E-puck robots in cyclic pursuit obtained in the set-up.

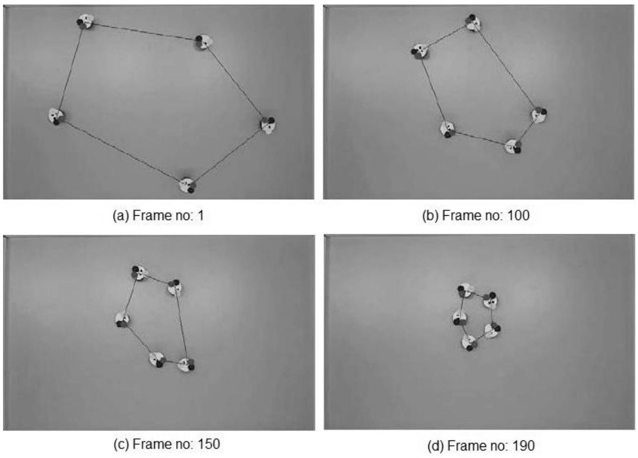

(a) Frame no: 1
(b) Frame no: 100
(c) Frame no: 150
(d) Frame no: 190

Figure 2.6: Cyclic Pursuit of 5 robots.

2.6 Conclusions

In this study we showed the convergence of the positions of n agents in cyclic pursuit with asynchronous dynamics to a common point. We assumed that the agents perform fundamental behaviors modeled by a finite state machine consisting of *wait*, *sense and compute*, and *move* states. To reach the proof of convergence of asynchronous pursuit we started with the convergence of synchronous cyclic pursuit. Then using the result for the synchronous case we showed that the asynchronous system will converge as well, despite the asynchronism and the time delays.

It is claimed that if convergence to a point is feasible, then more general formations are achievable as well [124]. However, it is not clear whether its possible or not to achieve convergence to any geometric formation using cyclic pursuit under asynchronism and time delays and this needs to be investigated further.

CHAPTER 3

Orientation Agreement Problem

3.1 Introduction

In this study, we compare three different orientation agreement strategies of multi-agent/particle systems under different conditions. Since in nature and in robotic applications the autonomous agents mostly act asynchronously, a model based on asynchronous actions of agents would be more realistic and implementable. Hence, in this study (based on our previous works [109, 129]) we will develop an asynchronous version of the model developed by Vicsek [56] (without the additive noise) and investigate the effects of asynchronism in the coordination of agents striving to travel with a common heading. We consider 3 different orientation rules (rules of dynamics to achieve a common heading) and compare the behavior of the self-propelled particle systems for these three different rules. Furthermore, we consider the effect of restricting the maximum turning angle of the particles and perform simulations for bounded and unbounded regions. We perform extensive simulations with many different initial conditions. Moreover, in the discussions section we provide some analytical explanations for the obtained simulation results.

Recently there have been some articles on extending the works in [63, 64, 65] to systems with time delays or systems operating asynchronously [130, 131, 132, 133, 134]. In [130] Angeli and Bliman provide an extension of the result by Moreau [64] by relaxing the convexity assumption and allowing for a known and bounded time-delay. In [132], besides discussing the some available results in the literature, some new results for systems/protocols with delays are presented as well. Asynchronous motion is not considered in [130, 132]. In [131] the recent results on synchronous consensus protocols are summarized, asynchronous protocols are discussed briefly, some open questions are posed , and some simulation based preliminary results on asynchronous protocols using a custom Java based simulator are shown. The more recent study in [133] presents some new results on the asynchronous agreement problem. No time delays are considered in [133]. The study in [134] considers the problem of asynchronous agreement of systems also incurring time delays and extends the results in [63, 64, 65].

Despite the fact that almost all of the above articles claim that they consider the Vicsek's model, in reality they consider only part of the dynamics of the model considered by Vicsek. The model considered is, in general, a linear averaging (or sometimes nonlinear convex/contracting) agreement model that does not include the agent position dynamics which are present in the Vicsek's model. Then the articles investigate the agreement properties in the dynamics of that partial model under some artificial connectivity assumptions. However, in the model by Vicsek the connectivity is an *emergent property* which depends also on the position dynamics of the agents. Different from the studies in [62, 63, 64, 65, 130, 131, 132, 133, 134], in this study we include also the position dynamics of the agents and study the performance of the system for three different agreement strategies for synchronous and asynchronous cases incurring also time delays. In addition, we impose also turn angle restrictions (a type of non-holonomic constraint) on the agents and investigate the performance for different levels of restrictions. For comparing the performances of the strategies we define several performance metrics, which include not only orientation agreement of the agents but also their clustering properties of the system. To the best of our knowledge, no study similar to this one has been performed so far in the literature.

To summarize, we investigate the behavior of multi-agent systems utilizing three strategies with different combinations of the following properties: (i) the multi-agent systems may be synchronous or asynchronous, (ii) they may travel in bounded or unbounded regions and (iii) the mobile agents may have turning speed restrictions. The agents/particles are assumed to move with constant speed and update their orientation of motion based on three different strategies. Based on these strategies simulations are performed and the effects on the clustering performance are investigated.

3.2 High Level Dynamics

We consider a multi-agent system consisting of n so called self-propelled or self-driven particles each of which, similar to the model by Vicsek [56], moves based on the dynamics

$$x_i(t+1) = x_i(t) + v\cos(\theta_i(t+1)) \tag{3.1}$$

$$y_i(t+1) = y_i(t) + v\sin(\theta_i(t+1)) \quad i = 1, ..., n \tag{3.2}$$

where $x_i(t), y_i(t) \in \mathbb{R}$ denote the cartesian position coordinates of agent i and $\theta_i(t) \in \mathbb{R}$ denotes its orientation angle at time t. We assume that v is constant and equal for all agents. In other words, we assume that all the agents move with the same constant speed in possibly different directions (determined by their orientation angles θ_i). Moreover, we assume that an agent has limited sensing capabilities and can "see" or "sense" the other agents that are within a circle of radius δ from it and call these agents its neighbors. Furthermore, it is assumed that an agent updates its orientation based on its current orientation and the orientation of its current neighbors. In particular, we will utilize three different orientation rules using which the agents will adjust their headings.

Many studies in the literature assume that the agents move synchronously and have perfect information about the orientations of their neighbors. In other words, it is assumed that the agents move simultaneously/synchronously and at each step they know the current positions/orientations of their neighbors. However, in real swarms this is hardly possible. Implementing such dynamics will require a global clock to be shared by all the agents. Therefore, an asynchronous model is more realistic in which each agent can move and reorient itself independently. Moreover, usually there might be time delays in the communication/sensing between the agents. Including such delays in the multi-agent systems will result in a more realistic approaches. In order to achieve such a realistic model we use a higher-level asynchronous model similar to the one used in [109].

For the asynchronous high level model we will refer to the study in [108] which considers the cyclic pursuit problem with asynchronous high level dynamics. In that study a finite state machine (FSM) is proposed for high level model. The architecture consists of three behaviors: *wait*, *sense and compute*, and *move*. In our model here the agents always move in the last updated direction with constant speed and a finite state machine works for the orientation dynamics. Therefore, the behavior considered here can be described with the *sense-compute-*

turn and *move straight* states (Figure 3.1).

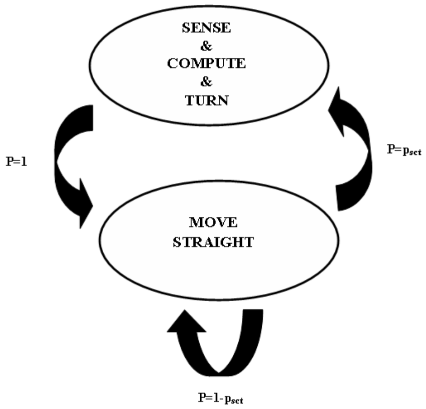


Figure 3.1: Finite State Machine Model

During the *sense-compute-turn* behavior the i^{th} agent gets (measures or receives by other means) the orientations of neighbor agents and computes its own next desired orientation and turns to the computed orientation. During the *move straight* behavior, the agent doesn't turn and basically moves along its last updated orientation. These behaviors are arbitrated by using a finite state machine, in an infinite loop as shown in Figure 3.1. As soon as the *sense-compute-turn* state completed agents immediately pass to the *move straight* state (with the probability of 1). However, *move straight* state is followed by the *sense-compute-turn* state depending on some probabilistic measure or reasoning. Let the following state of *move straight* state is *sense-compute-turn* state with probability of p_{sct} and again *move straight* state with probability of $1 - p_{sct}$.

We assume that each agent has a low level control which guarantees that the agent turns to the computed orientation with the specified angular speed. We are not concerned with the low level dynamics and how the low-level control is implemented. Therefore, the analysis below is applicable for many systems with variety of different low-level vehicle dynamics including heterogenous swarms/systems (i.e. swarms consisting of more than one type of agents). Moreover, in the current study we ignore the issue of collisions between the agents. The resulting sequence of behaviors can be summarized as:

- Turn to the computed orientation.
- Wait for a predetermined time interval.

- Sense the orientations of the neighbor agents and turn again in the computed orientation.

Recall that during the *sense-compute-turn* behavior the i^{th} agent gets the orientations of the neighbor agents and then computes its next desired orientation. However, during these sensing and computing processes of the i^{th} agent the neighbor agents may be in their *sense-compute-turn* state and therefore, the measured orientations of the neighbor agents may be outdated. Moreover, the measurement of the orientations of the neighbor agents may itself incure some delays. Whether any type of sensors or even communication are used the propagation delay of the signals may lead to measurement of old (outdated) orientations. Similarly delay will be also present even if the orientations are obtained by inter-agent communication. Therefore, the modeling of the dynamics of agents working for a common orientation problem should be designed taking into account the orientation sensing delays. Referring to this phenomena we introduce the variables $\tau^i_j(t)$ which satisfy $0 \leq \tau^i_j(t) \leq t$ in order to represent the delay in the orientation measurements. In other words, we assume that at time t agent i knows $\theta_j(\tau^i_j(t))$ instead of the actual $\theta_j(t)$ about the orientation of agent j. In other words, $\theta_j(\tau^i_j(t))$ is the *perceived orientation* of agent j by agent i at time t. Also since each agent operates on its local clock following the state machine cycle on Figure 3.1 without a need for synchronization with the other agents, we introduce a set of time indices T^i, $i = 1, 2, ..., n$, at which the agent i updates its orientation θ_i where the sets T^i are independent subsets of the set $\{0, 1, 2, ...\}$. It is assumed that at the other instances the agent i does not perform orientation calculation (it might be in one of the other states/behaviors at these time instants). Note that in the synchronous model $\tau^i_j(t) = t$ and $T^i = \{0, 1, 2, ...\}$ for all $t > 0$ and $i = 1, 2, ..., n$ and $j = 1, 2, ..., n$. In other words, in the synchronous case all the agents have the exact and current orientation information of their neighbors and perform updates at all time instants simultaneously/synchronously.

We believe that the asynchronous model is more realistic (compared to the studies performed earlier with synchronism assumption) since in real world applications (such as robots coordinating to achieve a common orientation) or animal flocks (such as schooling behavior of fishes) usually there is no synchrony between agents and time delays are also possible. In our model we utilize the study on the relation between the synchronism and asynchronism in the parallel computing systems in [92].

The asynchronism between the agents (i.e. robots, fish) may be at different levels due to the characteristics of each agent itself or some environmental disturbance. In some multi-agent systems the asynchrony may be negligible (leading to a synchronism assumption) and the behavior of these systems may be computable or predictable. On the other hand, in some systems asynchronism may drastically change the performance of the system. In these kind of systems the behaviors of agents may be difficult to predict. Nevertheless, unbounded or excessively long delays in the information flow or acting of the agents may result in the violation of "agent interaction" concept of multi-agent systems. In other words, it might be difficult to view the systems experiencing unbounded or excessively long delays or systems with agents some of which do not act for unbounded amount of time as a single multi-agent system. This is because an agent that never performs sensing or never acts cannot be considered as a member of the group. Therefore, the level of asynchronism should be limited in such a way that the agents still can interact and form a single multi-agent system. Hence, here we state an assumption (as utilized in the study [108]) which establishes a bound on the maximum possible time delay as well as guarantees uniformity in the updates of the agents.

Assumption 2 *There exists a positive integer B such that*
(a) For every i and every $t \geq 0$, at least one of the elements of the set $\{t, t+1, \ldots, t+B-1\}$ belongs to T^i.
(b) There holds $t - B < \tau^i_j(t) \leq t \ \ \forall i \ \ t \geq 0, t \in T^i$

This assumption basically states that (i) every agent performs update or change in its orientation in at most B time steps; (ii) the delay in sensing the orientations of the neighbors of the agent is bounded as well by at most B time steps. This assumption in a sense restricts the level of asynchronism in the multi-agent system. Assumption 1 is taken from [92] where it is utilized for parallel and distributed computing systems. Systems satisfying Assumption 2 are being referred to as partially asynchronous systems in the parallel and distribution computation literature [92].

Let

$$N_i(t) = \{j : \mid j \neq i, \ (x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 \leq \delta^2\}$$

denote the actual set of neighbors of agent i at time t and $|N_i(t)|$ denote the number of agents in the set $N_i(t)$.

For the asynchronous system considering also the time delays in sensing/communication the set of neighbors of agent i at time t considering the time delays can be described as

$$\mathbb{N}_i(t) = \{j : \mid j \neq i,\ (x_i(t) - x_j(\tau^i_j(t)))^2 + (y_i(t) - y_j(\tau^i_j(t)))^2 \leq \delta^2\}$$

where $\tau^i_j(t)$ represents the last instant at which agent i obtained the orientation information of agent j during the last *sense-compute-turn* state. Note that during the delays in the information gathering and computing states the neighbor agent j may leave the neighborhood region or any other agent may enter this region.

Below we describe three different strategies for the orientation computation of the agents.

3.3 Strategies for Orientation Agreement.

3.3.1 Strategy 1 (Averaging)

This strategy is based on the averaging of orientations of neighbor agents. The new orientation of agent i at step $t + 1$ is determined by the following equation.

$$\theta_i(t+1) = \frac{1}{1 + |\mathbb{N}_i(t)|}\left(\theta_i(t) + \sum_{j \in \mathbb{N}_i(t)} \theta_j(\tau^i_j(t))\right), \quad t \in T^i, \tag{3.3a}$$

$$\theta_i(t+1) = \theta_i(t), \quad t \notin T^i, \tag{3.3b}$$

As mentioned above, we assume that at time t agent i knows $\theta_j(\tau^i_j(t))$ instead of the actual $\theta_j(t)$ about the orientation of agent j. In other words, $\theta_j(\tau^i_j(t))$ is the *perceived orientation* of agent j by agent i at time t. Consequently, if agent i has not yet obtained any information about the j^{th} agent's orientation and still has the initial orientation information, then $\tau^i_j(t) = 0$ whereas $\tau^i_j(t) = t$ means that agent i has the current orientation information of the j^{th} agent. The difference between the current time t and the value of the variable $\tau^i_j(t)$ i.e., $(t - \tau^i_j(t))$ is the delay occurring due to the sensory, computing and/or communication processes or other reasons. Note from Assumption 1 that $t - \tau^i_j(t) > B$ should be satisfied.

One drawback with the rule in (3.3a) is that it may sometimes result in directions of motion that are not very intuitive. For example, assume that there are two agents with directions of motion $+5^o$ and $+355^o$. Based on the rule in (3.3a) on the next step both the agents will turn to 180^o (i.e. they will flip direction) while the intuitive direction is 0^o for both. Here we assumed that the orientation angles are defined between 0^o and 360^o. The situation will not change if they are defined between -180^o and $+180^o$ since the same problem will occur around 180^o this time. The reasons for such behavior are discussed more detailed in the Discussions section.

3.3.2 Strategy 2 (Relative Angles)

In this strategy the agents determine their new orientations by considering the orientation differences between themselves and their neighbors or basically the relative orientations. The next orientation of agents is found by the following equation.

$$\theta_i(t+1) = \theta_i(t) + \frac{\Theta}{1 + |\mathbb{N}_i(t)|} \quad t \in T^i \tag{3.4a}$$

$$\theta_i(t+1) = \theta_i(t), \quad t \notin T^i, \tag{3.4b}$$

where Θ is the total of differences between the orientation of the agent itself and the orientations of its neighbor agents. The orientation of an agent itself and its neighbor's orientation may appear in four different arrangements (Figure 3.2). Therefore, Θ is calculated considering the four different cases with the following pseudo code.

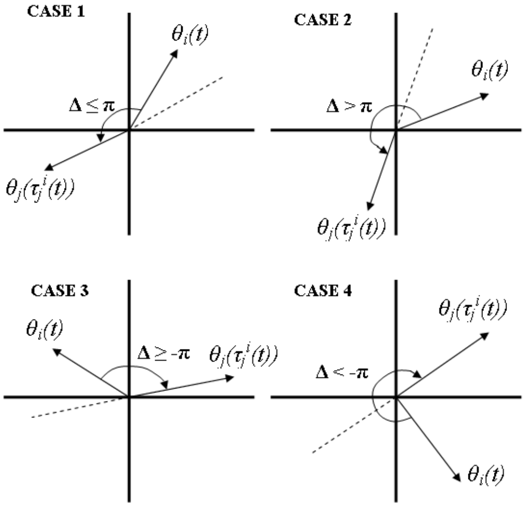


Figure 3.2: Possible arrangements of orientation of an agent and its neighbor's orientation

Table 3.1: Pseudocode for Calculating Θ

```
$\Theta = 0$
FOR $j \in \mathbb{N}_i(t)$
    $\Delta = \theta_j(\tau^i_j(t)) - \theta_i(t)$
    IF $\Delta \leq \pi$   OR   $\Delta \geq -\pi$    (CASE 1 OR 3)
        $\Theta = \Theta + \Delta$
    ELSE IF   $\Delta > \pi$    (CASE 2)
        $\Theta = \Theta + \Delta - 2\pi$
    ELSE IF   $\Delta < -\pi$    (CASE 4)
        $\Theta = \Theta + \Delta + 2\pi$
    END
END
```

where $\theta_i(t) \in [0, 2\pi)\ \ \forall i, t$. This pseudo-code is also equivalent to the equation

$$\Theta = \sum_{j \in \mathbb{N}_i(t)} mod(\theta_j(t) - \theta_i(t) + \pi, 2\pi) - \pi, \tag{3.5}$$

The rule in (3.4a) is more intuitive compared to rule (3.3a) since it considers the relative orientations and always chooses the smaller angle. However, it may also have problems in cases of some symmetries. For example if three agents are oriented such that the orientation difference between each pair is 120^o, they will continue their motion with their previous orientations and will not achieve orientation agreement. The details of such behavior are also discussed in the Discussions section.

3.3.3 Strategy 3 (Vector Sum)

Strategy 3 is based on the vectorial sum of unit vectors that lie along the orientations of agents. Let r_i denote the unit vector in the direction of motion of agent i. Then the orientation rule becomes

$$\theta_i(t+1) = angle\left(r_i(t) + \sum_{j \in \mathbb{N}_i(\tau_j(t))} r_j(\tau^i_j(t)) \right), \quad t \in T^i \tag{3.6a}$$

$$\theta_i(t+1) = \theta_i(t), \quad t \notin T^i, \tag{3.6b}$$

where $angle(v)$ is the function that returns the angle of any given vector v. In implementations it can be computed by using the $atan2(Py, Px)$ function where Px and Py are the components of the computed vector on the right hand side of (3.6a) along the x and y directions, respectively.

For instance in Figure 3.3 we see the calculation of new orientation, $\theta_i(t+1)$ of i^{th} agent with only one neighbor (j).

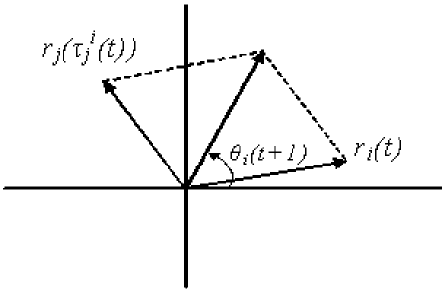


Figure 3.3: Orientation rule of Strategy 3 for only one neighbor (j) of agent i.

Rule (3.6a) is another intuitive rule that is used in determining the agent directions. However, as was the case with rule (3.4a) it may be difficult to decide the next orientation using rule (3.6a) in some cases with symmetry. However, these cases have very low probability in a swarm of many agents. In implementation if such a case occurs one may choose the turning direction randomly until the symmetry is broken. The limitations of Strategy 2 and 3 in cases of symmetries and the drawback of Strategy 1 explained in section 3.3.1 are investigated in more detail in the Discussions section.

3.4 Turn Angle Restrictions

As discussed above in this study we assume that the agents update their orientation based on their own orientation and the orientation of their neighbors. In this section additionally we assume that there is a restriction on the maximum possible turning angle of the agents due to mechanical or physical reasons. Therefore, the dynamics of the orientation angles of the agents are given by

$$\theta_i(t+1) = \theta_i(t) + min\,(abs(\phi_i(t)), \alpha) \times sign\,(\phi_i(t))\,, \quad t \in T^i, \tag{3.7a}$$

$$\theta_i(t+1) = \theta_i(t), \quad t \notin T^i \tag{3.7b}$$

where α is the maximum possible turn angle per step and $\phi_i(t)$ is the desired turn angle which is computed at time t by using one of the three strategies given above. Due to (3.7a), during an update i^{th} agent can turn at most the angle α in the direction of $\phi_i(t)$ (clockwise or counter clockwise).

Adding the turn angle restrictions we can rearrange the rules of the previous three strategies for computing $\phi_i(t)$ as

For Strategy 1

$$\phi_i(t) = \frac{1}{1 + |\mathbb{N}_i(t)|}\left(\theta_i(t) + \sum_{j \in \mathbb{N}_i(t)} \theta_j(\tau^i_j(t))\right) - \theta_i(t), \quad t \in T^i \tag{3.8}$$

For Strategy 2

$$\phi_i(t) = \frac{\Theta}{1 + |\mathbb{N}_i(t)|} \quad t \in T^i \tag{3.9}$$

where Θ is calculated by the pseudo code given in section 3.3.2.

For Strategy 3

$$\phi_i(t) = angle\left(r_i(t) + \sum_{j \in \mathbb{N}_i(\tau_j(t))} r_j(\tau^i_j(t))\right) - \theta_i(t), \quad t \in T^i \tag{3.10}$$

We would like to emphasize here that having turn angle restrictions is a very realistic assumption since most real agents will have such constraints. However, such restrictions have not been considered in the literature so far.

3.5 Simulation Results

We simulated the motion of $n = 50$ agents. Initially the agents are located in a square region of size 100 × 100 units and the constant speed of all of the agents is set to 1 unit/step. The simulations are performed for T time steps ($T = 500$ for unbounded region and $T = 1000$ for bounded region). The agents perform updates depending on a stochastic function of B. The probability of update of an agent is distributed along the interval $t \in (t - B, t]$ such that it is $1/(B - \Omega + 1)$ where Ω is the number of steps that the agent has not performed an update since the last update. In other words, if the agent has just performed an update in the previous step the probability that it will perform an update again is $1/B$, whereas if it has not performed an update for $3 < B$ steps then its probability of update is $1/(B - 2)$ If the agent has not performed an update for B time steps then its probability of update becomes 1. At each time step at which the agent does not perform an update the value of Ω is incremented by 1 and at each time step the agent performs an update the value of Ω is reset to zero. Note that this implementation is not a real discrete event based asynchronous system. Instead it mimics such systems and is sufficient for illustrating/verifying the performance of the asynchronous system discussed in this study. The initial positions and orientations of agents are generated randomly and for each simulation the same initial conditions are utilized. We provide 20 simulation results performed for different initial conditions generated randomly and present the mean and variance of the results. The algorithm performing these steps is presented in table 3.5. In order to measure the performance of the system we used the following performance metrics

$$
\begin{aligned}
e_d(t) &= \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|z_i(t) - z_j(t)\|, \quad t \geq 1 \\
e_\theta(t) &= \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \|\theta_i(t) - \theta_j(t)\|, \quad t \geq 1 \\
\dot{e}_\theta(t) &= \frac{1}{n} \sum_{i=1}^{n} \|\theta_i(t) - \theta_i(t-1)\|, \quad t \geq 2
\end{aligned}
$$

where $z_i(t) = [x_i(t), y_i(t)]^T$ is the position vector of agent i. Basically $e_d(t)$ is the *average distance between the agents*, and $e_\theta(t)$ is the *average of orientation differences between the agents*, and $\dot{e}_\theta(t)$ is the *average rate of change of orientation of the agents* at time t. The fourth performance metric is the number of clusters formed by the agents. A cluster is defined as the group of agents which are connected to (meaning are neighbors of) each other either directly or indirectly through other agent. (Note that agents i and j belong to the same cluster at time t if $\|z_i(t) - z_j(t)\| \leq \delta$). As $e_d(t)$ gets lower, the agents get closer to each other which implies that the multi-agent system performs better in clustering. The metric, number of clusters is also a performance criterion in determining the success of clustering of agents. Note that as the number of clusters decrease or size of clusters increase (resulting in lower $e_d(t)$) the number of agents traveling with the same heading increases. Therefore, if the performance in clustering is better, then the performance in orientation agreement is also better. On the other

Table 3.2: Pseudocode of Simulation Steps

```
Draw the initial positions and orientations of agents
randomly from a uniform distribution
FOR i=1:n DO
        set $\Omega_i$ = 1; ($\Omega_i$ is the number of steps
                that agent i has not performed an update.
                At the beginning $\Omega_i$ is one for every agent.)
END
FOR t=1:T DO (T = number of simulation steps)
  FOR i=1:n (n = number of agents) DO
          prbOfUpdate=1/(B-$\Omega_i$+1);
          c = randint(1,100); (Generate random integer from
                  uniform distribution to compare with prbOfUpdate)
          IF (c$\leq$prbOfUpdate*100) (performing update)
            FOR j=1:n DO
                    $\tau$ = randint(t-B,t)
                      ($\tau$ is the random step drawn between last
                      step orientation updated and current step
                      Consider $\tau$ as the last step that agent i
                      received/measured the orientation of agent j)
                    IF agent $j$ is neighbor of agent $i$
                      Use $\theta_j(\tau)$ in the computation of the new
                      orientation
                      (based on formula for the current strategy)
                    END
            END
            $\Omega_i$ = 1; Set to one since the agent is updated
          ELSE (No update will be performed)
                    $\Omega_i$ = $\Omega_i$ + 1; (increment number of steps
                      that no update performed)
                    No change in agent i's orientation
          END
  END
END
```

hand, as $e_\theta(t)$ gets lower, the agents are heading in closer orientations which means they have better performance in orientation agreement. The agents have more steady headings if $\dot{e}_\theta(t)$ converges to zero.

We have conducted simulations in order to determine the effects of asynchronism and time delays on the cluster formation by varying α (turn angle restrictions) in unbounded and bounded regions.

3.5.1 Effect of α for Unbounded Region

Here, we compare the effects of α and asynchronism on the performances of the three orientation strategies. δ is set to 20 units. Note that in all simulations $B = 0$ corresponds to the synchronous case and $B = 10$ corresponds to an asynchronous case.

3.5.1.1 Effect of α for Unbounded Region in Synchronous model

In Figures 3.4 and 3.5 we plotted $e_d(t)$ and $e_\theta(t)$ at the end of simulations ($t = T = 500$) versus α values. It is seen that as the α value increases (which means that the restriction on the turning angle decreases) the values of both $e_d(T)$ and $e_\theta(T)$ decrease in all strategies. The total distance between agents is smaller for higher α values. This is an expected result when we consider the fact that as the restriction gets weaker, the agents perform better turning motion and therefore, they aggregate better. On the other hand, the decrease in the sum of orientation differences $e_\theta(T)$ implies that the number of agents moving with different headings decrease and also the difference between the heading of clusters decreases, since the orientation adjustment becomes easier at lower turning restrictions. In Figure 3.6 we see the clustering performance of the three strategies with varying α. As seen all strategies get better as the turning angle restrictions get smaller.

In all Figures 3.4, 3.5, and 3.6 the first strategy has the best performance for all α values. The second and third strategies are close to each other.

The total of rate of change of orientations of all agents at each step, $\dot{e}_\theta(t)$ is plotted in Figures 3.7, 3.8, and 3.9 for strategies 1, 2, and 3, respectively. In all three figures the results for $\alpha = 1^o$ shows lower values than the ones for $\alpha = 180^o$ because as α gets lower the maximum

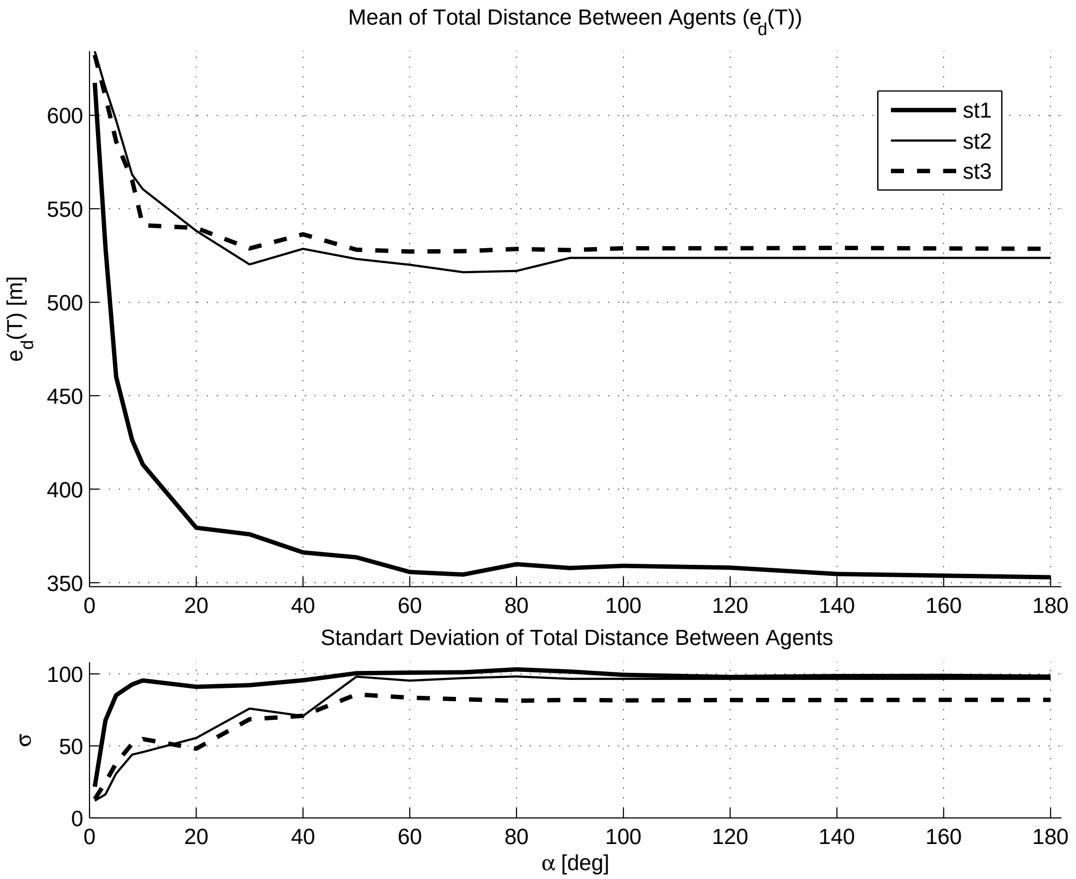


Figure 3.4: $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and unbounded region.

possible rate of change of orientations of each agent gets lower as well. The agents having lower α values cannot form large clusters as stated above. Therefore, they continue their motion in relatively small clusters (there are many of them) which are spreading away. As the clusters get out of neighborhood range of each other with different orientations then there exits no possibility for them to change their orientations to travel with the same heading. Therefore, in all figures, after some amount of steps $\dot{e}_\theta(t)$ settles to zero implying that no change of orientations of clusters or agents occur. Note that in Figure 3.7 for $\alpha = 1^o$ there are some peaks around 140^{th}, 270^{th} and 440^{th} steps that means some small clusters come across.

3.5.1.2 Effect of α for Unbounded Region in Asynchronous model

Here, we present the results of simulations of the three strategies for the asynchronous case. In Figure 3.10 and 3.11 we plotted $e_d(t)$ and $e_\theta(t)$ at the end of simulations ($t = T = 500$)

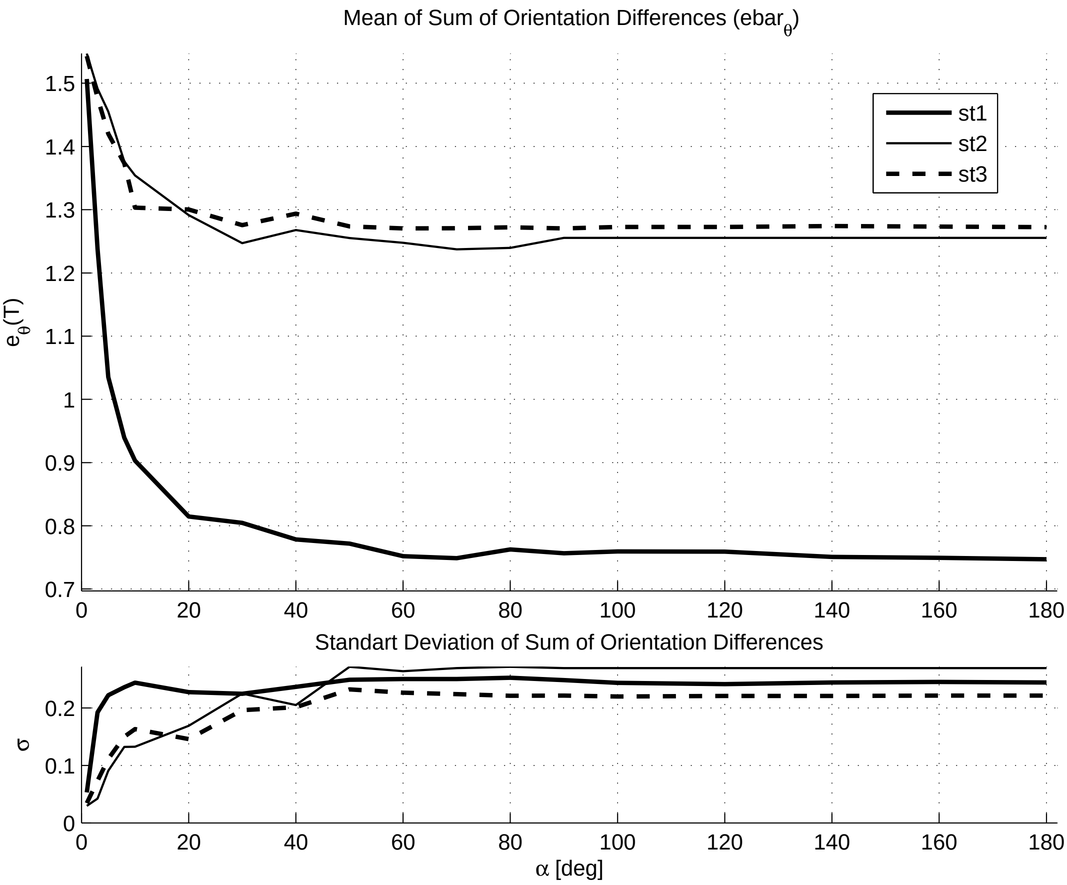


Figure 3.5: $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and unbounded region.

versus α values. Like the results for the synchronous case as the α value increases the values of both $e_d(T)$ and $e_\theta(T)$ decrease in all strategies. Comparing with the synchronous case, we can conclude that in the asynchronous case all strategies have worse performances. This is an expected result when we consider the delays in the sense and computing states (which results in lack of valid information about the orientations of its neighbor agents) of the agents and the fact that the agents may not perform orientation update at each time step. These two reasons make the performances worse. In Figure 3.12 we see the clustering performance of the three strategies with varying α for the asynchronous agents. As seen all strategies perform better as the turning angle restrictions get weaker. However, like the worse performances in $e_d(t)$ and $e_\theta(t)$ with respect to the synchronous case, the number of clusters for a specific α value is worse (higher) too, for the asynchronous dynamics.

Like the synchronous case all plots in Figures 3.10, 3.11, and 3.12 show that the first strategy

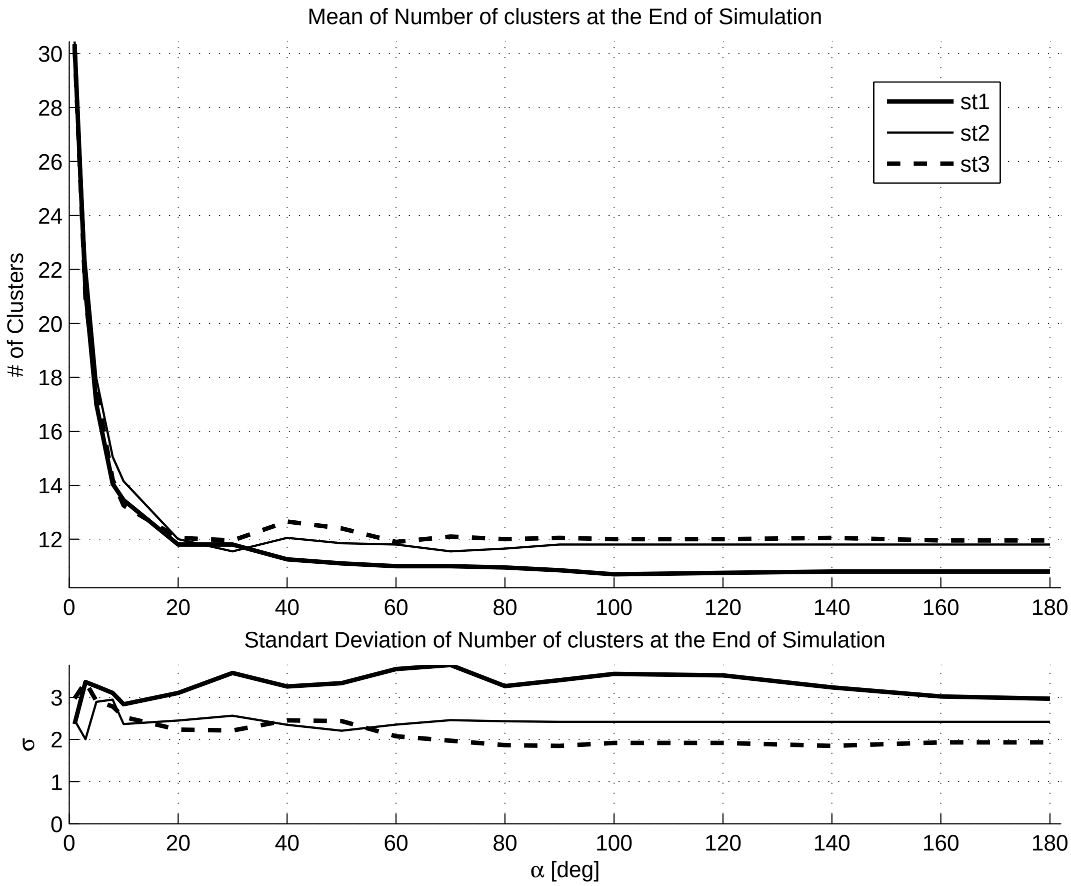


Figure 3.6: The number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and unbounded region.

again has the best performance for all α values. The performances of the second and the third strategies are close to each other.

The asynchronous results for $\dot{e}_\theta(t)$ are plotted in Figures 3.13, 3.14, and 3.15 for strategies 1, 2, and 3, respectively. As in the synchronous case all three figures show that results for $\alpha = 1^o$ have lower values than the ones for $\alpha = 180^o$ due to lower maximum possible turning angle values at each step. In all figures the amount of steps that $\dot{e}_\theta(t)$ settles to zero is higher with respect to synchronous case. Again note that since the clusters are spread away as time passes the possibility of orientation adjustment with respect to neighbor clusters diminishes. Therefore, the fluctuation amounts are diminishing in time. There are some out of order peaks at the different steps that may be caused by the asynchronism in the dynamics or some clusters come across with each other.

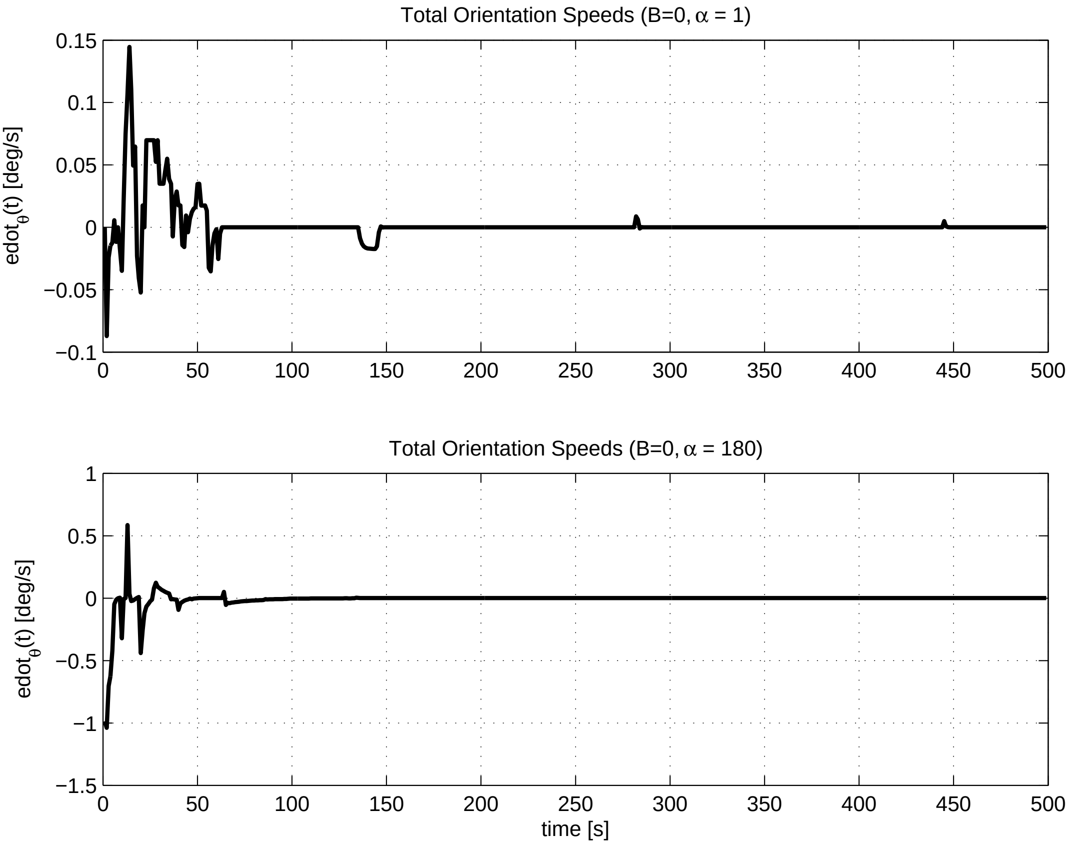


Figure 3.7: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Synchronous case - Unbounded region).

3.5.2 Effect of α for Bounded Region

The simulation parameters and initial conditions used in this section are the same with those used in the previous sections except that in this case we restrict the arena in which the agents move. They move in a 100 by 100 square region. When an agent faces any boundary it continues to its motion with the orientation that is the reflection of its previous orientation just like a light beam reflects on a mirror. In the following sections again we present the simulation results for synchronous and asynchronous cases.

3.5.2.1 Effect of α for Bounded Region in Synchronous model

The plots of $e_d(t)$ and $e_\theta(t)$ are presented in Figures 3.16 and 3.17, respectively. As in the previous results we see that as the α value increases the values of both $e_d(T)$ and $e_\theta(T)$ decrease

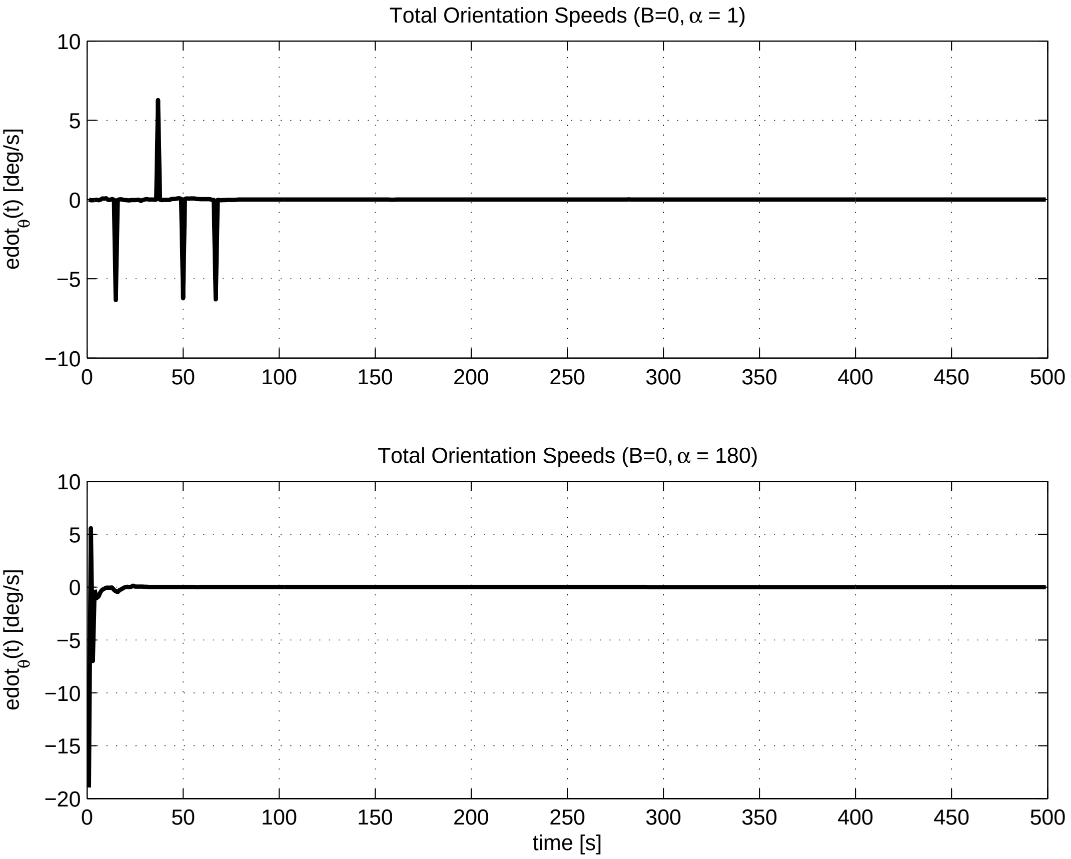


Figure 3.8: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Synchronous case - Unbounded region).

in all strategies. In Figure 3.18 we see the clustering performance of the three strategies with varying α. As seen all strategies get better in clustering as the turning angle restrictions get smaller.

The first strategy has again best performance as seen in Figures 3.16, 3.17, and 3.18. The second and third strategies perform close to each other. As expected, since the region is bounded the performances of this case are better than performances of its unbounded counterpart.

Figures 3.19, 3.20, and 3.21 are the plots of the $\dot{e}_\theta(t)$ for synchronous and bounded case. The amounts of rate of changes of the orientations of the agents are very high compared to the unbounded region results. In the unbounded region the agents settle down to some orientation and continue with this orientation for the rest of the simulation. However, if the region is bounded the agents have to turn (to the reflection angle) when they come across with the boundaries. Therefore, we see some sharp peaks in the plots that resulted from the reflection

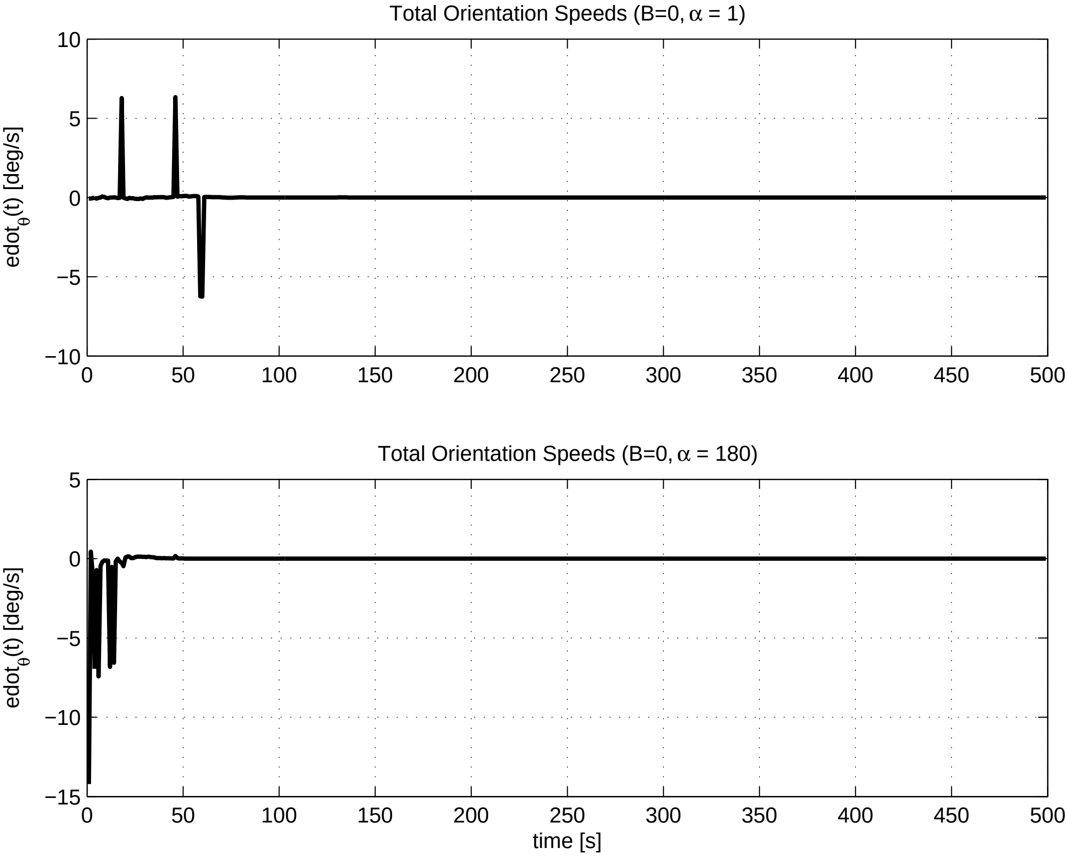


Figure 3.9: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Synchronous case - Unbounded region).

of clusters from walls. Note that during this process some of the agents come across with the boundaries before their neighbors that they travel with the same orientations. Since the leader agents turn to a reflection orientation, orientation strategies of all neighbor agents calculate new orientations -different from the one they all settled down. Hence, we see that for the high restriction of turn angles -low α value- the new orientation agreement of a cluster of agents may take more time.

3.5.2.2 Effect of α for Bounded Region in Asynchronous model

The results of the asynchronous version of the previous bounded region simulation is presented in Figures 3.22, 3.23, and 3.24. As seen the results are similar in terms of getting better as the turning angle restrictions weakens. Comparing the synchronous and asynchronous results, again we find that the asynchronism leads to worse performance. Here the second and

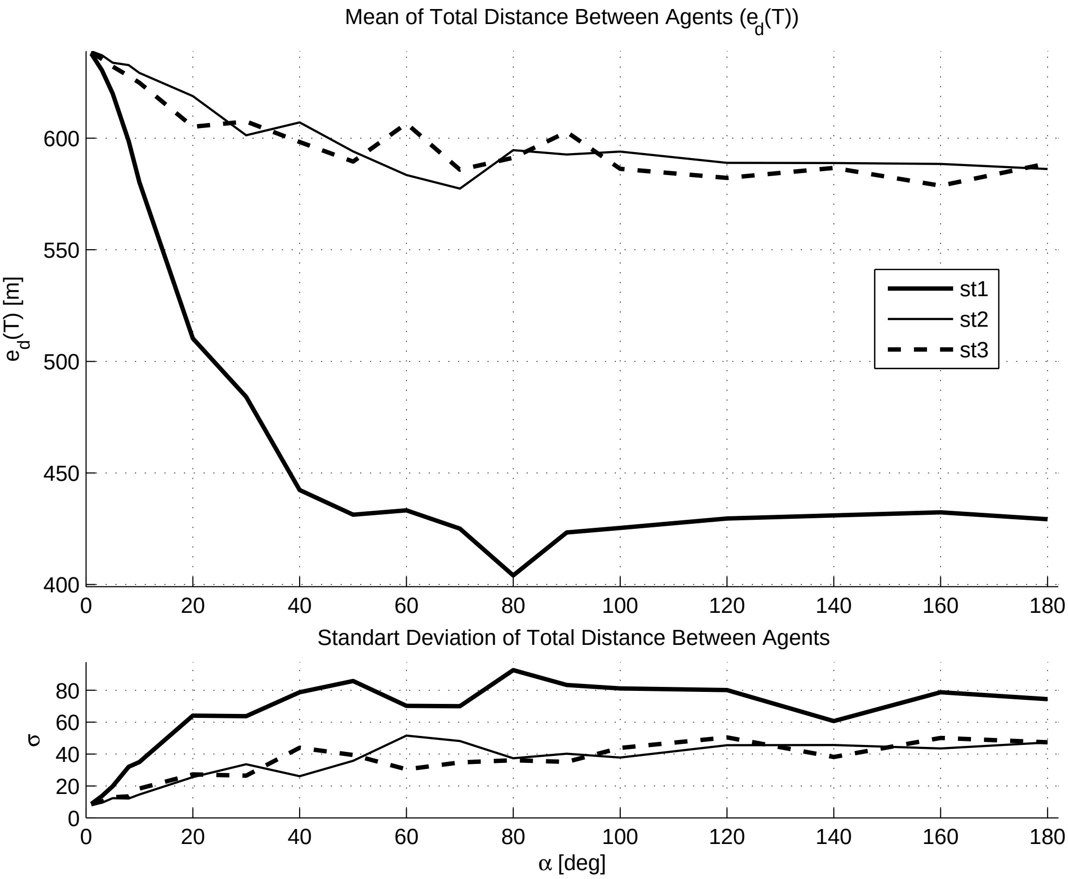


Figure 3.10: $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and unbounded region.

third strategies performs close to each other and worse than the first strategy again. Moreover, as expected since the region is bounded the performances of this case are better than performances of its unbounded counterpart.

The last three Figures 3.25, 3.26, and 3.27 are the plots of the $\dot{e}_\theta(t)$ for asynchronous and bounded case for strategies 1, 2, and 3, respectively. As seen from the figures for $\alpha = 1$ the agreement of orientations of agents is not achieved at all. Every agent changes its orientation due to its orientation strategy and/or facing of boundaries. The agents having $\alpha = 180^o$ performs better than agents $\alpha = 1^o$ in this case but they have still worse performance compared to the agents with synchronism assumption.

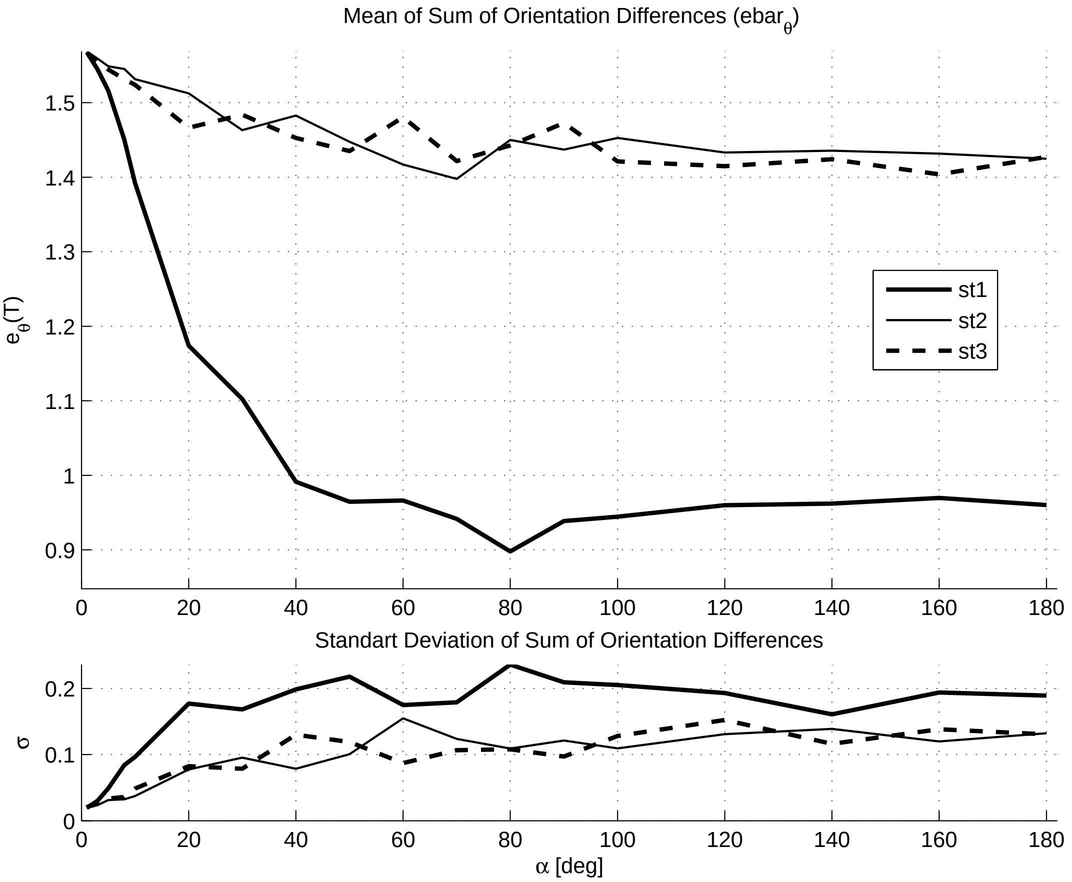


Figure 3.11: $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and unbounded region.

3.6 Discussions

In this study the initial conditions of the agents are set randomly with uniform probability in the interval $[0, 2\pi)$ since the initial orientations of agents are required not to be biased towards any direction. In other words, the probability of an agent i starting with orientation $\theta_{i1}(0) \in [0, 2\pi)$ is equal to the probability of agent i starting with any other orientation $\theta_{i2}(0) \in [0, 2\pi)$. Therefore, the probability distribution function of the initial orientations of the agents in the simulations is given by

$$f(\theta|0, 2\pi) = \begin{cases} \frac{1}{2\pi}, & if\ 0 \leq \theta < 2\pi \\ 0, & otherwise \end{cases} \tag{3.11}$$

Depending on the initial conditions (orientations and positions), the agents may show different behavior for each of the three strategies. Note that, the group of agents in the neighborhood of agent i, is in fact a sample set of the population distributed uniformly. Considering the

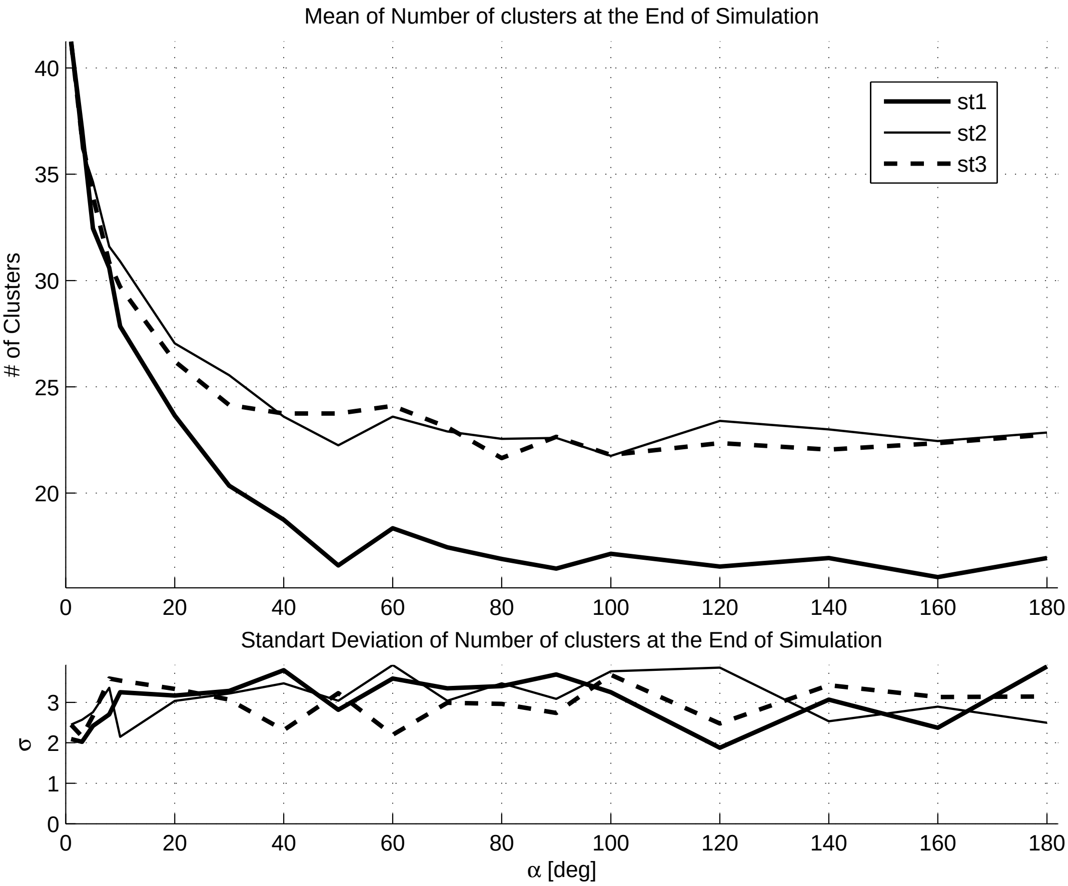


Figure 3.12: the number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and unbounded region.

asynchronism of updates, the orientations of the set of agents that agent i is utilizing their orientations in performing its orientation update is also a sample set of a uniformly distributed set.

The following analysis is valid for only $t \geq 1$. However, it is sufficient to illustrate the general tendencies of the strategies considered in this study.

In Strategy 1 (the averaging strategy), the agent in any cluster will update its orientation according to

$$\theta_i(t+1) = \frac{1}{N}\sum_{j=1}^{N}\theta_j(t), \tag{3.12a}$$

where N is the number of the agents in the neighborhood of agent i including itself (Note that, N is different from $|N_i(t)|$ which we used in previous sections. In fact N includes the number of neighbors and agent itself such that $N = |N_i(t)| + 1$). Recall that, the orientations of agents

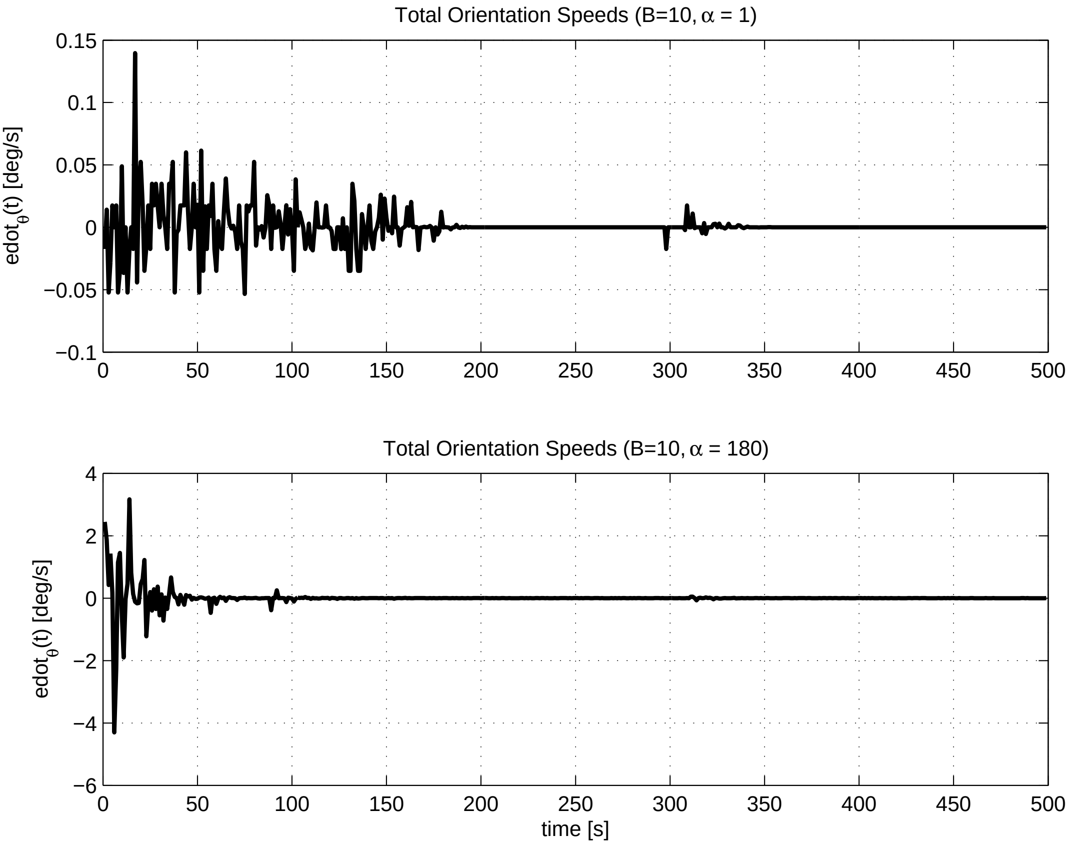


Figure 3.13: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Asynchronous case - Unbounded region).

at $t = 1$ in any neighborhood are in fact in a sample set of the uniform distribution we stated above. Therefore, the expected value of the outcome of the averaging rule is the mean of the uniform distribution. In other words, the expected value of the updated orientations for $t = 1$ (initial step) is

$$E[\theta_i(t+1)] = \mu = \frac{2\pi + 0}{2} = \pi \tag{3.13}$$

This shows that the agents utilizing the first strategy will tend to orient themselves towards or basically to agree upon an orientation close to π where the orientations of agents are distributed uniformly in the previous state ($t = 1$ in this case). In other words, the agents calculating their new orientation based on the averaging strategy (Strategy 1) will have a bias towards the angle π. This fact will not change even if the topology is not fully connected or synchronous or there are time delays or turn angle restrictions in the system. This is because taking convex combinations between a set of values (and that is what exactly the averaging rule does) cannot lead to values outside the initial set. Therefore, for any subgroup in the

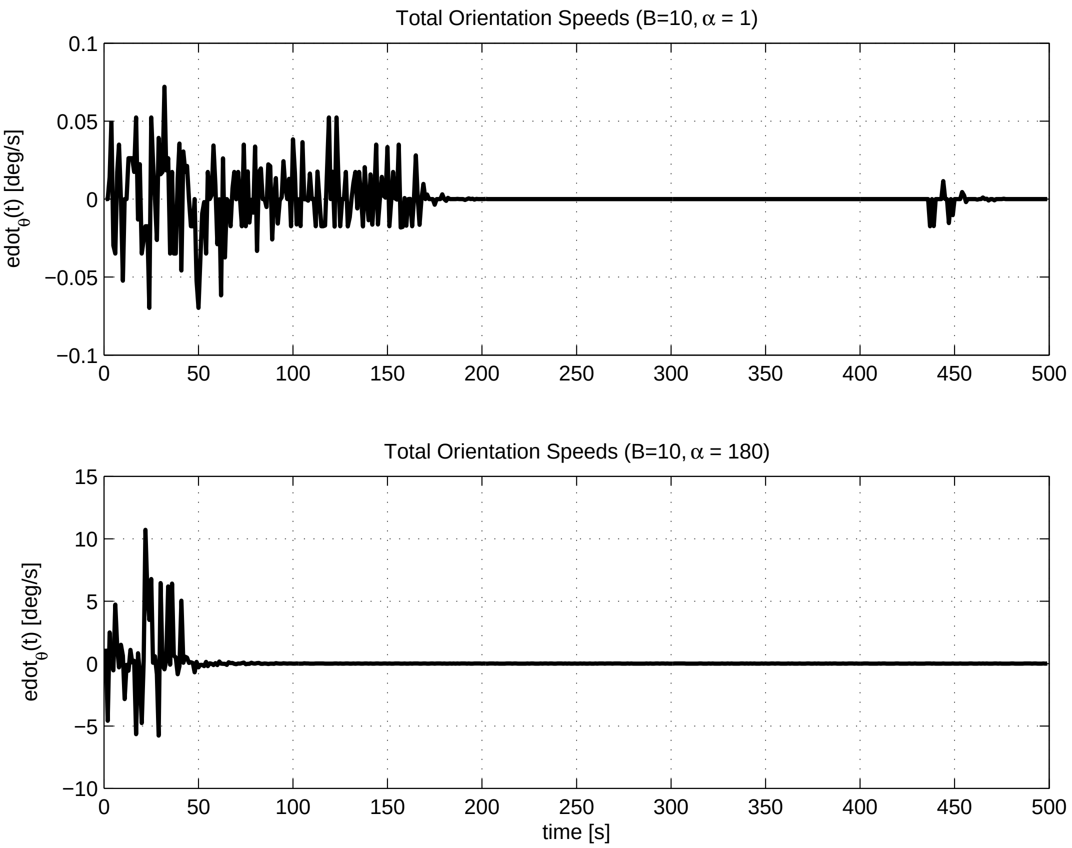


Figure 3.14: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Asynchronous case - Unbounded region).

swarm the bias will be towards the initial average orientations of the group and as the number of the members in the group increases or the groups join or disjoin this average will tend to be closer to π. In fact we have noticed in the simulations that the flocks for the first strategy always tend towards the left of the screen (which is the expected result based on the discussions above). The reason for the drawback/shortcoming of flipping directions in this strategy (mentioned before) is exactly due to this tendency towards π. Note also that the averaging strategy uses global orientations. In other words, for its implementation all the agents need to have means to measure global orientations (e.g., each of them needs to have a compass) and they have to agree a priori on a global reference frame (i.e., the compasses have to be calibrated properly). We believe that this is the main reason for the better performance (faster convergence) of the first strategy. The swarm under this strategy seems as a guided swarm with global bias and the model of Strategy 1 might be suitable for such applications. However, for many minimalist multi-agent applications it may not be possible to have global informa-

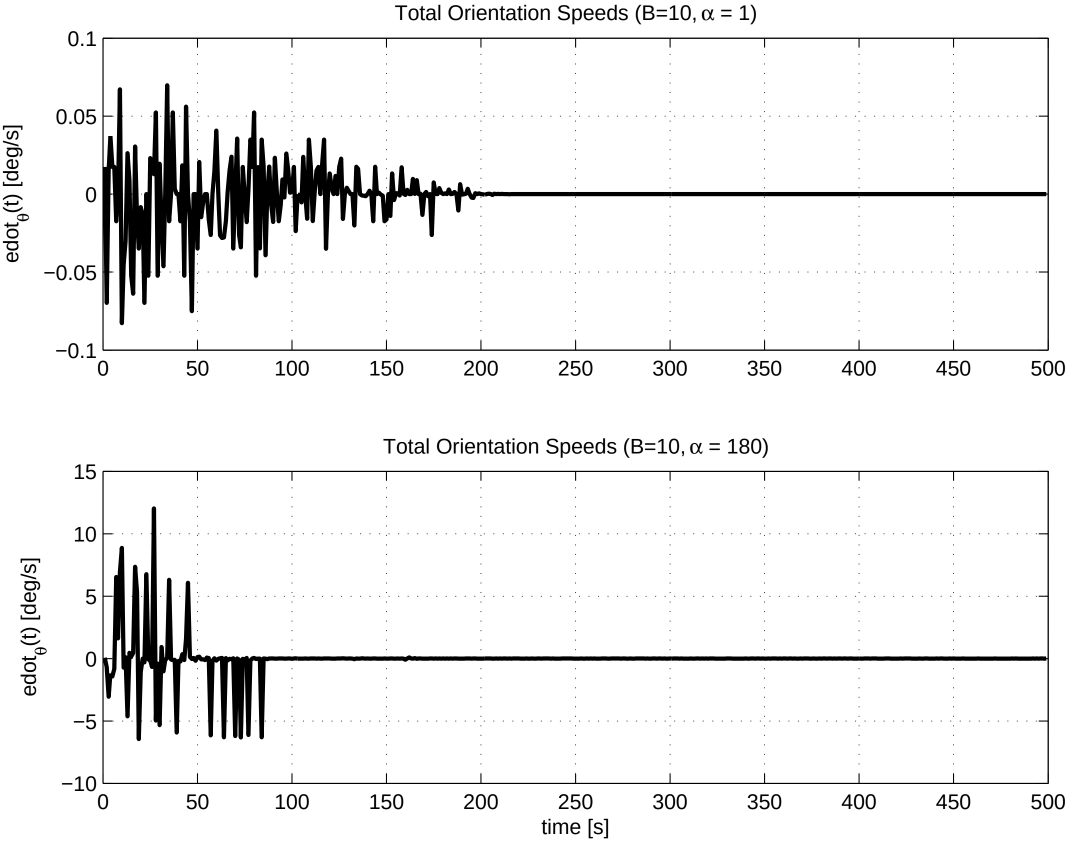


Figure 3.15: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Asynchronous case - Unbounded region).

tion (i.e., a global reference frame agreed upon a priori). Therefore, for such applications it may not be possible to implement Strategy 1 even though it converges faster.

We would like to also emphasize that if the orientations of agents were drawn from a uniform distribution of angles which is between $(-\pi, \pi]$ (instead of the set $[0, 2\pi)$) then the expected value of the updated orientations would be $\mu = \pi + (-\pi)/2 = 0$. This means that the agents are guided towards 0 of the global reference frame and the qualitative behavior does not change.

In Strategy 2, the agents will update their orientations according to

$$\theta_i(t+1) = \theta_i(t) + \frac{1}{|N_i(t)|} \sum_{j=1}^{|N_i(t)|} mod(\theta_j(t) - \theta_i(t) + \pi, 2\pi) - \pi, \tag{3.14}$$

where $|N_i(t)|$ is the number of agents in the neighborhood of agent i (not including agent itself). In equation (3.14), the part $\sum_{j=1}^{|N_i(t)|} mod(\theta_j(t) - \theta_i(t) + \pi, 2\pi) - \pi$ is in a sense the estimation of function $u(\theta - \theta_i(t)) = mod(\theta - \theta_i(t) + \pi, 2\pi) - \pi$. For simplicity lets call $\Theta = \theta - \theta_i(t)$. Then

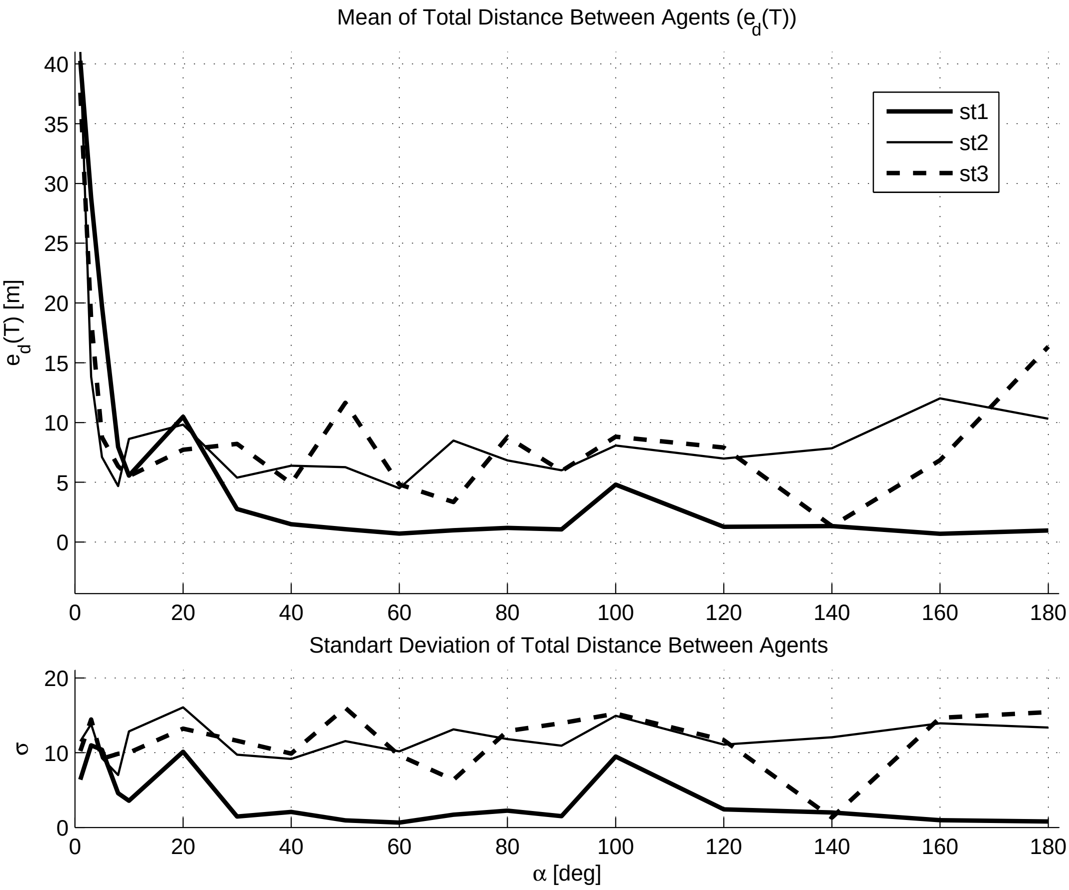


Figure 3.16: $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and bounded region.

$u(\Theta) = mod(\Theta + \pi, 2\pi) - \pi$. The expected value of $u(\Theta)$ is

$$E[u(\Theta)] = \int_{-\infty}^{\infty} u(\Theta)\frac{1}{2\pi}d\Theta \tag{3.15a}$$

where given the fact that the (absolute) orientation angles are uniformly distributed in the interval $[0, 2\pi)$ (initial - $t = 1$ - orientations in this case) one can show that the probability density function of Θ is given by

$$f(\Theta| - \theta_i(t), 2\pi - \theta_i(t)) = \begin{cases} \frac{1}{2\pi}, & if \ -\theta_i(t) \leq \theta < 2\pi - \theta_i(t) \\ 0, & otherwise \end{cases} \tag{3.16}$$

The function $u(\Theta)$ is a piecewise continuous function

$$u(\Theta) = \begin{cases} \Theta + 2\pi, & if \ -2\pi \leq \Theta < -\pi \\ \Theta, & if \ -\pi \leq \Theta < \pi \\ \Theta - 2\pi, & if \ \pi \leq \Theta < 2\pi \\ 0, & otherwise \end{cases} \tag{3.17}$$

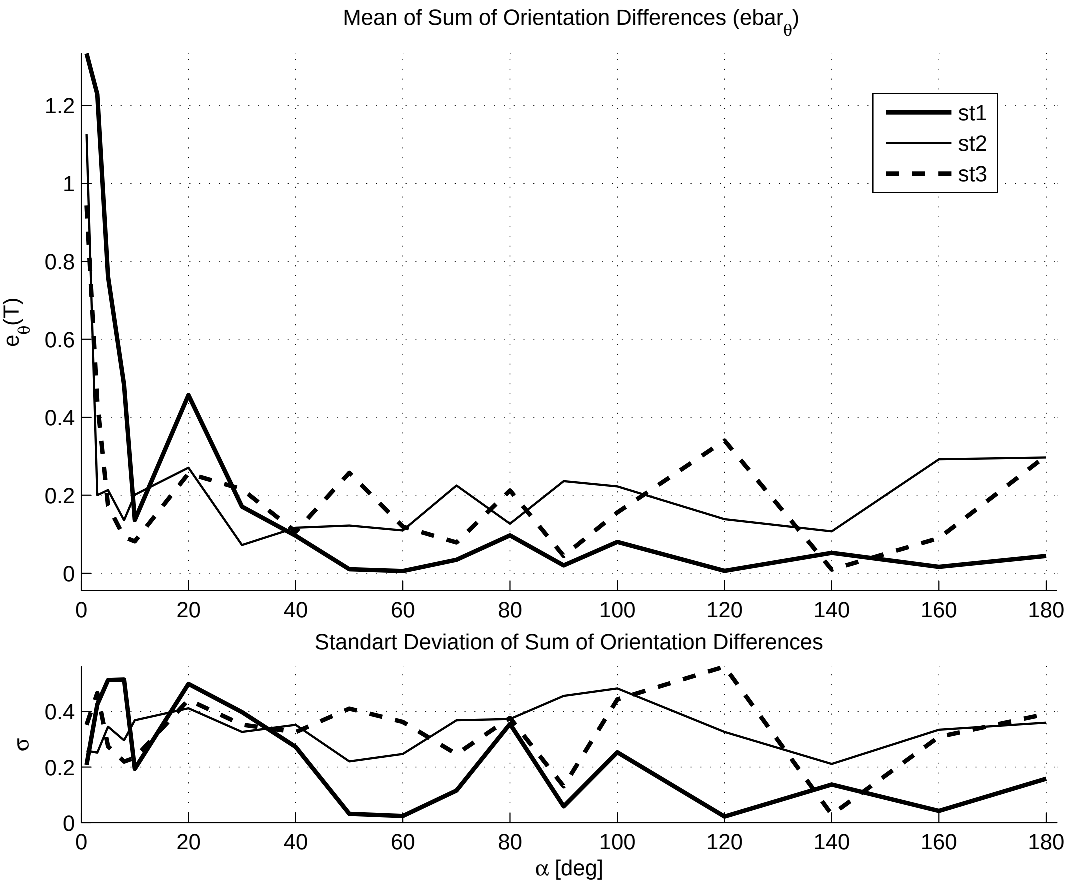


Figure 3.17: $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and bounded region.

Therefore, the expected value of $u(\Theta)$ is given by

$$E[u(\Theta)] = \int_{-2\pi}^{-\pi} (\Theta + 2\pi)\frac{1}{2\pi} d\Theta \tag{3.18a}$$

$$+ \int_{-\pi}^{\pi} \Theta \frac{1}{2\pi} d\Theta \tag{3.18b}$$

$$+ \int_{\pi}^{2\pi} (\Theta - 2\pi)\frac{1}{2\pi} d\Theta \tag{3.18c}$$

One can simply show that

$$E[u(\Theta)] = 0 \tag{3.19}$$

Therefore, for the case with fully connected topology and synchronous motion, the expected

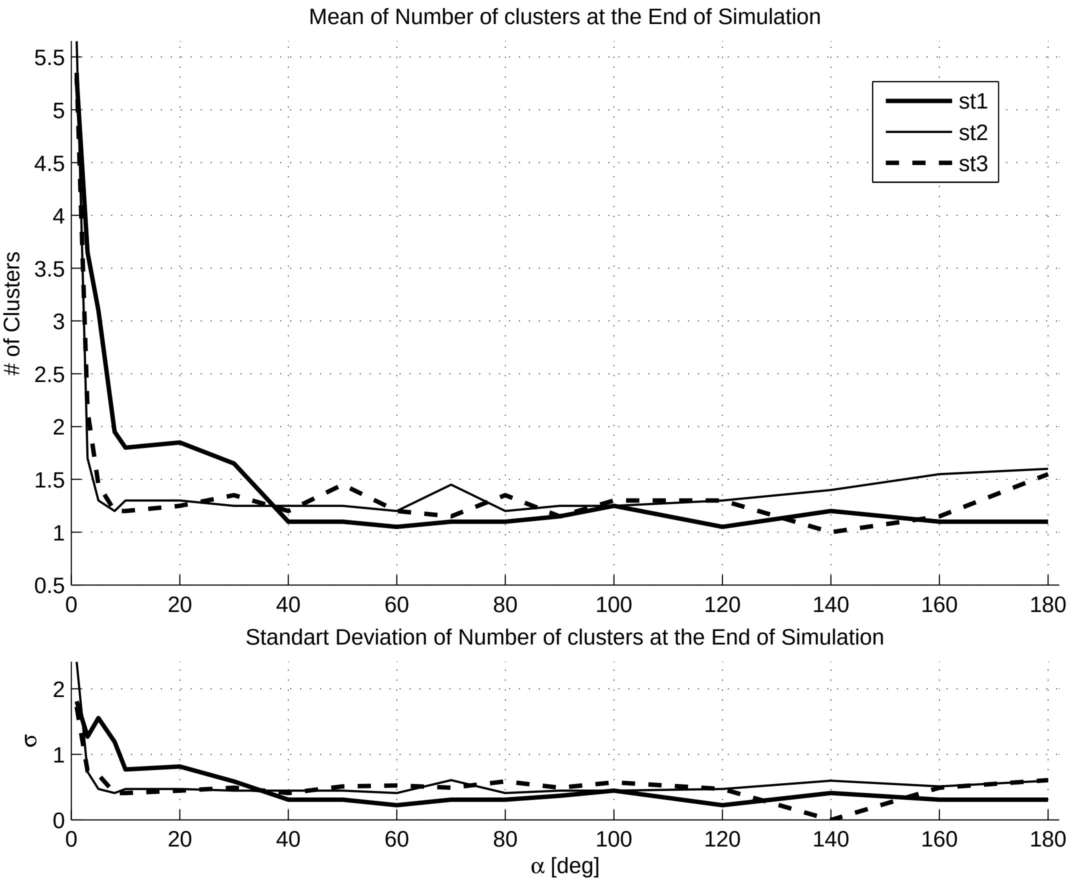


Figure 3.18: the number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for synchronous case and bounded region.

value of $\theta_i(t + 1)$ becomes

$$
\begin{aligned}
E[\theta_i(t+1)] &= E[\theta_i(t)] + E[u(\Theta)] && \text{(3.20a)}\\
&= \theta_i(t) + E[u(\Theta)] && \text{(3.20b)}\\
&= \theta_i(t) && \text{(3.20c)}
\end{aligned}
$$

which implies that agent *i* utilizing the second strategy will be tending to preserve its previous orientation. Since agent *i* was chosen arbitrarily, the same will hold for all the agents, which, on the other hand, implies that there is no global reference or bias towards which all the agents tend to converge. This observation explains why the agents perform worse in Strategy 2 compared to Strategy 1. Note that even though Strategy 2 performs worse than Strategy 1, it might be more suitable for many multi-agent applications. This is because first of all it does not require agreement on a global coordinate system between the agents. Second, in real applications in Strategy 1 the agents have to pass their global orientations to each other

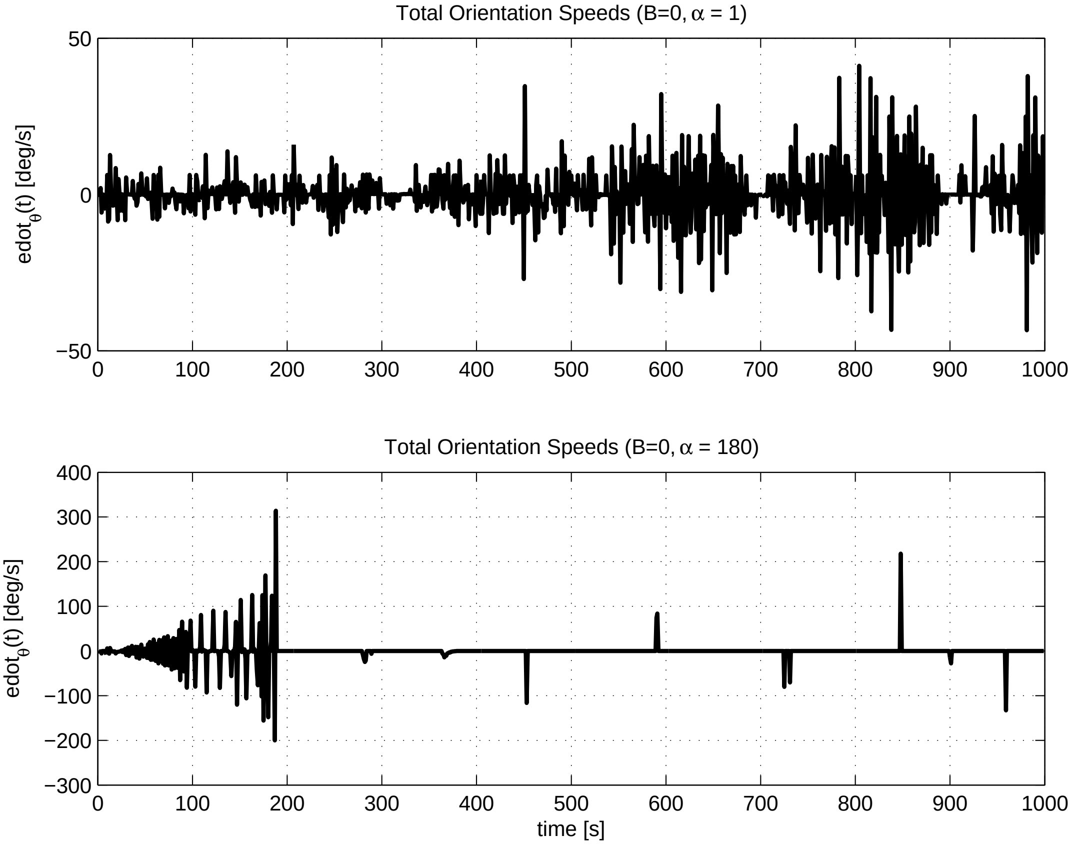


Figure 3.19: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Synchronous case - Bounded region).

by means of some kind of inter-agent communication. In contrast, in Strategy 2 the agents can determine themselves the relative orientations (in their local reference frame) of their neighbors by means of local sensing (without a need for inter-agent communication). In fact, even though Strategy 1 has been inspired by natural phenomena such as the global motion of schools of fish, flocks of birds, or swarms of bacteria or synchronization of the flushing of fireflies [63] and has been used to explain such phenomena, it is difficult to imagine that such natural systems operate based on global information (such as Strategy 1) and operation based on local (relative) information (such as Strategy 2) seems more realistic or natural.

For the third strategy, the next orientation is calculated as

$$\theta_i(t+1) = angle\left(\sum_{j=1}^{N} r_j(t)\right) \tag{3.21}$$

where N is the number of the agents in the neighborhood of agent i including itself and as one can recall that $r_j(t)$ denotes the unit vector in the direction of $\theta_j(t)$. Therefore, the x and y

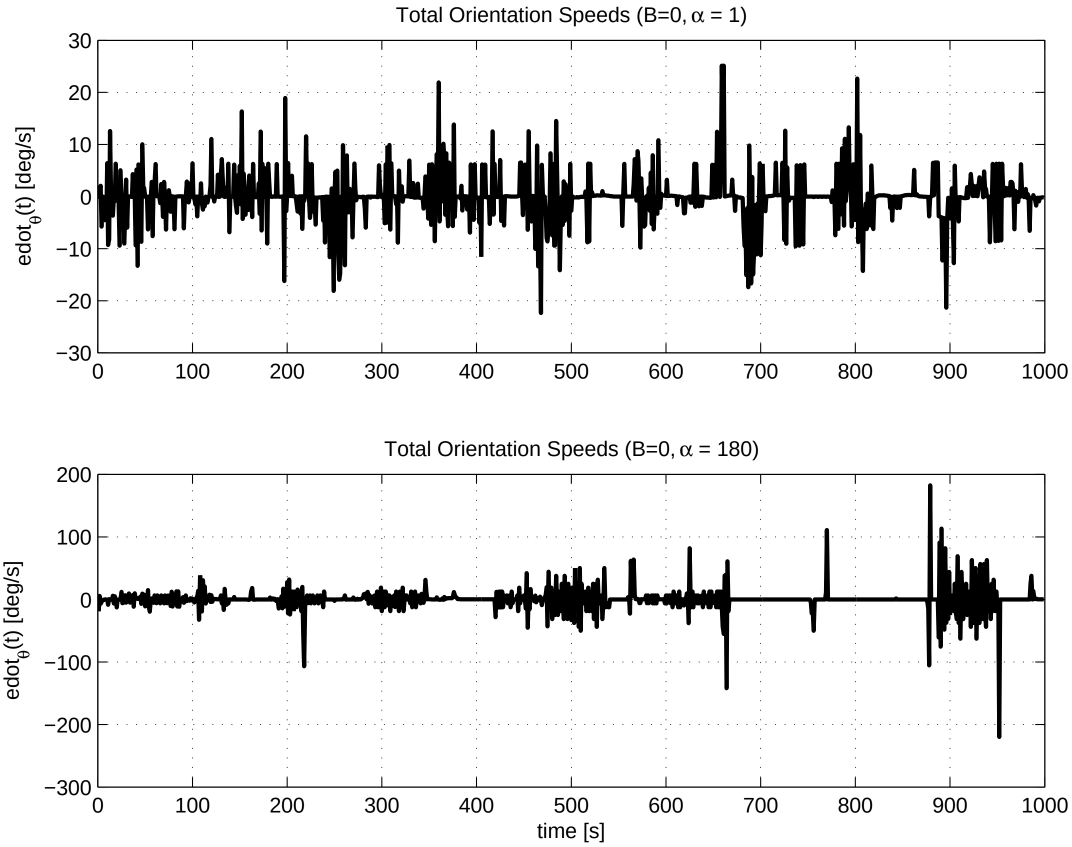


Figure 3.20: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Synchronous case - Bounded region).

components of unit vector $r_j(t)$ become

$$x_j(t) = cos(\theta_j(t)) \tag{3.22a}$$

$$y_j(t) = sin(\theta_j(t)) \tag{3.22b}$$

Substituting these into equation (3.21), the next orientation of agents is calculated as

$$\theta_i(t+1) = angle\left(\sum_{j=1}^{N}\begin{bmatrix} x_j(t) & y_j(t)\end{bmatrix}\right) \tag{3.23a}$$

$$= angle\left(\begin{bmatrix}\sum_{j=1}^{N}(x_j(t)) & \sum_{j=1}^{N}(y_j(t))\end{bmatrix}\right) \tag{3.23b}$$

$$= angle\left(\begin{bmatrix}\sum_{j=1}^{N}(cos(\theta_j(t))) & \sum_{j=1}^{N}(sin(\theta_j(t)))\end{bmatrix}\right) \tag{3.23c}$$

or

$$\theta_i(t+1) = angle\left(\begin{bmatrix}\frac{1}{N}\sum_{j=1}^{N}(cos(\theta_j(t))) & \frac{1}{N}\sum_{j=1}^{N}(sin(\theta_j(t)))\end{bmatrix}\right) \tag{3.24}$$

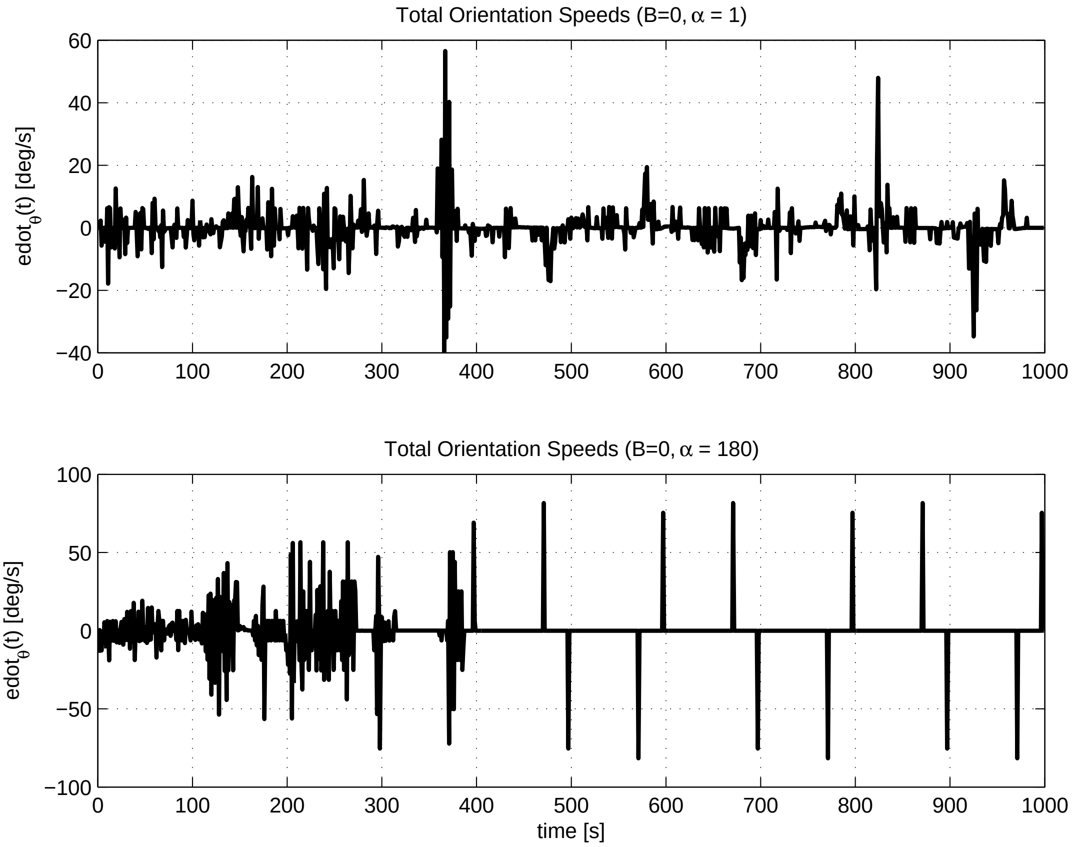


Figure 3.21: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Synchronous case - Bounded region).

where $\frac{1}{N}\sum_{j=1}^{N}(cos(\theta_j(t)))$ and $\frac{1}{N}\sum_{j=1}^{N}(sin(\theta_j(t)))$ are the estimations of means of $cos(\theta_j(t))$ and $sin(\theta_j(t))$, respectively. The expected $x = cos(\theta_i(t+1))$ and $y = sin(\theta_i(t+1))$ values of next orientation becomes

$$E[x] = E[cos(\theta_i(t+1))] = E\left[\frac{1}{N}\sum_{j=1}^{N}(cos(\theta_j(t)))\right] \tag{3.25a}$$

$$E[y] = E[sin(\theta_i(t+1))] = E\left[\frac{1}{N}\sum_{j=1}^{N}(sin(\theta_j(t)))\right] \tag{3.25b}$$

To find the estimations of means of the sinusoidal functions of $\theta_j(t)$ we will utilize the following relation [135, 136]. If θ is uniform in the interval $[0, 2\pi)$, the probability distribution of $\alpha = a\ sin(\theta + \gamma)$ is given by

$$f(\alpha) = \frac{1}{\pi\sqrt{a^2 - \alpha^2}}, \quad -a < y < a \tag{3.26}$$

Therefore, the probability distributions of $y = sin(\theta_j(t))$ and $x = cos(\theta_j(t)) = sin(\theta_j(t) + \pi/2)$

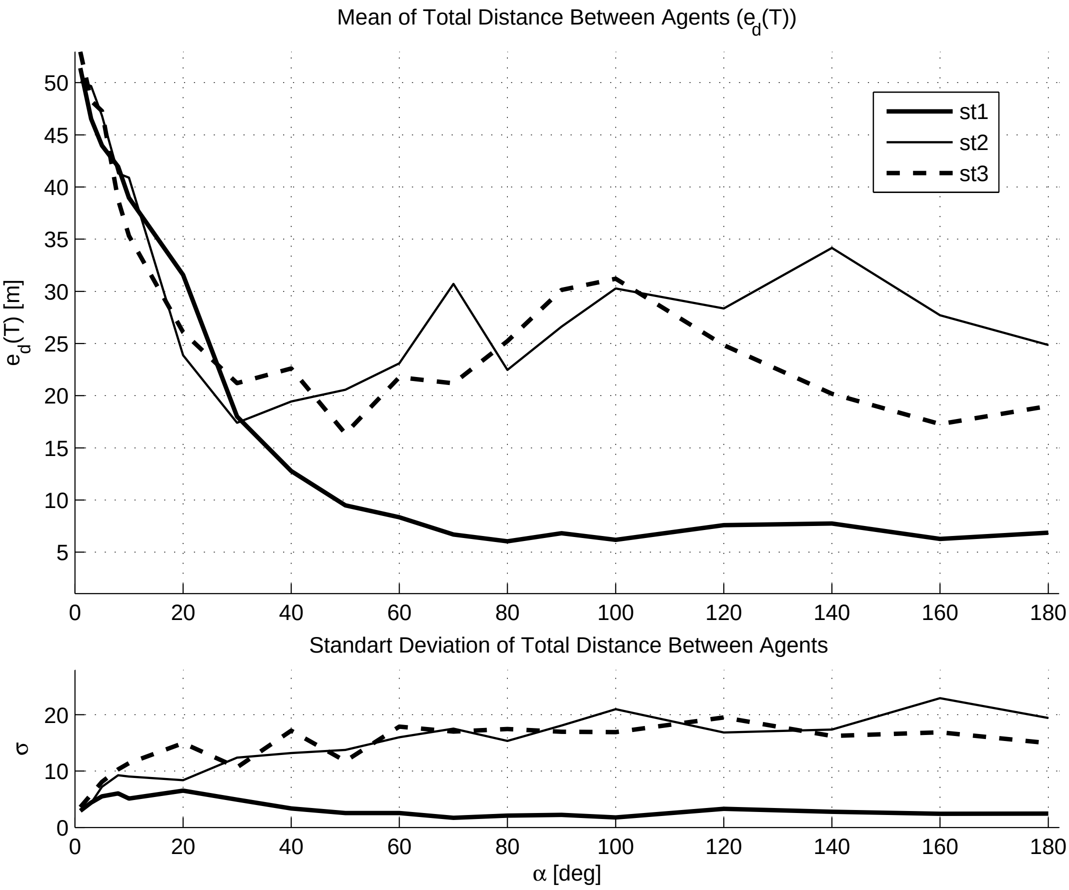


Figure 3.22: $e_d(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and bounded region.

are found as

$$f(x) = \frac{1}{\pi\sqrt{1-x^2}}, \quad -1 < x < 1 \tag{3.27a}$$

$$f(y) = \frac{1}{\pi\sqrt{1-y^2}}, \quad -1 < y < 1 \tag{3.27b}$$

The estimation of x is

$$E[x] = \int_{-1}^{1} x \frac{1}{\pi\sqrt{1-x^2}}\, dx \tag{3.28a}$$

$$= \frac{1}{2\pi}\sqrt{1-x^2}\,|_{-1}^{1} \tag{3.28b}$$

$$= 0 \tag{3.28c}$$

The same result can be found for the estimation of y as well. Hence, the expected results of means of x and y are 0. This result states that there is no particular expected output for the next orientation since there is no solution for θ satisfying $sin(\theta) = 0$ and $cos(\theta) = 0$. In other

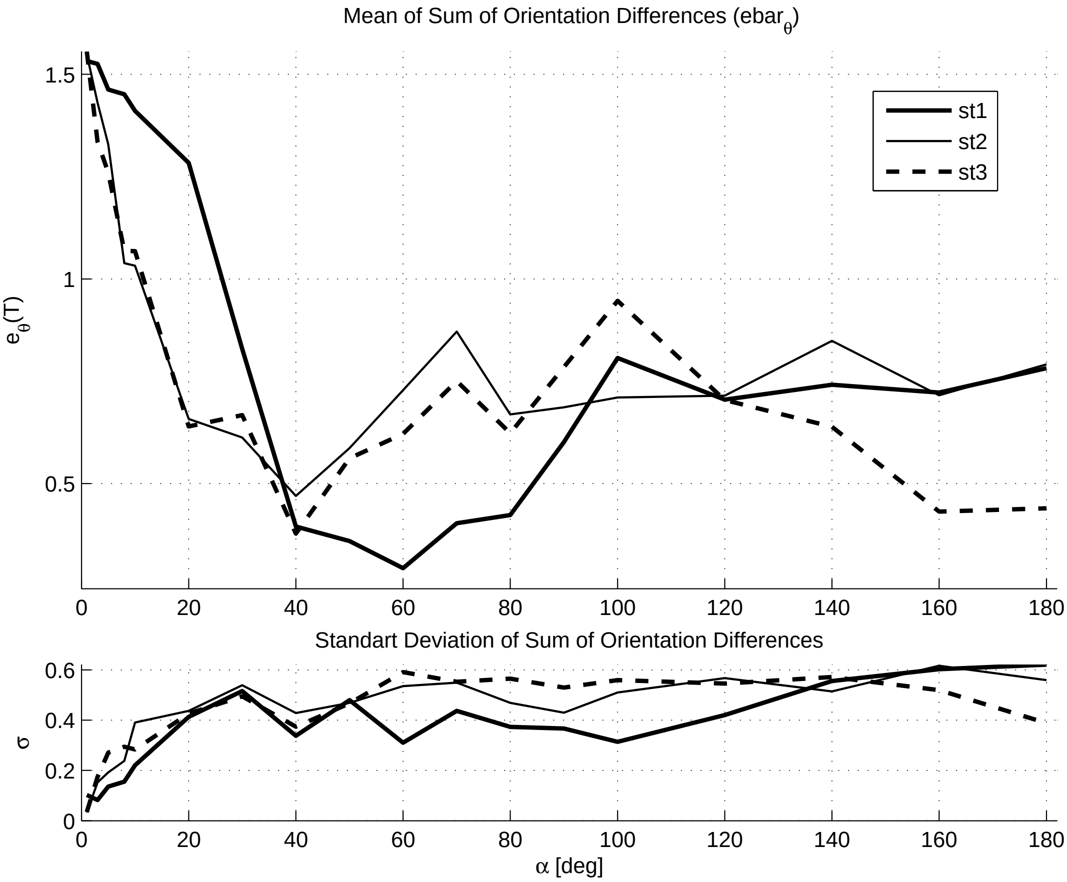


Figure 3.23: $e_\theta(T)$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and bounded region.

words $angle([0,0]) = \emptyset$. This result shows that there is no guided initial direction or bias for the agents using Strategy 3. In this sense Strategy 3 is similar to Strategy 2. Therefore, as was the case in Strategy 2, it is natural to expect that Strategy 3 as well is worse than Strategy 1 (which has a bias as towards π as one can recall). Hence, most of the comments for Strategy 2 hold for Strategy 3 as well.

3.7 Conclusions

In this study, we compared the performances of three different orientation agreement strategies. We analyzed the behavior of multi-agent systems utilizing these strategies with different combinations of the following properties: (i) the multi-agent systems are synchronous or asynchronous, (ii) they travel in bounded or unbounded regions and (iii) the mobile agents have various amount of turning speed restrictions. The agents try to orient themselves in the

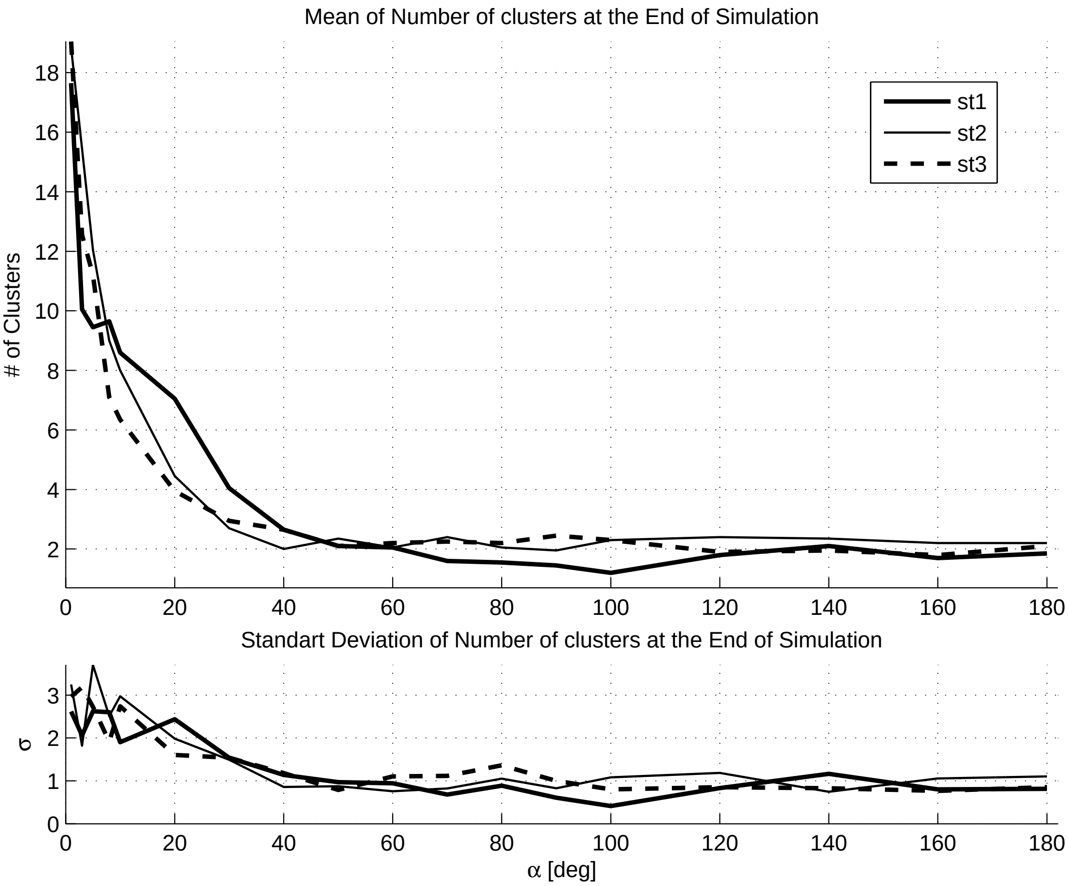


Figure 3.24: the number of clusters at $t = T$ for strategies 1 (bold solid line), 2 (solid line), 3(dash-dot line) for asynchronous case and bounded region.

same direction depending on three different interaction rules while at the same time moving with constant speed. We performed each simulation for various turn angle restrictions to observe the effects of non-holonomic dynamics. The agents exhibit best performance in orientation agreement in the case in which they are synchronous, holonomic and using Strategy 1 and in bounded region. In general for all simulations Strategy 1 has the best agreement performance. As discussed in the Discussion section we believe that this is mainly due to the fact that in Strategy 1 the agents are guided (i.e. have a bias) towards angle π. We showed that if the initial orientations of agents are drawn from a uniform distribution between $[0, 2\pi)$ the orientation updates of agents will be guided towards the angle π of global reference frame for Strategy 1. In contrast, the orientation updates of agents utilizing Strategy 2 and 3 are not guided towards any direction. Each of these strategies have advantages and disadvantages (as discussed in the Discussion section). While Strategy 1 performs best in orientation agreement, for implementation it needs agreement on a global reference frame and measurement

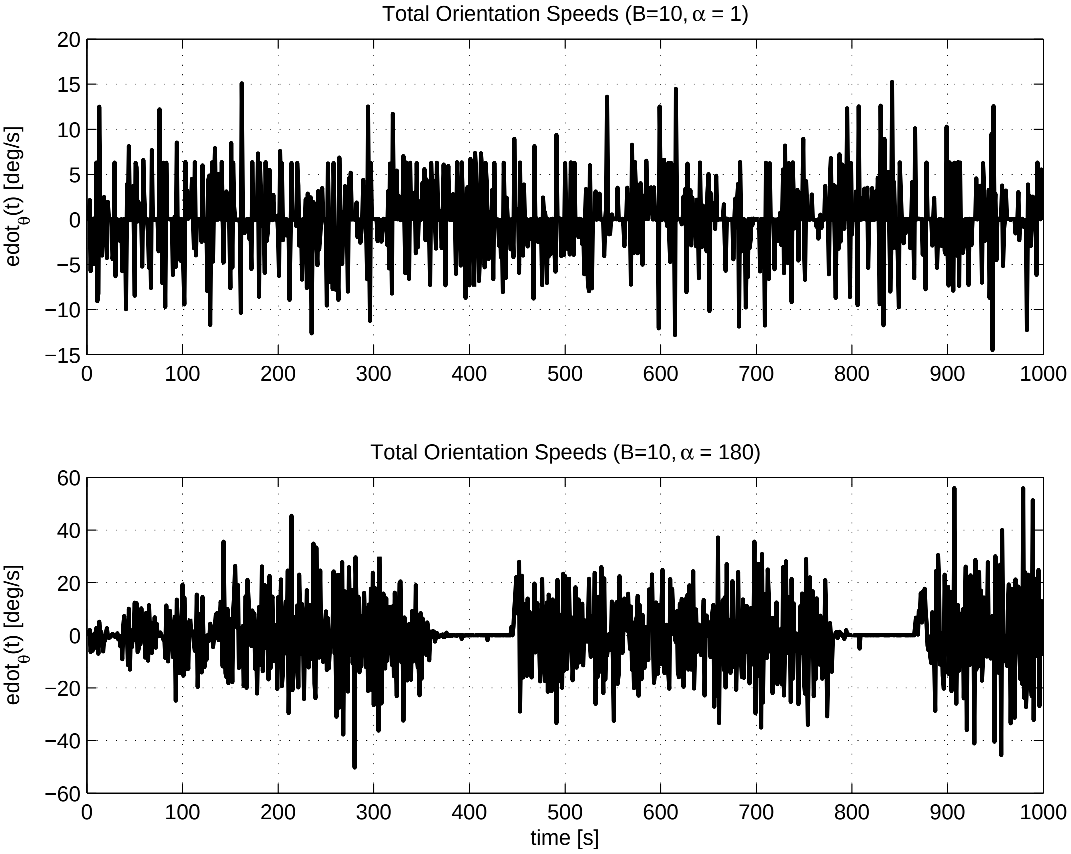


Figure 3.25: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 1 - Asynchronous case - Bounded region).

of the angles with respect to that frame, whereas Strategy 2 can be implemented using only relative information. For guided orientation agreement problems Strategy 1 would be a good choice. On the other hand, for the cases where global reference frame should not result in bias in orientations of swarms the second and third strategies are better to be utilized.

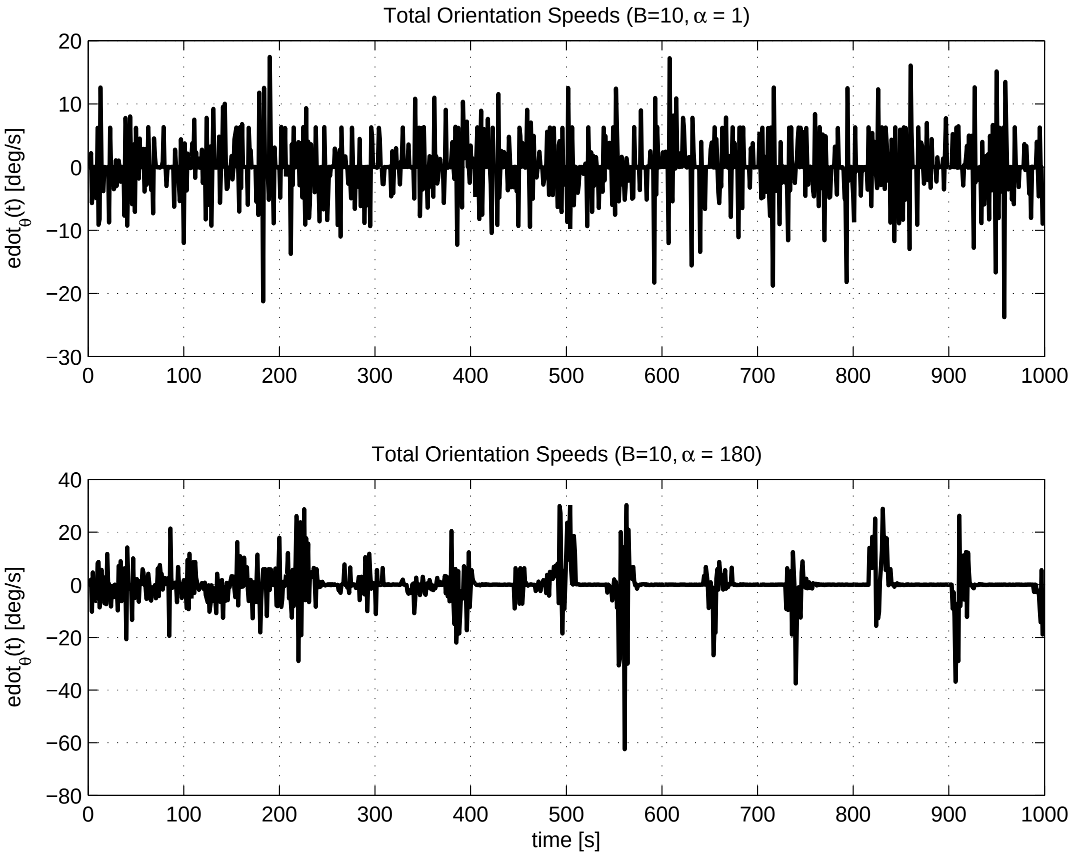


Figure 3.26: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 2 - Asynchronous case - Bounded region).

.

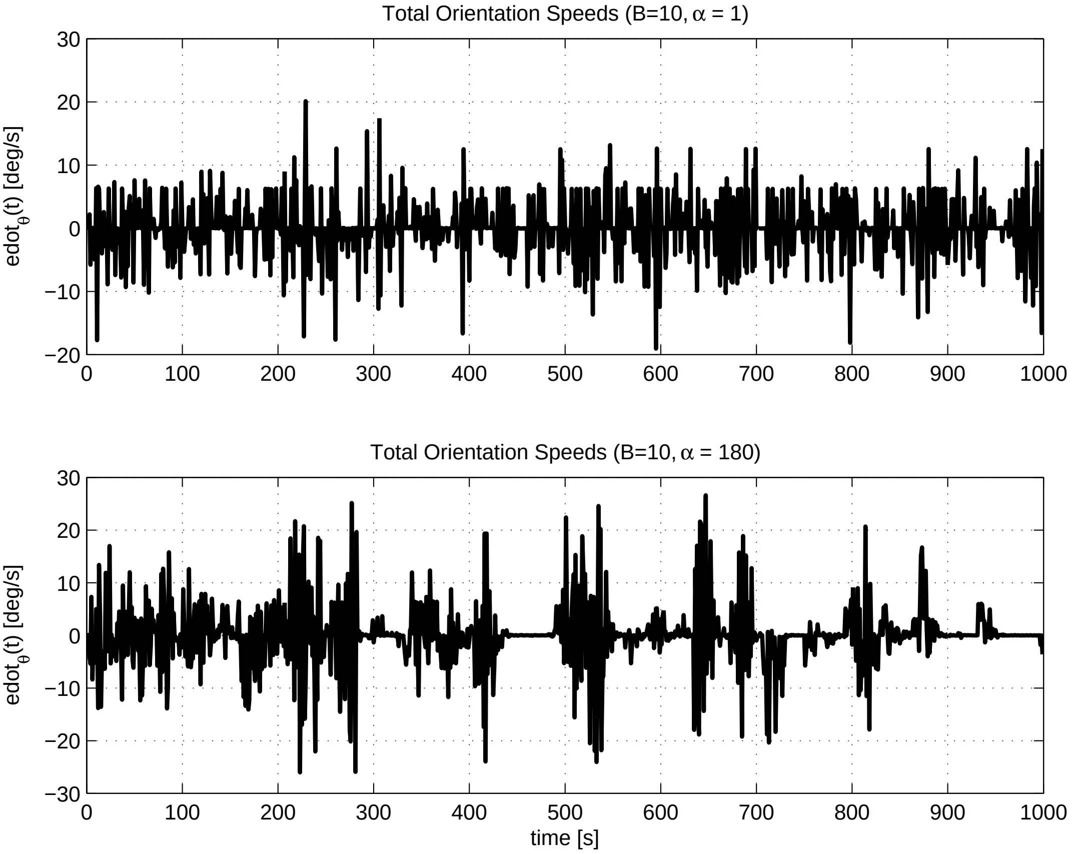


Figure 3.27: Total of rate of change of orientations of agents at each step for $\alpha = 1^o$ (upper subplot) and $\alpha = 180^o$ (lower subplot) for an arbitrary initial condition (Strategy 3 - Asynchronous case - Bounded region).

.

CHAPTER 4

Controllers for Tracking, Circling, and Line Following

4.1 Introduction

The trajectory generation and trajectory control methods for autonomous navigation of mobile robots have been studied in the literature. The studies in the literature mostly utilize a pre-generated path for tracking of mobile robots. By using these paths many researchers achieved the robots to follow straight lines, circular or randomly generated paths. In this study we propose new stable methods for a mobile robot to track a circular path of which the center -considered as the target- is given. The radius of the circular path can be adjusted by the controller parameters. We first analyze a target tracking method in relation with the circling behavior of a mobile robot. It is shown that by using the same proportional controller methods in determining the speed and angular velocity of robots the target tracking and circling around a target strategies can be achieved. A robot may have a target tracking behavior until a specified distance to the target and then switch to the circling around the target behavior without any external inputs. In the following sections, the mathematical background and stability analysis are presented. The results of the analytical derivations are validated in the next simulation study section. We lastly present the conclusions on the results of this part of the thesis study.

4.2 Mathematical Model

Here we consider the problem of target tracking with a non-holonomic agent with dynamics which obey the unicycle model with 3 degrees of freedom including the two translational and one rotational freedom which are given by

$$\dot{z}_1 = v\ \cos(\theta) \tag{4.1a}$$

$$\dot{z}_2 = v\ \sin(\theta) \tag{4.1b}$$

$$\dot{\theta} = u \tag{4.1c}$$

where v is the translational speed of the agent, θ is the orientation of the agent, and u is the controller for the angular speed of the agent. We assume that in the system there are one single robot and a target (see Figure 4.1). We will use a gradient vector ∇G which is a function of the agent and target positions. Moreover, we assume that if the distance between the agent and the target is larger than a predefined desired distance the gradient ∇G points from the agent towards the target, whereas if it is smaller than the desired distance ∇G points from the target towards the agent. It might represent the negative gradient of an artificial potential function which might be being utilized for attraction/repulsion purposes. In particular, the requirements on ∇G can easily be achieved with distance based potential functions such as considered in [17, 19, 18]. We will use a controller in which the translational speed and angular velocity of the agent are functions of the gradient vector in the form

$$u = u(\theta, \gamma) \tag{4.2a}$$

$$v = v(\nabla G) \tag{4.2b}$$

where $\gamma = angle(\nabla G)$ is the angle of the gradient vector ∇G. With appropriate transformation the dynamics of the system can be transformed into polar coordinates as (see Figure 4.1)

$$\dot{r} = v\ \cos(\theta - \phi) \tag{4.3a}$$

$$\dot{\phi} = \frac{1}{r} v\ \sin(\theta - \phi) \tag{4.3b}$$

$$\dot{\theta} = u(\theta, \gamma) \tag{4.3c}$$

where r is the distance from the origin to the position of the agent $r = \sqrt{x^2 + y^2}$ and ϕ is the angle of the vector along r, $\phi = atan2(y, x)$ and θ is the steering angle (orientation) of the agent as mentioned above.

For the time being we assume that the robot obtains the position of the target perfectly (exact values with synchronous timing i.e., no time delay in sensing) and the target is stationary. Consequently, the gradient vector is calculated without any errors and time delays and the change of the gradient vector just depends on the motion the robot. Furthermore, since we

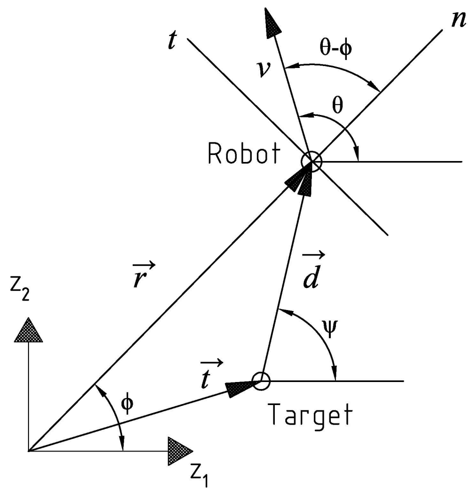


Figure 4.1: System model in polar coordinates.

are interested in the relative position of the robot with respect to the target, we will utilize the dynamics in relative coordinates. The reference coordinate frame (x_1-x_2 frame) has the origin on the target with parallel coordinate axes to the global coordinate frame axes. The relative coordinates are plotted in Figure 4.2 and can be described with the equations.

$$\begin{aligned} \dot{x}_1 &= v\ \cos(\theta) && (4.4a)\\ \dot{x}_2 &= v\ \sin(\theta) && (4.4b)\\ \dot{\theta} &= u(\theta, \gamma) && (4.4c) \end{aligned}$$

Again with the coordinate transformation $d = \sqrt{x_1^2 + x_2^2}$ and $\phi = atan2(x_2, x_1)$ and under the assumption that the target is stationary the dynamics of the robot in the polar coordinates relative to the target is obtained as

$$\begin{aligned} \dot{d} &= v\cos(\theta - \psi) && (4.5a)\\ \dot{\psi} &= \frac{1}{d}v\sin(\theta - \psi) && (4.5b)\\ \dot{\theta} &= u(\theta, \gamma) && (4.5c) \end{aligned}$$

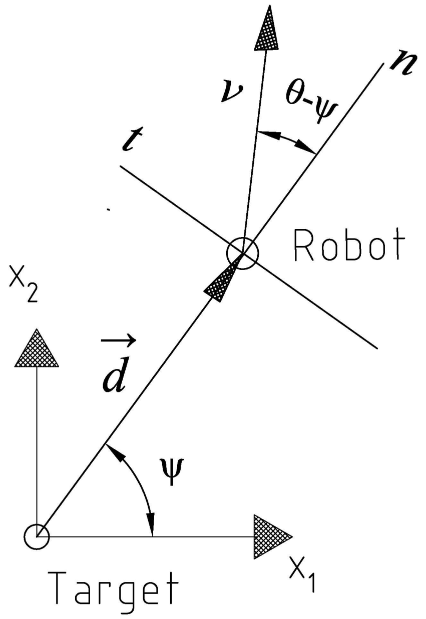


Figure 4.2: System model for relative coordinates.

4.3 Targeting and Circling Around a Target

Let us assume that the distance between the target and the agent is larger than the desired distance. As was mentioned before, for that case we design the controller such that the gradient vector points towards the target from the robot, i.e., in the opposite direction of the $\vec{d}$ vector in Figure 4.2. The angle of the gradient vector becomes

$$\gamma = angle(\nabla G) = \psi + \pi \tag{4.6}$$

On the other hand, for the case where the robot is close to the target more than the desired distance, the gradient angle is directed in the same direction as the $\vec{d}$ vector, and therefore, the angle of the gradient vector is $\gamma = angle(\nabla G) = \psi$. We will just examine the first case where the gradient vector is in the opposite direction of the $\vec{d}$ vector.

The angular velocity controller is selected as a proportional controller that directs the robot towards the gradient vector. However, at this point the question is from which side to turn towards the desired direction of motion. We choose the direction of the smaller relative angle for that purpose. In other words, the error is defined as the smaller relative angle difference for the P controller so that the robot rotates towards the smaller relative angle to attain the direction of the gradient vector. Therefore, we utilize for the controller function $u(\theta, \gamma)$

$$u(\theta, \gamma) = -\alpha \left[\mathrm{mod}(\theta - \gamma + \pi, 2\pi) - \pi\right] \tag{4.7}$$

where $\alpha > 0$ is the proportional controller coefficient. Equation (4.7) includes the modulo operator for selecting the smaller angle difference between the direction of the robot and the gradient vector. Now substituting equations (4.6) and (4.7) into (4.5) we get the system dynamics

$$\begin{aligned} \dot{d} &= v\cos(\theta - \psi) && (4.8a) \\ \dot{\psi} &= \frac{1}{d} v\sin(\theta - \psi) && (4.8b) \\ \dot{\theta} &= -\alpha \left[\mathrm{mod}(\theta - \psi, 2\pi) - \pi\right] && (4.8c) \end{aligned}$$

4.3.1 Constant Speed Controllers

If the speed v is a positive constant then the equilibrium of the system can be found as following: equilibrium occurs at points where $\dot{d} = 0$, $\dot{\psi} = 0$, and $\dot{\theta} = 0$ are satisfied simultaneously. Then from $\dot{d} = 0$ we have

$$v\ \cos(\theta - \psi) = 0 \Longrightarrow\ v = 0\ or\ \theta - \psi = \frac{\pi}{2} + k\pi \tag{4.9}$$

for $k = 0, \mp 1, \mp 2$. Similarly, from $\dot{\psi} = 0$ we have

$$\frac{1}{d} v\ \sin(\theta - \psi) = 0 \Longrightarrow\ v = 0\ or\ \theta - \psi = k\pi \tag{4.10}$$

for $k = 0, \mp 1, \mp 2$. Also, from $\dot{\theta} = 0$ one can obtain

$$-\alpha\ \left[\mathrm{mod}(\theta - \psi, 2\pi) - \pi\right] = 0 \Longrightarrow \theta - \psi = (2k+1)\pi \tag{4.11}$$

for $k = 0, \mp 1, \mp 2$. Then from the common solution of the equations (4.9), (4.10), (4.11) one can determine the set of equilibrium points E of the system as

$$E = \{v \in R^{+}, 0 < \theta, \psi < 2\pi | v = 0, \theta - \psi = (2k+1)\pi\} \tag{4.12}$$

where $k = 0, \mp 1, \mp 2....$ Moreover, in addition to the set of equilibrium points in (4.12) in the simulation studies of this system we observed periodic trajectories, which might be due to a limit cycle. Therefore, we will define a limit cycle at for $\dot{d} = 0$ and $\dot{\theta} - \dot{\psi} = 0$, and $v \neq 0$

the system may exhibit periodic behavior. Solving the system dynamics under the above constraints one obtains the solutions for these equations as

$$\dot{d} = v\ \cos(\theta - \psi) = 0 \Longrightarrow\ \theta - \psi = \frac{\pi}{2} + k\pi \quad for\, k = 0, \mp 1, \mp 2 \tag{4.13}$$

and

$$\dot{\theta} - \dot{\psi} = -\alpha\ [mod(\theta - \psi, 2\pi) - \pi] - \frac{1}{d} v\ \sin(\theta - \psi) = 0 \tag{4.14}$$

Solving equations (4.13) and (4.14) simultaneously, we get two independent solutions. The first solution is at $\theta - \psi = \pi/2$ at which (4.14) gives

$$-\alpha\ [\text{mod}(\pi/2, 2\pi) - \pi] - \frac{1}{d} v\ \sin(\pi/2) = 0 \tag{4.15a}$$

$$\Longrightarrow -\alpha \frac{-\pi}{2} = \frac{1}{d} v \tag{4.15b}$$

$$\Longrightarrow d = \frac{v}{\alpha\pi/2}. \tag{4.15c}$$

Similarly the second solution is at $\theta - \psi = 3\pi/2$, at which (4.14) becomes

$$-\alpha\ [\text{mod}(3\pi/2, 2\pi) - \pi] - \frac{1}{d} v\ \sin(3\pi/2) = 0 \tag{4.16a}$$

$$\Longrightarrow -\alpha \frac{\pi}{2} = -\frac{1}{d} v \tag{4.16b}$$

$$\Longrightarrow d = \frac{v}{\alpha\pi/2}. \tag{4.16c}$$

This results in the set of solutions

$$\left(\theta - \psi = \frac{\pi}{2}, d = \frac{2v}{\alpha\pi}\right) \text{ and } \left(\theta - \psi = \frac{3\pi}{2}, d = \frac{2v}{\alpha\pi}\right).$$

From the above solutions the periodic solution (or the limit cycle) can be defined as

$$L = \{v \in R, 0 < \theta, \psi < 2\pi | d = \frac{v}{\alpha\pi/2}, \theta - \psi = \frac{\pi}{2} + k\pi\} \tag{4.17}$$

where $k = 0, \mp 1, \mp 2$.

Note that for the case where the gradient vector points in the direction of the $\overrightarrow{d}$ vector and the robot is closer to the target than a pre-defined desired distance, the equilibrium set becomes

$$E' = \{v \in R^{+}, 0 < \theta, \psi < 2\pi | v = 0, \theta - \psi = 2k\pi\}\ for\ \ k = 0, \mp 1, \mp 2 \tag{4.18}$$

Also note that, there is no limit cycle set for this case. It is interesting to note that in both cases at equilibrium the steering angle (the orientation) of the robot points in the direction of the gradient while translational speed is zero.

Now we will examine the stability of the equilibrium set E and the limit cycle L. Therefore, the time derivatives of the distance between the robot and the target, d, (Equation (4.8a)) and the time derivatives of the difference between the orientation of the robot and the orientation of the distance vector $\overrightarrow{d}$ which is

$$\dot{\theta} - \dot{\psi} = -\alpha\ [mod(\theta - \psi, 2\pi) - \pi] - \frac{1}{d} v\ \sin(\theta - \psi) \tag{4.19}$$

will be examined for different values of controller parameters.

The roots of $\dot{\theta} - \dot{\psi} = 0$ can be found by solving the nonlinear equation (4.19), numerically. Depending on the values of the controller parameter α, the distance between the robot and the target d, and the speed of the robot v, two different cases are possible. The equation may have a single root or three different roots. In Figure 4.3 the plot of $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ is presented for different values of v. For small values of v there is only one root and for larger values of v there are three roots (the points intersecting the x-axis). The number of roots of the function changes when the slope of the function at $\theta - \psi = \pi$ changes from negative to positive. Solving for the slope at $\theta - \psi = \pi$ we obtain

$$\begin{aligned}
\frac{d(\dot{\theta} - \dot{\psi})}{d(\theta - \psi)} &= \frac{d}{d(\theta - \psi)}\left[-\alpha\ [\mathrm{mod}(\theta - \psi, 2\pi) - \pi] - \frac{1}{d} v\ \sin(\theta - \psi)\right] && (4.20a)\\
&= -\alpha - \frac{1}{d} v\ \cos(\theta - \psi) && (4.20b)\\
\frac{d(\dot{\theta} - \dot{\psi})}{d(\theta - \psi)}|_{\theta - \psi = \pi} &= -\alpha - \frac{1}{d} v\ \cos(\pi) = 0 \ \ \Rightarrow v = \alpha d && (4.20c)
\end{aligned}$$

Therefore, for $v \leq \alpha d$ the only root is at $\theta - \psi = \pi$. This point is a stable root since the slope is negative at this point. Note that at $\theta - \psi = \pi$, provided that the other conditions are also satisfied the system is at the equilibrium set E defined in (4.12). The angular velocity of the robot is $\dot{\theta} = -\alpha\ [\mathrm{mod}(\theta - \psi, 2\pi) - \pi] = 0$ and the other state variables are $\dot{d} = v\ \cos(\theta - \psi) = -v$, $\dot{\psi} = \frac{1}{d} v\ \sin(\theta - \psi) = 0$ at $\theta - \psi = \pi$. Therefore, if v is a positive variable which can easily be guaranteed the robot travels towards the target on a straight line. And if a controller is designed such that the speed v is set to zero at a specified distance from the target then it stops at that distance after having a straight line motion.

On the other hand, if $v > \alpha d$ then there are two more symmetric roots on the left and right of π. For this case the slope of the function is positive for $\theta - \psi = \pi$ and therefore, it is an unstable root. The slope is negative for the other two roots showing that they are stable roots. Note that for $v = \alpha d\pi/2$ the stable roots are $\theta - \psi = \pi/2$ and $\theta - \psi = 3\pi/2$ which is in fact the limit cycle defined in (4.17). For better understanding of the relations between the state variables

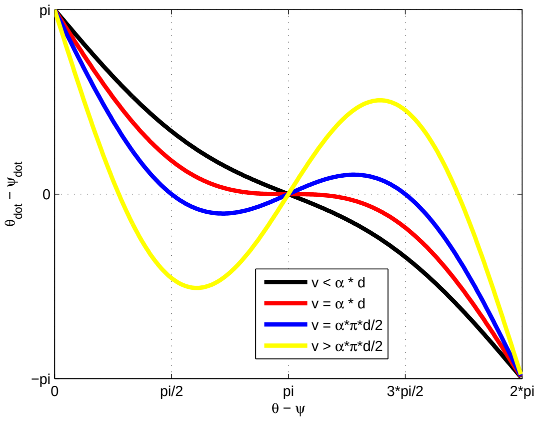


Figure 4.3: $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ for several v values.

and therefore, the system behavior, we should consider the change of $\dot{\theta} - \dot{\psi}$ with respect to both distance between the agent and the target and the relative angle difference $\theta - \psi$. Hence, a 3-dimensional plot of state variables $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ and the distance d is presented in Figure 4.4.

The speed of the robot is kept constant at $v = \bar{r}\alpha\pi/2$ for generating this figure. Here, $\bar{r}$ is the radius of the periodic motion when the system converges to the limit cycle. The roots or the points where the surface intersects the zero-plane are plotted with thick black curves. The straight lines (blue ones) are the lines at $\theta - \psi = \pi/2$ and $\theta - \psi = 3\pi/2$ for better comparison. Now let us consider the case in which the robot is at a distance of 100 [units] from the target. The figure shows that, provided that the distance $d = 100$ [units] is kept constant, the state variables will converge to the stable equilibrium point $\theta - \psi = \pi$ where the orientation angle of the robot is equal to the orientation of the gradient vector. Also note that the state variables $\dot{\theta}$ and $\dot{\psi}$ are zero at this equilibrium. However, as the robot gets closer to the target, i.e., as d decreases the equilibrium point $\theta - \psi = \pi$ becomes unstable. Therefore, if it is desired for the robot to move towards the target and stay at a predefined distance to it, one may prefer to choose the controller parameters (α and v) such that the robot stops before it reaches the region where $\theta - \psi = \pi$ root is unstable. This strategy may be utilized for just tracking a target from a specified distance. In other words, one may choose the parameters such that $v \leq \alpha d$ is always satisfied.

If the controller is designed/adapted for the robot to trace a circle around the target (circling in short) then the controller parameters should be selected for the robot to continue through the region where the equilibrium point $\theta - \psi = \pi$ is unstable and the other two symmetric roots are stable. In that region depending on the initial condition the difference $\theta - \psi$ will converge to one of the roots on the left or right of π which will result in circling the target at either CW[1] or CCW[2] direction. Note that the behavior of circling around a target might be a desired behavior for some UAV[3] applications This is because UAV's cannot hover or travel below some pre-specified speeds and in order to trace some ground vehicles/targets which might be moving with much slower velocities they might need to circle above the target.

Let us first consider that $\theta - \psi$ converges to a point where $\pi/2 < \theta - \psi < \pi$. Here we mean a point on the black curve since there we have $\dot{\theta} - \dot{\psi} = 0$. At this point the time derivative of the state variable $\dot{d} = v \ \cos(\theta - \psi)$ is negative. Therefore, the robot will continue to get closer to the target until $\theta - \psi = \pi/2$. In contrast, if it converges to a point (on the black curve) between $0 < \theta - \psi < \pi/2$, then $\dot{d} = v \ \cos(\theta - \psi)$ is positive in this case. Therefore, the roots $\theta - \psi = \pi/2$ and $\dot{d} = 0$ are a set of stable points of this system as mentioned for the limit cycle above in equation (4.17). For this limit cycle set the robot is circling in the CCW direction around the target. The same conclusion can be derived similarly for the other root on the right of π. For that case the state variables will converge to the set at which $\theta - \psi = 3\pi/2$ and $\dot{d} = 0$ which in fact represents the CW circling of the robot around the target.

Note that in the above derivations we assumed that the gradient vector is directed towards the target from the robot. Therefore, we utilized equation (4.6) in the analysis. However, some gradient vectors are designed to direct in the opposite direction when the robot is too close to the target especially for collision avoidance with a similar analysis. One can easily show that in that case the system will converge to the stable set E' that force the robot to get away from the target in the same direction with this new gradient vector. The derivation for that case is straightforward and therefore, left out.

4.3.1.1 Stability Analysis via Lyapunov Functions

Here, we will find a Lyapunov function candidate that shows the stability of the above system. Before we present the Lyapunov function we should present the second derivatives of the

---

[1] CW: clockwise

[2] CCW: counter clockwise

[3] Unmanned Air Vehicle

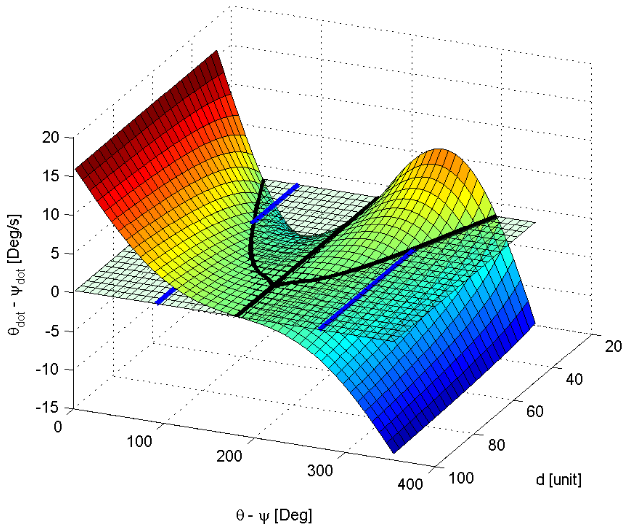


Figure 4.4: $\dot\theta - \dot\psi$ versus $\theta - \psi$ and the distance d for constant $v = r\alpha\pi/2$ where r is the radius of the circle on which the robot travels.

system variables to be utilized in the time derivative of the Lyapunov function. Consider the system dynamics in (4.5). Under the assumption that the linear speed v is constant the second derivatives with respect to time of the variables are

$$\begin{aligned} \ddot d &= -v\sin(\theta-\psi)(\dot\theta-\dot\psi) \\ &= -d\dot\psi(\dot\theta-\dot\psi) \qquad (4.21a) \\ \ddot\psi &= -\frac{\dot d}{d^2}v\sin(\theta-\psi) + \frac{1}{d}v\cos(\theta-\psi)(\dot\theta-\dot\psi) \\ &= -\frac{\dot d}{d}\dot\psi + \frac{\dot d}{d}(\dot\theta-\dot\psi) \\ &= \frac{\dot d}{d}(\dot\theta-2\dot\psi) \qquad (4.21b) \\ \ddot\theta &= -\alpha(\dot\theta-\dot\psi) \qquad (4.21c) \end{aligned}$$

The Lyapunov function candidate is

$$V(\dot d,\dot\theta,\dot\psi) = \frac{1}{2}\dot d^2 + \frac{1}{2}[d(\dot\theta-\dot\psi)]^2 \qquad (4.22)$$

where $D = \{\dot d,\dot\theta,\dot\psi \in \mathbb{R}\}$. $V(\dot d,\dot\theta,\dot\psi)$ is positive definite in D. Then the Lie derivative of the Lyapunov function becomes

$$\dot V = \dot d\ddot d + [d(\dot\theta-\dot\psi)][\dot d(\dot\theta-\dot\psi) + d(\ddot\theta-\ddot\psi)] \qquad (4.23)$$

which substituting the second derivatives in (4.21) becomes

$$\dot V = \dot d\left(-d\dot\psi(\dot\theta-\dot\psi)\right) + \left(d(\dot\theta-\dot\psi)\right)\left(\dot d(\dot\theta-\dot\psi) + d\left[-\alpha(\dot\theta-\dot\psi) - \frac{\dot d}{d}(\dot\theta-2\dot\psi)\right]\right) \qquad (4.24)$$

Then, rearranging

$$\begin{aligned}
\dot{V} &= -d\dot{d}\dot{\psi}(\dot{\theta} - \dot{\psi}) + d(\dot{\theta} - \dot{\psi})\Big(\dot{d}(\dot{\theta} - \dot{\psi}) - \alpha d(\dot{\theta} - \dot{\psi}) - \dot{d}(\dot{\theta} - 2\dot{\psi})\Big) && (4.25a)\\
&= -d\dot{d}\dot{\psi}(\dot{\theta} - \dot{\psi}) + d(\dot{\theta} - \dot{\psi})\Big(\dot{d}(\dot{\theta} - \dot{\psi} - \dot{\theta} + 2\dot{\psi}) - \alpha d(\dot{\theta} - \dot{\psi})\Big) && (4.25b)\\
&= -d\dot{d}\dot{\psi}(\dot{\theta} - \dot{\psi}) + d(\dot{\theta} - \dot{\psi})\Big(\dot{d}\dot{\psi} - \alpha d(\dot{\theta} - \dot{\psi})\Big) && (4.25c)\\
&= -d\dot{d}\dot{\psi}(\dot{\theta} - \dot{\psi}) + \Big(d\dot{d}\dot{\psi}(\dot{\theta} - \dot{\psi}) - \alpha d^2(\dot{\theta} - \dot{\psi})^2\Big) && (4.25d)\\
&= -\alpha d^2(\dot{\theta} - \dot{\psi})^2 \leq 0 && (4.25e)
\end{aligned}$$

The Lie derivative of the Lyapunov function, $\dot{V}$ is negative semi-definite within set D. The stability should be investigated by the use of Lasalle's principle.

4.3.2 Speed and Angular Velocity Controllers

For most of the mobile robotic applications and UGV/UAV applications the translational and angular velocity of the agents are adjusted with respect to the relative distance and relative angles between the agents or targets. The speed and angular velocity can be adjusted by many controller methods like PID and nonlinear controllers, having the relative distance and relative orientation as the controller parameters.

In this section we will examine the system behavior assuming that the speed v is a function of relative distance, $v = v(d)$ and the angular velocity is a function of relative distance $\dot{\theta} = f(\theta - \psi)$. Then the system dynamics can be expressed as

$$\begin{aligned}
\dot{d} &= v(d)\ \cos(\theta - \psi) && (4.26a)\\
\dot{\psi} &= \frac{1}{d}\, v(d)\ \sin(\theta - \psi) && (4.26b)\\
\dot{\theta} &= g(\theta - \psi) && (4.26c)
\end{aligned}$$

For simplicity of analysis we lump the states θ and ψ as $\gamma = \theta - \psi$. Then the two state system model becomes

$$\begin{aligned}
\dot{d} &= v(d)\ \cos(\gamma) && (4.27a)\\
\dot{\gamma} &= g(\gamma) - \frac{1}{d}\, v(d)\ \sin(\gamma) && (4.27b)
\end{aligned}$$

Utilizing appropriate controllers, the system is expected to converge targeting and circling behaviors. These behaviors can be expressed by the following equilibrium sets

$$E = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\} \tag{4.28}$$

$$L_1 = \{d \in R^+, 0 < \gamma < 2\pi | v(d_{eq}) = d_{eq}\, g(\pi/2),\ \ \gamma = \frac{\pi}{2}\} \tag{4.29}$$

$$L_2 = \{d \in R^+, 0 < \gamma < 2\pi | v(d_{eq}) = -d_{eq}\, g(3\pi/2),\ \ \gamma = \frac{3\pi}{2}\} \tag{4.30}$$

where E stands for the targeting behavior and L_1 and L_2 stands for the circling in CW and CCW directions, respectively.

We can examine the stability of these points by linearization methods.

4.3.2.1 Stability via Linearization

Here, we will examine the stability of the system at the equilibriums E, and L_1 and L_2 by linearizing around these points. Let

$$\dot{d} = f_1(d,\gamma) = v(d)\ \cos(\gamma) \tag{4.31a}$$

$$\dot{\gamma} = f_2(d,\gamma) = g(\gamma) - \frac{1}{d}\ v(d)\ \sin(\gamma) \tag{4.31b}$$

The Jacobian matrix is

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial d} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial d} & \frac{\partial f_2}{\partial \gamma} \end{bmatrix} \tag{4.32}$$

$$\frac{\partial f}{\partial x} = \begin{bmatrix} v' \cos(\gamma) & -v\ \sin(\gamma) \\ \frac{1}{d^2}\ v\ \sin(\gamma) - \frac{1}{d}\ v'\ \sin(\gamma) & g'(\gamma) - \frac{1}{d}\ v(d)\ \cos(\gamma) \end{bmatrix} \tag{4.33}$$

The value of the linearized coefficient matrix at equilibrium $L_1 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = d\, g(\pi/2),\ \ \gamma = \frac{\pi}{2}\}$ is

$$A = \left.\frac{\partial f}{\partial x}\right|_{L_1} = \begin{bmatrix} 0 & -d\, g(\pi/2) \\ \frac{1}{d^2}\ d\, g(\pi/2) - \frac{1}{d}\ v' & g'(\pi/2) \end{bmatrix} \tag{4.34}$$

$$A = \begin{bmatrix} 0 & -d\, g(\pi/2) \\ \frac{1}{d}\,(g(\pi/2) -\ v') & g'(\pi/2) \end{bmatrix} \tag{4.35}$$

The eigenvalues of this Jacobian matrix A are

$$|A - I\lambda| = \begin{vmatrix} -\lambda & -d\, g(\pi/2) \\ \frac{1}{d}\,(g(\pi/2) -\ v') & g'(\pi/2) - \lambda \end{vmatrix} = \lambda(\lambda - g'(\pi/2)) + d\, g(\pi/2)\frac{1}{d}\,(g(\pi/2) -\ v') \tag{4.36}$$

$$|A - I\lambda| = \lambda^2 - g'(\pi/2)\lambda + g(\pi/2)\,(g(\pi/2) -\ v') = 0 \tag{4.37}$$

solving the quadratic equation

$$\lambda_{1,2} = \frac{g'(\pi/2) \mp \sqrt{g'(\pi/2)^2 - 4g(\pi/2)\,(g(\pi/2) - v')}}{2} \tag{4.38}$$

The system is stable around L_1 if $real(\lambda_{1,2}) < 0$. This requires two conditions: $g'(\pi/2) < 0$ and

$$-\,4g(\pi/2)\,(g(\pi/2) - v') < 0 \tag{4.39}$$

$$g(\pi/2)\,(g(\pi/2) - v') > 0 \tag{4.40}$$

If $g(\pi/2) > 0$ then $g(\pi/2) > v'$ and if $g(\pi/2) < 0$ then $g(\pi/2) < v'$. Hence, the speed and angular velocity function satisfying $v' < g(\pi/2)$ and $g'(\pi/2) < 0$ at $\gamma = \frac{\pi}{2}$ would result in the stability at L_1.

The same solution is applied for set $L_2 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = -d\ g(3\pi/2),\ \ \gamma = \frac{3\pi}{2}\}$. Then the Jacobian matrix is

$$A = \left.\frac{\partial f}{\partial x}\right|_{L_2} = \begin{bmatrix} 0 & d\ g(3\pi/2) \\ -\frac{1}{d^2}\ d\ g(3\pi/2) + \frac{1}{d}\ v' & g'(3\pi/2) \end{bmatrix} \tag{4.41}$$

$$A = \begin{bmatrix} 0 & d\ g(3\pi/2) \\ -\frac{1}{d}\,(g(3\pi/2) -\ v') & g'(3\pi/2) \end{bmatrix} \tag{4.42}$$

The eigenvalues of this Jacobian matrix are

$$|A - I\lambda| = \begin{vmatrix} -\lambda & d\ g(3\pi/2) \\ -\frac{1}{d}\,(g(3\pi/2) -\ v') & g'(3\pi/2) - \lambda \end{vmatrix} = \lambda(\lambda - g'(3\pi/2)) + d\ g(3\pi/2)\frac{1}{d}\,(g(3\pi/2) -\ v') \tag{4.43}$$

$$|A - I\lambda| = \lambda^2 - g'(3\pi/2)\lambda + g(3\pi/2)\,(g(3\pi/2) -\ v') = 0 \tag{4.44}$$

solving the quadratic equation

$$\lambda_{1,2} = \frac{g'(3\pi/2) \mp \sqrt{g'(3\pi/2)^2 - 4g(3\pi/2)\,(g(3\pi/2) - v')}}{2} \tag{4.45}$$

The system is stable around L_2 if $real(\lambda_{1,2}) < 0$. The required conditions are

$$g'(3\pi/2) < 0 \tag{4.46}$$

and

$$-\,4g(3\pi/2)\,(g(3\pi/2) - v') < 0 \tag{4.47}$$

$$g(3\pi/2)\,(g(3\pi/2) - v') > 0 \tag{4.48}$$

Similar to the previous results, if $g(3\pi/2) > 0$ then $g(3\pi/2) > v'$ and if $g(3\pi/2) < 0$ then $g(3\pi/2) < v'$.

So, the speed and angular velocity function satisfying $v' < g(3\pi/2)$ and $g'(3\pi/2) < 0$ at $\gamma = \frac{3\pi}{2}$ would result in the stability at L_2.

Similarly for the equilibrium set $E = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\}$ the Jacobian matrix A is

$$A = \left.\frac{\partial f}{\partial x}\right|_E = \begin{bmatrix} -v' & 0 \\ 0 & g'(\pi) \end{bmatrix} \tag{4.49}$$

The eigenvalues of this state coefficient matrix A are

$$|A - I\lambda| = \begin{vmatrix} -v' - \lambda & 0 \\ 0 & g'(\pi) - \lambda \end{vmatrix} = (v' + \lambda)(\lambda - g'(\pi)) = 0 \tag{4.50}$$

$$|A - I\lambda| = (\lambda + v')(\lambda - g'(\pi)) = 0 \tag{4.51}$$

solving the quadratic equation

$$\lambda_1 = -v' \tag{4.52}$$

$$\lambda_2 = g'(\pi) \tag{4.53}$$

The system is stable around E if $real(\lambda_{1,2}) < 0$. The conditions to be satisfied for the stability are $v' > 0$ and $g'(\pi) < 0$

Then the speed function satisfying $v'(d) > 0$ and $g'(\pi) < 0$ while $v(d) = 0$ at $\gamma = \pi$ would result in the stability at E.

4.3.2.2 Stability Analysis via Lyapunov Functions

The dynamics of the system is highly nonlinear. And as derived in section 4.3.2 the system has bifurcation at certain points. Therefore, the dynamics will be examined for two cases. The first is the stability to the set E which in fact the targeting behavior of the system. The second is the case where the dynamics converges to the circling behaviors described by the sets L_1 and L_2.

For the targeting behavior the state γ converges to the point $\gamma = \pi$. The differential equation of this state can be represented as the summation of two periodic functions with periods 2π. For the targeting case the summation of these periodic functions results in a new periodic

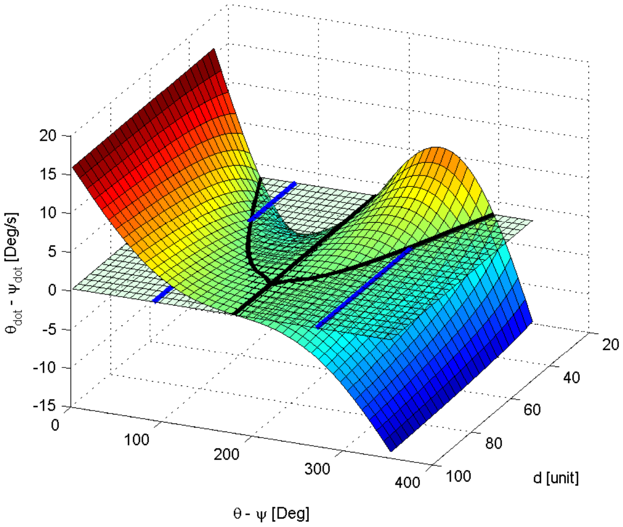


Figure 4.5: $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ and the distance d for constant $v = r\alpha\pi/2$ where r is the radius of the circle on which the robot travels.

function with period 2π. By the fourier transforms of the functions one can find that the time derivative of the state faces with two different types of functions due to the bifurcation. As seen in the Figure 4.6 there occurs different roots due to the relation between α, v, and d. We will examine the stability of these two different cases. The bifurcation can be seen on the Figure 4.5 clearly.

In the first case these is just one root at $\gamma = \pi$ and the function is an odd periodic function. Therefore, the function will be approximated by the first fourier transform term such that

$$\dot{\gamma} = K \ \sin(\gamma) \tag{4.54}$$

where K is a positive constant. The equilibrium point is $\gamma = \pi$. The Lyapunov function candidate for this system is

$$V = \frac{1}{2}(\gamma - \pi)^2 \tag{4.55}$$

The Lie derivative of this function is

$$\dot{V} = (\gamma - \pi)\dot{\gamma} = K \ (\gamma - \pi) \ \sin(\gamma) \leq 0 \tag{4.56}$$

Therefore, the system is locally asymptotically stable at the equilibrium point $\gamma = \pi$.

The second function occurring due to the bifurcation has three equilibrium points. The first

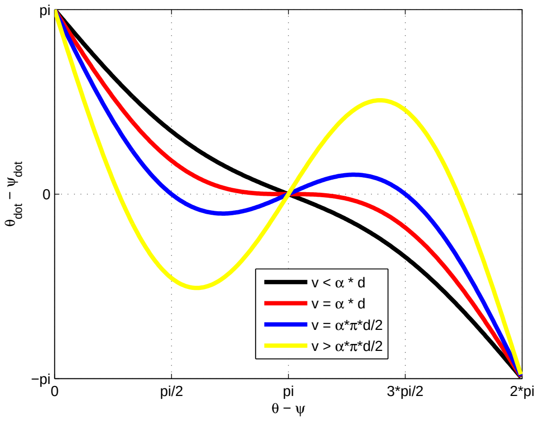


Figure 4.6: $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ for several v values.

fourier expansion term of these kind of functions would be the function $\sin(2\gamma)$. Therefore, the stability analysis of this function would imply the results for the original functions.

$$\dot{\gamma} = K \ \sin(2\gamma) \tag{4.57}$$

where K is a positive constant. The equilibrium points are $\gamma = \pi/2$ and $\gamma = 3\pi/2$. The Lyapunov function candidate for this system is

$$V = \frac{1}{2}\cos^2(\gamma) \tag{4.58}$$

Note that, This Lyapunov function is zero at the equilibrium. The Lie derivative of the function becomes

$$\dot{V} = -\dot{\gamma}\cos(\gamma)\sin(\gamma) = -2K \ \cos^2(\gamma)\sin^2(\gamma) \leq 0 \tag{4.59}$$

Therefore, the system is stable at the equilibrium points $\gamma = \pi/2$ and $\gamma = 3\pi/2$.

The other state d depends on the equilibrium of point of γ. If the γ state converges to the set E then $\dot{d} = -v$. Which means the distance between the agent and the target decreases. If the speed function is set to zero before the bifurcation region then the agent stops at a predetermined distance from the target. If not, the distance continues to decrease up to the bifurcation region and therefore, the system becomes unstable at E and stable at L_1 or L_2 due to the approximation angle. Therefore, agent circles around the target.

The above results show that the system will be stable at the targeting behavior and the limit cycle behaviors.

4.3.3 Angular Velocity Controllers

In this section we will propose linear and nonlinear angular velocity controllers.

4.3.3.1 Proportional Angular Velocity Controller

The angular velocity controller can be selected as the simple proportional controller that considers the shortest way of turn such that

$$\dot{\theta} = -\alpha\left[\mathrm{mod}(\gamma, 2\pi) - \pi\right] \tag{4.60}$$

Then the system dynamics can be expressed as

$$\begin{aligned} \dot{d} &= v(d)\ \cos(\gamma) && (4.61a) \\ \dot{\gamma} &= -\alpha\left[\mathrm{mod}(\gamma, 2\pi) - \pi\right] - \frac{1}{d}\,v(d)\ \sin(\gamma) && (4.61b) \end{aligned}$$

The equilibrium of this system occurs at points satisfying $\dot{d} = 0$, $\dot{\gamma} = 0$.

For $\dot{d} = 0$ we have

$$\dot{d} = v(d)\ \cos(\gamma) = 0 \Longrightarrow\ v(d) = 0\ or\ \gamma = \frac{\pi}{2} + k\pi \tag{4.62}$$

for $k = 0, 1$. Similarly, from $\dot{\gamma} = 0$ we have

$$\begin{aligned} \dot{\gamma} &= -\alpha\left[\mathrm{mod}(\gamma, 2\pi) - \pi\right] - \frac{1}{d}v(d)\ \sin(\gamma) = 0 \\ &\Longrightarrow\ v(d) = 0,\ \ \gamma = \pi && (4.63a) \\ &or \\ &\Longrightarrow\ v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{\pi}{2} && (4.63b) \\ &or \\ &\Longrightarrow\ v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{3\pi}{2} && (4.63c) \end{aligned}$$

The common solution of the equations (4.62), (4.63) yields three different equilibrium sets. The first one is

$$E = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\} \tag{4.64}$$

and the second is

$$L_1 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{\pi}{2}\} \tag{4.65}$$

and the last one is

$$L_2 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{3\pi}{2}\} \tag{4.66}$$

which are consistent with the general solutions derived in the previous sections on equilibrium points.

For the stability of these points we will consider the linearization methods. As found in the previous sections necessary conditions for the stability at L_1 is $g'(\pi/2) < 0$ which is satisfied since $g'(\pi/2) = -\alpha < 0$. Similarly the condition for the stability of L_2 is also satisfied $g'(3\pi/2) = -\alpha < 0$. The condition for the stability of E is $g'(\pi) = -\alpha < 0$ is again satisfied.

4.3.3.2 Nonlinear Angular Velocity Controller

The angular velocity controller selected in this section is a trigonometric function in (4.67) that behaves similar to the previous angular velocity controller. This controller again results in the shortest way of turn.

$$\dot{\theta} = \sqrt{2}\alpha\cos(\gamma/2) \tag{4.67}$$

The the system dynamics can be expressed as

$$\dot{d} = v(d)\ \cos(\gamma) \tag{4.68a}$$

$$\dot{\gamma} = \sqrt{2}\alpha\cos(\gamma/2) - \frac{1}{d}\ v(d)\ \sin(\gamma) \tag{4.68b}$$

The equilibrium of this system occurs at points satisfying $\dot{d} = 0$, $\dot{\gamma} = 0$.

For $\dot{d} = 0$ we have

$$\dot{d} = v(d)\ \cos(\gamma) = 0 \Longrightarrow\ v(d) = 0\ or\ \gamma = \frac{\pi}{2} + k\pi \tag{4.69}$$

for $k = 0, 1$. Similarly, from $\dot{\gamma} = 0$ we have

$$\begin{aligned} \dot{\gamma} &= \sqrt{2}\alpha\cos(\gamma/2) - \frac{1}{d}v(d)\ \sin(\gamma) = 0 \\ &\Longrightarrow\ v(d) = 0,\ \ \gamma = \pi && (4.70a) \\ &or \\ &\Longrightarrow\ v(d) = \alpha d,\ \ \gamma = \frac{\pi}{2} && (4.70b) \\ &or \\ &\Longrightarrow\ v(d) = \alpha d,\ \ \gamma = \frac{3\pi}{2} && (4.70c) \end{aligned}$$

The common solution of the equations (4.69), (4.70) yields three different equilibrium sets. The first one is

$$E = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\} \tag{4.71}$$

and the second is

$$L_1 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha d,\ \ \gamma = \frac{\pi}{2}\} \tag{4.72}$$

and the last one is

$$L_2 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha d,\ \ \gamma = \frac{3\pi}{2}\} \tag{4.73}$$

which are appropriate to the general solutions derived in the previous sections.

For the stability of these points we will consider the linearization methods. As found in the previous sections necessary conditions for the stability at L_1 is $g'(\pi/2) < 0$ which is satisfied since $g'(\pi/2) = -\sqrt{2}/2\alpha\sin(\pi/4) = -\alpha/2 < 0$. Similarly the condition for the stability of L_2 is also satisfied $g'(3\pi/2) = -\sqrt{2}/2\alpha\sin(3\pi/4) = -\alpha/2 < 0$. The condition for the stability of E is $g'(\pi) = -\sqrt{2}/2\alpha\sin(\pi/2) = -\alpha < 0$ is again satisfied.

4.3.4 Speed Controllers

The speed of the agent can be controlled by many different methods. Here we will show some simple and complicated controllers.

4.3.4.1 Constant Speed Controller

This controller is the most simple and applicable controller for mobile robots. The speed is taken as constant resulting in

$$\dot{d} = v\ \cos(\gamma) \tag{4.74}$$

We will use this controller with two angular velocity controllers derived in the previous section.

The system dynamics with the proportional angular velocity controller can be expressed as

$$\begin{aligned} \dot{d} &= v\ \cos(\gamma) && (4.75a) \\ \dot{\gamma} &= -\alpha\left[\operatorname{mod}(\gamma, 2\pi) - \pi\right] - \frac{1}{d}\, v\ \sin(\gamma) && (4.75b) \end{aligned}$$

The possible equilibrium sets are

$$E = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\} \tag{4.76}$$

$$L_1 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{\pi}{2}\} \tag{4.77}$$

$$L_2 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{3\pi}{2}\} \tag{4.78}$$

Note that since v is a positive constant the system cannot converge to the $\dot{d} = 0$ state for the set E. However the γ dynamics will converge to $\gamma = \pi$ that means the agent heading is towards the target however does not stop at a predetermined distance. For the circling behaviors the radius of the circles are $d_{eq} = v/(\alpha\pi/2)$.

Applying the results for stability from the linearization section we get

$$g(\pi/2) = -\alpha\pi/2 < 0 \ \mathit{and}\ v'(d_{eq}) = 0 > g(\pi/2) \tag{4.79}$$

and

$$g(3\pi/2) = \alpha\pi/2 > 0 \ \mathit{and}\ v'(d_{eq}) = 0 < g(3\pi/2) \tag{4.80}$$

Then the system is stable at the equilibrium sets L_1 and L_2. The phase potrait of the system is shown in Figure 4.7. As seen from the figure the two equilibrium sets L_1 and L_2 are locally asymptotically stable equilibrium points.

The simulation result for this case is represented in Figure 4.8

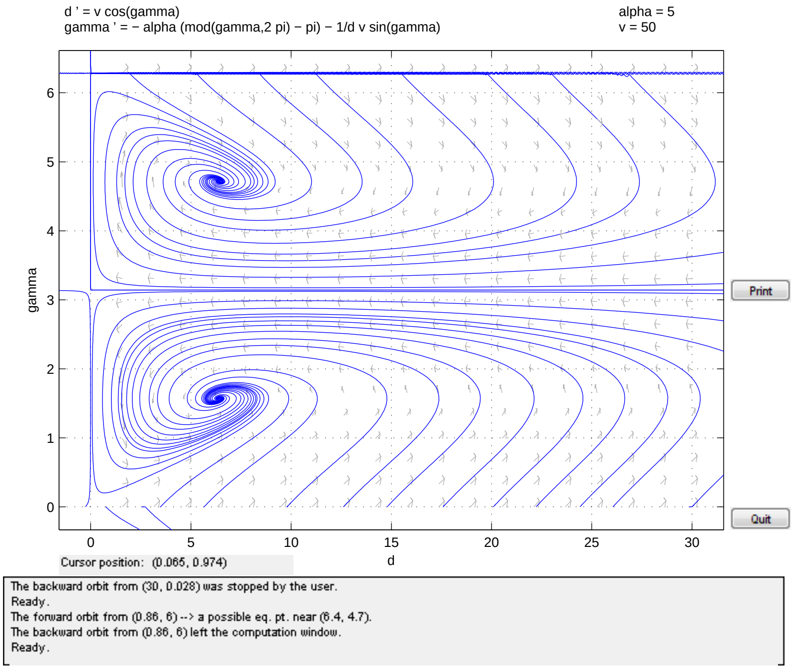
4.3.4.2 Linear Speed Controller - Type I

The next speed controller for mobile robots is the controller that relates the speed linearly with the distance given in the following equation.

$$v(d) = Ad + B \tag{4.81}$$

Where A and B are positive constants. The system dynamics with the proportional angular velocity controller can be expressed as

$$\begin{aligned} \dot{d} &= v(d)\ \cos(\gamma) && (4.82a) \\ \dot{\gamma} &= -\alpha\left[\operatorname{mod}(\gamma, 2\pi) - \pi\right] - \frac{1}{d}\ v(d)\ \sin(\gamma) && (4.82b) \end{aligned}$$



```
d ' = v cos(gamma)
gamma ' = − alpha (mod(gamma,2 pi) − pi) − 1/d v sin(gamma)
```

```
alpha = 5
v = 50
```

```
Cursor position:  (0.065, 0.974)
```

```
The backward orbit from (30, 0.028) was stopped by the user.
Ready.
The forward orbit from (0.86, 6) --> a possible eq. pt. near (6.4, 4.7).
The backward orbit from (0.86, 6) left the computation window.
Ready.
```

Figure 4.7: Phase Portrait for Constant Speed and Proportional Angular Velocity Controllers.

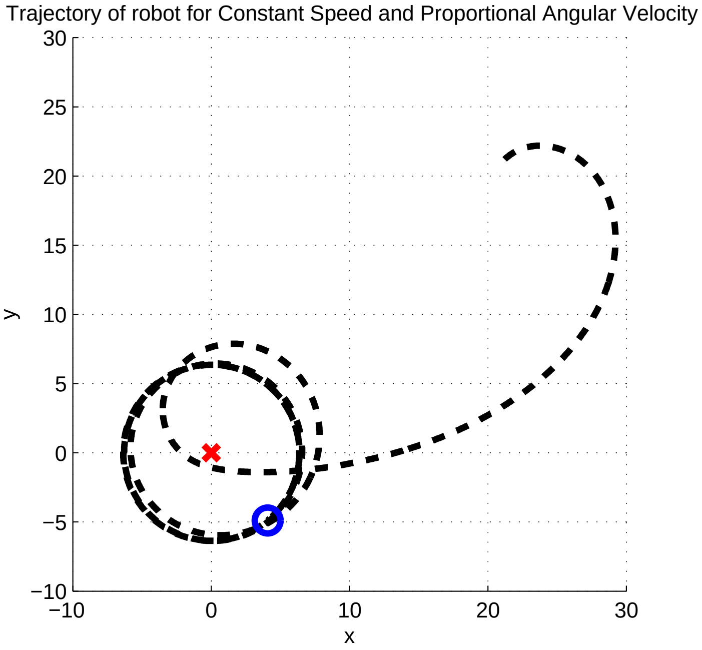


Figure 4.8: Simulation result for Constant Speed and Proportional Angular Velocity Controllers.

The possible equilibrium sets are

$$E = \{d \in R^{+}, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\} \tag{4.83}$$

$$L_1 = \{d \in R^{+}, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{\pi}{2}\} \tag{4.84}$$

$$L_2 = \{d \in R^{+}, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{3\pi}{2}\} \tag{4.85}$$

Note that since $Ad + B > 0$ the system cannot converge to the $\dot{d} = 0$ state for the set E. However the γ dynamics will converge to $\gamma = \pi$ that means the agent heading is towards the target however does not stop at a predetermined distance. For the circling behaviors the radius of the circles are

$$Ad_{eq} + B = d_{eq}\alpha\frac{\pi}{2} \Longrightarrow d_{eq} = \frac{B}{\alpha\frac{\pi}{2} - A} \tag{4.86}$$

Applying the results for stability from the linearization section we get

$$g(\pi/2) = -\alpha\pi/2 < 0 \textit{ and } v'(d_{eq}) = A > g(\pi/2) \tag{4.87}$$

and

$$g(3\pi/2) = \alpha\pi/2 > 0 \textit{ and } v'(d_{eq}) = A < g(3\pi/2) \tag{4.88}$$

Then the system is locally stable at the equilibrium sets L_1 and L_2 if $A < \alpha\pi/2$. The phase portraits of the system for $A < \alpha\pi/2$ and $A > \alpha\pi/2$ are shown in Figures 4.9, 4.10. As seen from the figure the two equilibrium sets L_1 and L_2 are stable equilibrium points.

The simulation result for this case is represented in Figure 4.11

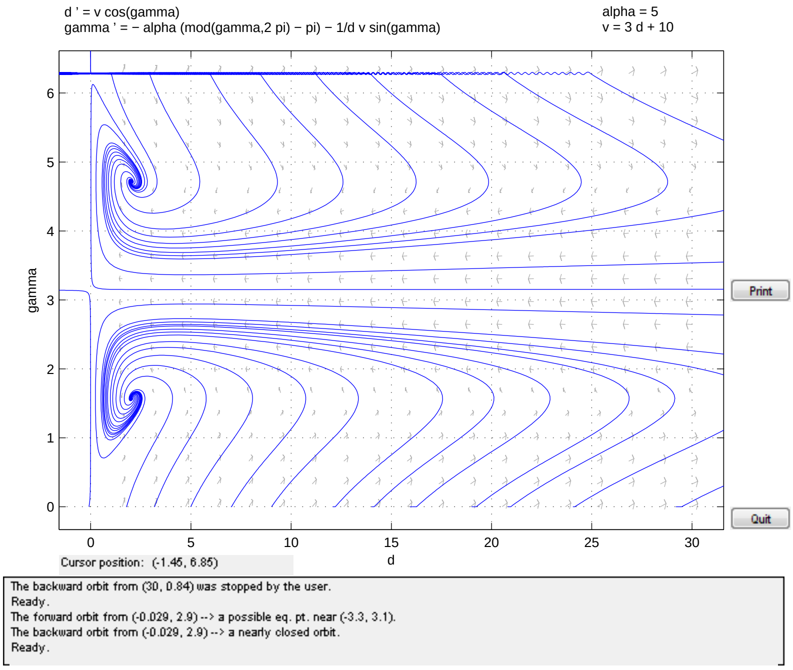
4.3.4.3 Linear Speed Controller - Type II

This second type speed controller relates the speed linearly with the distance as

$$v(d) = Ad - B \tag{4.89}$$

where A and B are positive constants. The system dynamics with the proportional angular velocity controller can be expressed as

$$\begin{aligned} \dot{d} &= v(d)\ \cos(\gamma) && \text{(4.90a)} \\ \dot{\gamma} &= -\alpha\left[\text{mod}(\gamma, 2\pi) - \pi\right] - \frac{1}{d}\ v(d)\ \sin(\gamma) && \text{(4.90b)} \end{aligned}$$



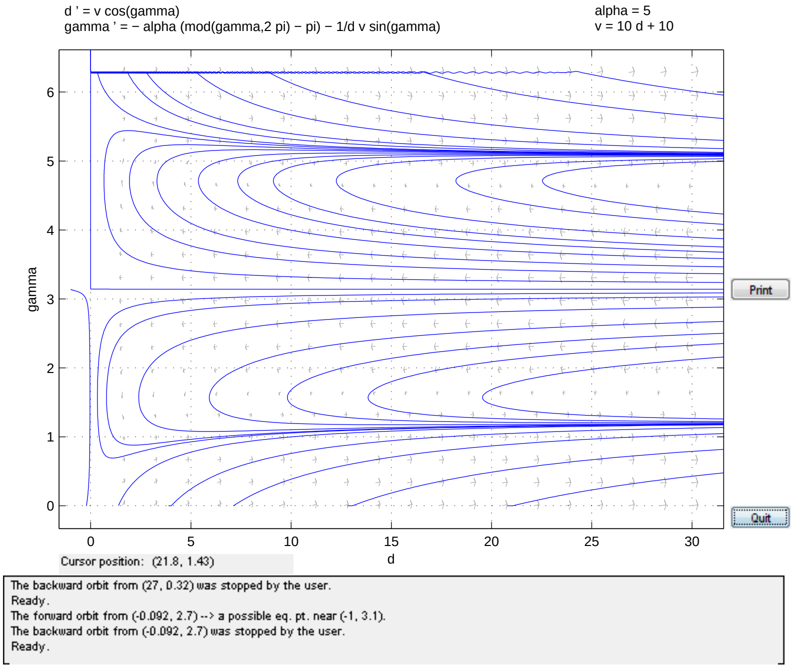
d ' = v cos(gamma)
gamma ' = − alpha (mod(gamma,2 pi) − pi) − 1/d v sin(gamma)

alpha = 5
v = 3 d + 10

Cursor position: (-1.45, 6.85)

```
The backward orbit from (30, 0.84) was stopped by the user.
Ready.
The forward orbit from (-0.029, 2.9) --> a possible eq. pt. near (-3.3, 3.1).
The backward orbit from (-0.029, 2.9) --> a nearly closed orbit.
Ready.
```

Figure 4.9: Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers, $A < \alpha\pi/2$.



d ' = v cos(gamma)
gamma ' = − alpha (mod(gamma,2 pi) − pi) − 1/d v sin(gamma)

alpha = 5
v = 10 d + 10

Cursor position: (21.8, 1.43)

```
The backward orbit from (27, 0.32) was stopped by the user.
Ready.
The forward orbit from (-0.092, 2.7) --> a possible eq. pt. near (-1, 3.1).
The backward orbit from (-0.092, 2.7) was stopped by the user.
Ready.
```

Figure 4.10: Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers, $A > \alpha\pi/2$.

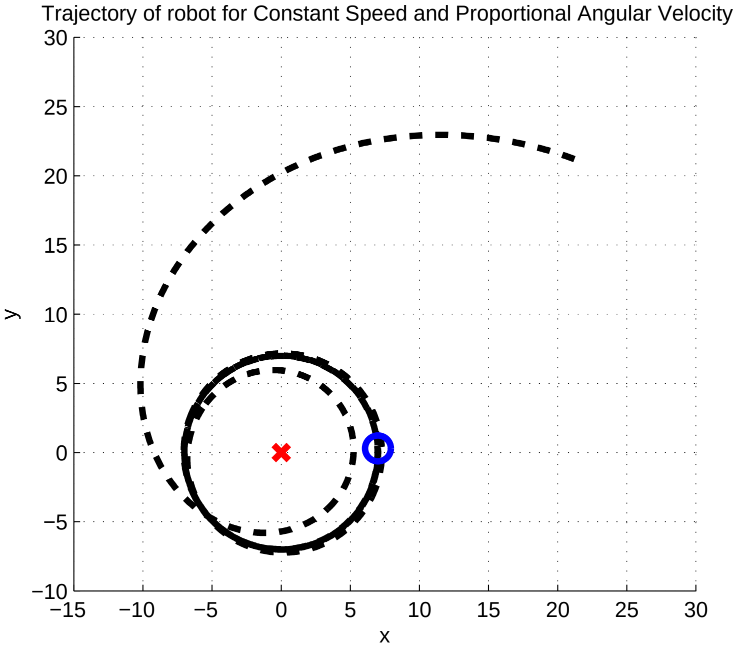


Figure 4.11: Simulation result for Linear Speed and Proportional Angular Velocity Controllers, $A < \alpha\pi/2$.

The possible equilibrium sets are

$$E = \{d \in R^{+}, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\} \tag{4.91}$$

$$L_1 = \{d \in R^{+}, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d, \ \ \gamma = \frac{\pi}{2}\} \tag{4.92}$$

$$L_2 = \{d \in R^{+}, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d, \ \ \gamma = \frac{3\pi}{2}\} \tag{4.93}$$

For the equilibrium set E,

$$Ad_{eq} - B = 0 \Longrightarrow d_{eq} = B/A$$

. The agent converges to targeting behavior and stops at the distance $d_{eq} = B/A$. For the circling behaviors the radius of the circles are

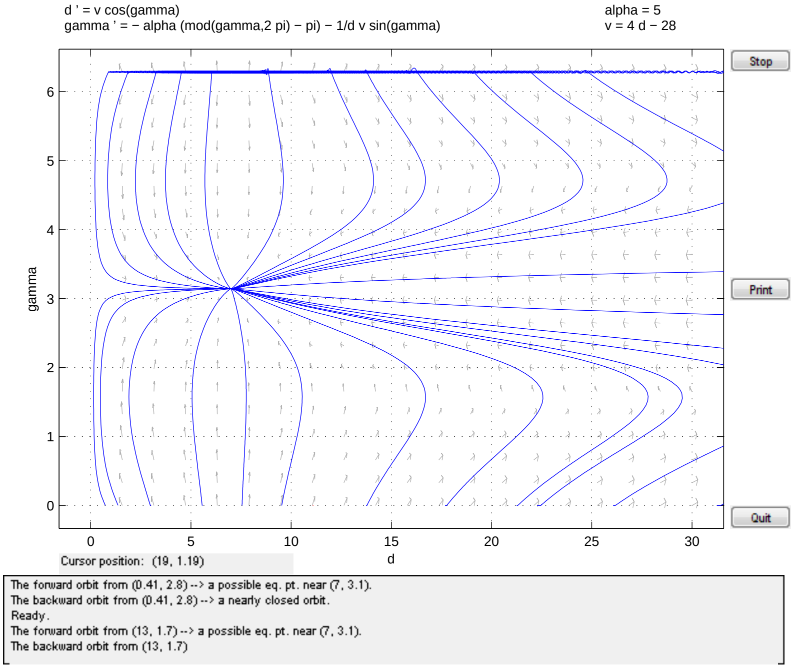
$$Ad_{eq} - B = d_{eq}\alpha\frac{\pi}{2} \Longrightarrow d_{eq} = \frac{B}{A - \alpha\frac{\pi}{2}} \tag{4.94}$$

Applying the results for stability from the linearization section we get

$$g(\pi/2) = -\alpha\pi/2 < 0 \ and \ v'(d_{eq}) = A > g(\pi/2) \tag{4.95}$$

and

$$g(3\pi/2) = \alpha\pi/2 > 0 \ and \ v'(d_{eq}) = A < g(3\pi/2) \tag{4.96}$$


d ' = v cos(gamma)
gamma ' = − alpha (mod(gamma,2 pi) − pi) − 1/d v sin(gamma)

alpha = 5
v = 4 d − 28

Cursor position: (19, 1.19)

```
The forward orbit from (0.41, 2.8) --> a possible eq. pt. near (7, 3.1).
The backward orbit from (0.41, 2.8) --> a nearly closed orbit.
Ready.
The forward orbit from (13, 1.7) --> a possible eq. pt. near (7, 3.1).
The backward orbit from (13, 1.7)
```

Figure 4.12: Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers.

However, note that for the circling radius we found $A - \alpha\frac{\pi}{2} > 0$ which contradicts with the above result. Then the system is not stable at the equilibrium sets L_1 and L_2. The phase portrait of the system is shown in Figure 4.12. As seen from the figure the two equilibrium sets L_1 and L_2 are not stable equilibrium points. The system converges to the equilibrium E.

The simulation result for this case is represented in Figure 4.13

4.3.4.4 Quadratic Speed Controller

This speed controller in this section is

$$v(d) = (d - C)^2 \tag{4.97}$$

Where C is a positive constants. A sample plot of this speed function is in Figure 4.14

The system dynamics with the proportional angular velocity controller can be expressed as

$$\begin{aligned} \dot{d} &= v(d)\ \cos(\gamma) && \text{(4.98a)} \\ \dot{\gamma} &= -\alpha\left[\mathrm{mod}(\gamma, 2\pi) - \pi\right] - \frac{1}{d}\ v(d)\ \sin(\gamma) && \text{(4.98b)} \end{aligned}$$

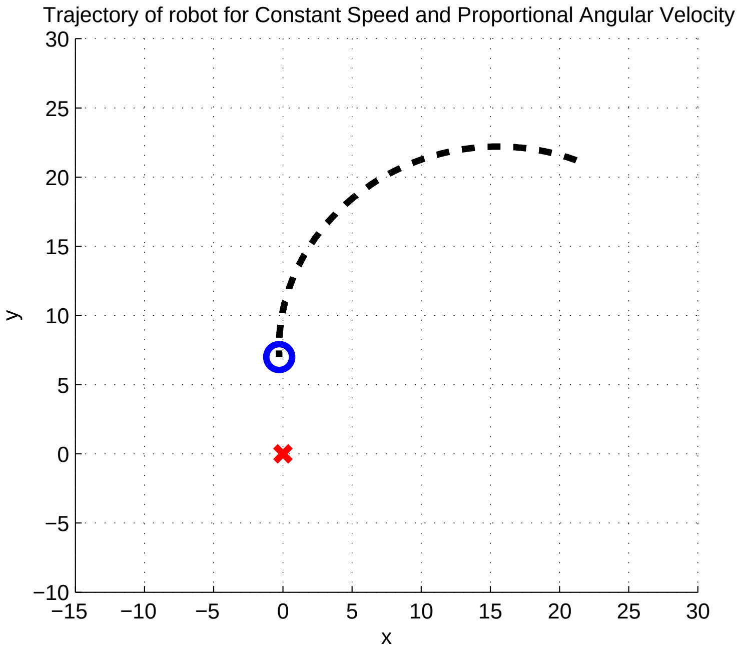


Figure 4.13: Simulation result for Linear Speed and Proportional Angular Velocity Controllers.

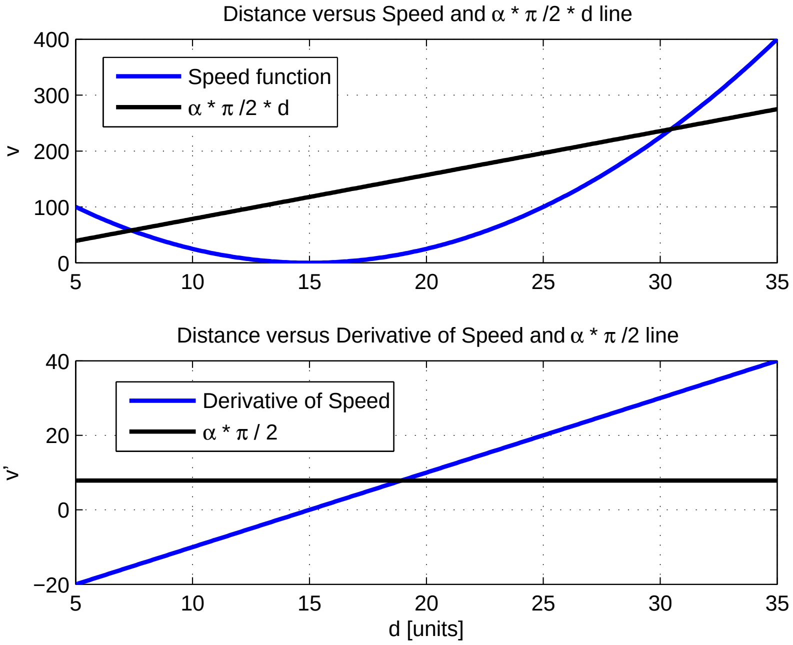


Figure 4.14: Quadratic Speed Function and its derivative with respect to distance.

The possible equilibrium sets are

$$E = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = 0, \gamma = \pi\} \tag{4.99}$$

$$L_1 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{\pi}{2}\} \tag{4.100}$$

$$L_2 = \{d \in R^+, 0 < \gamma < 2\pi | v(d) = \alpha\frac{\pi}{2}d,\ \ \gamma = \frac{3\pi}{2}\} \tag{4.101}$$

For the equilibrium set E,

$$(d - C)^2 = 0 \implies d_{eq} = C$$

. The agent converges to targeting behavior and stops at the distance $d_{eq} = C$. For the circling behaviors the radius of the circles are

$$(d - C)^2 = d_{eq}\alpha\frac{\pi}{2} \implies d_{eq1,2} = \frac{2C + \alpha\frac{\pi}{2} - \sqrt{(2C + \alpha\frac{\pi}{2})^2 - 4C^2}}{2} \tag{4.102}$$

Applying the results for stability from the linearization section the equations to be satisfied are

$$g(\pi/2) = -\alpha\pi/2 < 0 \ and\ v'(d_{eq1}) > g(\pi/2) \tag{4.103}$$

and

$$g(3\pi/2) = \alpha\pi/2 > 0 \ and\ v'(d_{eq1}) < g(3\pi/2) \tag{4.104}$$

Then the equilibrium for circling is stable at lower root d_{eq1}. And the equilibrium set E at $d_{eq} = C$ is also stable. The phase portrait of the system is shown in Figure 4.15. As seen from the figure the two equilibrium sets L_1 and L_2 at d_{eq1} are stable equilibrium points. The system converges to the equilibrium E at $d_{eq} = C$.

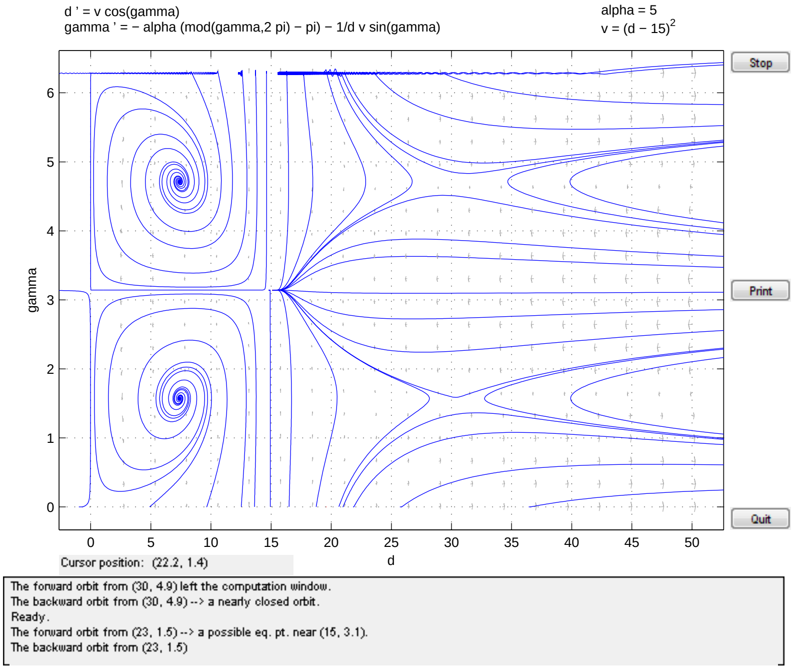
The simulation result for this case is represented in Figure 4.16 and 4.17

4.3.4.5 Stability via Lyapunov Functions

In this section, we will examine a Lyapunov function candidate that supports the above results. Before we present the Lyapunov function we should present the second derivatives of the system variables to be utilized in the time derivative of the Lyapunov function. Consider the system dynamics in (4.61). Under the assumption that the linear speed v is a function of distance the second derivatives with respect to time are

$$\ddot{d} = \dot{v}\cos(\gamma) - v\dot{\gamma}\sin(\gamma) \tag{4.105a}$$

$$\ddot{\gamma} = -\alpha\dot{\gamma} - \frac{\dot{v}}{d}\sin(\gamma) + \frac{\dot{d}}{d^2}v\sin(\gamma) - \frac{1}{d}v\dot{\gamma}\cos(\gamma) \tag{4.105b}$$



```
Cursor position:  (22.2, 1.4)
```

```
The forward orbit from (30, 4.9) left the computation window.
The backward orbit from (30, 4.9) --> a nearly closed orbit.
Ready.
The forward orbit from (23, 1.5) --> a possible eq. pt. near (15, 3.1).
The backward orbit from (23, 1.5)
```

Figure 4.15: Phase Portrait for Linear Speed and Proportional Angular Velocity Controllers.

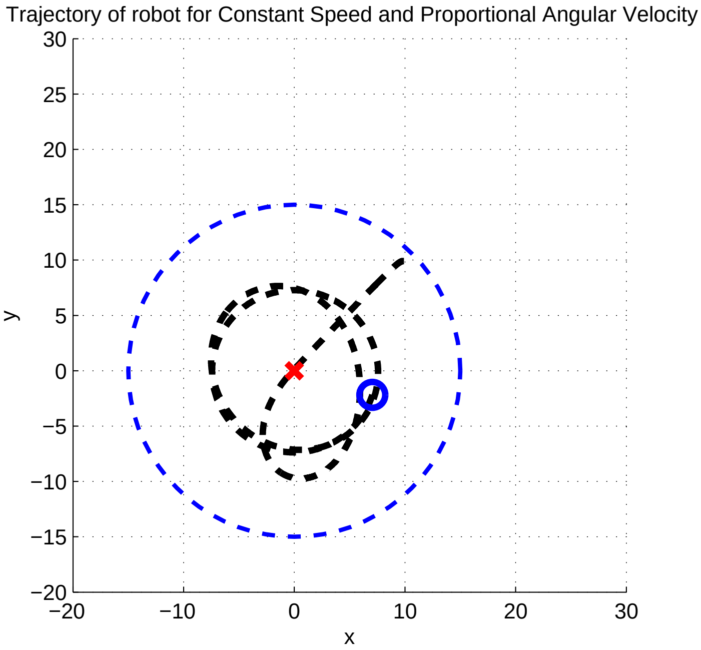


Figure 4.16: Simulation result for Quadratic Speed and Proportional Angular Velocity Controllers.

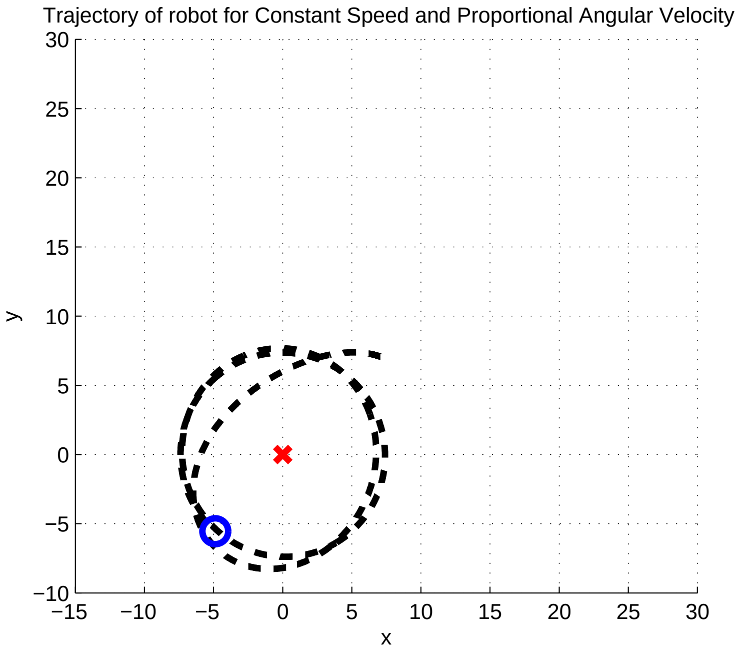


Figure 4.17: Simulation result for Quadratic Speed and Proportional Angular Velocity Controllers.

The Lyapunov function candidate is

$$V = \frac{1}{2}\dot{d}^2 + \frac{1}{2}(d\dot{\gamma})^2 \tag{4.106}$$

where $D = \{\dot{d}, \dot{\theta}, \dot{\psi} \in \mathbb{R}\}$. $V(\dot{d}, \dot{\theta}, \dot{\psi})$ is positive definite in D. Then the Lie derivative of the Lyapunov function becomes

$$\dot{V} = \dot{d}\ddot{d} + d\dot{\gamma}(\dot{d}\dot{\gamma} + d\ddot{\gamma}) \tag{4.107}$$

which substituting the second derivatives in (4.21) becomes

$$\dot{V} = \dot{d}\left[\dot{v}\cos(\gamma) - v\dot{\gamma}\sin(\gamma)\right] + d\dot{\gamma}\left[\dot{d}\dot{\gamma} + d\left(-\alpha\dot{\gamma} - \frac{\dot{v}}{d}\sin(\gamma) + \frac{\dot{d}}{d^2}v\sin(\gamma) - \frac{1}{d}v\dot{\gamma}\cos(\gamma)\right)\right] \tag{4.108}$$

Then, rearranging

$$\begin{aligned}
\dot{V} &= \dot{d}\left[\dot{v}\cos(\gamma) - v\dot{\gamma}\sin(\gamma)\right] + d\dot{\gamma}\left[\dot{d}\dot{\gamma} - \alpha d\dot{\gamma} - \dot{v}\sin(\gamma) + \frac{\dot{d}}{d}v\sin(\gamma) - v\dot{\gamma}\cos(\gamma)\right] \\
&= \dot{v}\dot{d}\cos(\gamma) - v\dot{d}\dot{\gamma}\sin(\gamma) + d\dot{\gamma}\left[\cancel{\dot{d}\dot{\gamma}} - \alpha d\dot{\gamma} - \dot{v}\sin(\gamma) + \frac{\dot{d}}{d}v\sin(\gamma) - \cancel{\dot{d}\dot{\gamma}}\right] \\
&= \dot{v}\dot{d}\cos(\gamma) - \cancel{v\dot{d}\dot{\gamma}\sin(\gamma)} - \alpha d^2\dot{\gamma}^2 - \dot{v}d\dot{\gamma}\sin(\gamma) + \cancel{v\dot{d}\dot{\gamma}\sin(\gamma)} \\
&= \dot{v}\dot{d}\cos(\gamma) - \alpha d^2\dot{\gamma}^2 - \dot{v}d\dot{\gamma}\sin(\gamma) \\
&= -\alpha d^2\dot{\gamma}^2 - \dot{v}\left(d\dot{\gamma}\sin(\gamma) - \dot{d}\cos(\gamma)\right)
\end{aligned} \tag{4.109}$$

substituting the variables in (4.61)

$$\begin{aligned}\dot{V} &= -\alpha d^2\dot{\gamma}^2 - \dot{v}\left(d\dot{\gamma}\sin(\gamma) - \dot{d}\cos(\gamma)\right)\\
&= -\alpha d^2\left[-\alpha\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] - \frac{1}{d}\,v\ \sin(\gamma)\right]^2\\
&\quad -\dot{v}\left\{d\sin(\gamma)\left[-\alpha\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] - \frac{1}{d}\,v\ \sin(\gamma)\right] - v\cos^2(\gamma)\right\}\\
&= -\alpha d^2\left[-\alpha\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] - \frac{1}{d}\,v\ \sin(\gamma)\right]^2\\
&\quad -\dot{v}\left\{-\alpha d\sin(\gamma)\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] -\ v\ \sin^2(\gamma) - v\cos^2(\gamma)\right\}\\
&= -\alpha d^2\left[-\alpha\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] - \frac{1}{d}\,v\ \sin(\gamma)\right]^2\\
&\quad +\dot{v}\left\{\alpha d\sin(\gamma)\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] +\ v\right\} \qquad (4.110)\end{aligned}$$

using the relation $\dot{v} = v'\dot{d}$ where v' is the derivative of speed with respect to distance d.

$$\begin{aligned}\dot{V} &= -\alpha d^2\left[-\alpha\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] - \frac{1}{d}\,v\ \sin(\gamma)\right]^2\\
&\quad +v'\dot{d}\left\{\alpha d\sin(\gamma)\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] +\ v\right\}\\
&= -\alpha d^2\left[-\alpha\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] - \frac{1}{d}\,v\ \sin(\gamma)\right]^2\\
&\quad +v'v\cos(\gamma)\left\{\alpha d\sin(\gamma)\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] +\ v\right\} \qquad (4.111)\end{aligned}$$

The functions of speed satisfying the Lie derivative in (4.111) to be less than or equal to zero, around the equilibrium sets E will result in a stable targeting behavior and the ones satisfying same condition around set L will result in a circling behavior.

Lyapunov Analysis for Constant Speed Controller

The speed function is a positive constant for all d. Then the Lie derivative becomes

$$\dot{V} = -\alpha d^2\left[-\alpha\left[\mathrm{mod}(\gamma,2\pi)-\pi\right] - \frac{1}{d}\,v\ \sin(\gamma)\right]^2 \leq 0$$

Since Lie derivative is less than or equal to zero then the equilibriums L_1 and L_2 are stable equilibrium points. In Figure 4.18 the plot of Lie derivative with respect to d and γ is shown. The cross sections of Lie derivative of Lyapunov function at $\gamma = 90^o$ and $d = d_{eq}$ are shown in the figures 4.19 and 4.20

Lyapunov Analysis for Linear Speed Controller - Type I

The speed function is

$$v(d) = Ad + B \qquad (4.112)$$

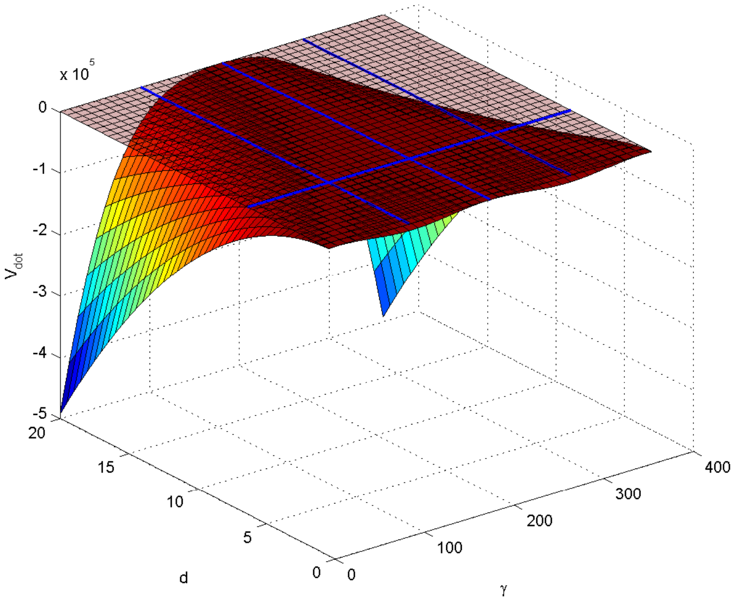


Figure 4.18: Lie Derivative for Speed function v : *constant*

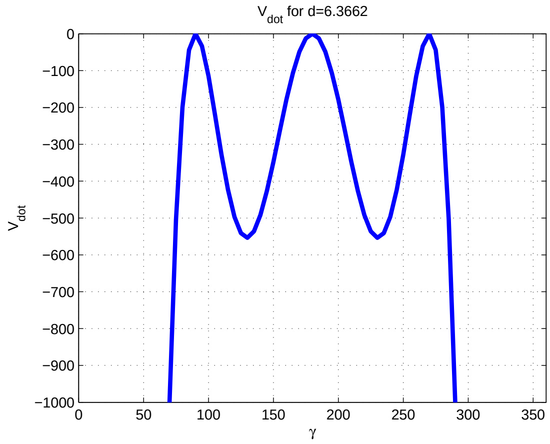


Figure 4.19: Lie Derivative at $d = d_{eq}$ for Speed function v : *constant*

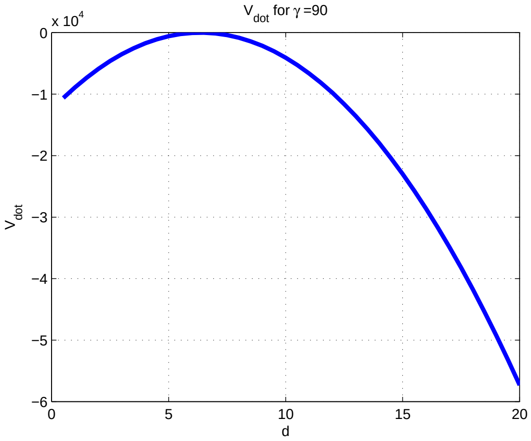


Figure 4.20: Lie Derivative at $\gamma = 90^o$ for Speed function v : *constant*

In Figure 4.21 the plot of Lie derivative with respect to d and γ is shown.

The cross sections of Lie derivative of Lyapunov function at $\gamma = 90^o$ and $d = d_{eq}$ are shown in the figures 4.22 and 4.23. As seen from the figures, the Lie derivatives are negative semi-definite. Therefore, the equilibriums are stable nodes.

Lyapunov Analysis for Linear Speed Controller - Type II

The speed function is

$$v(d) = Ad - B \tag{4.113}$$

In Figure 4.24 the plot of Lie derivative with respect to d and γ is shown.

The cross sections of Lie derivative of Lyapunov function at $\gamma = 90^o$ and $d = d_{eq}$ are shown in the figures 4.25 and 4.26. The figures shows that the Lie derivatives are negative semi-definite. Therefore, the equilibriums are stable nodes.

Lyapunov Analysis for Quadratic Speed Controller

The speed function is

$$v(d) = (d - C)^2 \tag{4.114}$$

In Figure 4.27 the plot of Lie derivative with respect to d and γ is shown.

The cross sections of Lie derivative of Lyapunov function at $\gamma = 90^o$ and $d = d_{eq}$ are shown in the figures 4.28 and 4.29. As seen from the figures, the Lie derivatives are negative semi-

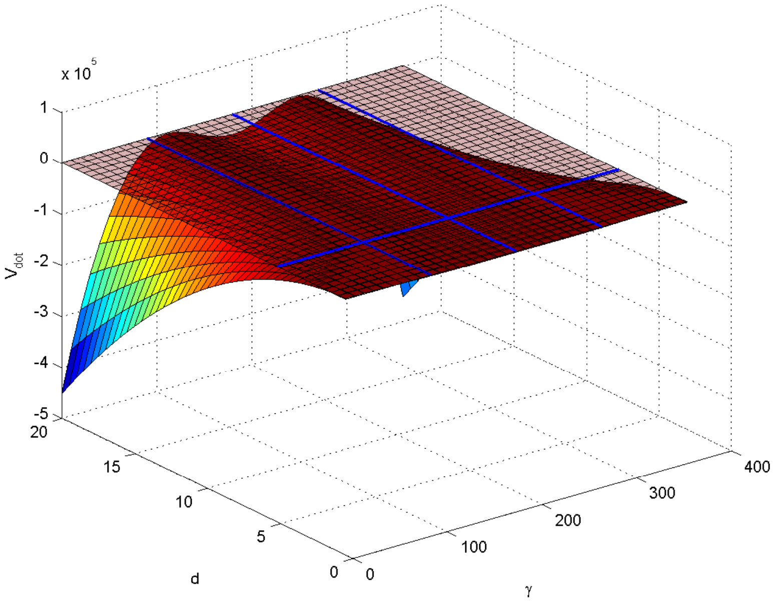


Figure 4.21: Lie Derivative for Speed function $v(d) = Ad + B$

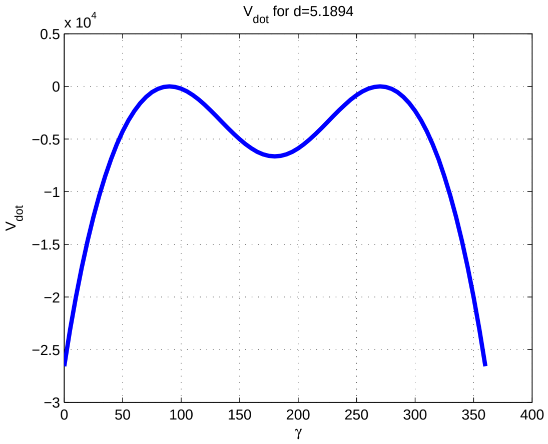


Figure 4.22: Lie Derivative at $d = d_{eq}$ for Speed function $v(d) = Ad + B$

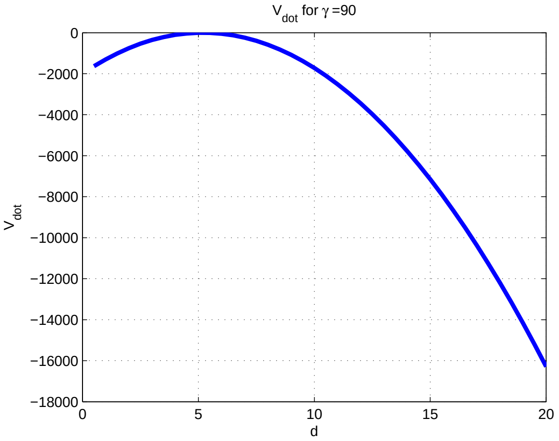


Figure 4.23: Lie Derivative at $\gamma = 90^o$ for Speed function $v(d) = Ad + B$

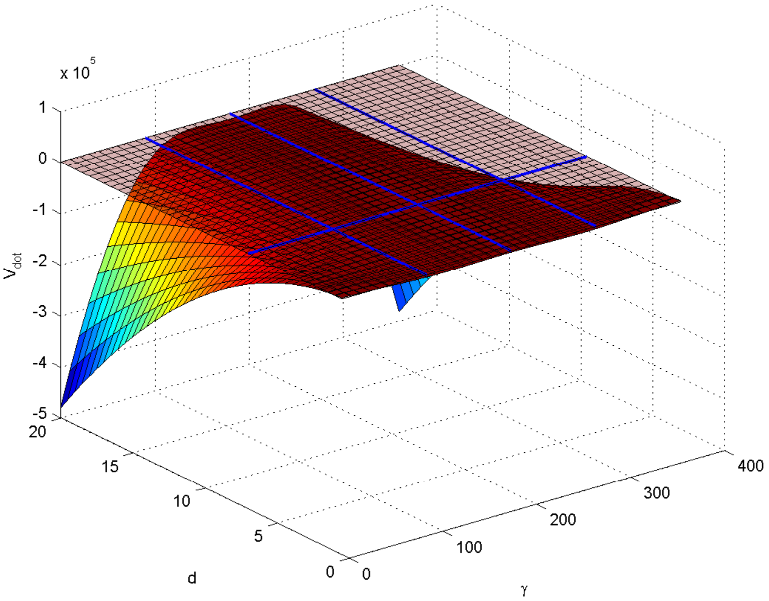


Figure 4.24: Lie Derivative for Speed function $v(d) = Ad - B$

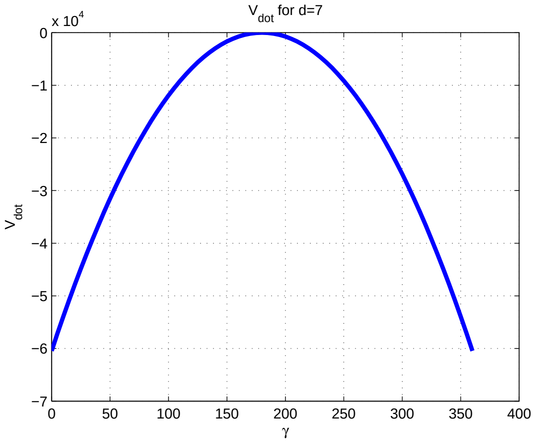


Figure 4.25: Lie Derivative at $d = d_{eq}$ for Speed function $v(d) = Ad - B$

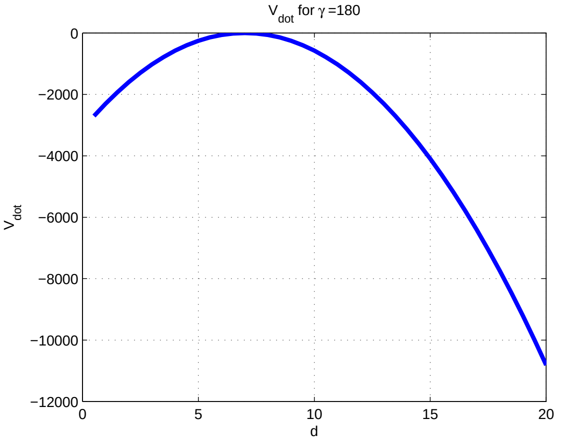


Figure 4.26: Lie Derivative at $\gamma = 90^o$ for Speed function $v(d) = Ad - B$

definite. Therefore, the equilibriums are stable nodes.

4.3.5 Simulation Results

In this section we present simulation results confirming the findings in the previous sections. We examine a robot tracking and circling around static and dynamic targets in the following sub-sections. Note that although we designed the controller assuming the target is stationary we also present the performance of the controller for a dynamic target.

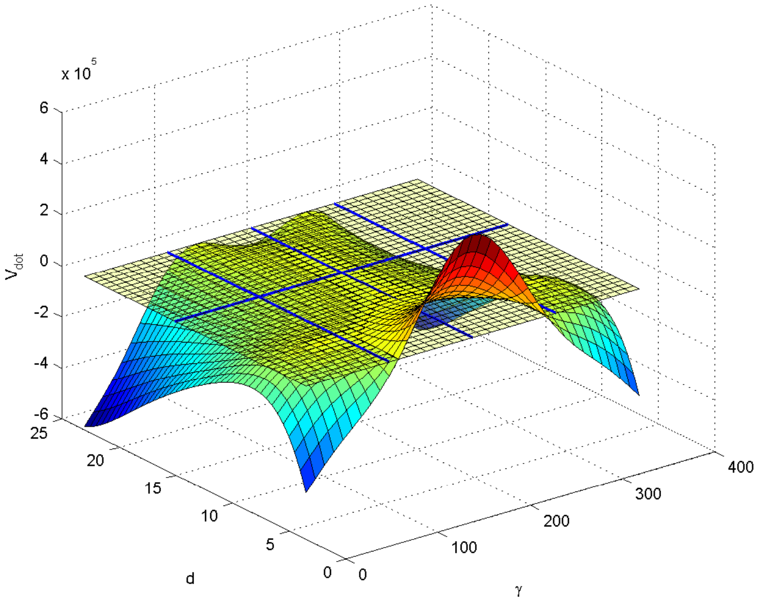


Figure 4.27: Lie Derivative for Speed function $v(d) = (d - C)^2$

4.3.5.1 Target Tracking

The first simulation is performed for target tracking. The controller is designed to force the robot to keep a specified distance to the target. Using the results from the previous section the controller just adjusts the speed of the robot such that the robot traces the target at a specified distance without circling around it. In other words, the system converges to the equilibrium E in (4.12). The sufficient condition for the convergence to the equilibrium E is found to be $v < \alpha d$. Therefore, for the first simulation we will utilize a very simple function for the speed that is $v = \alpha d - \beta < \alpha d$ where β is a constant selected as $\beta = 60$.

In Figure 4.30 the trajectories of the robot for several initial positions and orientations are presented. The small circles are the initial positions and the triangles are the final positions of the robot. In Figure 4.31 the distances between the robot and the target is plotted for the same simulation. As seen from the figures each trajectory reaches to the same distance from the target, which is represented with a cross sign, and stops at that distance. For better visualization a circle with radius equal to the desired distance is drawn around the target. The angular velocity controller coefficient is chosen as $\alpha = 3$. Therefore, the robot stops at $v = \alpha d - \beta = 0 \Rightarrow d = \beta / \alpha = 20[units]$. Note that the angle of approach depends on the initial

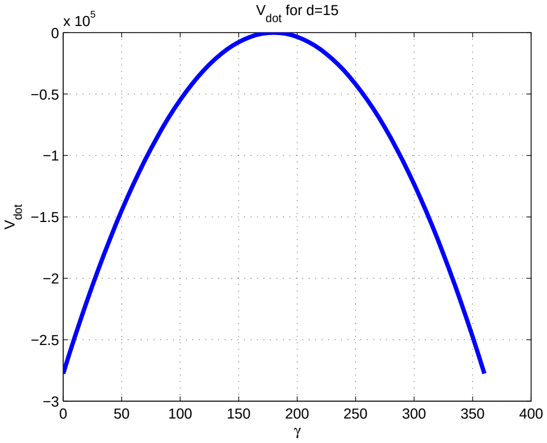


Figure 4.28: Lie Derivative at $d = d_{eq}$ for Speed function $v(d) = (d - C)^2$

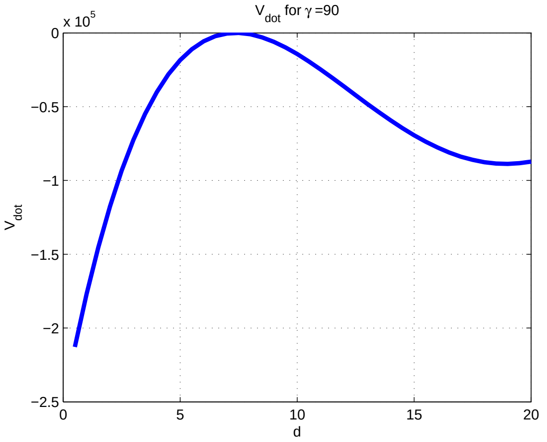


Figure 4.29: Lie Derivative at $\gamma = 90^o$ for Speed function $v(d) = (d - C)^2$

position and orientation of the robot. In order to achieve different robot-to-target distance different combination of α and β parameters can be used.

Trajectories and distance between robot and target of the robot for different initial orientations are represented in 4.32 and 4.33.

Next, we examine the tracking performance of the controller for a dynamic target which has the motion function of the form

$$Z_{target} = [20; 20] + [6t; 3t - 8\sin(t)]; \tag{4.115}$$

The path of the target is presented in Figure 4.34. The cross and diamond stand for the start and final positions of the target, respectively.

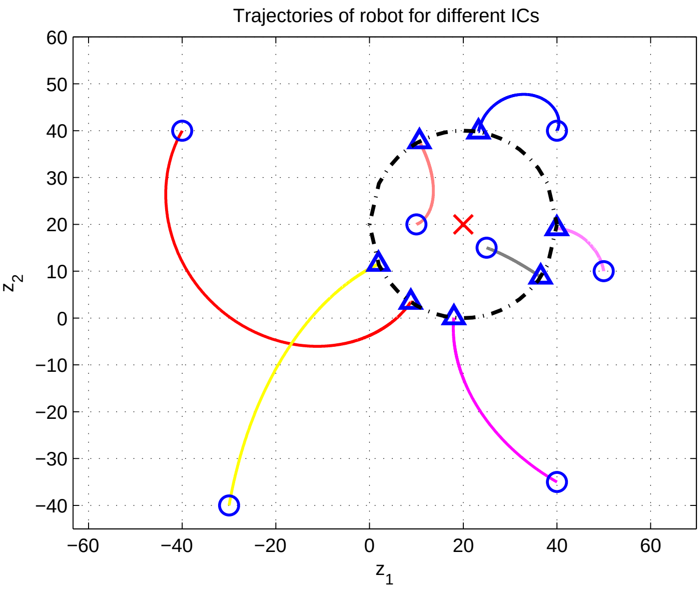


Figure 4.30: Trajectories of the robot for different initial positions and orientations.

The systems response, i.e., path of the robot, for this moving target is presented in Figure 4.35 where the path of the target is also plotted as the dashed curve. The distance between the target and the robot is plotted in Figure 4.36. For this simulation $\alpha = 6$ and $\beta = 30$ are selected and therefore, the distance at which the robot follows the target is $d = \beta/\alpha = 5[units]$. As seen from the figures the robot achieves to trace the target at a distance around $5[units]$, i.e. there is small error. Note that it does not converge to 5 since the controller is designed assuming zero target speed.

4.3.5.2 Circling Around the Target

In this part of the simulation we change the speed controller so that we have circling behavior of the robot around the target. Again we will use the simplest speed controller for demonstration. We set the speed to a constant so that the set of equilibrium points E is attractive for the robot dynamics when the robot is far from the target whereas as it gets closer to the target the set E becomes repulsive and the limit cycle L becomes attractive. Therefore, switching of the behavior occurs at $v = \alpha d$. During $v < \alpha d$ the robot dynamics are converging towards the equilibrium set E and d gets smaller. When d is small enough to satisfy $v > \alpha d$ then the dynamics start to converge to the limit cycle which is the circling of the robot around the

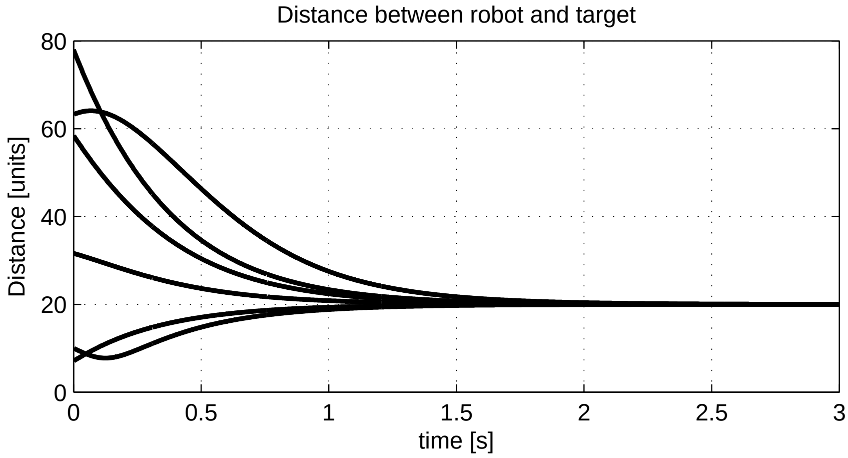


Figure 4.31: Distance between robot and target for different initial positions and orientations.

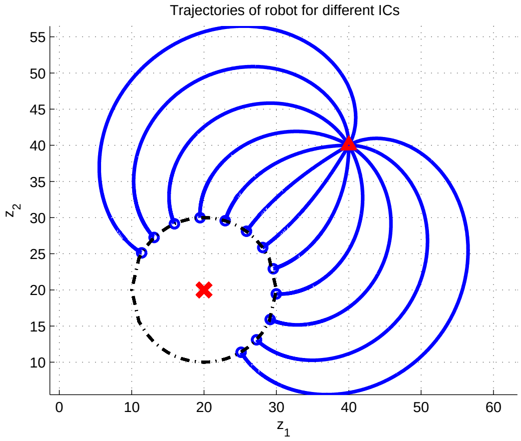


Figure 4.32: Trajectories of the robot for different initial orientations.

target with the radius $d = v/(\alpha\pi/2)$.

In Figures 4.37 and 4.38 the trajectories of the robot and the distance between the robot and the target are presented for different initial positions and orientations. The simulation parameters are selected as $\alpha = 4$, $v = 40\pi$ $[units/s]$. All of the trajectories converge to the circling behavior with the radius $d = v/(\alpha\pi/2) = 40\pi/4/(\pi/2) = 20$ $[units]$. As can be seen from Figure 4.37 some of the trajectories circle around the target in CW direction while some are rotating in CCW direction. The direction of the rotation depends on the angle of approach towards the target and is not aimed to be controlled in the scope of this study.

Next, the dynamic target presented in the previous section is again utilized to observe the circling behavior of the robot. The controller parameters are selected to obtain a circling

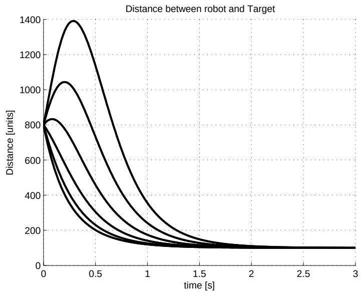


Figure 4.33: Distance between robot and target for different initial orientations.

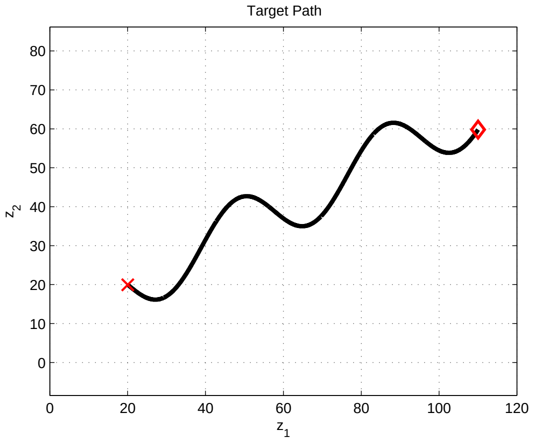


Figure 4.34: Path of the Target.

radius of 10 [*units*] ($\alpha = 5$, $v = 25\pi$). The robot trajectory and the distance of the robot from the target are presented in the Figures 4.39 and 4.40, respectively. As can be seen from the results the robot achieves the objective of circling around the target in motion (the dashed curve is the path of the target) with errors which are at most 5 [units]. This error is due to the fact that the controller was designed based on the assumption of a static target. Extending of the analytical study for dynamic targets is outside of the scope of this study.

4.4 Switching Gradient Method for Circling Around a Target

In this section, a different strategy for circling around a target is presented. For this strategy we will develop the system response for the case where the gradient vector directs from the target towards the robot or in the same direction of $\vec{d}$. Such cases usually occur when the robot

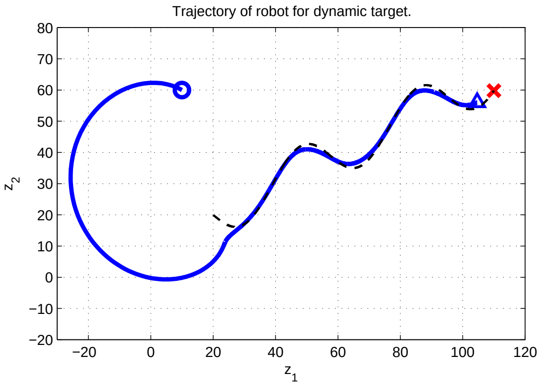


Figure 4.35: Robot trajectory tracking a dynamic target.

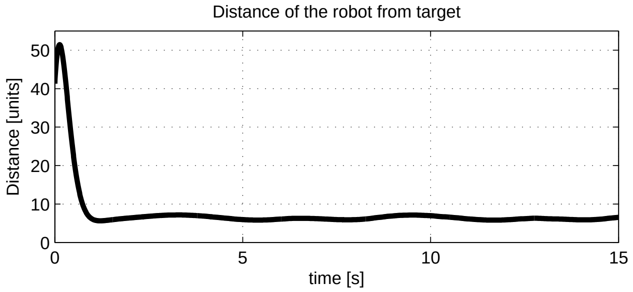


Figure 4.36: Distance of the robot from the dynamic target.

is close to the target in order to avoid collisions. Therefore, the angle of the gradient vector is

$$\gamma = angle(\nabla G) = \psi \tag{4.116}$$

Then the system dynamics becomes

$$\begin{aligned} \dot{d} &= v\cos(\theta - \psi) && (4.117a) \\ \dot{\psi} &= \frac{1}{d} v\sin(\theta - \psi) && (4.117b) \\ \dot{\theta} &= -\alpha[\text{mod}(\theta - \psi + \pi, 2\pi) - \pi] && (4.117c) \end{aligned}$$

The equilibrium of this system is at $\dot{d} = 0$, $\dot{\psi} = 0$, and $\dot{\theta} = 0$. Then for $\dot{d} = 0$ we have a similar solution of (4.9)

$$v\ \cos(\theta - \psi) = 0 \Longrightarrow\ v = 0,\ \ \theta - \psi = \frac{\pi}{2} + k\pi\ \ for\ \ k = 0, \mp1, \mp2... \tag{4.118}$$

Similarly, for $\dot{\psi} = 0$

$$\frac{1}{d} v\ \sin(\theta - \psi) = 0 \Longrightarrow\ v = 0,\ \ \theta - \psi = k\pi\ \ for\ \ k = 0, \mp1, \mp2... \tag{4.119}$$

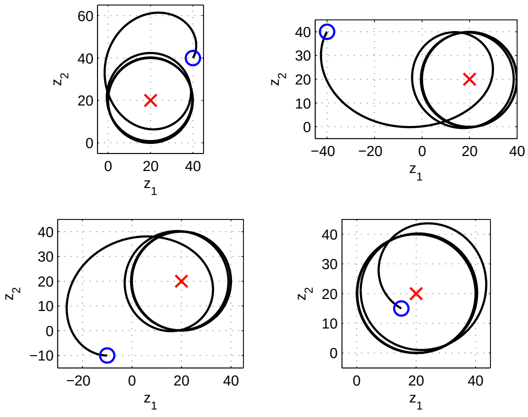


Figure 4.37: Robot trajectories circling around a static target at [20, 20].

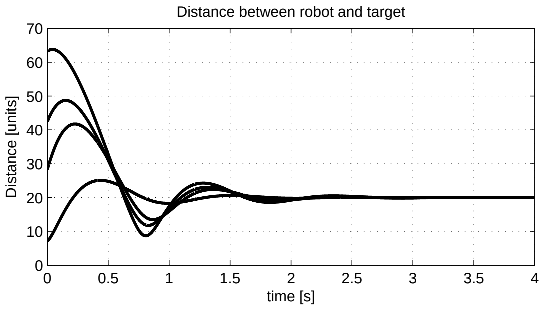


Figure 4.38: Distance between robot and static target at [20, 20] for circling around the target strategy.

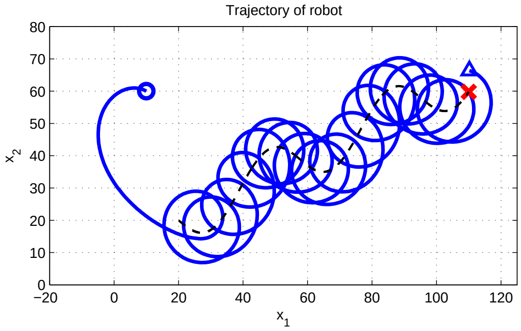


Figure 4.39: Robot trajectory circling around a dynamic target.

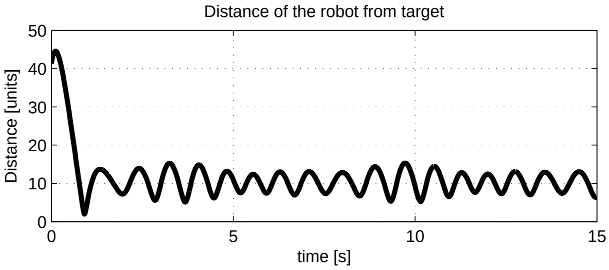


Figure 4.40: Distance between target and robot circling around a dynamic target.

And, for $\dot{\theta} = 0$ we obtain

$$-\,\alpha\,[\mathrm{mod}(\theta - \psi + \pi, 2\pi) - \pi] = 0 \Longrightarrow \theta - \psi = 2k\pi \;\; for \;\; k = 0, \mp 1, \mp 2... \tag{4.120}$$

The common solution of the equations (4.118), (4.119), (4.120) is the set E

$$E = \{v \in R^{+}, 0 < \theta, \psi < 2\pi | v = 0, \theta - \psi = 2k\pi\} \;\; for \;\; k = 0, \mp 1, \mp 2... \tag{4.121}$$

We will check for the existence of the limit cycle similar to the previous section. The possible limit cycle occurs at $\dot{d} = 0$ and $\dot{\theta} - \dot{\psi} = 0$, and $v \neq 0$. Solving the system dynamics under the above constraints we obtain the solutions as

$$\dot{d} = v\;\cos(\theta - \psi) = 0 \Longrightarrow \;\theta - \psi = \frac{\pi}{2} + k\pi \;\;\; for\, k = 0, \mp 1, \mp 2... \tag{4.122}$$

and

$$\dot{\theta} - \dot{\psi} = -\alpha\,[\mathrm{mod}(\theta - \psi + \pi, 2\pi) - \pi] - \frac{1}{d} v\;\sin(\theta - \psi) = 0 \tag{4.123}$$

There are two independent solutions of $\dot{d} = 0$ which are $\theta - \psi = \pi/2$ and $\theta - \psi = 3\pi/2$. For $\theta - \psi = \pi/2$, (4.123) gives

$$-\alpha\,[\mathrm{mod}(\pi/2 + \pi, 2\pi) - \pi] - \frac{1}{d} v\;\sin(\pi/2) = 0 \tag{4.124a}$$

$$\Longrightarrow -\alpha\frac{\pi}{2} = \frac{1}{d} v \tag{4.124b}$$

$$\Longrightarrow d = -\frac{v}{\alpha\pi/2}. \tag{4.124c}$$

which is not a valid solution since d is a nonnegative physical measurement. Similarly for the second solution is $\theta - \psi = 3\pi/2$, (4.123) becomes

$$-\alpha\,[\mathrm{mod}(3\pi/2 + \pi, 2\pi) - \pi] - \frac{1}{d} v\;\sin(3\pi/2) = 0 \tag{4.125a}$$

$$\Longrightarrow \alpha\frac{\pi}{2} = -\frac{1}{d} v \tag{4.125b}$$

$$\Longrightarrow d = -\frac{v}{\alpha\pi/2}. \tag{4.125c}$$

which is again not a valid solution. Therefore, there is no periodic solution at $\dot{d} = 0$ and $\dot{\theta} - \dot{\psi} = 0$.

4.4.1 Stability Analysis via Lyapunov Functions

Now we will examine the stability of the equilibrium set E in (4.121). As in the previous section we will investigate the behavior of the time derivatives of the difference between the orientation of the robot and the orientation of the distance vector $\overrightarrow{d}$ which is

$$\dot{\theta} - \dot{\psi} = -\alpha\,[\mathrm{mod}(\theta - \psi + \pi, 2\pi) - \pi] - \frac{1}{d} v\, \sin(\theta - \psi) \tag{4.126}$$

The numerical solution of this equation is presented for different values of v, α, and d in Figure 4.41. As seen from the figure there is only one solution of the equation $\dot{\theta} - \dot{\psi} = 0$ which is $\theta - \psi = 2k\pi$ for $k = 0, \mp 1, \mp 2....$ The slope of $\dot{\theta} - \dot{\psi}$ at $\theta - \psi = 2k\pi$ is negative which means the root is a stable root. We can also reach the same result by the following analytical solution. The derivative $\frac{d(\dot{\theta} - \dot{\psi})}{d(\theta - \psi)}$ at the root $\theta - \psi = 2k\pi$ can be derived as

$$\frac{d(\dot{\theta} - \dot{\psi})}{d(\theta - \psi)} = \frac{d}{d(\theta - \psi)}[-\alpha\,[\mathrm{mod}(\theta - \psi + \pi, 2\pi) - \pi] - \frac{1}{d} v\, \sin(\theta - \psi)] \tag{4.127a}$$

$$= -\alpha - \frac{1}{d} v\, \cos(\theta - \psi) \tag{4.127b}$$

$$\frac{d(\dot{\theta} - \dot{\psi})}{d(\theta - \psi)}|_{\theta - \psi = 2k\pi} = -\alpha - \frac{1}{d} v\, \cos(2k\pi) \tag{4.127c}$$

$$= -\alpha - \frac{1}{d} v < 0 \tag{4.127d}$$

Since $\frac{d(\dot{\theta} - \dot{\psi})}{d(\theta - \psi)}|_{\theta - \psi = 2k\pi} < 0$, the root at $\theta - \psi = 2k\pi$ is a stable root.

For better understanding of the relations between the state variables and therefore, the system behavior, we should consider the change of $\dot{\theta} - \dot{\psi}$ with respect to both distance between the agent and the target and the relative angle difference $\theta - \psi$. Hence, a 3-dimensional plot of state variables $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ and the distance d is presented in Figure 4.42. The speed of the robot is kept constant at $v = r\alpha\pi/2$ for generating this figure. As seen from the figure the only stable root for several system parameters is at $\theta - \psi = 2k\pi$. Therefore, if the gradient vector is in the same direction of vector $\overrightarrow{d}$ then the robot motion will converge to the direction of the gradient vector. The robot will escape from the target in the same direction of the vector from target to robot.

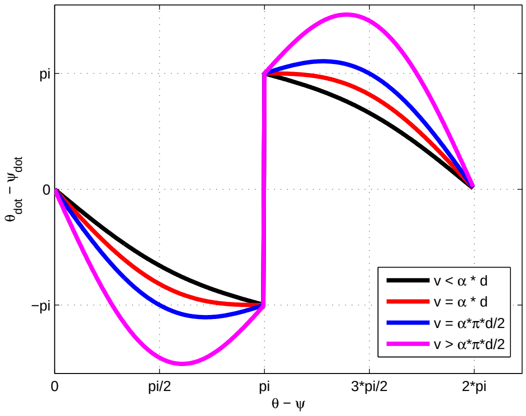


Figure 4.41: $\dot{\theta} - \dot{psi}$ versus $\theta - \psi$ for several v values.

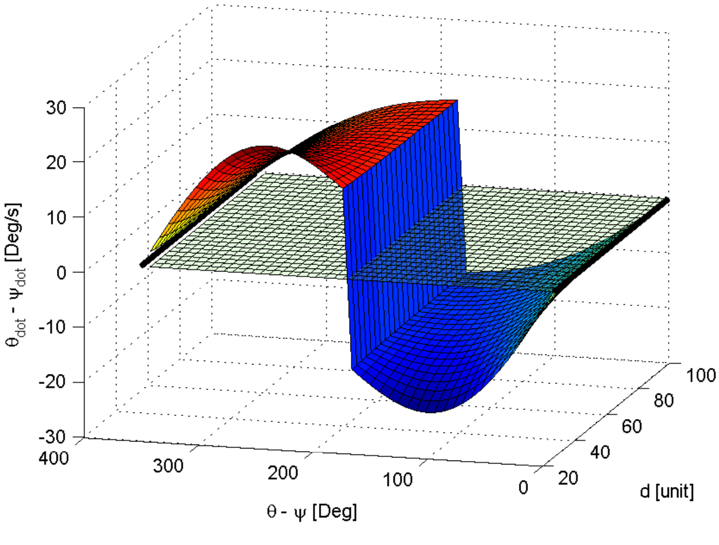


Figure 4.42: $\dot{\theta} - \dot{\psi}$ versus $\theta - \psi$ and the distance d for constant $v = r\alpha\pi/2$ where r is the radius of the circle on which the robot travels.

This strategy may be utilized for just making the robot escape from a target. However, in this study we will utilize a combination of two strategies presented in the previous and this section. Recall that in the previous section we showed that if the gradient vector directs from the robot to the target and if $v < \alpha d$ then the robot moves towards the target by converging to the axes between the robot and the target. In addition we showed in this section that if the gradient vector is from the target to the robot then for any value of the robot speed, v and proportional controller gain α the robot escapes from the target converging to the axes between robot and target. Therefore, we propose a controller that changes the direction of the gradient vector at a pre-specified distance from the target so that the robot circles around the target. In the following section we present the simulation results for this controller strategy.

4.4.2 Simulation Results

In this section we present simulation results confirming the findings for circling around a target with variable gradient strategy. Note that although we designed the controller assuming the target is stationary we also present the performance of the controller for a dynamic target.

In this simulation the speed of the robot is taken as $v = 0.3\alpha d < \alpha d$. The controller is designed such that the gradient vector changes direction at $d = 30[units]$. For larger distances the gradient vector directs from the robot to the target so that the equilibrium set in (4.12) is a stable point and robot gets closer to the target by converging to the axes from the robot to the target. For smaller distances ($d < 30[units]$) the direction of gradient vector is from the target to the robot so that the equilibrium set in (4.121) is a stable equilibrium and thus the robot gets away from the target by converging to the axes from the target to the robot. Therefore, the change in the direction of the gradient vector at $d = 30[units]$ results in the circling behavior of the robot around the target.

In Figures 4.43 and 4.44 the trajectories of the robot and the distance between the robot and the target are shown for different initial positions and orientations. The simulation parameters are selected as $\alpha = 3$, $v = 0.3\alpha d$. All of the trajectories converge to the circular path centered at $[x, y] = [20, 20]$ with radius $\bar{r} = 30[units]$. As can be seen from Figure 4.43 some of the trajectories circle around the target in CW direction while some are rotating in CCW direction. The direction of the rotation depends on the angle of approach towards the target and is not aimed to be controlled in the scope of this study.

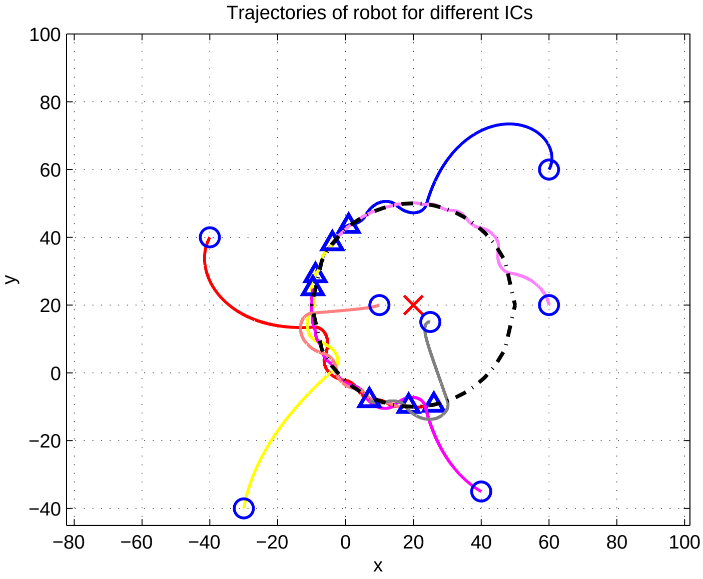


Figure 4.43: Robot trajectories circling around a static target at $[20, 20]$.

In Figure 4.45 the robot trajectories for the case in which the robot is located at the same position with different initial orientation angles are presented. As seen from the figure all trajectories converge to the same circle but the rotations vary in CW and CCW directions depending on the angle of approach to the target.

Next, the dynamic target presented in the previous section is again utilized to observe the circling behavior of the robot. The controller parameters are selected to obtain a circling radius of 10 $[units]$ ($\alpha = 5$, $v = 0.6\alpha d$). The robot trajectory and the distance of the robot from the target are presented in Figures 4.46 and 4.47, respectively. As can be seen from the results the robot achieves the objective of circling around the target in motion (the dashed curve is the path of the target) with errors which are at most 0.5 [units]. This error is due to the fact that the controller was designed based on the assumption of a static target.

4.5 Switching Gradient Method for Line Formation

Here we present a controller for line formation of robots. In fact the robots get far away from the target by converging to a predefined axes. The axes passes through the target and the angle of the axes depends on the deviation angle β applied to the gradient vector, which is a pre-specified controller parameter. The original gradient vector directs from the robot towards

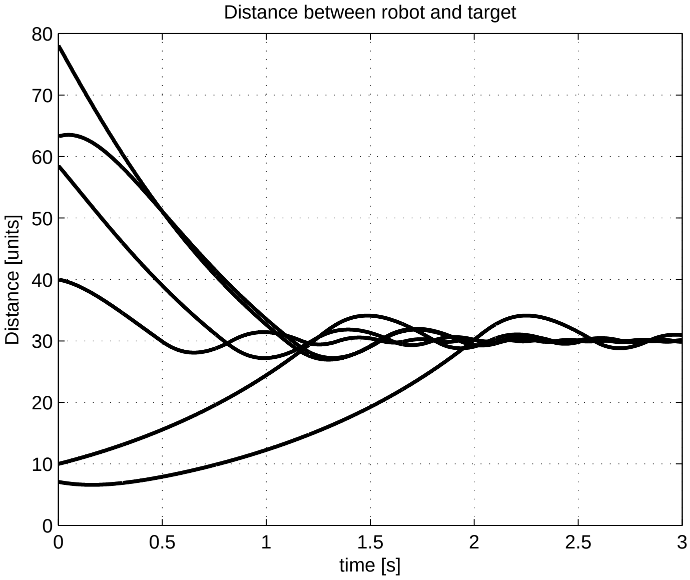


Figure 4.44: Distance between robot and static target at [20, 20] for circling around the target strategy.

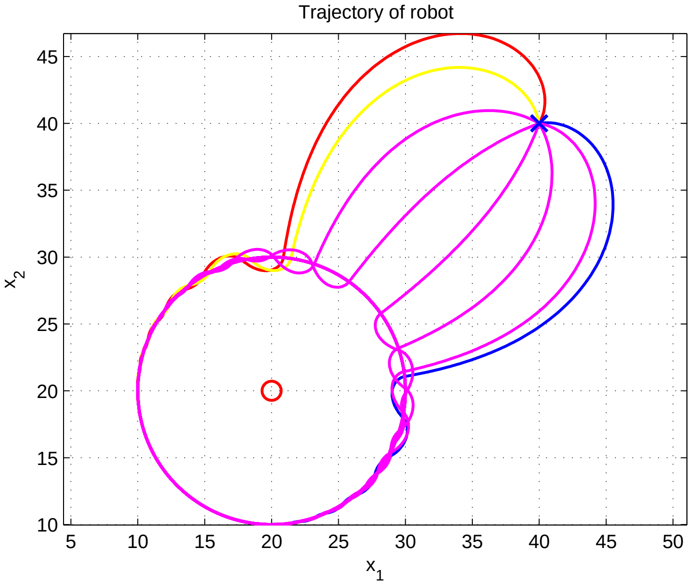


Figure 4.45: Robot trajectories circling around a static target at [20, 20] for different initial angles.

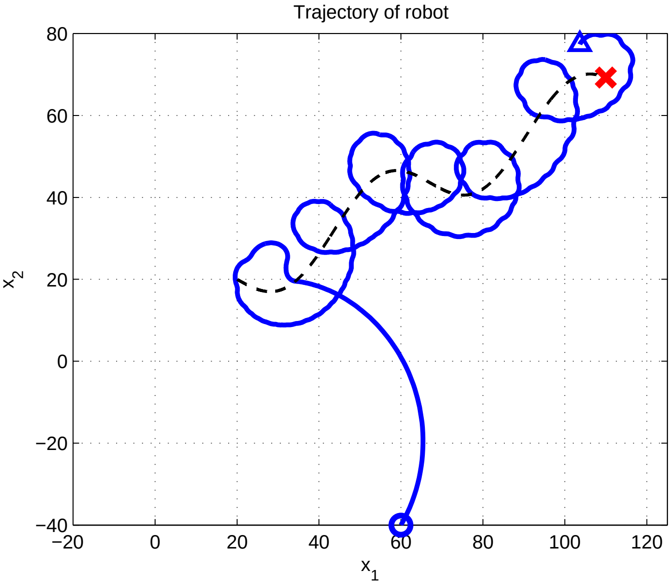


Figure 4.46: Robot trajectory circling around a dynamic target.

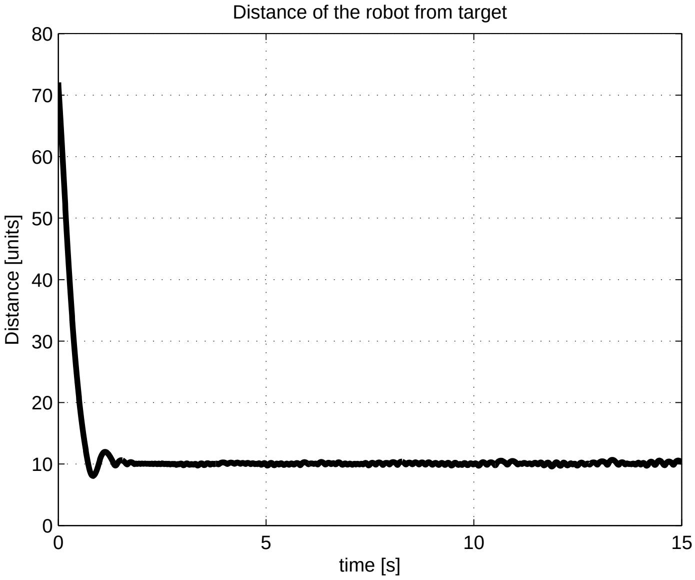


Figure 4.47: Distance between target and robot circling around a dynamic target.

the target or in the opposite direction of $\vec{d}$ as in the first section. Therefore, the angle of the modified gradient vector is

$$\gamma' = \beta - angle(\nabla G) = \beta - \psi - \pi \tag{4.128}$$

The modified dynamics of the system becomes

$$\begin{aligned}
\dot{d} &= v\cos(\theta - \psi) && (4.129a)\\
\dot{\psi} &= \frac{1}{d}v\sin(\theta - \psi) && (4.129b)\\
\dot{\theta} &= -\alpha[\mathrm{mod}(\theta - \gamma' + \pi, 2\pi) - \pi] && (4.129c)\\
&= -\alpha[\mathrm{mod}(\theta - \beta + \psi, 2\pi) - \pi] && (4.129d)
\end{aligned}$$

The steady state of this system occurs at $\dot{d} = v$, $\dot{\psi} = 0$, and $\dot{\theta} = 0$. Then for $\dot{d} = v$ we have

$$v\ \cos(\theta - \psi) = v \Longrightarrow\ \theta - \psi = 2k\pi\ \ for\ \ k = 0, \mp 1, \mp 2... \tag{4.130}$$

Similarly, for $\dot{\psi} = 0$

$$\frac{1}{d}v\ \sin(\theta - \psi) = 0 \Longrightarrow\ v = 0,\ \ \theta - \psi = k\pi\ \ for\ \ k = 0, \mp 1, \mp 2... \tag{4.131}$$

And, for $\dot{\theta} = 0$ we obtain

$$-\alpha[\mathrm{mod}(\theta - \beta + \psi, 2\pi) - \pi] = 0 \Longrightarrow \theta - \beta + \psi = (2k + 1)\pi\ \ for\ \ k = 0, \mp 1, \mp 2... \tag{4.132}$$

The common solution of the equations (4.130), (4.131), (4.132) is the set S

$$S = \left\{ v \in R^{+}, 0 < \theta, \psi < 2\pi \| v > 0, \theta = \psi = \frac{\pi + \beta}{2} \right\}\ \ for\ \ k = 0, \mp 1, \mp 2... \tag{4.133}$$

Now we will examine the stability of the steady state set S in (4.133) by Lyapunov Stability theorem.

4.5.1 Stability Analysis via Lyapunov Functions

In this section, we will examine a Lyapunov function candidate that shows the attractiveness of the set in (4.133). Before we present the Lyapunov function we should present the second derivatives of the system variables to be utilized in the time derivative of the Lyapunov function. The system dynamics is repeated in the following equation

$$\begin{aligned}
\dot{d} &= v\cos(\theta - \psi) && (4.134a)\\
\dot{\psi} &= \frac{1}{d}v\sin(\theta - \psi) && (4.134b)\\
\dot{\theta} &= -\alpha[\mathrm{mod}(\theta - \beta + \psi, 2\pi) - \pi] && (4.134c)
\end{aligned}$$

The second time derivatives of the variables are given by

$$
\begin{aligned}
\ddot{d} &= -v\sin(\theta-\psi)(\dot{\theta}-\dot{\psi}) \\
&= -d\dot{\psi}(\dot{\theta}-\dot{\psi}) && (4.135\text{a}) \\
\ddot{\psi} &= -\frac{\dot{d}}{d^2}v\sin(\theta-\psi)+\frac{1}{d}v\cos(\theta-\psi)(\dot{\theta}-\dot{\psi}) \\
&= -\frac{\dot{d}}{d}\dot{\psi}+\frac{\dot{d}}{d}(\dot{\theta}-\dot{\psi}) \\
&= \frac{\dot{d}}{d}(\dot{\theta}-2\dot{\psi}) && (4.135\text{b}) \\
\ddot{\theta} &= -\alpha(\dot{\theta}+\dot{\psi}) && (4.135\text{c})
\end{aligned}
$$

The Lyapunov function candidate is

$$
V(\dot{d},\dot{\theta},\dot{\psi})=\frac{1}{2}\alpha(\dot{d}-v)^2+\frac{1}{2}\alpha(d\dot{\psi})^2+\frac{1}{2}vd\dot{\theta}^2 \tag{4.136}
$$

where $D=\{\dot{d},\dot{\theta},\dot{\psi}\in\mathbb{R}\}$. $V(\dot{d},\dot{\theta},\dot{\psi})$ is positive definite in D. Then the Lie derivative of the Lyapunov function becomes

$$
\dot{V}=\alpha(\dot{d}-v)\ddot{d}+\alpha(d\dot{\psi})\left[\dot{d}\dot{\psi}+d\ddot{\psi}\right]+v\left[\frac{1}{2}\dot{d}\dot{\theta}^2+d\dot{\theta}\ddot{\theta}\right] \tag{4.137}
$$

Then, substituting the second derivatives in (4.135) into (4.137) one obtains

$$
\dot{V}=\alpha(\dot{d}-v)\left[-d\dot{\psi}(\dot{\theta}-\dot{\psi})\right]+\alpha(d\dot{\psi})\left[\dot{d}\dot{\psi}+d\frac{\dot{d}}{d}(\dot{\theta}-2\dot{\psi})\right]+v\left[\frac{1}{2}\dot{d}\dot{\theta}^2+d\dot{\theta}\left(-\alpha(\dot{\theta}+\dot{\psi})\right)\right] \tag{4.138}
$$

which can be rearranged as

$$
\begin{aligned}
\dot{V} &= -\alpha d(\dot{d}-v)\dot{\psi}(\dot{\theta}-\dot{\psi}) && (4.139\text{a}) \\
&\quad +\alpha d\dot{\psi}\dot{d}(\dot{\theta}-\dot{\psi})+ && (4.139\text{b}) \\
&\quad +\frac{1}{2}\dot{d}v\dot{\theta}^2-dv\dot{\theta}\alpha(\dot{\theta}+\dot{\psi}) && (4.139\text{c})
\end{aligned}
$$

and further as

$$
\begin{aligned}
\dot{V} &= -\alpha d\dot{d}\dot{\psi}(\dot{\theta}-\dot{\psi})+\alpha dv\dot{\psi}\dot{\theta}-\alpha dv\dot{\psi}^2 && (4.140\text{a}) \\
&\quad +\alpha d\dot{d}\dot{\psi}(\dot{\theta}-\dot{\psi}) && (4.140\text{b}) \\
&\quad +\frac{1}{2}\dot{d}v\dot{\theta}^2-\alpha dv\dot{\theta}^2-\alpha dv\dot{\psi}\dot{\theta} && (4.140\text{c})
\end{aligned}
$$

After appropriate calculations we obtain

$$
\dot{V}=-\alpha dv\dot{\psi}^2-\alpha dv\dot{\theta}^2+\frac{1}{2}\dot{d}v\dot{\theta}^2 \tag{4.141}
$$

which can be written as

$$\dot{V} = -\alpha d v \dot{\psi}^2 - v\dot{\theta}^2 \left(\alpha d - \frac{1}{2}\dot{d}\right) \tag{4.142}$$

The Lie derivative of the Lyapunov function, $\dot{V}$ is negative semi-definite for $\alpha d - \frac{1}{2}\dot{d} > 0$. Note that $\dot{d}$ is at most v since $\max(v\cos(\theta - \psi)) = v$. Therefore, if $\alpha d > v/2$ then $\dot{V}$ is negative semi-definite. Also note that if $\dot{d}$ is negative then $\alpha d - \frac{1}{2}\dot{d}$ is positive and if $\dot{d}$ is positive then $\alpha d - \frac{1}{2}\dot{d}$ may be negative but as time passes d increases and so $\alpha d - \frac{1}{2}\dot{d}$ becomes positive. The stability should be investigated by the use of Lasalle's principle.

4.5.2 Simulation Results

In this section we present simulation results confirming the findings for the motion of the robot that converges to a line.

In this simulation the speed of the robot is taken as constant $v = 10[units/sec]$. In Figures 4.48 and 4.49 the trajectories of the robot and the distance between the robot and the line are given for different initial positions and orientations. The simulation parameters are selected as $\alpha = 3, \beta = 3\pi/2$. As seen from the Figure 4.48 the line that the robots are converging to passes through the target position ($[x, y] = [20, 20]$) with the slope $(\pi + \beta)/2 = 5\pi/4$. In Figure 4.49 the distance from the robot to the line is plotted with respect to time. All of the distances for different initial positions are converging to zero. Note that the trajectories of the robot are towards the two ends of the line. The initial position and orientation of the robot determines to which end the robot will converge to.

In Figure 4.50 the robot trajectory located at the same position with different initial orientations are presented. β is selected to be 150 degrees. As seen from the figure all trajectories converge to the line passing through the target with slope of $(180 + \beta)/2 = 165$ degrees. In Figure 4.51 the distance between the robot and the line is shown. As seen all distance values converge to zero.

4.6 Conclusion

In this study we considered the unicycle mobile robot kinematics and developed simple proportional controllers for the speed and angular velocity of the robot for achieving approaching

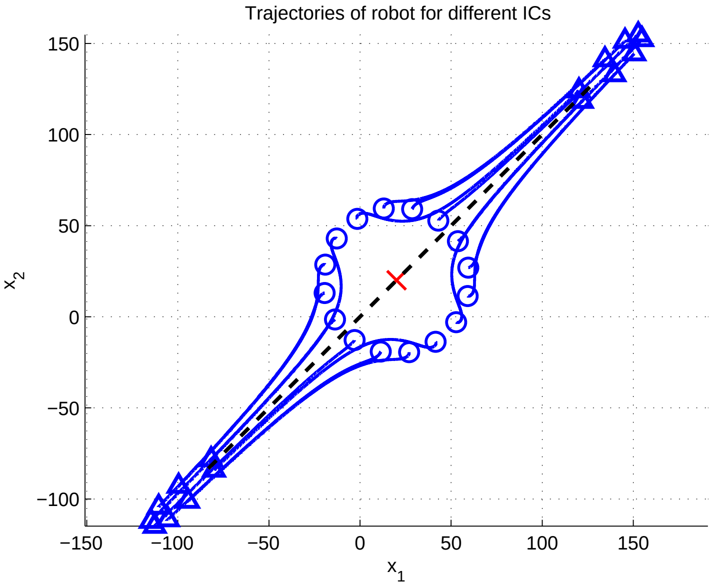


Figure 4.48: Robot trajectories converging to line passing through a static target at $[20, 20]$.

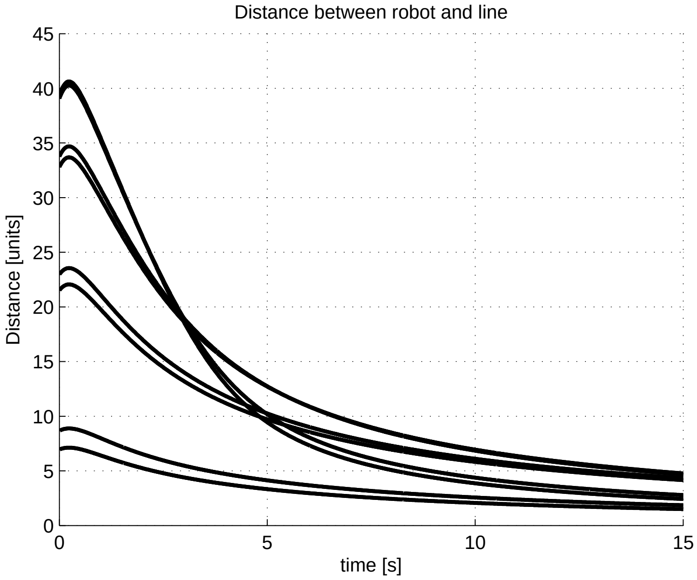


Figure 4.49: Distance between robot and line for different initial positions.

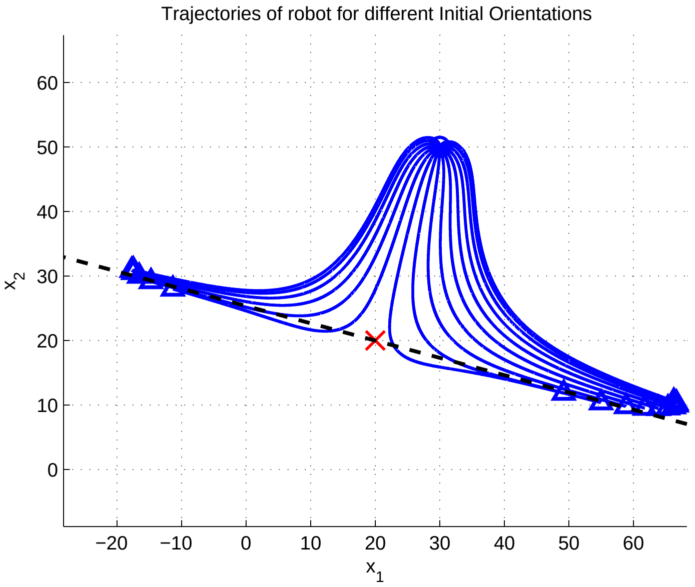


Figure 4.50: Robot trajectories converging to line passing through a static target at $[20, 20]$ for different initial orientations.

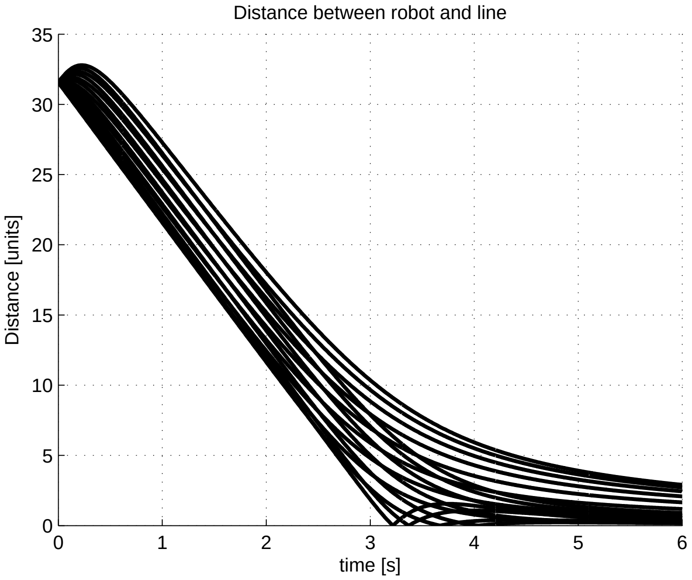


Figure 4.51: Distance between robot and line for different initial orientations.

and circling behavior around a static target and approach to the line passing through the target with a pre-specified orientation. For these controllers the stability of the system is investigated and some conditions on the controller coefficients are derived. In this study we propose controllers that achieve circling and line formation behavior of the mobile robot without any pre-specified path. We also show that a mobile robot can easily achieve a switching between the target tracking and circling around a target controller strategies. Hence, a mobile robot is able to first approach the target and then start to circle around that target without any external switching input or external decision mechanism. We validate the controller performances using numerical simulations.

Although the controllers designed in this study are developed for static targets we also mention their performances for dynamic targets. In this chapter we utilized just proportional controllers and performed analytical studies on the stabilities of the controllers. Some of our simulation based studies show that a controller that achieves divergence of a mobile robot at a pre-specified axes passing through a given target location can also be developed with a similar logic explained in this study. Moreover, there may be other gradient vector functions to be utilized for the controllers, especially a gradient vector that points biased directions from the target may also lead to interesting but useful trajectories.

CHAPTER 5

Circling Controllers Developed by Feedback Linearization

5.1 Introduction

Several trajectory generation and trajectory control methods for autonomous navigation of mobile robots have been developed in the literature. Many nonlinear control theories like Lyapunov stability theory, Lasalle's invariance principle, feedback linearization, output regulation, backstepping, sliding mode controllers etc. have been utilized for the path tracking problems.

One of the earlier studies on the tracking problem of autonomous mobile robots is [137]. Kanayama et.al. propose a stable tracking control rule for non-holonomic vehicles and prove the stability of the rule through the use of a Lyapunov function in addition to the experimental validations. In [138], the authors study on the closed loop control laws for unicycle-like vehicles. They show that the use of the simplest quadratic form as candidate Lyapunov function directly leads to the definition of very simple, smooth and effective closed loop control laws suitable to be used for steering, path following, and navigation. In [139], the control of a wheeled mobile robot to track a moving target with limited control inputs is studied. Based on a proper system modeling, a Lyapunov based controller design approach is used to determine the robot's angular and linear velocities to achieve the asymptotic convergence of the tracking errors. Simulation results are provided to verify the effectiveness of the proposed approach. In [140] the authors propose a vehicle command system for the wheeled autonomous mobile robots with a high capability in describing the navigation task in the real environment and describe a feedback control method to track along the given paths by the vehicle commands. Another study on the target tracking problem of unicycle type mobile robots that utilizes Lyapunov functions in the stabilization is [141]. The study in [142], deals with the formulation and solution of the tracking control problem with saturation constraint for a class of unicycle-modeled mobile robots. The authors utilize the backstepping technique and the idea from the LaSalle's invariance principle in the solutions and show that with the proposed control laws, the robot can globally follow any path specified by a straight line, a circle or a path approaching the origin using a single controller. In [143], a target tracking controller is designed with four possible moving directions of linear and angular velocities using Lyapunov stability theory. The proposed controller is experimentally demonstrated under high velocity and acceleration conditions with different control parameters. Fukao et.al. [144] study on a method to design an adaptive tracking controller for the dynamic model of a non-holonomic mobile robot with unknown parameters in the kinematic parts by adaptive backstepping. In [145], a first-state contractive model predictive control algorithm is developed for the trajectory tracking, point stabilization and formation problems of nonholonomic mobile robots. The authors also presents simulation results showing the proposed control algorithm can generate satisfactory system responses.

Another recent study on the tracking control is [146] that utilizes the sliding-mode controllers for wheeled-mobile robots in polar coordinates. They design two controllers that asymptotically stabilize the tracking errors in position and heading direction so that they achieve the stabilization and trajectory tracking for reference trajectories. The target tracking problem is solved by fuzzy controller methods in [147]. The authors utilize two mobile robots where one is the target mobile robot with infrared transmitters and the other one is the tracker mobile robot with infrared receivers and reflective sensors. The latter is designed to track the former mobile robot which is designed to drive in a specific trajectory. In [148] and [149], the artificial potential methods and sliding mode controllers are utilized for tracking maneuvering targets. In [150], a control scheme that combines a kinematic controller and a sliding mode dynamic controller with external disturbances is designed for an automatic guided vehicle to track a desired trajectory with a specified constant velocity. The control law is obtained based on the backstepping technique and system stability is proved using the Lyapunov stability theory. The simulation and experimental results are presented to illustrate the effectiveness of the proposed controller. The paper [151] describes a strategy for trajectory tracking and path-following control of vehicles using sliding surface. In [152], the trajectory tracking control problem for the wheeled mobile robot is solved using the sliding mode control.Four control laws are modeled and the system performances are investigated. The sliding mode control laws for the trajectory tracking problem are simulated and then implemented on a laboratory mobile Robot. In [153], a robust tracking control of nonholonomic wheeled mobile robots using sliding mode is proposed. The posture of a mobile robot is represented by polar coordinates and the dynamic equation of the robot is feedback-linearized by the computed-torque method. The study is supported by experimental validations.

The feedback linearization techniques are also widely utilized nonlinear controller design methods in the path following problems of mobile robots. One of the early studies is [154] where the authors design controllers for both unicycle-type and two-steering-wheels mobile robots. They use the exact output feedback linearization and decoupling techniques and a Lyapunov oriented approach in the design of controllers. The controllers are associated with a specific parametrization of the relative path to vehicle distance and orientation and achieve the path following convergence. In [155], authors use the dynamic feedback linearization techniques in the controller design for both trajectory tracking and target point regulation problems of wheeled mobile robots. They support their work with the experiments on a mobile robot. Yang et.al., considers the state constraints in the control of nonholonomic car-like mobile robots in [156]. They achieve point-to-point tracking and circular trajectory tracking in the simulations. The controllers are designed by dynamic feedback linearization techniques and the steering angle is constrained. In [157] the unified polar-space kinematics control approach for set-point stabilization, trajectory tracking, and path following of an omnidirectional mobile robot. The unified kinematics controller has been designed using feedback linearization to simultaneously achieve three basic navigation problems. Computer simulations and experimental results have shown that the proposed kinematics controller is capable of accomplishing stabilization, trajectory tracking, and path following missions at slow speeds. In [158], the point stabilization of the mobile robot is formulated by utilizing the state-space exact feedback linearization. The case in which the mobile robot moves to a target point without heading angle constraints and the control of the robot to a target point with heading angle constraint is formulated. In [159] an output-feedback tracking controller for the unicycle-type mobile robot is considered assuming that only the measurements of the x and y position coordinates are available. Using cascaded systems theory it is shown that local exponential stability by means of Lyapunov is obtained for the resulting closed loop system. A verfication of these results is demonstrated by implementation on an experimental mobile robot. In [160], the authors are interested in solving the problem of tracking with stability of a reference trajectory, by means of linearizing ”static” and ”dynamic” state feedback laws. They give conditions to avoid possible singularities of the feedback laws. In [161], authors design nonlinear adaptive control laws by Lyapunov theory and backstepping techniques. The designed control law for path following deals with vehicle dynamics and plant parameter uncertainty. Furthermore, it overcomes stringent initial condition constraints by controlling explicitly the rate of progression of a ”virtual target” to be tracked along the path, thus bypassing the problems that arise when the position of the virtual target is simply defined by the projection of the actual vehicle on that path.

In [162], the authors propose a number of guidance laws driving wheeled robots towards stationary and moving targets. The control scheme utilizes just the distance measurements between the vehicles and targets. The vehicles approach the targets along an equiangular spiral trajectory and stays steady on a circular path centered at the target. The authors support the analytical results with experimental and simulation results.

The studies in the literature mostly utilize pre-generated paths for tracking by mobile robots. By using these paths many researchers achieved the robots to follow straight lines, circular or randomly generated paths. In this study first the circling around stationary and non-stationary target problems for ground and air vehicles are considered as periodic system problems and the states in the system kinematics are forced to converge to periodical trajectories. We propose new stable methods for mobile robots to track circular paths of which the center -considered as the target- is given. The radius of the circular path and the angular velocity can be adjusted by the controller parameters. We propose different controllers for three different cases. In the first one, the only controllable input is the angular velocity of the robot (SISO-Stationary Target Controller). The translational speed of the robot is taken as constant. This case can represent applications such as autonomous air vehicles (UAV) circling around a stationary target. The only required global info is the distance to the target. In the second case in addition to the angular velocity input the translational speed is taken as an input (MIMO-Stationary Target Controller). So that one can adjust both the radius of the circular path and the angular velocity by controlling these inputs. The only required global info is the distance to the target. The last controller developed is similar to the MIMO controller but this time the target is not stationary (MIMO-NonStationary Target Controller). The controller utilizes the target velocity vector in the manipulations and achieves much better performance with respect to the stationary target controllers.

The circling behavior is achieved without pre-specified paths to be followed by the robots. In fact robots use the target position and velocity to achieve the mentioned behaviors. The usage of the only target information instead of pre-specified paths is mostly beneficial in the multi-robot applications. In multi-robot applications usually each robot is a target for the others; therefore, the controllers that use target information is preferable with respect to pre-generated paths since in most of the applications the paths of the robots in the swarm are emergent paths that are hard to predict. The controllers developed for single agents to circle around specified targets are applied to multi-agent systems. The agents in the swarm utilized the same controllers targeting the centroid of the swarm to achieve the rotational motion on the same circle. A minor modification is performed in the MIMO-stationary target controller to achieve circling. The MIMO controllers worked well in doing the circling behavior of swarms with various number of agents.

In the following sections, the mathematical background and development of the controllers are presented. The results of the analytical derivations are validated in the simulation study sections.

5.2 Mathematical Model

In this chapter, the control problem is constructed by the first-order kinematic model of the unicycle vehicles since in most of the applications the control architecture of mobile robots does not allow the user to impose acceleration or torque inputs; they rather use the speed and position inputs. The unicycle dynamics of a mobile robot in cartesian coordinates are the same with the ones in the previous Chapter 4 section 4.2 (see equations (4.1) and (4.3)). However, we will present the model again for the reader to follow up easily. The kinematics are represented as

$$\dot{z}_1 = v\ \cos(\theta) \tag{5.1a}$$

$$\dot{z}_2 = v\ \sin(\theta) \tag{5.1b}$$

$$\dot{\theta} = u \tag{5.1c}$$

where v is the translational speed, θ is the steering angle, and u is the controller for the angular speed of the agent. The agent and the target are shown on Figure 5.1. With appropriate

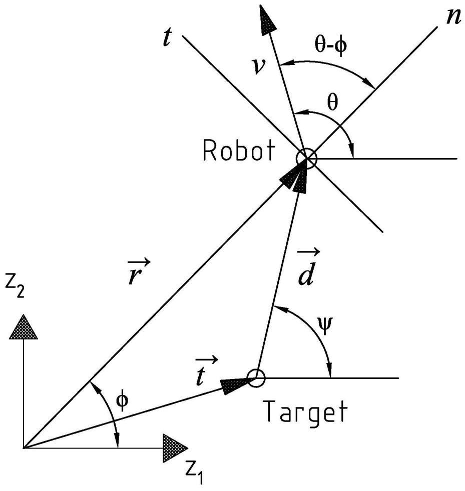


Figure 5.1: System model in polar coordinates.

transformation the dynamics of the system can be transformed into polar coordinates as (see Figure 5.1)

$$\begin{aligned} \dot{r} &= v\ \cos(\theta - \phi) && (5.2a) \\ \dot{\phi} &= \frac{1}{r} v\ \sin(\theta - \phi) && (5.2b) \\ \dot{\theta} &= u(\theta, \gamma) && (5.2c) \end{aligned}$$

where r is the distance from the origin to the position of the agent $r = \sqrt{x^2 + y^2}$ and ϕ is the angle of the vector along r, $\phi = atan2(y, x)$ and θ is the steering angle (orientation) of the agent as mentioned above.

5.2.1 Stationary Target

For the time being we assume that the robot obtains the position of the target perfectly (exact values with synchronous timing i.e., no time delay in sensing) and the target is stationary. Furthermore, since we are interested in the relative position of the robot with respect to the target, we will utilize the dynamics in relative coordinates. The reference coordinate frame

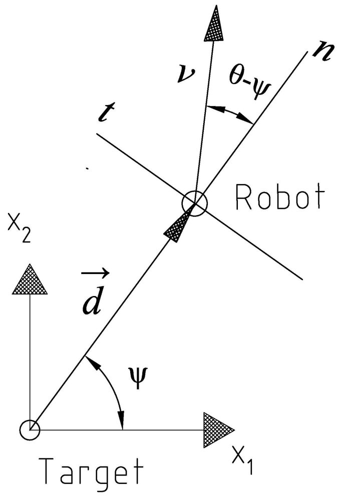


Figure 5.2: System model for relative coordinates.

(x_1-x_2 frame) has the origin on the target with parallel coordinate axes to the global coordinate frame axes. The relative coordinates are plotted in Figure 5.2 and can be described with the equations.

$$\dot{x}_1 = v\ \cos(\theta) \tag{5.3a}$$

$$\dot{x}_2 = v\ \sin(\theta) \tag{5.3b}$$

$$\dot{\theta} = u(\theta, \gamma) \tag{5.3c}$$

Again with the coordinate transformation $d = \sqrt{x_1^2 + x_2^2}$ and $\phi = atan2(x_2, x_1)$ and under the assumption that the target is stationary the dynamics of the robot in the polar coordinates relative to the target is obtained as

$$\dot{d} = v\cos(\theta - \psi) \tag{5.4a}$$

$$\dot{\psi} = \frac{1}{d} v\sin(\theta - \psi) \tag{5.4b}$$

$$\dot{\theta} = u(\theta, \gamma) \tag{5.4c}$$

5.2.2 Non-Stationary Target

For many cases the vehicles will face with non-stationary targets. The kinematics of the vehicles does not change in the global reference frame; however they do in the relative coordinate

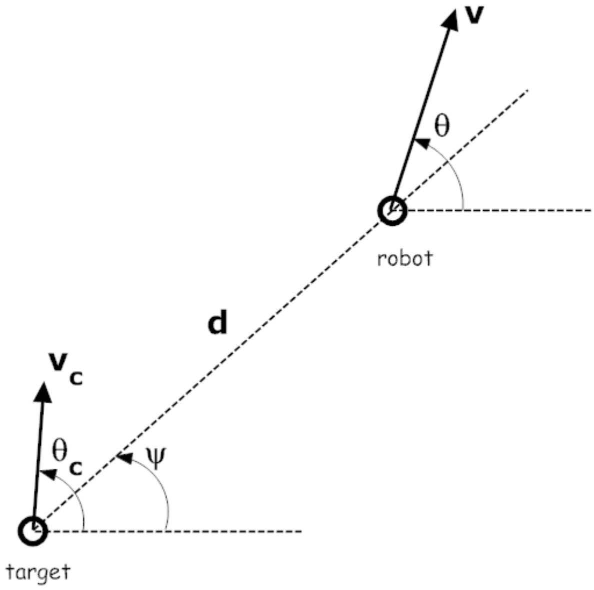


Figure 5.3: System model with non-stationary target in relative coordinates.

frame. The kinematics of the agent relative to the non-stationary target becomes (see Figure 5.3)

$$\begin{aligned}
\dot{d} &= v\cos(\theta-\psi) - v_c\cos(\theta_c-\psi) && \text{(5.5a)}\\
\dot{\psi} &= \frac{1}{d}\left[v\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right] && \text{(5.5b)}\\
\dot{\theta} &= u(\theta,\gamma) && \text{(5.5c)}
\end{aligned}$$

where v_c and θ_c are the speed and angular velocity of the target, respectively.

Using the kinematics above we will solve the the feedback linearization problems for "Circling Around a Target" behavior, in the following sections.

5.3 Circling Around a Target

In this section the circling around a target problem for ground and air vehicles is considered as a periodic system problem and the states in the system kinematics are forced to converge to periodical trajectories. We propose a new stable method for a mobile robot to track a circular path of which the center -considered as the target- is given. The radius of the circular path and the angular velocity can be adjusted by the controller parameters. We propose different controllers for two different cases. In the first one, the only controllable input is the angular velocity of the robot. The translational speed of the robot is taken as constant. We consider this case especially for the air vehicles (like UAV) circling around a target with constant speed. In the second case in addition to the angular velocity input the translational speed is taken as an input. So that one can adjust both the radius of the circular path and the angular velocity by controlling these inputs. We use static feedback linearization methods in the first case and dynamic feedback linearization in the second one. The first will be called as "Circling-SISO" where the only controlled system state is the angular velocity of the robot ($\dot{\theta} = u$). The robot travels at constant translational speed v. The second case called as "Circling-MIMO" is the case in which both translational and rotational speeds are inputs of the system ($v = u_1$ and $\dot{\theta} = u_2$). In the following sections we will derive the feedback linearization laws for these cases and present some simulation results. In the following sections, the feedback linearization results are presented. The results of the analytical derivations are validated in the simulation study sections. We lastly present the conclusions on the results of this part and mention some future work.

5.3.1 Circling - SISO Case - Stationary Target

In this section the circling around a target controller for the unicycle kinematics of a UGV or UAV is developed. For this purpose we use the feedback linearization techniques. The only controlled input is the angular velocity of the vehicles ($u = \dot{\theta}$). The translational velocity of the vehicle is assumed to be constant at v. Therefore, we write the dynamics as

$$\begin{bmatrix} \dot{d} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta - \psi) \\ \frac{1}{d} v \sin(\theta - \psi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{5.6}$$

For simplicity we will lump the two states θ and ψ as $\gamma = \theta - \psi$. Note that, for circling behavior only the relative dynamics $(\theta - \psi)$ is important. Then the system kinematics become

$$\begin{bmatrix} \dot{d} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} v\cos(\gamma) \\ -\frac{1}{d} v \sin(\gamma) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \Rightarrow \dot{x} = f(x) + g(x)u \tag{5.7}$$

For the circling behavior of a vehicle the equilibrium of the states are desired to be at

$$\dot{d} \;=\; v\cos(\gamma) = 0 \Rightarrow \gamma = \frac{\pi}{2}(2k+1) \tag{5.8}$$

Let the radius of the circling be $d = d_0$. At this equilibrium the other state's steady value becomes

$$\dot{\gamma} \;=\; u - \frac{1}{d}v\sin(\gamma) = 0 \Rightarrow u = \pm\frac{v}{d_0} \tag{5.9}$$

To achieve this equilibrium, we will use the feedback linearization method to linearize the system and then design linear system controllers.

The output of the system is selected as

$$y = k(x) = d \tag{5.10}$$

Then the lie derivatives along the nonlinear system functions $f(x)$ and $g(x)$ are

$$L_g k(x) \;=\; \frac{\partial k}{\partial x}g(x) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = 0 \tag{5.11a}$$

$$L_f k(x) \;=\; \frac{\partial k}{\partial x}f(x) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}v\cos(\gamma)\\-\frac{1}{d}v\sin(\gamma)\end{bmatrix} = v\cos(\gamma) \tag{5.11b}$$

$$L_g L_f k(x) \;=\; \frac{\partial L_f k(x)}{\partial x}g(x) = \begin{bmatrix}0 & -v\sin(\gamma)\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = -v\sin(\gamma) \tag{5.11c}$$

At equilibrium ($\gamma = \pi/2$)

$$L_g L_f k(x)|_{\gamma=\pi/2} = -v\sin(\pi/2) = -v \neq 0 \tag{5.12}$$

Therefore, around the equilibrium point ($\gamma = \pi/2$), the relative degree is 2. In fact, it is $\rho = 2$ in $(0, \pi)$ and $(-\pi, 0)$. To derive the linearizing input we first find second order Lie derivative of $k(x)$ along $f(x)$ as

$$L_f L_f k(x) = \frac{\partial L_f k(x)}{\partial x}f(x) = \begin{bmatrix}0 & -v\sin(\gamma)\end{bmatrix}\begin{bmatrix}v\cos(\gamma)\\-\frac{1}{d}v\sin(\gamma)\end{bmatrix} = \frac{1}{d}v^2\sin^2(\gamma) \tag{5.13}$$

Then the linearizing input is calculated as

$$\begin{aligned} u &= \frac{1}{L_g L_f k(x)}\left[-L_f L_f k(x) + w\right] \\ &= \frac{1}{-v\sin(\gamma)}\left[-\frac{1}{d}v^2\sin^2(\gamma) + w\right] \\ &= -\frac{w}{v\sin(\gamma)} + \frac{1}{d}v\sin(\gamma) \end{aligned} \tag{5.14}$$

The normal form of the system dynamics according to the derivations above is the following

$$\xi_1 = k(x) = d \tag{5.15a}$$

$$\xi_2 = L_f k(x) = v\cos(\gamma) \tag{5.15b}$$

The time derivatives of these normal form states with the input in equation (5.14) becomes

$$\dot{\xi}_1 = v\cos(\gamma) \tag{5.16a}$$

$$\begin{aligned}\dot{\xi}_2 &= -v\dot{\gamma}\sin(\gamma)\\ &= -v\sin(\gamma)\left[u - \frac{1}{d}v\sin(\gamma)\right]\\ &= -v\sin(\gamma)\left[-\frac{w}{v\sin(\gamma)} + \cancel{\frac{1}{d}v\sin(\gamma)} - \cancel{\frac{1}{d}v\sin(\gamma)}\right]\\ &= w\end{aligned} \tag{5.16b}$$

The new transformed dynamics of the system becomes

$$\dot{\xi}_1 = \xi_2 \tag{5.17a}$$

$$\dot{\xi}_2 = w \tag{5.17b}$$

Note that the desired radius of the circling behavior is $d = d_0$. Therefore, we will shift the first state as $\xi_1 = d - d_0$. Now, the stabilizing controller input w can be designed by many linear system controller design methods. Here we will just derive the sufficient conditions on the input function parameters and present some results.

The jacobian matrix of the transformed model becomes

$$J = \begin{bmatrix} 0 & 1 \\ \frac{\partial w}{\partial \xi_1} & \frac{\partial w}{\partial \xi_2} \end{bmatrix} \tag{5.18}$$

The characteristic polynomial of the Jacobian matrix is

$$\begin{aligned}\begin{vmatrix} \lambda & -1 \\ -\frac{\partial w}{\partial \xi_1} & \lambda - \frac{\partial w}{\partial \xi_2} \end{vmatrix} &= \lambda\left(\lambda - \frac{\partial w}{\partial \xi_2}\right) - \frac{\partial w}{\partial \xi_1}\\ &= \lambda^2 - \frac{\partial w}{\partial \xi_2}\lambda - \frac{\partial w}{\partial \xi_1} = 0\end{aligned} \tag{5.19}$$

Then the eigenvalues are

$$\lambda_{1,2} = \frac{\frac{\partial w}{\partial \xi_2} \pm \sqrt{\left[\frac{\partial w}{\partial \xi_2}\right]^2 + 4\frac{\partial w}{\partial \xi_1}}}{2} \tag{5.20}$$

For a stable system

$$\frac{\partial w}{\partial \xi_2} < 0 \ \& \ \frac{\partial w}{\partial \xi_1} < 0 \tag{5.21}$$

For an underdamped response

$$\left[\frac{\partial w}{\partial \xi_2}\right]^2 + 4\frac{\partial w}{\partial \xi_1} < 0 \Rightarrow$$
$$\frac{\partial w}{\partial \xi_1} < -\frac{\left[\frac{\partial w}{\partial \xi_2}\right]^2}{4} \tag{5.22}$$

For an overdamped response

$$0 < \left[\frac{\partial w}{\partial \xi_2}\right]^2 + 4\frac{\partial w}{\partial \xi_1} < \left[\frac{\partial w}{\partial \xi_2}\right]^2 \Rightarrow$$
$$-\frac{\left[\frac{\partial w}{\partial \xi_2}\right]^2}{4} < \frac{\partial w}{\partial \xi_1} < 0 \tag{5.23}$$

Let w be a linear state feedback

$$w = K_1\xi_1 + K_2\xi_2 \tag{5.24}$$

where $K_1 = \frac{\partial w}{\partial \xi_1}$ and $K_2 = \frac{\partial w}{\partial \xi_2}$. Now selecting the parameters K_1 and K_2 according to the conditions above one can achieve circling behaviors.

Note that, in equation (5.14) in order for the feedback linearizing controller to be well defined, it is required that $\gamma = \theta - \psi = k\pi$, $k = 0, 1, 2, ...$ to be avoided. Physically this corresponds to the case in which the robot is heading towards the target or heading in the opposite direction along the $\overrightarrow{d}$ vector in Figure 5.2. Depending on the initial conditions, disturbances, unmodeled dynamics, controller parameters, and the value of the constant speed v the robot may converge to the region where the controller is undefined (which is not desirable). In Figure 5.4 a sample phase plane of the system in (5.7) with linearizing input in (5.14) is shown. The parameters for the phase plane are $v = 20$, $K_1 = -2$, $K_2 = -1$, and $d_0 = 20$. As seen in the figure in some neighborhood of the equilibrium $d = d_0$ and $\gamma = \mp\pi/2$ the states are converging to the equilibrium. However, when the robot is far away form the target (i.e. higher d values), the states converge to the undefined region of the controller. Therefore, special care must be taken to avoid the second situation. If we examine the linearized system (equation (5.17)) we can derive the exponential time functions of the normal states as

$$\xi(t) = C_1V_1e^{\lambda_1 t} + C_2V_2e^{\lambda_2 t} \tag{5.25}$$

where λ_i and V_i ($i = 1, 2$) are the eigenvalues and eigenvectors, respectively. The constants C_i are functions of the initial conditions. Using these time dependent equations one can examine the transient characteristics (damping ratio, maximum overshoot, rise time, settling

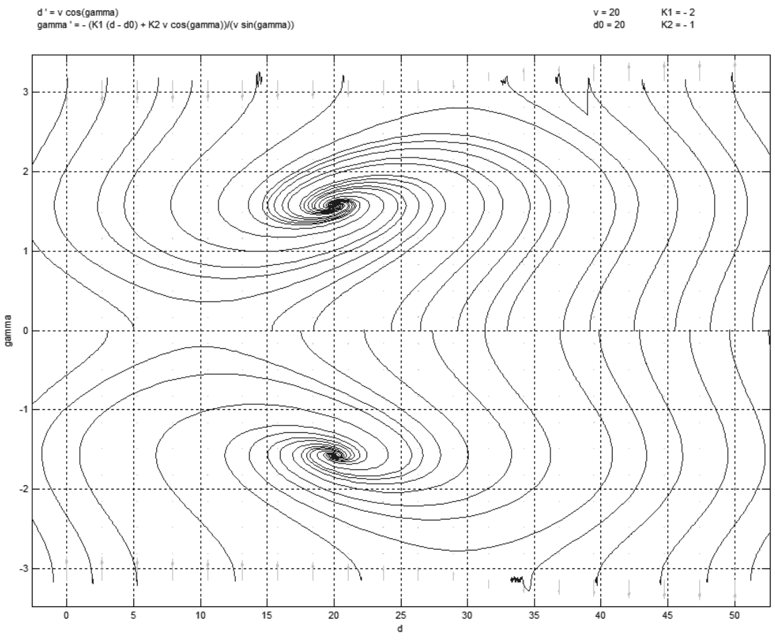


Figure 5.4: A sample phase plane of Circling-SISO system.

time, peak time etc.) of the system and consider the selection of appropriate constant speed, and controller parameters in the design of the controller. For example, at least the conditions $\xi_1 = d - d_0 > -d_0$ and $\gamma = \arccos(\xi_2/v) \neq \mp\pi$ should be satisfied by the equation (5.25). To find more specific relations between these conditions and the equations one should consider finding the maximum and minimum values of ξ_1 and ξ_2 by taking the time derivative of (5.25). By solving the resulting implicit equations one may adjust the controller parameters to design a convergent circling behavior.

5.3.2 Circling - SISO - Stationary Target - Simulation Results

In this section we will present the results of the developed controller in the previous section. As mentioned above we may select underdamped or overdamped controller parameters. The first example is the overdamped circling behavior. For this behavior we select the parameters as $K_2 = -4 < 0$ and $-|K_2|^2/4 < K_1 = -2 < 0$. In the upper part of the Figure 5.5 the change of the distance between robot and target (d) is presented. As seen the robot approaches the target with an overdamped response and stays steady at $d = d_0 = 20[units]$. The lower part of the figure shows the change of the angle difference γ. This variable reaches 90^o which means the robot is rotating around the target. The path of the robot can be followed in the Figure 5.6.

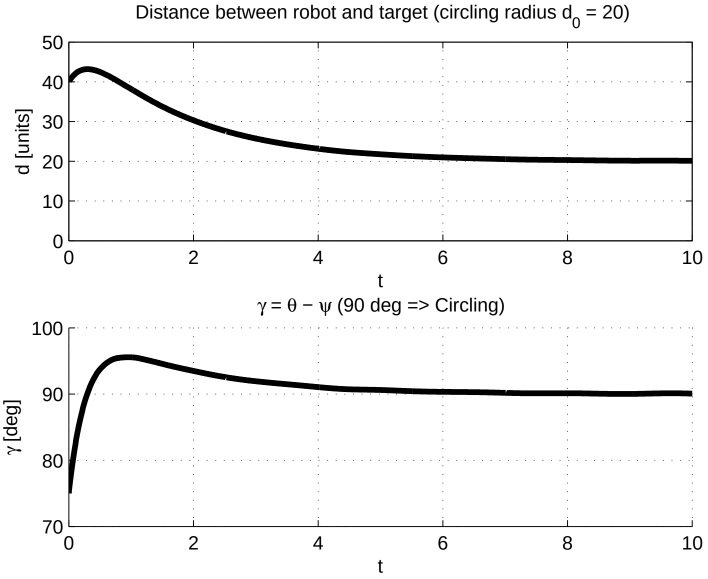


Figure 5.5: The change of the distance between robot and target, and the angle difference γ with respect to time. Overdamped response for circling behavior.

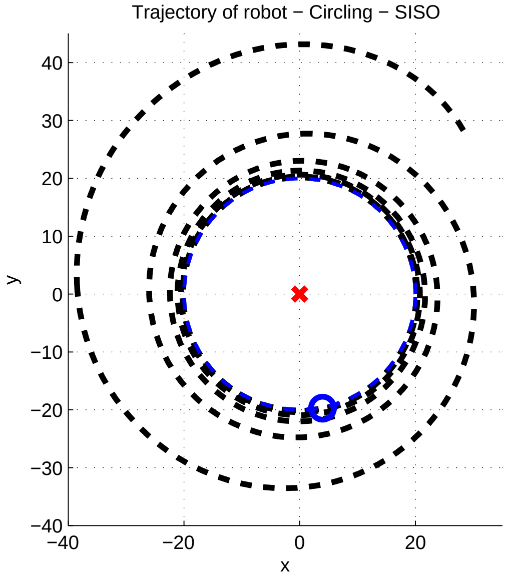


Figure 5.6: The path of the robot. Overdamped response for circling behavior.

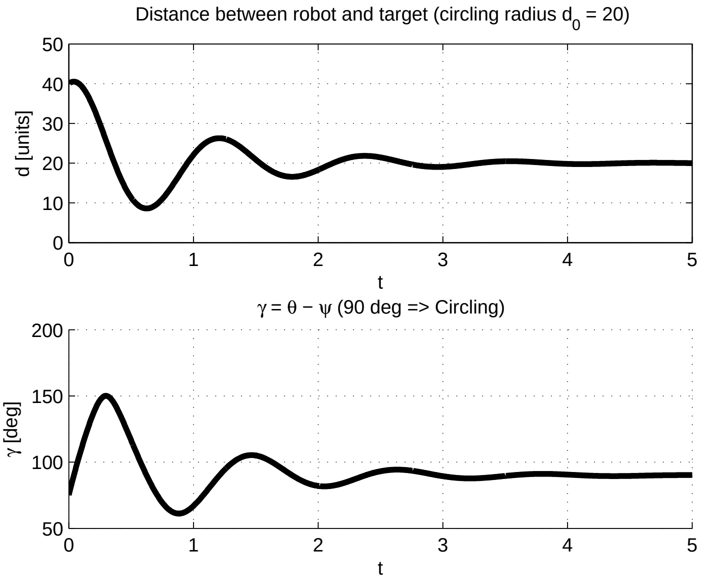


Figure 5.7: The change of the distance between robot and target, and the angle difference γ with respect to time. Underdamped response for circling behavior.

The underdamped response is obtained by the controller parameters $K_2 = -2 < 0$ and $K_1 = -30 < -|K_2|^2/4$. The upper part of Figure 5.7 shows the change of d with respect to time. As seen the response is underdamped. The lower part of the figure presents the underdamped response of the state γ. The path of the robot can be followed in the Figure 5.8.

5.3.3 Circling - MIMO - Stationary Target Case

In this section we will develop the feedback linearization of circling dynamics by taking speed v and angular velocity $\dot{\theta}$ as the inputs. Lets call $v = u_1$ and $\dot{\theta} = u_2$. Then the system dynamics becomes

$$\dot{d} = u_1 \cos(\theta - \psi) \tag{5.26a}$$

$$\dot{\psi} = \frac{1}{d} u_1 \sin(\theta - \psi) \tag{5.26b}$$

$$\dot{\theta} = u_2 \tag{5.26c}$$

At the circling behavior, the vehicle follows a circular path with radius $d = d_0$. Then the equilibrium for the distance variable becomes

$$\dot{d} = 0 \Rightarrow \theta - \psi = (2k+1)\pi/2 \;\; \& \;\; d = d_0 \tag{5.27}$$

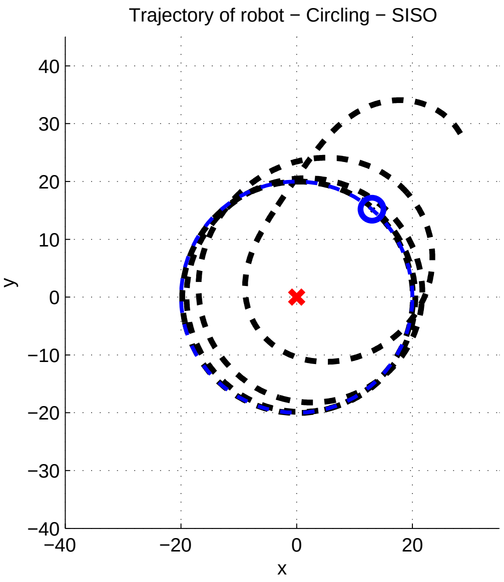


Figure 5.8: The path of the robot. Underdamped response for circling behavior.

The angular velocities $\dot{\psi}$ and $\dot{\theta}$ will be equal at a reference speed (α) while the vehicle is in circling behavior. However, the derivations show that the static feedback linearization is not possible for this system. Therefore, we will use the dynamic feedback linearization technique and instead of using $\dot{\psi} = \dot{\theta} = u_2 = \alpha$ we will state that the angular acceleration $\ddot{\psi}$ is zero at the circling dynamics. The equilibrium set for the circling dynamics is

$$d = d_0 \tag{5.28a}$$

$$\dot{\psi} = u_2 = \alpha \Rightarrow \ddot{\psi} = 0 \tag{5.28b}$$

Since the $\ddot{\psi}$ is not present at current system dynamics we add the integrators for the input u_1 and state ψ as

$$u_1 = z_1 \tag{5.29a}$$

$$\dot{z}_1 = w_1 \tag{5.29b}$$

$$\dot{\psi} = z_2 \tag{5.29c}$$

$$\dot{z}_2 = \frac{\dot{d}}{d^2} z_1 \sin(\theta - \psi) + \frac{1}{d} w_1 \sin(\theta - \psi) + \frac{1}{d} z_1 (\dot{\theta} - \dot{\psi}) \cos(\theta - \psi) \tag{5.29d}$$

For the convenience let us call $u_2 = w_2$. Then the system dynamics becomes

$$\begin{aligned}
\dot{d} &= z_1 \cos(\theta - \psi) && (5.30a)\\
\dot{z}_1 &= w_1 && (5.30b)\\
\dot{\psi} &= z_2 && (5.30c)\\
\dot{z}_2 &= -\frac{{z_1}^2}{d^2} \cos(\theta - \psi) \sin(\theta - \psi) + \frac{1}{d} w_1 \sin(\theta - \psi) \\
&\quad + \frac{1}{d} z_1 w_2 \cos(\theta - \psi) - \frac{1}{d} z_1 z_2 \cos(\theta - \psi) && (5.30d)\\
\dot{\theta} &= w_2 && (5.30e)
\end{aligned}$$

The equations in (5.30a)-(5.30e) may be represented in matrix form as in equation (5.31).

$$\begin{bmatrix} \dot{d} \\ \dot{z}_1 \\ \dot{\psi} \\ \dot{z}_2 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} z_1 \cos(\theta - \psi) \\ 0 \\ z_2 \\ -\frac{{z_1}^2}{d^2} \cos(\theta - \psi) \sin(\theta - \psi) - \frac{1}{d} z_1 z_2 \cos(\theta - \psi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{1}{d} \sin(\theta - \psi) \\ 0 \end{bmatrix} w_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{d} z_1 \cos(\theta - \psi) \\ 1 \end{bmatrix} w_2 \quad (5.31)$$

Defining functions of x we may write

$$\dot{x} = f(x) + g_1(x) w_1 + g_2(x) w_2 \quad (5.32)$$

And the outputs are selected as

$$y_1 = k_1(x) = d \quad (5.33)$$

$$y_2 = k_2(x) = z_2 \quad (5.34)$$

so that in the resulting linearized system the state d will be forced to converge to d_0 and the state $\dot{\psi} = z_2$ will be forced to converge to a constant angular velocity α (α is positive for CCW and negative for CW rotations).

Now, we examine the first derivatives of the outputs

$$\begin{aligned}
\dot{y}_1 &= \dot{d} = z_1\cos(\theta-\psi) && \text{(5.35a)}\\
\dot{y}_2 &= \dot{z}_2 = -\frac{z_1{}^2}{d^2}\cos(\theta-\psi)\sin(\theta-\psi) + \frac{1}{d}w_1\sin(\theta-\psi) + \frac{1}{d}z_1w_2\cos(\theta-\psi) - \frac{1}{d}z_1z_2\cos(\theta-\psi) && \text{(5.35b)}
\end{aligned}$$

in matrix form

$$\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\end{bmatrix} = T_1(x) + A_1\begin{bmatrix}w_1\\ w_2\end{bmatrix}$$

where

$$T_1(x) = \begin{bmatrix} z_1\cos(\theta-\psi)\\ -\frac{z_1{}^2}{d^2}\cos(\theta-\psi)\sin(\theta-\psi) - \frac{1}{d}z_1z_2\cos(\theta-\psi)\end{bmatrix} \tag{5.36}$$

$$A_1 = \begin{bmatrix} 0 & 0\\ \frac{1}{d}\sin(\theta-\psi) & \frac{1}{d}z_1\cos(\theta-\psi)\end{bmatrix} \tag{5.37}$$

For linearization of the system, the coefficient matrix A_1 should have full rank. However, it has rank 1. Note that, the inputs do not appear in the first input's time derivative. Therefore, we take the second time derivative of the first input y_1

$$\begin{aligned}
\ddot{y}_1 &= \dot{z}_1\cos(\theta-\psi) - z_1(\dot{\theta}-\dot{\psi})\sin(\theta-\psi)\\
&= w_1\cos(\theta-\psi) - z_1w_2\sin(\theta-\psi) + z_1z_2\sin(\theta-\psi) && \text{(5.38)}
\end{aligned}$$

then

$$\begin{bmatrix}\ddot{y}_1\\ \dot{y}_2\end{bmatrix} = T_2(x) + A_2\begin{bmatrix}w_1\\ w_2\end{bmatrix} \tag{5.39}$$

where

$$T_2(x) = \begin{bmatrix} z_1z_2\sin(\theta-\psi)\\ -\frac{z_1{}^2}{d^2}\cos(\theta-\psi)\sin(\theta-\psi) - \frac{1}{d}z_1z_2\cos(\theta-\psi)\end{bmatrix} \tag{5.40}$$

$$A_2 = \begin{bmatrix} \cos(\theta-\psi) & -z_1\sin(\theta-\psi)\\ \frac{1}{d}\sin(\theta-\psi) & \frac{1}{d}z_1\cos(\theta-\psi)\end{bmatrix} \tag{5.41}$$

The determinant of A_2 is $Det(A_2) = z_1/d$. At the steady state $z_1 = v \neq 0$ and $d = d_0 \neq 0$, $Det(A_2) = v/d_0 \neq 0$. Therefore, matrix A_2 has full rank at steady state. Note that, matrix A_2

has full rank during transient response if $z_1 \neq 0$. Now, we can derive the normal states as

$$\begin{aligned} \xi_1^1 &= k_1(x) = d && \text{(5.42a)} \\ \xi_2^1 &= L_f k_1(x) && \text{(5.42b)} \\ \xi_1^2 &= k_2(x) = z_2 && \text{(5.42c)} \end{aligned}$$

where

$$\begin{aligned} \xi_2^1 &= L_f k_1(x) = \frac{\partial k_1(x)}{\partial x} f(x) \\ &= [1\ 0\ 0\ 0\ 0] \begin{bmatrix} z_1 \cos(\theta - \psi) \\ 0 \\ z_2 \\ -\frac{z_1{}^2}{d^2} \cos(\theta - \psi) \sin(\theta - \psi) - \frac{1}{d} z_1 z_2 \cos(\theta - \psi) \\ 0 \end{bmatrix} \\ &= z_1 \cos(\theta - \psi) = \dot{\xi}_1^1 \end{aligned} \tag{5.43}$$

and the time derivative of the second state is

$$\dot{\xi}_2^1 = w_1 \cos(\theta - \psi) - z_1 w_2 \sin(\theta - \psi) + z_1 z_2 \sin(\theta - \psi) \tag{5.44}$$

The feedback linearizing inputs are calculated as

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = -A_2{}^{-1} T_2(x) + A_2{}^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{5.45}$$

The calculation is performed by a symbolic equation solver. The results are

$$w_1 = z_1^2 \frac{\cos(\psi - \theta) - \cos(3\psi - 3\theta)}{4d} + p_1 \cos(\psi - \theta) - d p_2 \sin(\psi - \theta) \tag{5.46}$$

$$w_2 = z_2 + \frac{p_2 \cos(\psi - \theta) d + p_1 \sin(\psi - \theta)}{z_1} - z_1 \frac{\sin(\psi - \theta) + \sin(3\psi - 3\theta)}{4d} \tag{5.47}$$

Examining the linearizing inputs we observe that the required info to form these inputs are the agent's own speed z_1, and orientation θ, the orientation ψ and length d of the vector from target to agent, the and the speed of ψ which is z_2. However, note that the inputs w_1 and w_2 consists just the difference $\psi - \theta$ which is the relative orientation of agent. Therefore, the agent does not need the global orientations θ and ψ, instead it requires the relative orientation

$\psi - \theta$ which simplifies the data acquirement during practical applications. Note that, z_2 is the time derivative of ψ and it again depends on just the relative orientation $\psi - \theta$.

Substituting the linearizing inputs we get $\dot{\xi}_2^1 = p_1$ and $\dot{\xi}_1^2 = p_2$.

The normal states are decoupled.

$$\dot{\xi}_1^1 = \xi_2^1 \tag{5.48a}$$

$$\dot{\xi}_2^1 = p_1 \tag{5.48b}$$

$$\dot{\xi}_1^2 = p_2 \tag{5.49}$$

As in the SISO case, we again shift the states as $\xi_1^1 = d - d_0$ and $\xi_1^2 = z_2 - \alpha$. Now, we can use any linear system controller design method to have stable systems at $d = d_0$ and $\dot{\psi} = \alpha$. Consider the inputs p_1 and p_2 are linearly dependent on the normal states. Then the normal state space becomes

$$\begin{bmatrix} \dot{\xi}_1^1 \\ \dot{\xi}_2^1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ K_1^1 & K_2^1 \end{bmatrix} \begin{bmatrix} \xi_1^1 \\ \xi_2^1 \end{bmatrix} \tag{5.50}$$

and

$$\dot{\xi}_1^2 = K_1^2 \xi_1^2 \tag{5.51}$$

Selecting the proper values for the controller parameters, one can obtain stable responses. Note also that in this system once the linearizing controller in (5.46) and (5.47) is employed, two of the states become unobservable. The stability of the unobservable states at the desired behavior or basically the zero dynamics is an issue which needs to be investigated.

Since the relative degree (3) is less than the number of states (5) we will select the new states η_1 and η_2 independent of input i.e. orthogonal to the function $g(x)$.

$$\frac{\partial \eta}{\partial x} g(x) = 0 \Rightarrow \begin{bmatrix} \frac{\partial \eta_1}{\partial d} & \frac{\partial \eta_1}{\partial z_1} & \frac{\partial \eta_1}{\partial \psi} & \frac{\partial \eta_1}{\partial z_2} & \frac{\partial \eta_1}{\partial \theta} \\ \frac{\partial \eta_2}{\partial d} & \frac{\partial \eta_2}{\partial z_1} & \frac{\partial \eta_2}{\partial \psi} & \frac{\partial \eta_2}{\partial z_2} & \frac{\partial \eta_2}{\partial \theta} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ \frac{1}{d}\sin(\theta - \psi) & \frac{1}{d} z_1 \cos(\theta - \psi) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.52}$$

$$\frac{\partial \eta_1}{\partial z_1} + \frac{\partial \eta_1}{\partial z_2} \frac{1}{d} \sin(\theta - \psi) = 0 \tag{5.53}$$

$$\frac{\partial \eta_1}{\partial z_2}\frac{1}{d}z_1\cos(\theta-\psi)+\frac{\partial \eta_2}{\partial \theta}=0 \tag{5.54}$$

which means η_1 should be independent of z_1 and z_2, and η_2 should be independent of θ and z_2. We select $\eta_1 = \theta - \psi - \pi/2$ and $\eta_2 = \psi$. Let us examine the zero dynamics of these variables.

The time derivative of η_1 is

$$\begin{aligned}\dot{\eta_1} &= w_2 - z_2\\ &= \frac{2p_2\cos(\psi-\theta)d^2+2p_1\sin(\psi-\theta)d-\cos(\psi-\theta)\sin(2\psi-2\theta)z_1^2}{2dz_1}\end{aligned} \tag{5.55}$$

We first substitute the equations in (5.48) and (5.50) into this equation. And then we do the following replacement of parameters to obtain equations in reduced system domain

$$d = \xi_1^1 + d_0 \tag{5.56}$$

$$z_2 = \xi_1^2 + \alpha \tag{5.57}$$

$$\theta = \eta_1 + \pi/2 + \eta_2 \tag{5.58}$$

$$\psi = \eta_2 \tag{5.59}$$

Note that in the above equations the equilibrium is set for $\alpha > 0$ and so $\theta - \psi = \pi/2$. However, the other equilibrium $\alpha < 0$ and $\theta - \psi = 3\pi/2$ may also be utilized in the following analysis.

Now we use the zero dynamics of the system at $\xi_1^1 = \xi_2^1 = \xi_1^2 = 0$, then $\dot{\eta_1}$ becomes [1]

$$\dot{\eta_1} = \alpha\sin(\eta_1)^2 - \frac{K_2^1\sin(2\eta_1)}{2} \tag{5.60}$$

Note that the zero dynamics of η_1 is independent of η_2. The stability of η_1 may be examined by linearization around the equilibrium $\eta_1 = 0$. The derivative of $\dot{\eta_1}$ with respect to η_1 at $\eta_1 = 0$ is

$$\frac{\partial \dot{\eta_1}}{\partial \eta_1}|_{\eta_1=0} = -K_2^1 \tag{5.61}$$

This is the eigenvalue of the minimal dynamics of η_1, and since it is definitely negative ($K_2^1 > 0$) the state is locally exponentially stable. Physically this means the system is converging to $\theta = \psi + \pi/2$ that is the CCW rotation of the agent around the target.

The time derivative of second selected state $\eta_2 = \psi$ becomes

$$\begin{aligned}\dot{\eta_2} &= z_2\\ &= \xi_1^2 + \alpha\end{aligned} \tag{5.62}$$

---

[1] The intermediate steps are performed with symbolic equation solver toolbox of Matlab

The zero dynamics becomes

$$\dot{\eta}_2 = \alpha$$

Here note that this state is not converging to a constant, instead it is increasing with a velocity of α but never diverging due to $\psi = mod(\psi, 2\pi)$. The behavior is indeed a tracking problem of the state. The orientation variable ψ tracks the constant speed rotation behavior at the zero dynamics. This is completely consistent with the controlled dynamics of the system. Furthermore, if we select the state as $\eta_i = \theta$ the zero dynamics becomes $\dot{\eta}_i = \dot{\theta} = \dot{\psi} = \alpha$. Note that for this selection of the derived states the problem becomes a tracking of states problem and again it is consistent with the steady state values of original states.

The last state we should examine is the speed of the agent z_1. This is directly related to the orientations and rotational speeds as in the following equation

$$z_1 = \frac{z_2 d}{\sin(\theta - \psi)} \tag{5.63}$$

At the steady state ($z_2 = \alpha, d = d_0, \theta - \psi = \pi/2$) it becomes

$$z_1|_{eq} = \alpha d_0 \tag{5.64}$$

which is again consistent with the controlled dynamics of the system. This state may be selected as one of the η states, however the linearization of the reduced system becomes inconclusive with a zero eigenvalue.

5.3.4 Circling - MIMO - Stationary Target - Simulation Results

In this section we will present the results of the developed controller in the previous section. As in the SISO case we can again select underdamped or overdamped controller parameters. The first example is the overdamped circling behavior. For this behavior we select the parameters as $K_2^1 = -5 < 0$ and $-\left|K_2^1\right|^2/4 < K_1^1 = -4 < 0$ and $K_1^2 = -1 < 0$. In the upper part of the Figure 5.9 the change of the distance between robot and target (d) is presented. As seen the robot approaches the target with an overdamped response and stays steady at $d = d_0 = 20[units]$ and $\alpha = -2$ (α is positive for CCW and negative for CW rotations). The lower part of the figure shows the change of the angle difference γ. This variable reaches 270^o which means the robot is rotating around the target in CW direction. The path of the robot can be followed in the Figure 5.10.

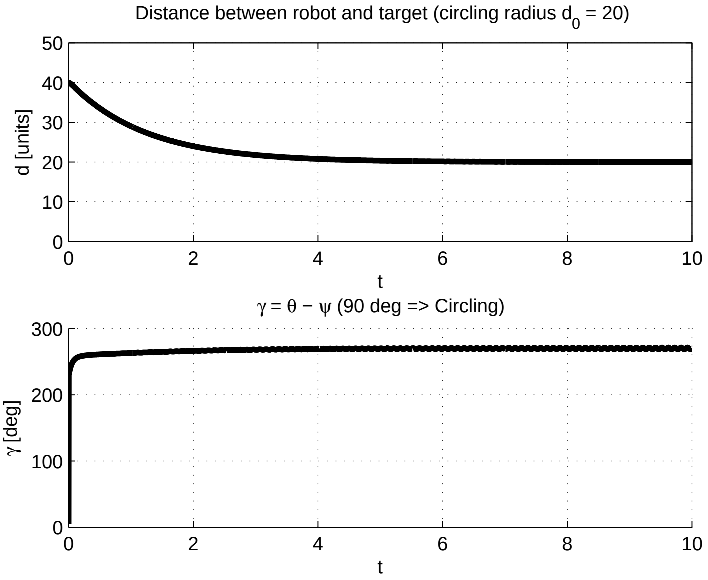


Figure 5.9: The change of the distance between robot and target, and the angle difference γ with respect to time. Overdamped response for circling behavior.

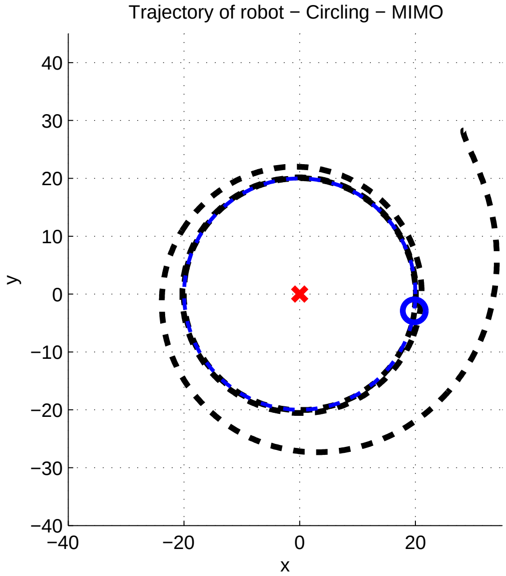


Figure 5.10: The path of the robot. Overdamped response for circling behavior.

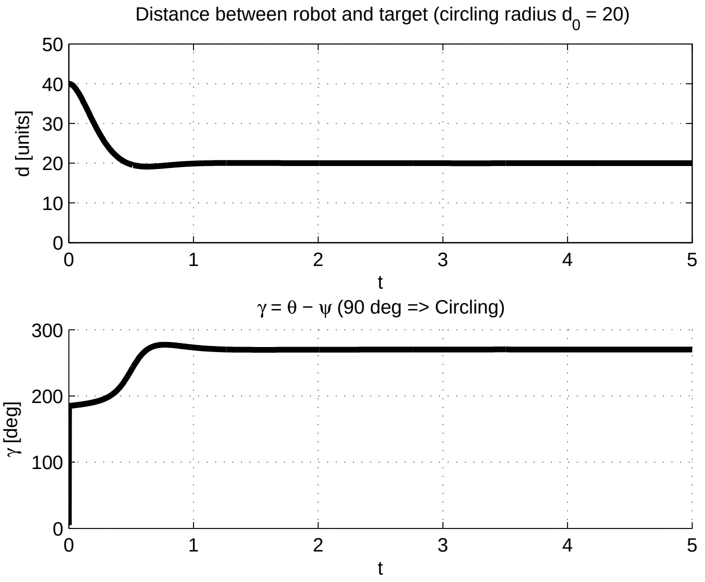


Figure 5.11: The change of the distance between robot and target, and the angle difference γ with respect to time. Underdamped response for circling behavior.

The underdamped response is obtained by the controller parameters $K_2^1 = -2 < 0$ and $K_1^1 = -10 < -\left|K_2^1\right|^2/4$ and $K_1^2 = -1 < 0$. The upper part of Figure 5.11 shows the change of d with respect to time. As seen the response is underdamped. The lower part of the figure presents the underdamped response of the state γ. The path of the robot can be followed in the Figure 5.12.

5.3.5 Circling - MIMO - Non-Stationary Target Case

In this section we will develop the feedback linearization of circling dynamics around a non-stationary target by taking speed v and angular velocity $\dot{\theta}$ as the inputs. Lets call $v = u_1$ and $\dot{\theta} = u_2$. Then the system dynamics becomes

$$\begin{aligned}
\dot{d} &= u_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) && \text{(5.65a)}\\
\dot{\psi} &= \frac{1}{d}\left[u_1 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] && \text{(5.65b)}\\
\dot{\theta} &= u_2 && \text{(5.65c)}
\end{aligned}$$

At the circling behavior, the vehicle follows a circular path with radius $d = d_0$. The angular velocities $\dot{\psi}$ and $\dot{\theta}$ will be equal at a reference speed (α) while the vehicle is in circling behavior. However, the derivations show that the static feedback linearization is not possible for

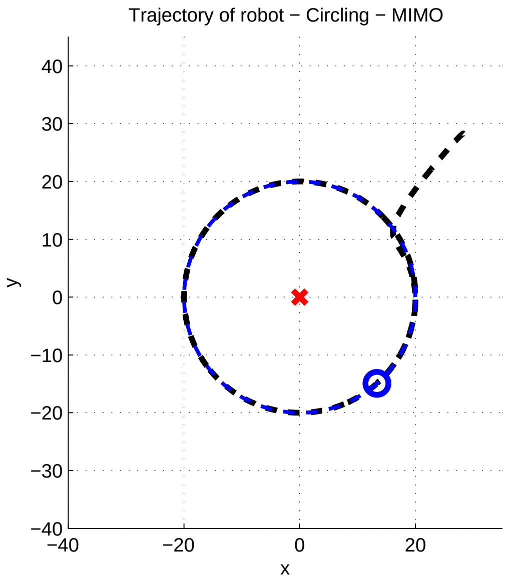


Figure 5.12: The path of the robot. Underdamped response for circling behavior.

this system. Therefore, we will use the dynamic feedback linearization technique and instead of using $\dot{\psi} = \dot{\theta} = u_2 = \alpha$ we will state that the angular acceleration $\ddot{\psi}$ is zero at the circling dynamics. The equilibrium set for the circling dynamics is

$$\begin{aligned} d &= d_0 && (5.66a) \\ \dot{\psi} &= u_2 = \alpha \Rightarrow \ddot{\psi} = 0 && (5.66b) \end{aligned}$$

Since the $\ddot{\psi}$ is not present at current system dynamics we add the integrators for the input u_1 and state ψ as

$$\begin{aligned} u_1 &= z_1 && (5.67a) \\ \dot{z}_1 &= w_1 && (5.67b) \\ \dot{\psi} &= z_2 && (5.67c) \\ \dot{z}_2 &= -\frac{\dot{d}}{d^2}\left[z_1 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] \\ &\quad + \frac{1}{d}\left[w_1 \sin(\theta - \psi) + z_1 (w_2 - z_2) \cos(\theta - \psi)\right. \\ &\qquad \left. - a_c \sin(\theta_c - \psi) - v_c(\omega_c - z_2)\cos(\theta_c - \psi)\right] && (5.67d) \end{aligned}$$

where a_c and ω_c are the acceleration and angular velocity of the target, respectively. For convenience let us call $u_2 = w_2$. Then the system dynamics becomes

$$\begin{aligned}
\dot{d} &= z_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) && (5.68a)\\
\dot{z_1} &= w_1 && (5.68b)\\
\dot{\psi} &= z_2 && (5.68c)\\
\dot{z_2} &= -\frac{1}{d^2}\left[z_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi)\right]\left[z_1 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right]\\
&\quad +\frac{1}{d}\left[w_1 \sin(\theta - \psi) + z_1(w_2 - z_2)\cos(\theta - \psi)\right.\\
&\quad \left.-a_c \sin(\theta_c - \psi) - v_c(\omega_c - z_2)\cos(\theta_c - \psi)\right] && (5.68d)\\
\dot{\theta} &= w_2 && (5.68e)
\end{aligned}$$

The equations in (5.68a)-(5.68e) may be represented in matrix form as in equation (5.69).

$$\begin{bmatrix} \dot{d} \\ \dot{z_1} \\ \dot{\psi} \\ \dot{z_2} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} z_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) \\ 0 \\ z_2 \\ f_{14} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{1}{d}\sin(\theta - \psi) \\ 0 \end{bmatrix} w_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{d} z_1 \cos(\theta - \psi) \\ 1 \end{bmatrix} w_2 \qquad (5.69)$$

where

$$\begin{aligned}
f_{14} &= -\frac{1}{d^2}\left[z_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi)\right]\left[z_1 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right]\\
&\quad +\frac{1}{d}\left[-z_1 z_2 \cos(\theta - \psi) - a_c \sin(\theta_c - \psi) - v_c(\omega_c - z_2)\cos(\theta_c - \psi)\right]
\end{aligned} \qquad (5.70a)$$

Defining functions of x instead of the vectors we write

$$\dot{x} = f(x) + g_1(x) w_1 + g_2(x) w_2 \qquad (5.71)$$

And the outputs are selected as

$$y_1 = k_1(x) = d \qquad (5.72)$$

$$y_2 = k_2(x) = z_2 \qquad (5.73)$$

so that in the resulting linearized system the state d will be forced to converge to d_0 and the state $\dot{\psi} = z_2$ will be forced to converge to a constant angular velocity α (α is positive for CCW and negative for CW rotations).

Now, we examine the first derivatives of the outputs

$$\begin{aligned}
\dot{y}_1 &= \dot{d} = z_1\cos(\theta-\psi) - v_c\cos(\theta_c-\psi) && (5.74a)\\
\dot{y}_2 &= -\frac{1}{d^2}\left[z_1\cos(\theta-\psi) - v_c\cos(\theta_c-\psi)\right]\left[z_1\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right]\\
&\quad +\frac{1}{d}\left[w_1\sin(\theta-\psi) + z_1(w_2-z_2)\cos(\theta-\psi)\right.\\
&\quad \left.-a_c\sin(\theta_c-\psi) - v_c(\omega_c-z_2)\cos(\theta_c-\psi)\right] && (5.74b)
\end{aligned}$$

in matrix form

$$\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\end{bmatrix} = T_1(x) + A_1\begin{bmatrix}w_1\\ w_2\end{bmatrix}$$

where

$$\begin{aligned}
T_1(x) &= \begin{bmatrix} z_1\cos(\theta-\psi) - v_c\cos(\theta_c-\psi)\\ f_{14}\end{bmatrix} && (5.75)\\
A_1 &= \begin{bmatrix} 0 & 0\\ \frac{1}{d}\sin(\theta-\psi) & \frac{1}{d}z_1\cos(\theta-\psi)\end{bmatrix} && (5.76)
\end{aligned}$$

For linearization of the system, the coefficient matrix A_1 should have full rank. However, it has rank 1. Note that, the inputs do not appear in the first input's time derivative. Therefore, we take the second time derivative of the first input y_1

$$\begin{aligned}
\ddot{y}_1 &= \dot{z}_1\cos(\theta-\psi) - z_1(\dot{\theta}-\dot{\psi})\sin(\theta-\psi) - a_c\cos(\theta_c-\psi) + v_c(\dot{\theta}_c-\dot{\psi})\sin(\theta_c-\psi)\\
&= w_1\cos(\theta-\psi) - z_1w_2\sin(\theta-\psi) + z_1z_2\sin(\theta-\psi)\\
&\quad -a_c\cos(\theta_c-\psi) + v_c\omega_c\sin(\theta_c-\psi) - v_cz_2\sin(\theta_c-\psi) && (5.77)
\end{aligned}$$

then

$$\begin{bmatrix}\ddot{y}_1\\ \dot{y}_2\end{bmatrix} = T_2(x) + A_2\begin{bmatrix}w_1\\ w_2\end{bmatrix} \qquad (5.78)$$

where

$$\begin{aligned}
T_2(x) &= \begin{bmatrix} z_1z_2\sin(\theta-\psi) - a_c\cos(\theta_c-\psi) + v_c\omega_c\sin(\theta_c-\psi) - v_cz_2\sin(\theta_c-\psi)\\ f_{14}\end{bmatrix} && (5.79)\\
A_2 &= \begin{bmatrix} \cos(\theta-\psi) & -z_1\sin(\theta-\psi)\\ \frac{1}{d}\sin(\theta-\psi) & \frac{1}{d}z_1\cos(\theta-\psi)\end{bmatrix} && (5.80)
\end{aligned}$$

The determinant of A_2 is $Det(A_2) = z_1/d$. At the steady state $z_1 = v_0 \neq 0$ and $d = d_0 \neq 0$, $Det(A_2) = v_0/d_0 \neq 0$. Therefore, matrix A_2 has full rank at steady state and at transient response as long as $z_1 \neq 0$. Now, we can derive the normal states as

$$\xi_1^1 = k_1(x) = d \tag{5.81a}$$

$$\xi_2^1 = L_f k_1(x) \tag{5.81b}$$

$$\xi_1^2 = k_2(x) = z_2 \tag{5.81c}$$

where

$$\begin{aligned}
\xi_2^1 &= L_f k_1(x) = \frac{\partial k_1(x)}{\partial x} f(x) \\
&= [1\ 0\ 0\ 0\ 0] \begin{bmatrix} z_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) \\ 0 \\ z_2 \\ f_{14} \\ 0 \end{bmatrix} \\
&= z_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) = \dot{\xi}_1^1
\end{aligned} \tag{5.82}$$

The time derivative of the second state is

$$\begin{aligned}
\dot{\xi}_2^1 &= w_1 \cos(\theta - \psi) - z_1 w_2 \sin(\theta - \psi) + z_1 z_2 \sin(\theta - \psi) \\
&\quad - a_c \cos(\theta_c - \psi) + v_c \omega_c \sin(\theta_c - \psi) - v_c z_2 \sin(\theta_c - \psi)
\end{aligned} \tag{5.83}$$

The time derivative of the last state is

$$\begin{aligned}
\dot{\xi}_1^2 &= -\frac{1}{d^2} \left[z_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) \right] \left[z_1 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi) \right] \\
&\quad + \frac{1}{d} \left[w_1 \sin(\theta - \psi) + z_1 (w_2 - z_2) \cos(\theta - \psi) \right. \\
&\quad \left. - a_c \sin(\theta_c - \psi) - v_c (\omega_c - z_2) \cos(\theta_c - \psi) \right]
\end{aligned} \tag{5.84}$$

The feedback linearizing inputs are calculated as

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = -A_2{}^{-1} T_2(x) + A_2{}^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{5.85}$$

The calculation is performed by a symbolic equation solver. The results are

$$\begin{aligned}
w_1 &= \frac{1}{4d} \left[z_1^2 \cos(\psi - \theta) - v_c^2 \cos(3\psi - \theta - 2\theta_c) - z_1^2 \cos(3\psi - 3\theta) \right. \\
&\quad + v_c^2 \cos(\psi + \theta - 2\theta_c) + 4 a_c d \cos(\theta - \theta_c) + 4 d p_1 \cos(\psi - \theta) \\
&\quad - 2 v_c z_1 \cos(\psi - \theta_c) + 2 v_c z_1 \cos(3\psi - 2\theta - \theta_c) - 4 d^2 p_2 \sin(\psi - \theta) \\
&\quad \left. + 4 d \omega_c v_c \sin(\theta - \theta_c) - 4 d v_c z_2 \sin(\theta - \theta_c) \right]
\end{aligned} \tag{5.86}$$

$$\begin{aligned} w_2 &= \frac{1}{4dz_1}\Big[2v_c z_1 \sin(3\psi - 2\theta - \theta_c) + 4dz_1 z_2 + 2v_c z_1 \sin(\psi - \theta_c) \\ &\quad -v_c^2 \sin(3\psi - \theta - 2\theta_c) - v_c^2 \sin(\psi + \theta - 2\theta_c) - 4a_c d \sin(\theta - \theta_c) \\ &\quad +4dp_1 sin(\psi - \theta) + 4d^2 p_2 \cos(\psi - \theta) + 4d\omega_c v_c \cos(\theta_c - \theta) \\ &\quad -4dv_c z_2 \cos(\theta_c - \theta) - z_1^2 \sin(\psi - \theta) - z_1^2 \sin(3\psi - 3\theta)\Big] \end{aligned} \tag{5.87}$$

Examining the linearizing inputs we observe that the required info to form these inputs are the agent's own speed z_1, and orientation θ, the orientation ψ and length d of the vector from target to agent, and the speed of ψ which is z_2 as in the stationary target case. Additionally, this controller needs the velocity vector (magnitude and orientation) of the dynamic target. All these info is hard to acquire in the practical applications.

Substituting the linearizing inputs we get $\dot{\xi}_2^1 = p_1$ and $\dot{\xi}_1^2 = p_2$

The normal states are decoupled.

$$\dot{\xi}_1^1 = \xi_2^1 \tag{5.88a}$$

$$\dot{\xi}_2^1 = p_1 \tag{5.88b}$$

$$\dot{\xi}_1^2 = p_2 \tag{5.89}$$

As in the SISO case, we again shift the states as $\xi_1^1 = d - d_0$ and $\xi_1^2 = z_2 - \alpha$. Now, we can use any linear system controller design method to have stable systems at $d = d_0$ and $\dot{\psi} = \alpha$. Consider the inputs p_1 and p_2 are linearly dependent on the normal states. Then the normal state space becomes

$$\begin{bmatrix} \dot{\xi}_1^1 \\ \dot{\xi}_2^1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ K_1^1 & K_2^1 \end{bmatrix} \begin{bmatrix} \xi_1^1 \\ \xi_2^1 \end{bmatrix} \tag{5.90}$$

and

$$\dot{\xi}_1^2 = K_1^2 \xi_1^2 \tag{5.91}$$

Selecting the proper values for the controller parameters, one can obtain stable responses.

The minimal dynamics should also be investigated. Since the relative degree (3) is less than the number of states (5) we will select a the new states η_1 and η_2 independent of input i.e.

orthogonal to the function $g(x)$.

$$\frac{\partial \eta}{\partial x} g(x) = 0 \Rightarrow \begin{bmatrix} \frac{\partial \eta_1}{\partial d} & \frac{\partial \eta_1}{\partial z_1} & \frac{\partial \eta_1}{\partial \psi} & \frac{\partial \eta_1}{\partial z_2} & \frac{\partial \eta_1}{\partial \theta} \\ \frac{\partial \eta_2}{\partial d} & \frac{\partial \eta_2}{\partial z_1} & \frac{\partial \eta_2}{\partial \psi} & \frac{\partial \eta_2}{\partial z_2} & \frac{\partial \eta_2}{\partial \theta} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ \frac{1}{d}\sin(\theta - \psi) & \frac{1}{d} z_1 \cos(\theta - \psi) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.92}$$

$$\frac{\partial \eta_1}{\partial z_1} + \frac{\partial \eta_1}{\partial z_2} \frac{1}{d} \sin(\theta - \psi) = 0 \tag{5.93}$$

$$\frac{\partial \eta_1}{\partial z_2} \frac{1}{d} z_1 \cos(\theta - \psi) + \frac{\partial \eta_2}{\partial \theta} = 0 \tag{5.94}$$

which means in addition to ξ_1^1, ξ_2^1, and ξ_1^2, the state η_1 should be independent of z_1 and z_2. Similarly η_2 should be independent of θ and z_2 or it should satisfy the partial differential equation in (5.94) . We select $\eta_1 = \psi$ and $\eta_2 = \frac{z_1 \sin(\theta - \psi)}{d} - z_2$. Let us examine the zero dynamics of these variables.

The time derivative of η_1 is

$$\dot{\eta}_1 = z_2 \tag{5.95}$$

We first substitute the equations in (5.48) and (5.50) into this equation. And then we do the following replacement of parameters to obtain equations in reduced system domain

$$d = \xi_1^1 + d_0 \tag{5.96}$$

$$z_2 = \xi_1^2 + \alpha \tag{5.97}$$

$$\theta = \arccos\left(\frac{\xi_2^1 + v_c \cos(\theta_c - \psi)}{z_1}\right) + \psi \tag{5.98}$$

$$\psi = \eta_1 \tag{5.99}$$

Now we use the zero dynamics of the system at $\xi_1^1 = \xi_2^1 = \xi_1^2 = 0$, then $\dot{\eta}_1$ becomes[2]

$$\dot{\eta}_1|_{eq} = \alpha \tag{5.100}$$

Here note that this state is not converging to a constant, instead it is increasing with a velocity of α but never diverging due to $\psi = mod(\psi, 2\pi)$. The behavior is indeed a tracking problem

---

[2] The intermediate steps are performed with symbolic equation solver toolbox of Matlab, and not represented here due to the space requirements.

of the state. The orientation variable ψ tracks the constant speed rotation behavior at the zero dynamics. This is completely consistent with the controlled dynamics of the system. Opposite to the stationary target case, in the dynamic targeting system θ does not change constantly. The change of θ with respect to time depends on the motion of the target. Furthermore, the velocity of the agent is not tangent to the circle centered at target if target is not stationary. The velocity of the agent converges to a vector consisting of normal and tangential components. The normal component of agent velocity is equal to normal component of the target velocity to make $\dot{d}$ zero (normal direction is along ψ direction). The tangential component of the agent velocity is equal to the tangential component of the target velocity plus the rotational speed ($z_2 d$). Therefore, we can write

$$z_1 \cos(\theta - \psi) = v_c \cos(\theta_c - \psi) \tag{5.101}$$

and

$$z_1 \sin(\theta - \psi) = v_c \sin(\theta_c - \psi) + z_2 d \tag{5.102}$$

If we consider the relation between $\eta_2 = \frac{z_1 \sin(\theta - \psi)}{d} - z_2$ and the equations above we can simply say that η_2 should converge to $v_c \sin(\theta_c - \psi)/d$ at the equilibrium. In the analysis of the zero dynamics of η_2 it is hard to solve for this convergence. Therefore, here we use a simplifying method and examine the dynamics of a new variable say $\bar{\eta}_2 = \eta_2 - v_c \sin(\theta_c - \psi)/d$. The time derivative of this variable is

$$\begin{aligned}
\dot{\bar{\eta}}_2 \;=\; & (a_c \sin(\theta_c - \psi) - w_1 \sin(\theta - \psi) + v_c \cos(\theta_c - \psi)(\omega_c - z_2) \\
& -z_1 \cos(\theta - \psi)(w_2 - z_2))/d - (v_c \cos(\theta_c - \psi) - z_1 \cos(\theta - \psi)) \\
& ((v_c \sin(\theta_c - \psi))/d^2 - (z_1 \sin(\theta - \psi))/d^2) + z_2((v_c \cos(\theta_c - \psi))/d \\
& -(z_1 \cos(\theta - \psi))/d) + ((v_c \cos(\theta_c - \psi) - z_1 \cos(\theta - \psi)) \\
& (v_c \sin(\theta_c - \psi) - z_1 \sin(\theta - \psi)))/d^2 - (a_c \sin(\theta_c - \psi))/d \\
& +(w_1 \sin(\theta - \psi))/d - (\omega_c v_c \cos(\theta_c - \psi))/d + (w_2 z_1 \cos(\theta - \psi))/d
\end{aligned} \tag{5.103}$$

Performing the substitutions done for η_1 again in the above equation we would get

$$\dot{\bar{\eta}}_2 = 0$$

which means η_2 converges to $v_c \sin(\theta_c - \psi)/d$ at the equilibrium. Similar to the stationary target case, here we have two tracking variables. The results are all consistent with the controlled circling behavior of the agents.

5.3.6 Circling - MIMO - Non-Stationary Target - Simulation Results

In this section we will present the results of the developed controller for non-stationary targets. We select an overdamped controller for which the parameters are $K_2^1 = -20 < 0$ and $-\left|K_2^1\right|^2/4 < K_1^1 = -50 < 0$ and $K_1^2 = -10 < 0$. The target is stationary at the beginning of the simulation to show that the controller is successful when the target is stationary, after a while the target starts a complex motion and stops towards the end of the simulation. The target motion is modeled by particle kinematics similar to the agents.

$$\begin{aligned} \dot{x}_c &= v_c \cos(\theta_c) && (5.104a) \\ \dot{y}_c &= v_c \sin(\theta_c) && (5.104b) \\ \dot{v}_c &= a_c && (5.104c) \\ \dot{\theta}_c &= \omega_c && (5.104d) \end{aligned}$$

where for this simulation the acceleration and angular velocity are selected to be partial functions as

$$a_c = \begin{cases} 0 & t \leq 5 \;\&\; t \geq 26 \\ 5\sin(0.3(t-5)) & 5 < t < 26 \end{cases} \qquad (5.105)$$

and

$$\omega_c = \begin{cases} 0 & t \leq 5 \;\&\; t \geq 26 \\ 1.5\sin(2.8(t-5)) & 5 < t < 26 \end{cases} \qquad (5.106)$$

Using the above dynamic target model, the simulation is performed for 30 seconds. The reference inputs are $d_0 = 20$ and $\alpha = -4$ (CW rotation). In Figure 5.13 the change of the distance between robot and target (d) is presented. As seen the robot approaches the target with an overdamped response and stays steady at $d = d_0 = 20[units]$ through the simulation although the target moves between $t = 15s$ and $t = 26s$. The translational velocities of the target and agent are shown in Figure 5.14. The translational speed of the agent is converging to $z_1 = 80[units/s]$ when the target is stationary. The path of the agent can be followed in the Figure 5.15.

For better understanding of the performance of non-stationary target following controller, we added the results of the same simulation for the controller developed in the previous section for stationary targets. The controller parameters are the same for both controllers. The agents

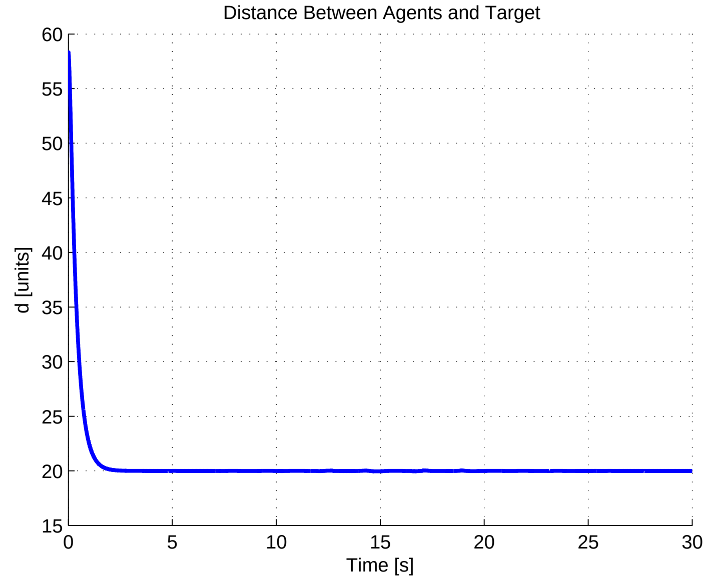


Figure 5.13: The distance between agent and target d with respect to time. Overdamped response for circling behavior with non-stationary target follower controller.

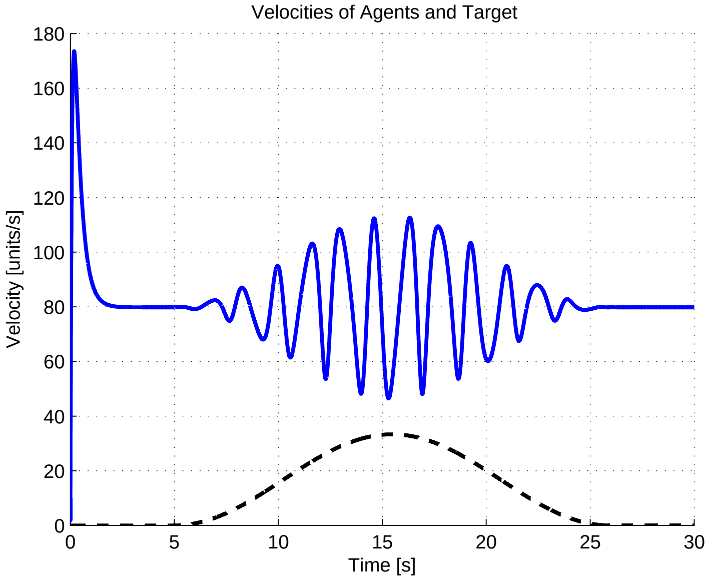


Figure 5.14: Velocities of agent and target. Overdamped response for circling behavior with non-stationary target follower controller.

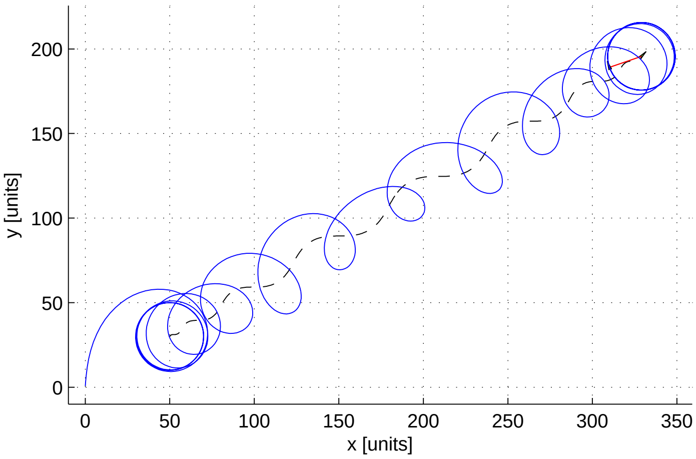


Figure 5.15: Path of the agent and target. Agent path: solid blue, target path: dashed black.

start with the same initial conditions for better comparison. In Figure 5.16 the distances between the agents and the target are plotted for both cases. At the begining of the simulation both agents follow the same path since the target is stationary. When target starts to move the controller for stationary target cannot keep the desired reference value $d = d_0$. When the target stops both agents again converge to the same desired distance value. The velocities of the agents can be compared in Figure 5.17. As seen the non-stationary controller utilizes less speed in inputs. The angular velocities are not presented here. However, we observed that the angular velocity inputs of non-stationary target following controller are less with respect to the stationary target follower one. The paths of both controllers are presented in the Figure 5.18.

5.3.7 Application of Controllers to Multi-agent Systems

In this section, the controllers developed for circling behaviors are applied to multi-agent systems. The developed controllers are designed for a single agent to circle around a specified target. If the objective is to make all agents of a swarm to circle around a specified target, then the controllers are successful in doing so. However, in many multi-agent applications the aim is to force agents to perform a specified task independent of external inputs like leader, target,

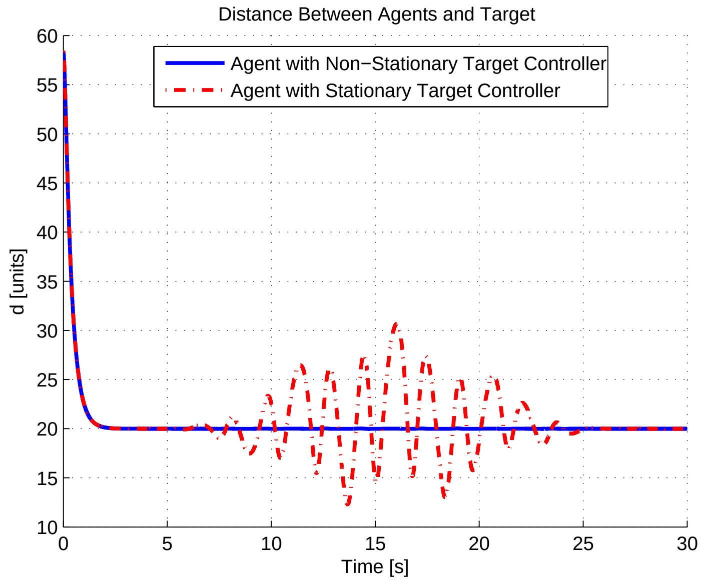


Figure 5.16: The distance between agents and target *d*. The responses of non-stationary and stationary target follower controllers.

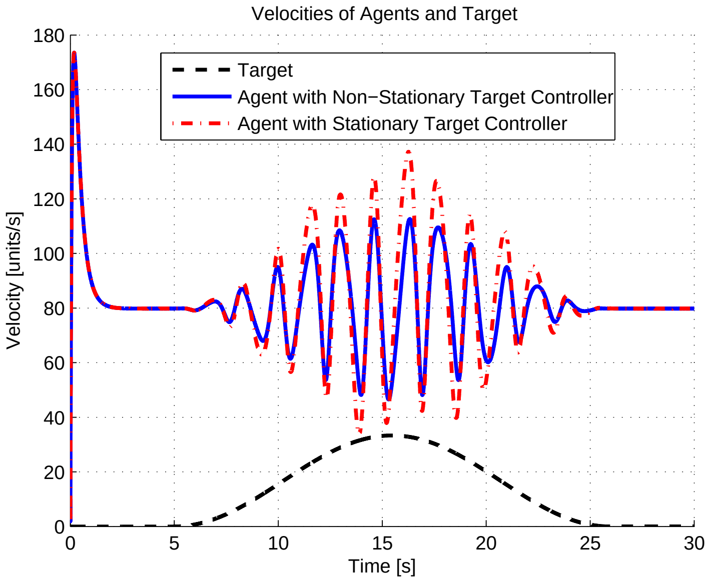


Figure 5.17: Velocities of agents with non-stationary and stationary target follower controllers.

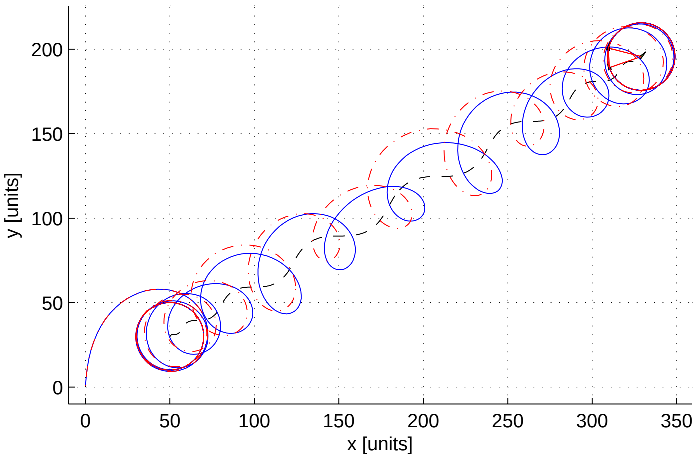


Figure 5.18: Path of the agents and target. Non-stationary controller path: solid blue, stationary controller path: dash-dotted red, target path: dashed black.

centralized controllers etc.. Therefore, in this part of the study we will develop methods for agents to circle around centroid of the swarm. We will start with the SISO controller for a stationary target.

5.3.7.1 Multi-Agent System with SISO - Stationary Target Following Controller

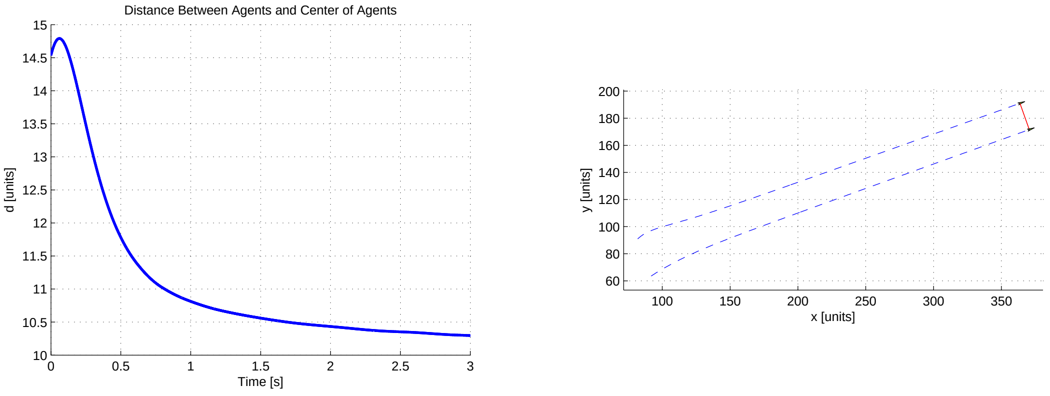
In the analysis of the controller we stated that depending on the initial conditions, disturbances, unmodeled dynamics, controller parameters, and the value of the constant speed v the agents may converge to the undefined region of the controller. Regarding this, in the simulations of multi-agent systems there occurred two conditions that the agents do not converge to circling behavior.

1. The agents are located symmetrically with respect to centroid and they are moving along parallel paths
2. The initial conditions are resulting in opposite rotational velocities

For example for two agents targeting their centroid, the first condition may occur when they are moving along parallel paths. Note that, two agents are always located symmetrically with respect to their centroid. Once the agents converge to this situation they continue their parallel motion with stable dynamics. In Figure 5.19 the results of simulation for parallel motion is presented (Controller parameters are $p = -100\xi_1^1 - 10\xi_1^2$. As seen from the figure the change of distance between agents and the center of agents is same for both agents. They follow symmetric paths and approach each other. The distance is converging to reference input $d = d_0 = 20[units]$. In fact the controller is successful since it achieves the only desired output $d = d_0$ by just regulating the angular velocities.

The simulation results for the second undesired condition is represented for two agents in Figure 5.20. The agents rotate in opposite directions and collide at the end.

The desired circular motion for 2 and 3 number of agents are represented in Figures 5.21 and 5.22. The path and the distance between agents are plotted in these figures.



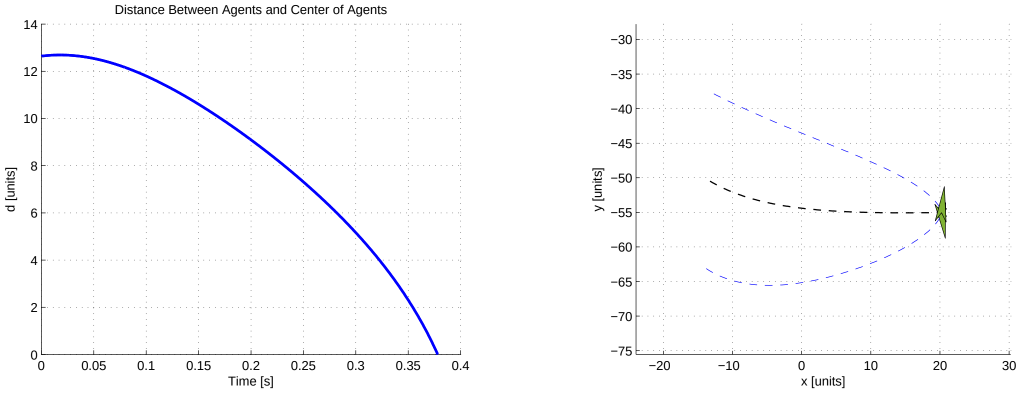
(a) (b)

Figure 5.19: Distance (a) and path (b) of parallel motion of 2 agents with Circling SISO-Stationary Targeting controllers.

For three agents, the second condition is occurring

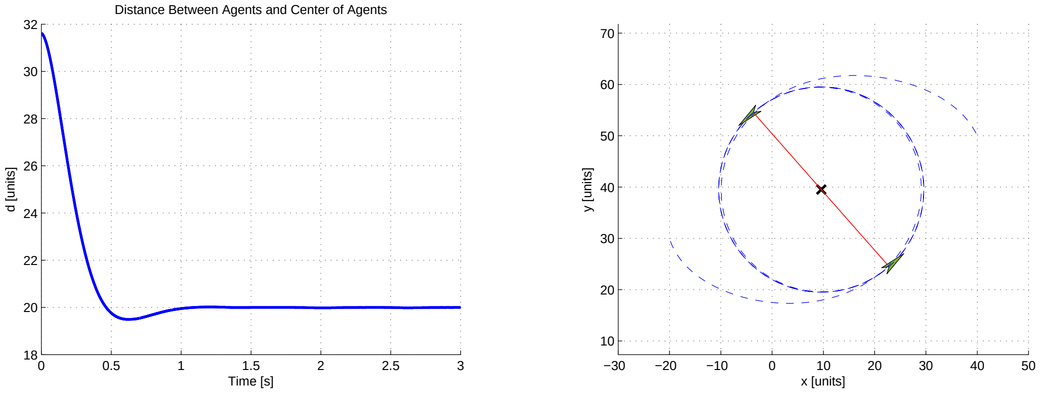
5.3.7.2 Multi-Agent System with MIMO - Stationary Target Following Controller

In this section the MIMO controller designed for circling behavior with translational acceleration input $\dot{z}_1 = w_1$ and angular velocity input $\dot{\theta} = w_2$ for stationary targets is studied for multi-agent systems. Remember that the feedback linearizing inputs are referencing the distance d and angular velocity of the vector from target to the agent $\dot{\psi} = z_2$.



(a) (b)

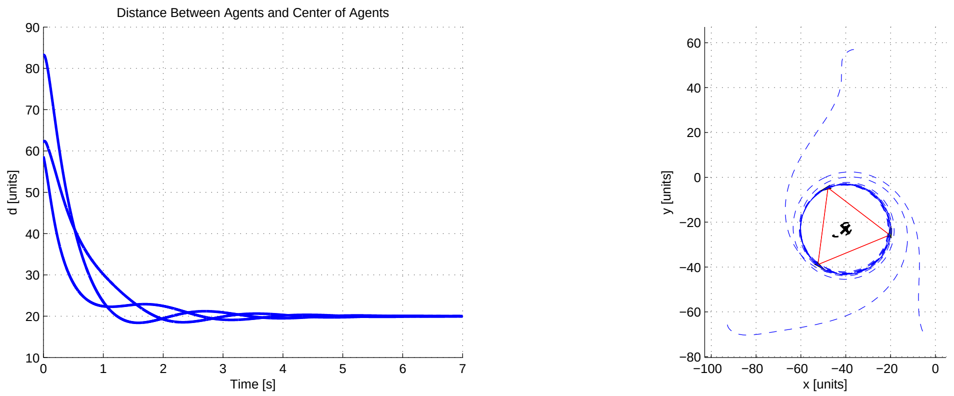
Figure 5.20: Distance (a) and path (b) of coinciding motion of 2 agents with Circling SISO-Stationary Targeting controllers.



(a) (b)

Figure 5.21: Distance (a) and path (b) of regular circling motion of 2 agents with Circling SISO-Stationary Targeting controllers.

In Figures 5.24 and 5.25 the simulation results are presented for 2 agents trying to circle around their centroid. As seen from the figures the agents are not converging to the same circle. In fact the controllers are successful in reaching the reference inputs d and z_2. However, since the centroid is also rotating the agents are converging to the circles which are dependent to this emerging rotation circle of the centroid. Therefore, the MIMO controller for the stationary target is not a good choice for the circling behavior of multi-agent system when the agents are trying to rotate around the centroid. However, if the target of the agents to rotate around is selected as the rotational axis of the vector from centroid to the agent (see Figure 5.23), then the simulations show that the controller achieves the desired behavior.



(a) (b)

Figure 5.22: Distance (a) and path (b) of regular circling motion of 3 agents with Circling SISO-Stationary Targeting controllers.

The first required info to calculate the axis of rotation is the velocity vector of the center of rotation which is simply the mean of the velocity vectors of all agents in the swarm calculated as

$$\vec{v}_c = \frac{1}{M} \sum_{i}^{M} \vec{v}_i \tag{5.107}$$

where M is the number of agents in the swarm.

Each agent should calculate the distance L_i and should use the distance to target value as $d_i + L_i$ instead of just d_i. The geometric relation between the distance L_i and the velocity vectors is

$$L_i = \frac{d_i v_c \sin(\theta_c - \psi)}{v_i \sin(\theta_i - \psi) - v_c \sin(\theta_c - \psi)} \tag{5.108}$$

Note that, this new controller method brings that each agent should acquire the velocity vectors of other agents to calculate the velocity vector of centroid. Remember that the controller is indeed designed for stationary targets. And in the application of the controller for multi-agent systems it was supposed to not require the motion parameters (velocity, acceleration, etc.) of the center of swarm, however the modification in the controller for using distance L_i instead of d_i requires the speed of the center of the swarm. Therefore, in practical applications, in addition to the positions, the speeds of all of the agents should be broadcast to every agent.

In Figures 5.26 and 5.27 the results of this new controller is presented. The same initial conditions and controller parameters ($K_1^1 = -50$, $K_1^2 = -20$, $K_2^1 = -10$) in the previous 2

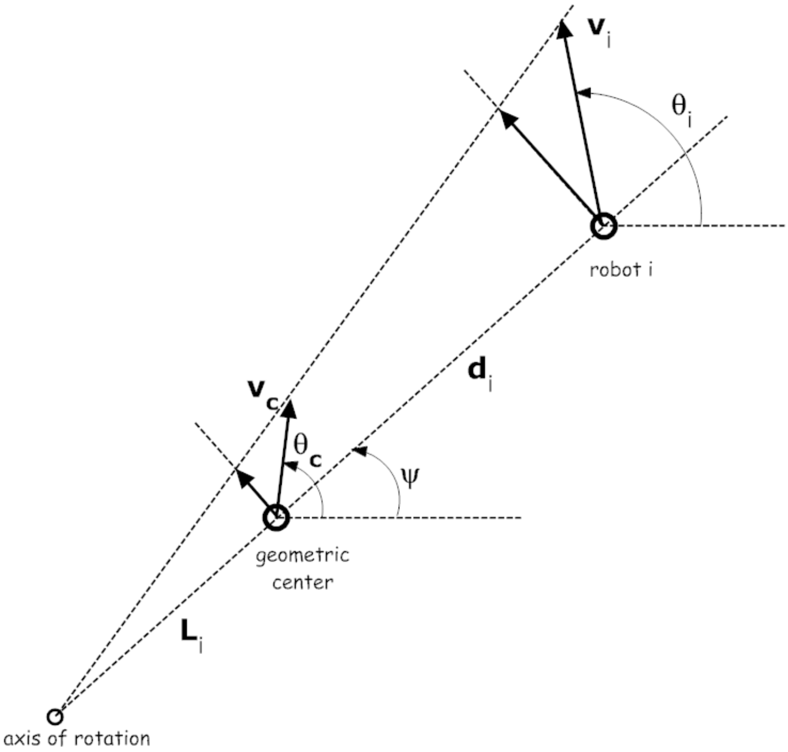
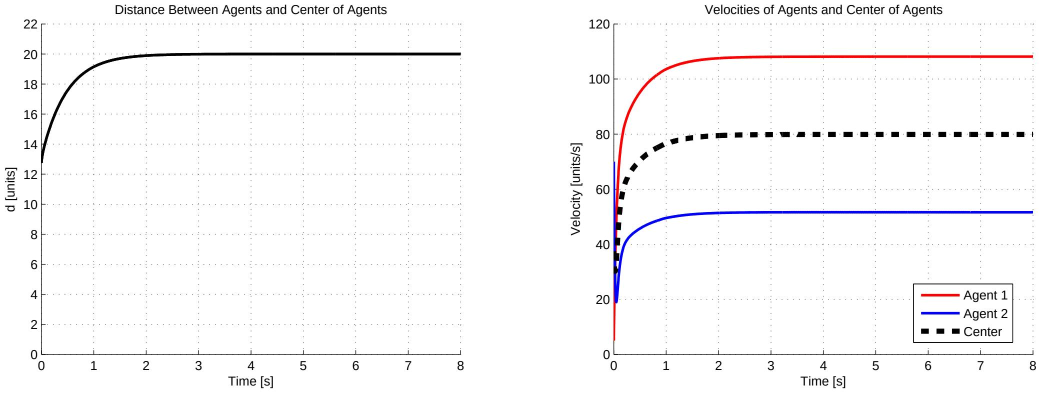


Figure 5.23: Axis of rotation



(a)

(b)

Figure 5.24: Distance (a) and velocities (b) of orbital circling motion of 2 agents with Circling MIMO-Stationary Targeting controllers.

agent simulation is utilized in this one. As seen from the figures the agents are converging to the reference $d = d_0 = 20[units]$ (5.26a) and the speeds of the agents and centroid are converging to $v = 80[units/s]$ (5.26b). The path of the agents and the centroid is plotted in Figure 5.27.

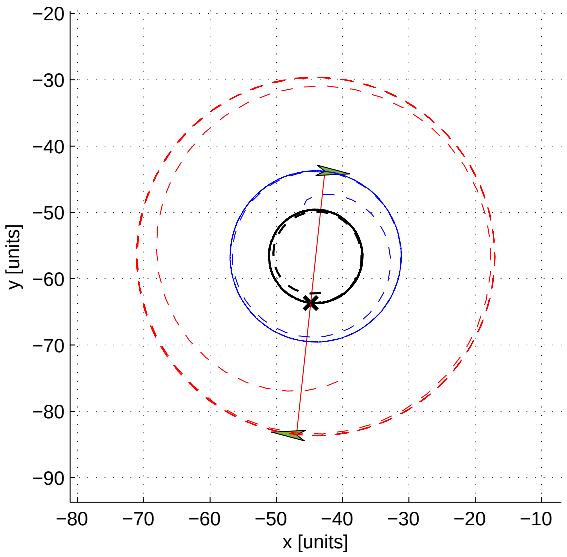
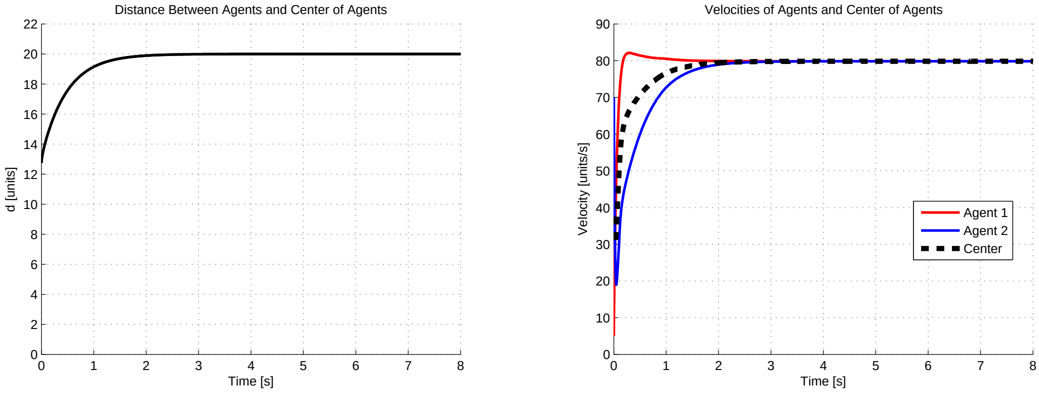


Figure 5.25: Path of the agents and center for orbital circling with MIMO-Stationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black.



(a)

(b)

Figure 5.26: Distance (a) and velocities (b) of regular circling motion of 2 agents with Circling MIMO-Stationary Targeting controllers.

The same controller parameters are utilized in the simulations of 3 and 10 number of agents. The results of the simulations are presented in Figures 5.28, 5.29, 5.30, 5.31. The ones presented and the other simulations show that the selection of the axis of rotation as the target of agents to rotate around yields always convergent dynamics.

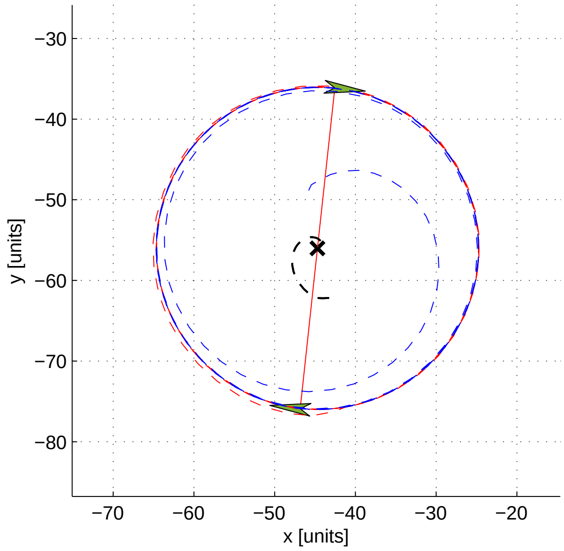
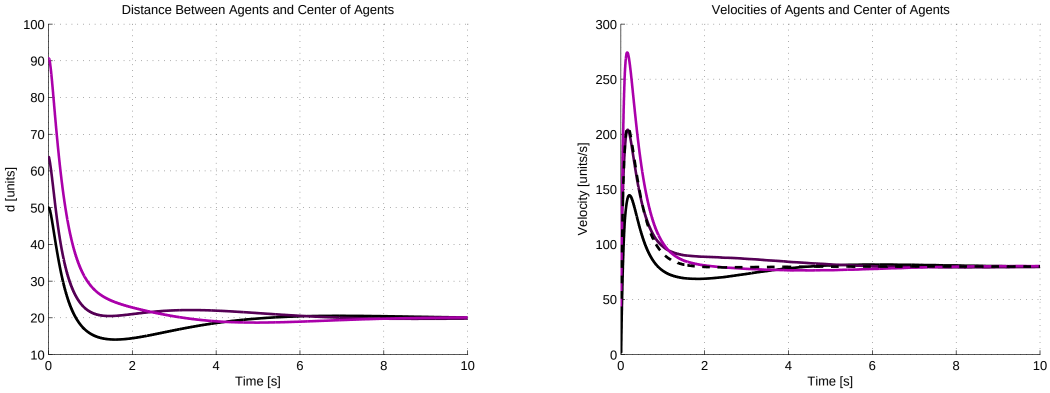


Figure 5.27: Path of the 2 agents and center for regular circling with MIMO-Stationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black.



(a)

(b)

Figure 5.28: Distance (a) and velocities (b) of regular circling motion of 3 agents with Circling MIMO-Stationary Targeting controllers.

5.3.7.3 Multi-Agent System with MIMO - Non-Stationary Target Following Controller

In this section we will develop the MIMO feedback linearization of circling dynamics around the center of the swarm. The center of the swarm is taken to be the target of the agents.

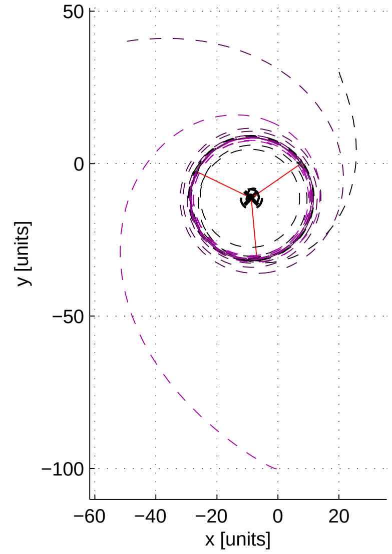
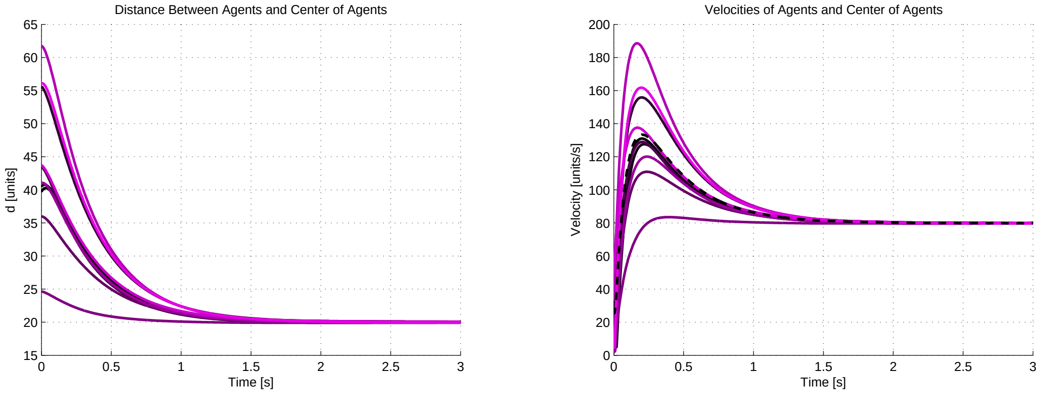


Figure 5.29: Path of the 3 agents and center for regular circling with MIMO-Stationary Targeting controllers. Agent paths: dashed colored, center path: dashed black.



(a)

(b)

Figure 5.30: Distance (a) and velocities (b) of regular circling motion of 10 agents with Circling MIMO-Stationary Targeting controllers.

Therefore, the target dynamics are coupled to the agent dynamics such that

$$\vec{v}_c = \frac{1}{M}\sum_{j=1}^{M}\vec{v}_j \tag{5.109a}$$

$$\vec{a}_c = \frac{1}{M}\sum_{j=1}^{M}\vec{a}_j \tag{5.109b}$$

where v_c and a_c are the velocity and acceleration vectors of the center of swarm, and v_j and a_j are the velocity and acceleration vectors of the jth agent. M is the number of agents in the

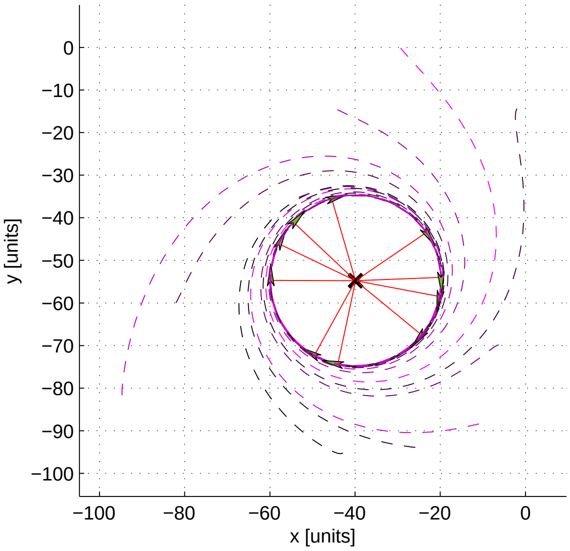


Figure 5.31: Path of the 10 agents and center for regular circling with MIMO-Stationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black.

swarm. And note that the velocity component along the line passing through the target (center of swarm) and the agent itself becomes

$$v_c \cos(\theta_c - \psi_i) = \frac{1}{M} \sum_{j=1}^{M} v_j \cos(\theta_j - \psi_i) \tag{5.110a}$$

$$v_c \sin(\theta_c - \psi_i) = \frac{1}{M} \sum_{j=1}^{M} v_j \sin(\theta_j - \psi_i) \tag{5.110b}$$

The controller that is going to be developed in this section is designed for agents to circle around the stationary center of the swarm. The controller requires the velocity and acceleration vectors of the center of the swarm. However, in the simulations it is observed that there may occur some kind of dead center position if the controllers utilize the exact values of the velocity and acceleration vectors of the center. In Figure 5.32 the simplest case for two agents is shown. Two agents should move in opposite directions at any instance of circling behavior around their geometric center (at the equilibrium). However, the agents are stuck in the motion satisfying the equilibrium of the controlled system as shown in the figure. The steady state values ($\dot{d} = 0$, $z_2 = \alpha d_0$) for the upper right agent is

$$z_1^i \sin(\theta - \psi) - v_c \sin(\theta_c - \psi) = \alpha d_0 \tag{5.111}$$

$$z_1^i \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) = 0 \tag{5.112}$$

and for the lower left agent

$$z_1^j \sin(\theta - \psi - \pi) - v_c \sin(\theta_c - \psi - \pi) = \alpha d_0 \tag{5.113}$$

$$z_1^j \cos(\theta - \psi - \pi) - v_c \cos(\theta_c - \psi - \pi) = 0 \tag{5.114}$$

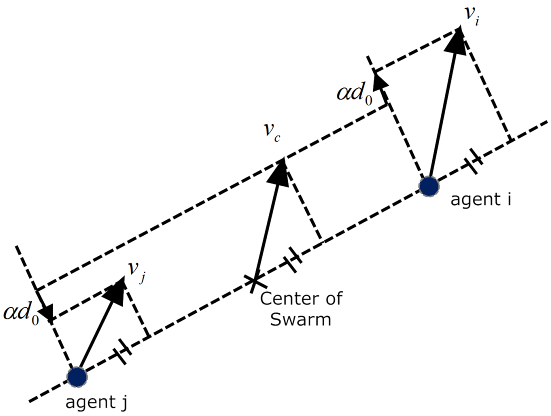


Figure 5.32: Two agents moving with the equilibrium of controller when $K = 1$.

where $v_c = \frac{z_1^i + z_1^j}{2}$ and $\alpha d_0 = \frac{z_1^i - z_1^j}{2} \sin(\theta_c - \psi)$. The controller force agent motions to converge to these equilibriums. This condition is also valid for more than 2 number of agents in the swarm. Therefore, the controller is designed such that it does not utilize the exact value of the center of flock, instead it utilizes some part of the center velocity. The ratio will be called as K and it will satisfy $0 \leq K < 1$.

The controller design will be similar to the one in MIMO-dynamic target case in section 5.3.5. Therefore, we will use the state equation in (5.65) and obtain the following set of equations for each agent i

$$\dot{d}_i = u_1^i \cos(\theta_i - \psi_i) - v_c \cos(\theta_c - \psi_i) \tag{5.115a}$$

$$\dot{\psi}_i = \frac{1}{d_i}\left[u_1^i \sin(\theta_i - \psi_i) - v_c \sin(\theta_c - \psi_i)\right] \tag{5.115b}$$

$$\dot{\theta}_i = u_2^i \tag{5.115c}$$

As in the previous controllers at the circling behavior, the vehicle follows a circular path with radius $d = d_0$ and the angular velocities $\dot{\psi}$ and $\dot{\theta}$ will be equal to a reference speed (α). Again we will use the dynamic feedback linearization technique and state that the angular acceleration $\ddot{\psi}$ is zero at the circling dynamics. The equilibrium set for each agent becomes

$$\begin{aligned} d_i &= d_0 && (5.116a)\\ \dot{\psi}_i &= u_2^i = \alpha \Rightarrow \ddot{\psi}_i = 0 && (5.116b) \end{aligned}$$

We add the integrators for the input u_1 and state ψ as in the previous designs. After the necessary arrangements agent dynamics become

$$\begin{aligned} \dot{d}_i &= z_1^i \cos(\theta_i - \psi_i) - K v_c \cos(\theta_c - \psi_i) && (5.117a)\\ \dot{z}_1^i &= w_1^i && (5.117b)\\ \dot{\psi}_i &= z_2^i && (5.117c)\\ \dot{z}_2^i &= -\frac{1}{d_i^2}\left[z_1^i \cos(\theta_i - \psi_i) - K v_c \cos(\theta_c - \psi_i)\right]\left[z_1^i \sin(\theta_i - \psi_i) - K v_c \sin(\theta_c - \psi_i)\right] \\ &\quad +\frac{1}{d_i}\Big[w_1^i \sin(\theta_i - \psi_i) + z_1^i (w_2^i - z_2^i)\cos(\theta_i - \psi_i) \\ &\quad -K a_c \sin(\theta_c - \psi_i) - K v_c(\omega_c - z_2^i)\cos(\theta_c - \psi_i)\Big] && (5.117d)\\ \dot{\theta}_i &= w_2^i && (5.117e) \end{aligned}$$

If we extract the velocity components (Eq. (5.110)) belonging to agent i in the equation (5.117a) we get

$$\dot{d}_i = z_1^i \cos(\theta_i - \psi_i) - K\frac{1}{M}\left[v_i \cos(\theta_i - \psi_i) + \sum_{j=1, j\neq i}^{M} v_j \cos(\theta_j - \psi_i)\right] \qquad (5.118)$$

Rearranging the terms

$$\dot{d}_i = z_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - K\frac{1}{M} \sum_{j=1, j\neq i}^{M} v_j \cos(\theta_j - \psi_i) \qquad (5.119)$$

and defining

$$\begin{aligned} \vec{v}_c^{\,i} &= \frac{1}{M}\sum_{j=1, j\neq i}^{M} \vec{v}_j && (5.120a)\\ \theta_c^i &= angle(\vec{v}_c^{\,i}) && (5.120b) \end{aligned}$$

which are the mean of the velocity vectors out of ith agent, and its orientation. Note that these variables are independent of the states of ith agent. Let us define $v_c^i = norm(\vec{v}_c^{\,i})$. Substituting these into state equation we may write

$$\dot{d}_i = z_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - K v_c^i \cos(\theta_c^i - \psi_i) \qquad (5.121)$$

If we do the same arrangements for the orientation variable ψ, we get

$$\dot{\psi}_i = \frac{1}{d_i}\left[z_1^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] \tag{5.122}$$

Note that at the equilibrium, the velocity vector of the *i*th agent is opposite to the mean of the rest of the agents.

$$\begin{aligned} v_c^i &= \frac{1}{M}z_1^i = \frac{1}{M}\alpha d_0 && (5.123)\\ \theta_c^i &= \theta_i + \pi && (5.124)\end{aligned}$$

substituting these into state equations the states become $\dot{d} = 0$ and $\dot{\psi}_i = \alpha$.

Utilizing the above relations the agent dynamics become

$$\begin{aligned}
\dot{d}_i &= z_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - Kv_c^i\cos(\theta_c^i-\psi_i) && (5.125a)\\
\dot{z}_1^i &= w_1^i && (5.125b)\\
\dot{\psi}_i &= z_2^i && (5.125c)\\
\dot{z}_2^i &= -\frac{1}{d_i^2}\left[z_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - Kv_c^i\cos(\theta_c^i-\psi_i)\right]\\
&\quad \left[z_1^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right]\\
&\quad +\frac{1}{d_i}\left[w_1^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) + z_1^i\left(1-\frac{K}{M}\right)(w_2^i-z_2^i)\cos(\theta_i-\psi_i)\right.\\
&\quad \left. -Ka_c^i\sin(\theta_c-\psi_i) - Kv_c^i(\omega_c^i-z_2^i)\cos(\theta_c^i-\psi_i)\right] && (5.125d)\\
\dot{\theta}_i &= w_2^i && (5.125e)
\end{aligned}$$

where a_c^i is the time derivative of the speed of *i*th agent, $a_c^i = \frac{dv_c^i}{dt}$ and ω_c^i is the angular velocity of the *i*th agent. These variables can be calculated with the following set of equations

$$\begin{aligned}
ac_x^i &= \frac{1}{M}\sum_{j=1,j\neq i}^{M} w_1^j\cos(\theta_j) - z_1^j w_2^j\sin(\theta_j) && (5.126)\\
ac_y^i &= \frac{1}{M}\sum_{j=1,j\neq i}^{M} w_1^j\sin(\theta_j) + z_1^j w_2^j\cos(\theta_j) && (5.127)
\end{aligned}$$

Here, ac_x^i and ac_y^i are the x and y components of the mean acceleration of the agents different from *i*th agent. Then, the relation of ω_c^i and a_c^i with the above acceleration components can be written as

$$ac_r^i = \sqrt{(ac_x^i)^2 + (ac_y^i)^2} \tag{5.128}$$

$$\psi_c^i = atan2(ac_y, ac_x) \tag{5.129}$$

$$ac_n^i = ac_r^i \sin(\psi_c^i - \theta_c^i) \tag{5.130}$$

$$ac_t^i = ac_r^i \cos(\psi_c^i - \theta_c^i) \tag{5.131}$$

$$\omega_c^i = \frac{ac_n^i}{v_c^i} \tag{5.132}$$

$$a_c^i = ac_t^i \tag{5.133}$$

The equations in (5.125) are represented in matrix form as

$$\begin{aligned}
\begin{bmatrix} \dot{d}_i \\ \dot{z}_1^i \\ \dot{\psi}_i \\ \dot{z}_2^i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} z_1^i\left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - Kv_c^i \cos(\theta_c^i - \psi_i) \\ 0 \\ z_2^i \\ f_{14}^i \\ 0 \end{bmatrix} &+ \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{1}{d_i}\left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) \\ 0 \end{bmatrix} w_1 \\
&+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{d_i} z_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) \\ 1 \end{bmatrix} w_2
\end{aligned} \tag{5.134}$$

where

$$\begin{aligned}
f_{14}^i \quad = \quad & -\frac{1}{d_i^2}\left[z_1^i\left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - Kv_c^i \cos(\theta_c^i - \psi_i)\right] \\
& \left[z_1^i\left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) - Kv_c^i \sin(\theta_c^i - \psi_i)\right] \\
& + \frac{1}{d_i}\left[z_1^i\left(1 - \frac{K}{M}\right)(-z_2^i)\cos(\theta_i - \psi_i)\right. \\
& \left. - Ka_c^i \sin(\theta_c - \psi_i) - Kv_c^i(\omega_c^i - z_2^i)\cos(\theta_c^i - \psi_i)\right]
\end{aligned} \tag{5.135a}$$

Defining functions of x we write

$$\dot{x}_i = f(x_i) + g_1^i(x_i)w_1^i + g_2^i(x_i)w_2^i \tag{5.136}$$

where $x = [d_i, z_1^1, \psi_i, z_2^1, \theta_i]^T$ is the state vector.

For linearization as in Section 5.3.5 the outputs are selected as

$$y_1^i = k_1^i(x) = d_i \tag{5.137}$$

$$y_2^i = k_2^i(x) = z_2^i \tag{5.138}$$

In the resulting linearized system the state d_i will be forced to converge to d_0 and the state $\dot{\psi}_i = z_2$ will be forced to converge to a constant angular velocity α (α is positive for CCW and negative for CW rotations).

Now, we examine the first derivatives of the outputs

$$\begin{aligned}
\dot{y}_1^i &= \dot{d}_i = z_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - Kv_c^i\cos(\theta_c^i-\psi_i) && (5.139a)\\
\dot{y}_2^i &= -\frac{1}{d_i^2}\left[z_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - Kv_c^i\cos(\theta_c^i-\psi_i)\right]\\
&\quad \left[z_1^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right]\\
&\quad +\frac{1}{d_i}\left[w_1^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) + z_1^i\left(1-\frac{K}{M}\right)(w_2^i-z_2^i)\cos(\theta_i-\psi_i)\right.\\
&\quad \left.-Ka_c^i\sin(\theta_c-\psi_i) - Kv_c^i(\omega_c^i-z_2^i)\cos(\theta_c^i-\psi_i)\right] && (5.139b)
\end{aligned}$$

in matrix form

$$\begin{bmatrix}\dot{y}_1^i\\ \dot{y}_2^i\end{bmatrix} = T_1^i(x_i) + A_1^i\begin{bmatrix}w_1^i\\ w_2^i\end{bmatrix}$$

where

$$T_1^i(x_i) = \begin{bmatrix} z_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - Kv_c^i\cos(\theta_c^i-\psi_i)\\ f_{14}^i\end{bmatrix} \tag{5.140}$$

$$A_1^i = \begin{bmatrix} 0 & 0\\ \frac{1}{d_i}\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) & \frac{1}{d_i}z_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i)\end{bmatrix} \tag{5.141}$$

For linearization the coefficient matrix A_1^i should have full rank. However, it has rank 1. Note that, the inputs do not appear in the first input's time derivative. Therefore, we take the second time derivative of the first input y_1^i

$$\begin{aligned}
\ddot{y}_1^i &= \dot{z}_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - z_1^i\left(1-\frac{K}{M}\right)(\dot{\theta}_i-\dot{\psi}_i)\sin(\theta_i-\psi_i) - Ka_c^i\cos(\theta_c^i-\psi_i)\\
&\quad +Kv_c^i(\dot{\theta}_c^i-\dot{\psi}_i)\sin(\theta_c^i-\psi_i)\\
&= w_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - z_1^iw_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) + z_1^iz_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)\\
&\quad -Ka_c^i\cos(\theta_c^i-\psi_i) + Kv_c^i\omega_c^i\sin(\theta_c^i-\psi_i) - Kv_c^iz_2^i\sin(\theta_c^i-\psi_i) && (5.142)
\end{aligned}$$

then

$$\begin{bmatrix} \dot{y}_1^i \\ \dot{y}_2^i \end{bmatrix} = T_2^i(x) + A_2^i \begin{bmatrix} w_1^i \\ w_2^i \end{bmatrix} \tag{5.143}$$

where

$$T_2^i(x) = \begin{bmatrix} \rho_i \\ f_{14}^i \end{bmatrix} \tag{5.144}$$

$$A_2^i = \begin{bmatrix} \left(1 - \frac{K}{M}\right)\cos(\theta - \psi) & -z_1\left(1 - \frac{K}{M}\right)\sin(\theta - \psi) \\ \frac{1}{d_i}\left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) & \frac{1}{d_i} z_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) \end{bmatrix} \tag{5.145}$$

where

$$\begin{aligned} \rho_i = z_1^i z_2^i \left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) - K a_c^i \cos(\theta_c^i - \psi_i) \\ + K v_c^i \omega_c^i \sin(\theta_c^i - \psi_i) - K v_c^i z_2^i \sin(\theta_c^i - \psi_i) \end{aligned} \tag{5.146}$$

The determinant of A_2^i is $Det(A_2^i) = z_1^i (K - M)^2 M^2 d_i$. At the steady state $z_1^i = \alpha d_0 \neq 0$ and $d_i = d_0 \neq 0$, $Det(A_2^i) = \alpha\left(1 - \frac{K}{M}\right)^2 \neq 0$. Therefore, matrix A_2^i has full rank at steady state. Now, we can derive the normal states as

$$\xi_1^{1^i} = k_1^i(x) = d_i \tag{5.147a}$$

$$\xi_2^{1^i} = L_f k_1^i(x) \tag{5.147b}$$

$$\xi_1^{2^i} = k_2^i(x) = z_2^i \tag{5.147c}$$

where

$$\begin{aligned} \xi_2^{1^i} &= L_f k_1^i(x) = \frac{\partial k_1^i(x)}{\partial x_i} f(x_i) \\ &= [1\,0\,0\,0\,0] \begin{bmatrix} z_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - K v_c^i \cos(\theta_c^i - \psi_i) \\ 0 \\ z_2^i \\ f_{14}^i \\ 0 \end{bmatrix} \\ &= z_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - K v_c^i \cos(\theta_c^i - \psi_i) = \dot{\xi}_1^1 \end{aligned} \tag{5.148}$$

The time derivative of the second state is

$$\begin{aligned} \dot{\xi}_2^{1^i} &= w_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - z_1^i \left(1 - \frac{K}{M}\right) w_2^i \sin(\theta_i - \psi_i) + z_1^i z_2^i \left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) \\ &\quad - K a_c^i \cos(\theta_c^i - \psi_i) + K v_c^i \omega_c^i \sin(\theta_c^i - \psi_i) - K v_c^i z_2^i \sin(\theta_c^i - \psi_i) \end{aligned} \tag{5.149}$$

The time derivative of the last state is

$$\begin{aligned}\dot{\xi}_1^{2^i} \;=\;& -\frac{1}{d_i^2}\left[z_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i)-Kv_c^i\cos(\theta_c^i-\psi_i)\right]\\ &\left[z_1\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)-v_c^i\sin(\theta_c^i-\psi_i)\right]\\ &+\frac{1}{d_i}\left[w_1^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)+z_1^i\left(1-\frac{K}{M}\right)(w_2^i-z_2^i)\cos(\theta_i-\psi_i)\right.\\ &\quad\left.-Ka_c^i\sin(\theta_c^i-\psi_i)-v_c^i(\omega_c^i-z_2^i)\cos(\theta_c^i-\psi_i)\right] \qquad (5.150)\end{aligned}$$

The feedback linearizing inputs are calculated as

$$\begin{bmatrix}w_1^i\\ w_2^i\end{bmatrix}=-{A_2^i}^{-1}T_2^i(x_i)+{A_2^i}^{-1}\begin{bmatrix}p_1^i\\ p_2^i\end{bmatrix} \qquad (5.151)$$

The calculation is performed by a symbolic equation solver. The results are

$$\begin{aligned}w_1^i \;=\;& -M\cos(\psi_i-\theta_i)/(K-M)\Big(z_2^i(Kv_c^i\sin(\theta_c^i-\psi_i)+z_1^i\sin(\theta_i-\psi_i)(K/M-1))\\ &+Ka_c^i\cos(\theta_c^i-\psi_i)-K\omega_c^iv_c^i\sin(\theta_c^i-\psi_i)\Big)\\ &-(Mp_1^i\cos(\psi_i-\theta_i))/(K-M)+Md_i\sin(\psi_i-\theta_i)/(K-M)\Big(((Ka_c^i\sin(\theta_c^i-\psi_i)\\ &-z_1^iz_2^i\cos(\theta_i-\psi_i)(K/M-1)+Kv_c^i\cos(\theta_c^i-\psi_i)(\omega_c^i-z_2^i))/d\\ &+((Kv_c^i\cos(\theta_c^i-\psi_i)+z_1^i\cos(\theta_i-\psi_i)(K/M-1))(Kv_c^i\sin(\theta_c^i-\psi_i)\\ &+z_1^i\sin(\theta_i-\psi_i)(K/M-1)))/d^2)\Big)\\ &+(Mdp_2^i\sin(\psi_i-\theta_i))/(K-M) \qquad (5.152)\end{aligned}$$

$$\begin{aligned}w_2^i \;=\;& -M\sin(\psi_i-\theta_i)/(z_1^i(K-M))(z_2^i(Kv_c^i\sin(\theta_c^i-\psi_i)\\ &+z_1^i\sin(\theta_i-\psi_i)(K/M-1))+Ka_c^i\cos(\theta_c^i-\psi_i)-K\omega_c^iv_c^i\sin(\theta_c^i-\psi_i))\\ &-(Mp_1^i\sin(\psi_i-\theta_i))/(z_1^i(K-M))-(Mdp_2^i\cos(\psi_i-\theta_i))/(z_1^i(K-M))\\ &-(Md\cos(\psi_i-\theta_i)((Ka_c^i\sin(\theta_c^i-\psi_i)-z_1^iz_2^i\cos(\theta_i-\psi_i)(K/M-1)\\ &+Kv_c^i\cos(\theta_c^i-\psi_i)(\omega_c^i-z_2^i))/d+((Kv_c^i\cos(\theta_c^i-\psi_i)+z_1^i\cos(\theta_i-\psi_i)(K/M-1))\\ &(Kv_c^i\sin(\theta_c^i-\psi_i)+z_1^i\sin(\theta_i-\psi_i)(K/M-1)))/d^2))/(z_1^i(K-M)) \qquad (5.153)\end{aligned}$$

Examining the linearizing inputs the required info to calculate these inputs are the agent's own speed z_1^i, and relative orientation $\theta_i-\psi_i$, the relative orientation $\theta_c^i-\psi_i$ and distance d_i of the vector from target to agent, and the speed of ψ_i which is z_2^i. Additionally, this controller needs the velocity vector (magnitude, and orientation, and time derivative) of the center of swarm or of all of the agents. All these info, especially the global variables, are hard to acquire in the practical applications. In the future studies on practical applications, we have to examine how to use relative variables instead of the global ones.

Substituting the linearizing inputs we get $\dot{\xi}_2^{1^i} = p_1^i$ and $\dot{\xi}_1^{2^i} = p_2^i$

The normal states are decoupled.

$$\dot{\xi}_1^{1^i} = \xi_2^{1^i} \tag{5.154a}$$

$$\dot{\xi}_2^{1^i} = p_1^i \tag{5.154b}$$

$$\dot{\xi}_1^{2^i} = p_2^i \tag{5.155}$$

As in the previous controllers, we again shift the states as $\xi_1^{1^i} = d_i - d_0$ and $\xi_1^{2^i} = z_2^i - \alpha$. Now, we can use any linear system controller design method to have stable systems at $d_i = d_0$ and $\psi_i = \alpha$.

Consider the inputs p_1^i and p_2^i are linearly dependent on the normal states. Then the normal dynamics become

$$\begin{bmatrix} \dot{\xi}_1^{1^i} \\ \dot{\xi}_2^{1^i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ K_1^1 & K_2^1 \end{bmatrix} \begin{bmatrix} \xi_1^{1^i} \\ \xi_2^{1^i} \end{bmatrix} \tag{5.156}$$

and

$$\dot{\xi}_1^{2^i} = K_1^2 \xi_1^{2^i} \tag{5.157}$$

Selecting the proper values for the controller parameters, one may obtain stable underdamped or overdamped responses. Utilizing the above states we can show that the center of the flock is converging to an emergent point (depending on the initial conditions). The proof of this convergence starts with examining the position of each agent in global coordinates

$$x_i = x_c + d_i \cos(\psi_i) \tag{5.158}$$

$$y_i = y_c + d_i \sin(\psi_i) \tag{5.159}$$

Summing positions of all agents

$$\sum_{i=1}^{M} x_i = \sum_{i=1}^{M} x_c + \sum_{i=1}^{M} d_i \cos(\psi_i) = M x_c + \sum_{i=1}^{M} d_i \cos(\psi_i) \tag{5.160}$$

$$\sum_{i=1}^{M} y_i = \sum_{i=1}^{M} y_c + \sum_{i=1}^{M} d_i \sin(\psi_i) = M y_c + \sum_{i=1}^{M} d_i \sin(\psi_i) \tag{5.161}$$

using $\sum_{i=1}^{M} x_i = M x_c$ and $\sum_{i=1}^{M} y_i = M y_c$

$$\sum_{i=1}^{M} d_i \cos(\psi_i) = 0 \tag{5.162}$$

$$\sum_{i=1}^{M} d_i \sin(\psi_i) = 0 \tag{5.163}$$

The time derivative of these equations are

$$\sum_{i=1}^{M} \dot{d}_i \cos(\psi_i) - \sum_{i=1}^{M} d_i \dot{\psi}_i \sin(\psi_i) = 0 \tag{5.164}$$

$$\sum_{i=1}^{M} \dot{d}_i \sin(\psi_i) + \sum_{i=1}^{M} d_i \dot{\psi}_i \cos(\psi_i) = 0 \tag{5.165}$$

At the steady state of ξ values $d_i = d_0$, and $\dot{d}_i = 0$, and $\dot{\psi}_i = \alpha$ for each agent. At the steady state the above equations become

$$\sum_{i=1}^{M} \sin(\psi_i) = 0 \tag{5.166}$$

$$\sum_{i=1}^{M} \cos(\psi_i) = 0 \tag{5.167}$$

Therefore, at the steady state of i.e. two agents, the agents are moving in the opposite directions with a distance of d_0 from the center of flock. Or for three agents, the agents are located on the corners of a equilateral triangle and moving tangent to the line passing through center of flock and the agent. The motions of two and three number of members in a flock are represented in Figure 5.33.

The velocity of the center of swarm can be expressed as

$$\dot{x}_c = \frac{1}{M} \sum_{i=1}^{M} z_1^i \cos(\theta_i) \tag{5.168}$$

$$\dot{y}_c = \frac{1}{M} \sum_{i=1}^{M} z_1^i \sin(\theta_i) \tag{5.169}$$

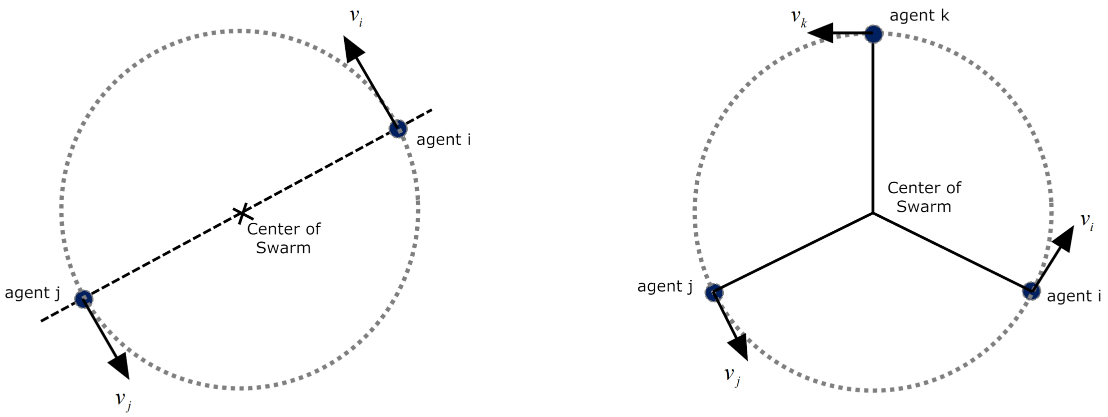


Figure 5.33: Agent positions and velocity vectors at equilibrium. Samples for 2 and 3 number of agents.

at the CCW rotational equilibrium $\theta_i = \psi_i + \pi/2$ and $z_1^i = \alpha d_0$. Substituting these into the equations

$$\dot{x}_c = \frac{1}{M}\sum_{i=1}^{M} \alpha d_0 \cos(\psi_i + pi/2) \tag{5.170}$$

$$\dot{y}_c = \frac{1}{M}\sum_{i=1}^{M} \alpha d_0 \sin(\psi_i + pi/2) \tag{5.171}$$

and rearranging

$$\dot{x}_c = \frac{1}{M}\alpha d_0 \sum_{i=1}^{M} \sin(\psi_i) \tag{5.172}$$

$$\dot{y}_c = -\frac{1}{M}\alpha d_0 \sum_{i=1}^{M} \cos(\psi_i) \tag{5.173}$$

and using the results in equation (5.166)

$$\dot{x}_c = 0 \tag{5.174}$$

$$\dot{y}_c = 0 \tag{5.175}$$

Therefore, the center of flock is stationary at the steady state. Furthermore, the velocity and orientation components in equation (5.120) can be derived at the steady state as

$$v_c^i = z_1^i/M = \frac{\alpha d_0}{M} \tag{5.176}$$

$$\theta_c^i = \theta_i + \pi \tag{5.177}$$

Therefore, at the steady state the mean velocity of the other agents (different from *i*th agent) is equal to $\alpha d_0/M$ in the opposite direction of *i*th agents velocity.

The minimal dynamics should also be investigated. Since the relative degree (3) is less than the number of states (5) we will select the new states η_1^i and η_2^i independent of input i.e. orthogonal to the function $g(x)$.

$$\frac{\partial \eta^i}{\partial x_i} g(x_i) = 0 \Rightarrow$$

$$\begin{bmatrix} \frac{\partial \eta_1^i}{\partial d_i} & \frac{\partial \eta_1^i}{\partial z_1^i} & \frac{\partial \eta_1^i}{\partial \psi_i} & \frac{\partial \eta_1^i}{\partial z_2^i} & \frac{\partial \eta_1^i}{\partial \theta_i} \\ \frac{\partial \eta_2^i}{\partial d_i} & \frac{\partial \eta_2^i}{\partial z_1^i} & \frac{\partial \eta_2^i}{\partial \psi_i} & \frac{\partial \eta_2^i}{\partial z_2^i} & \frac{\partial \eta_2^i}{\partial \theta_i} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ \frac{1}{d_i}\left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) & \frac{1}{d_i}\left(1 - \frac{K}{M}\right) z_1^i \cos(\theta_i - \psi_i) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.178}$$

$$\frac{\partial \eta_1^i}{\partial z_1^i} + \frac{\partial \eta_1^i}{\partial z_2^i} \frac{1}{d_i}\left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) = 0 \tag{5.179}$$

$$\frac{\partial \eta_2^i}{\partial z_2^i} \frac{1}{d_i} z_1^i \left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) + \frac{\partial \eta_2^i}{\partial \theta_i} = 0 \tag{5.180}$$

which means in addition to ${\xi_1^1}^i$, ${\xi_2^1}^i$, and ${\xi_1^2}^i$, the state η_1^i should be independent of z_1^i and z_2^i. Similarly η_2^i should be independent of θ_i and z_2^i or it should satisfy the partial differential equation in (5.180) . We select $\eta_1^i = \theta_i - \psi_i - \pi/2$ and $\eta_2^i = \psi_i$. Let us examine the zero dynamics of these variables.

The time derivative of η_1^i is

$$\dot{\eta}_1^i = \dot{\theta}_i - \dot{\psi}_i = w_2^i - z_2^i \tag{5.181}$$

We first substitute the equations in (5.154) and (5.156) into this equation. And then we do the following replacement of parameters to obtain equations in reduced system domain

$$d_i = {\xi_1^1}^i + d_0 \tag{5.182}$$

$$z_2^i = {\xi_1^2}^i + \alpha \tag{5.183}$$

$$\psi_i = \eta_2^i \tag{5.184}$$

$$\theta_i = \eta_1^i + \eta_2^i + \pi/2 \tag{5.185}$$

Note that in the above equations the equilibrium is set for $\alpha > 0$ and so $\theta_i - \psi_i = \pi/2$. However, the other equilibrium $\alpha < 0$ and $\theta_i - \psi_i = 3\pi/2$ may also be utilized in the following analysis.

Now we use the zero dynamics of the system at $\xi_1^{1^i} = \xi_2^{1^i} = \xi_1^{2^i} = 0$. In addition we use the results in (5.176). Then $\dot{\eta}_1^i$ becomes [3]

$$\begin{aligned}\dot{\eta}_1^i &= \alpha\cos(\eta_1^i)^2 - \alpha + (K_2^1\sin(2\eta_1^i)(M + KM - 1))/(2(K - M)) - (Md_0\sin(\eta_1^i)\\ &\quad ((z_1^2\sin(2\eta_1^i)(M - K + KM)^2)/(2M^2d0^2)\\ &\quad -(\alpha z_1^i\sin(\eta_1^i)(K - M))/(Md_0)))/(z_1^i(K - M)) \qquad (5.186)\end{aligned}$$

Note that the zero dynamics of η_1^i is independent of η_2^i. The local stability of η_1^i may be examined by linearization around the equilibrium $\eta_1^i = 0$. The derivative of $\dot{\eta}_1^i$ with respect to η_1^i at $\eta_1^i = 0$ is

$$\frac{\partial\dot{\eta}_1{}^i}{\partial\eta_1^i}|_{\eta_1^i=0} = -\frac{K_2^1(M(1+K) - K)}{M - K} \qquad (5.187)$$

This is the eigenvalue of the minimal dynamics of η_1^i, and since it is definitely negative ($K_2^1 > 0$, $0 \leq K < 1$, and $M > K$) the state is locally asymptotically stable. Physically this means the system is converging to $\theta_i = \psi_i + \pi/2$ that is the CCW rotation of the agent around the target.

The time derivative of second selected state $\eta_2^i = \psi_i$ becomes

$$\begin{aligned}\dot{\eta}_2^i &= z_2^i\\ &= \xi_1^{2^i} + \alpha \qquad (5.188)\end{aligned}$$

The zero dynamics becomes

$$\dot{\eta_2}^i = \alpha$$

Here note that this state is not converging to a constant, instead it is increasing with a velocity of α but never diverging due to $\psi_i = mod(\psi, 2\pi)$. The behavior is indeed a tracking problem of the state. The orientation variable ψ_i tracks the constant speed rotation behavior at the zero dynamics. This is completely consistent with the controlled dynamics of the system. Furthermore, if we select the state as $\eta_i = \theta$ the zero dynamics becomes $\dot{\eta}_i = \dot{\theta}_i = \dot{\psi}_i = \alpha$. Note that for this selection of the derived states the problem becomes a tracking of states problem and again it is consistent with the steady state values of original states.

The last state we should examine is the speed of the agent z_1^i. This is directly related to the orientations and rotational speeds as in the following equation

$$z_1^i = \frac{z_2^i d_i + Kv_c^i\sin(\theta_i - \psi_i)}{\left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i)} \qquad (5.189)$$

---

[3] The intermediate steps are performed with symbolic equation solver toolbox of Matlab.

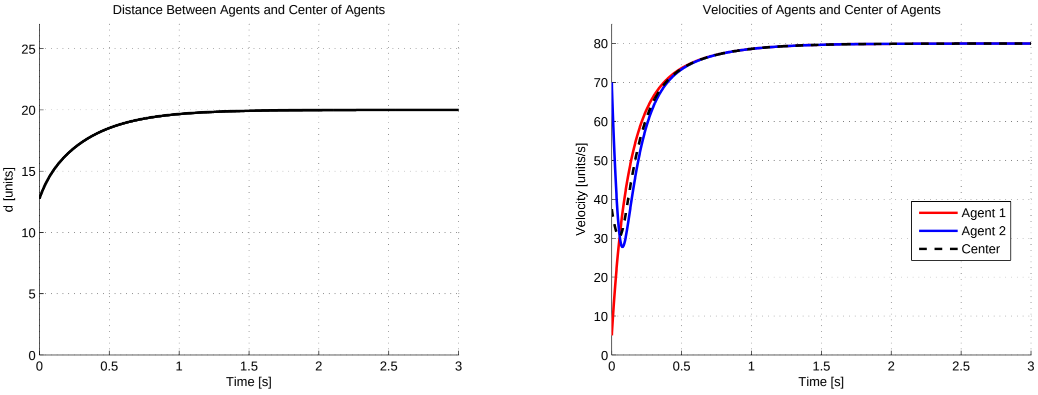
At the steady state ($z_2^i = \alpha$, $d_i = d_0$, $v_c^i = \alpha d_0$, $\theta_c^i = \theta + \pi$, $\theta_i - \psi_i = \pi/2$) it becomes

$$z_1^i|_{eq} = \alpha d_0 \tag{5.190}$$

which is again consistent with the controlled dynamics of the system. This state may be selected as one of the η^i states, however the linearization of the reduced system becomes inconclusive with a zero eigenvalue.

5.3.7.4 Multi-Agent System with MIMO - Non-Stationary Target Following Controller - Simulations

In this section the MIMO controller developed for the non-stationary targets is applied for the Multi-agent systems. The controller parameters and the initial conditions of the simulations presented in the following figures are the same with ones performed in stationary target controller section, for better comparison. The parameter K is selected to be $K = 0.8$ for these simulations. As seen from the figures 5.34, 5.35, 5.36, 5.37, 5.38, 5.39, the controllers are successful in achieving the desired circling behaviors.



(a) (b)

Figure 5.34: Distance (a) and velocities (b) of regular circling motion of 2 agents with Circling MIMO-NonStationary Targeting controllers.

5.3.8 Concluding Remarks

In the literature most of the studies utilize a pre-generated path for tracking setpoints, straight lines, circular or randomly generated paths by mobile robots. In this study the circling around

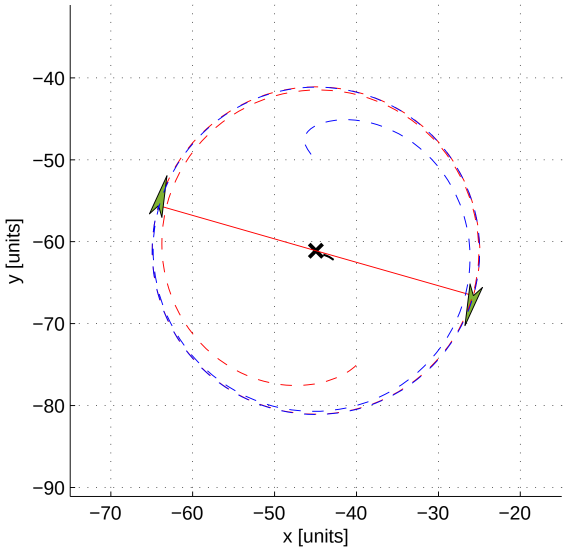
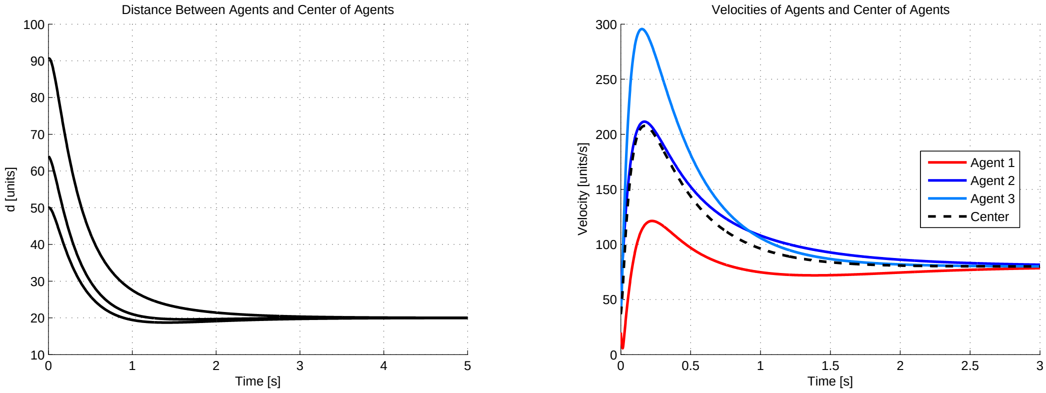


Figure 5.35: Path of the 2 agents and center for regular circling with MIMO-NonStationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black.



(a) (b)

Figure 5.36: Distance (a) and velocities (b) of regular circling motion of 3 agents with Circling MIMO-NonStationary Targeting controllers.

a target problem for ground and air vehicles is considered as a periodic system problem and the states in the system kinematics are forced to converge to periodical trajectories. Especially we propose controllers for the circling around a target behavior of mobile robots. These controllers are developed by static and dynamic feedback linearization techniques. We study two different cases: (i) Circling-SISO:input is angular speed and translational speed is constant, and (ii) Circling-MIMO: both angular and translational speeds are inputs. We derive the conditions on the underdamped and overdamped stable responses of the systems and present the

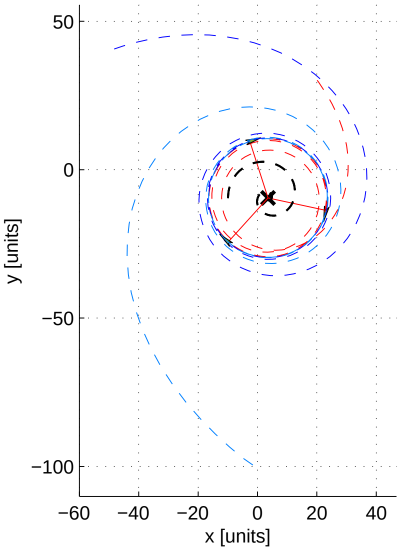
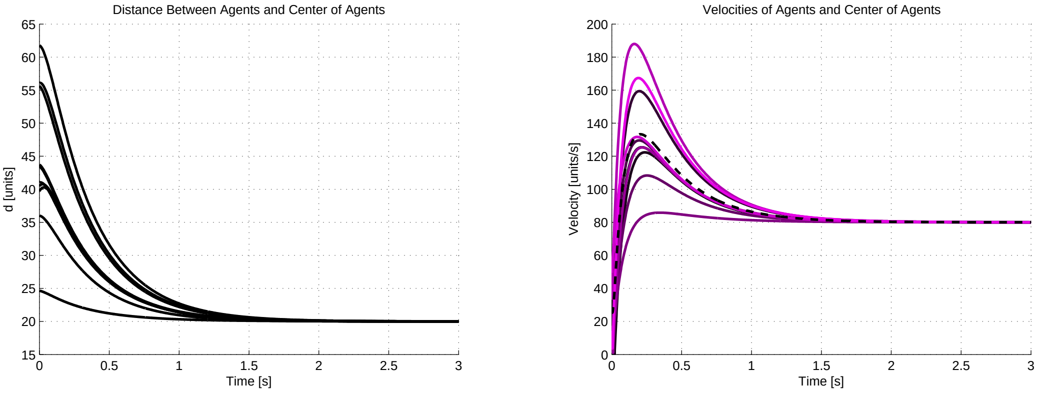


Figure 5.37: Path of the 3 agents and center for regular circling with MIMO-NonStationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black.



(a)

(b)

Figure 5.38: Distance (a) and velocities (b) of regular circling motion of 10 agents with Circling MIMO-NonStationary Targeting controllers.

simulations results for each case.

The controllers developed in this study may be utilized in UGV, UAV, and UWV applications dedicated to tracking of targets by circling around them. Furthermore, the controllers may be adapted for the swarm of these vehicles to circle around targets without collision. The inclusion of the vehicle dynamics to the kinematic model of this study is also a possible future work.

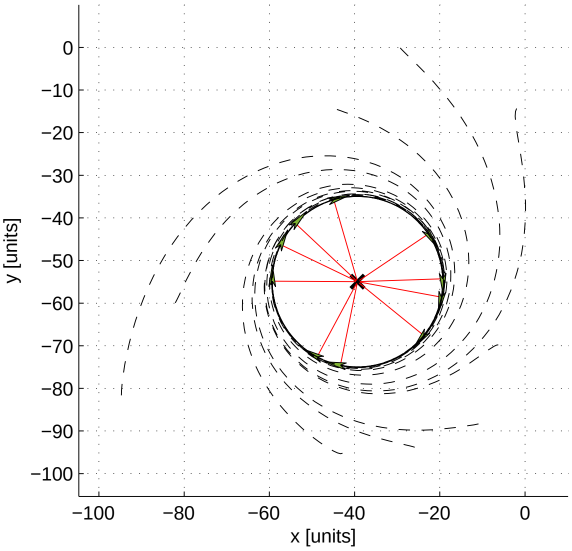


Figure 5.39: Path of the 10 agents and center for regular circling with MIMO-NonStationary Targeting controllers. Agent paths: dashed blue and dashed red, center path: dashed black.

.

CHAPTER 6

Line Following Controllers Developed by Feedback Linearization

6.1 Introduction

As mentioned in the previous chapter on circling controller development, the studies in the literature mostly utilize pre-generated paths for tracking by mobile robots. By using these paths many researchers achieved the robots to follow straight lines, circular or randomly generated paths. In this chapter, controllers for the line following behavior are developed by feedback linearization techniques. The controllers are first developed for single agents using stationary, and non-stationary targets and using reference orientations. The agent positions are controlled to converge to lines and travel at reference speeds on these lines. As in the circling controllers, again we propose different controllers for three different cases. In the first one, the only controllable input is the angular velocity of the robot (SISO-Stationary Target Controller). The translational speed of the robot is taken as constant. This case can represent applications such as autonomous air vehicles (UAV) traveling on a straight path. The only required global info is the distance to the target. In the second case in addition to the angular velocity input the translational speed is taken as an input (MIMO-Stationary Target Controller). The only required global info is the distance to the target. The last controller developed is similar to the MIMO controller but this time the target is not stationary (MIMO-NonStationary Target Controller). The controller utilizes the target velocity vector in the manipulations and achieves much better performance with respect to the stationary target controllers.

The line following behavior is achieved without pre-specified paths to be followed by the robots. In fact robots use the target position, prespecified orientation and velocity to achieve the mentioned behaviors. The usage of the only target information instead of pre-specified paths is mostly beneficial in the multi-robot applications. In multi-robot applications usually each robot is a target for the others; therefore, the controllers that use target information is preferable with respect to pre-generated paths since in most of the applications the paths of the robots in the swarm are emergent paths that are hard to predict. The controllers developed for single agents to follow a line are applied to multi-agent systems. The agents in the swarm utilized the same controllers targeting the centroid of the swarm to achieve the line following behavior. A minor modification in the specification of target is done for the MIMO-stationary target controller to achieve the behavior. The MIMO Non-stationary Target Controller is redeveloped for the multi-agent system.

In the following sections, the mathematical background and development of the controllers are presented. The results of the analytical derivations are validated in the simulation study sections. Lastly the discussions on the results of some modifications on the developed controllers and the future directions of this thesis study are given.

The mathematical model for agents is again the first-order kinematic model of the unicycle vehicles. The model is represented in the previous chapter in section 5.2. Using that kinematic model we will solve the feedback linearization problems for “Line Following” behaviors, in this chapter.

6.2 Following a line passing through a Target with a specified slope

In this section the problem to be solved is the behavior of the robot that converges to a line passing through a target. The slope of the line is specified as a parameter in the controller. The agent approaches the line and travels on it with the given constant linear speed. This strategy is beneficial especially for multi-agent systems. In these systems each agent is a target for the others. If the common controller achieves the above behavior. All agents will align on the same line traveling in serial formation which is a very beneficial formation for many multi-agent tasks like passing through a thin gap.

We propose different controllers for two different cases. In the first one, the only controllable input is the angular velocity of the robot. The translational speed of the robot is taken as constant. We consider this case especially for the air vehicles (like UAV) following a target with constant speed. In the second case in addition to the angular velocity input the translational speed is taken as an input. We use static feedback linearization methods in the first case and dynamics feedback linearization in the second one. The first will be called as "Line Following - SISO" where the only controlled system state is the angular velocity of the robot ($\dot{\theta} = u$). The robot travels at constant translational speed v. The second case called as "Line Following - MIMO" is the case in which both translational and rotational speeds are inputs of the system ($v = u_1$ and $\dot{\theta} = u_2$). In the following sections we will derive the feedback linearization laws for these cases and present some simulation results. In the following sections, the feedback linearization results are presented. The results of the analytical derivations are validated in the simulation study sections. We lastly present the conclusions on the results of this part and mention some future work.

6.2.1 Line Following - SISO - Stationary Target Case

To achive the line following behavior we will again linearize the system by nonlinear feedback signals. For the SISO case the only controlled input is the angular velocity of the vehicles ($u = \dot{\theta}$). The linear velocity of the vehicle is assumed to be constant at v. Therefore, we write the dynamics as

$$\begin{bmatrix} \dot{d} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta - \psi) \\ \frac{1}{d} v \sin(\theta - \psi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{6.1}$$

Since vehicle is desired to converge to a line, we will define the line with the target position and the slope. The new state of the system is the distance between the vehicle and the line h (Figure 6.1).

$$h = d\sin(\psi - \beta) \tag{6.2}$$

The time derivative of this state can be derived by simply taking the derivative of equation (6.2) or from figure it can be easily observed that

$$\dot{h} = v\sin(\theta - \beta) \tag{6.3}$$

Using this new state we can summarize the system dynamics as

$$\begin{bmatrix} \dot{h} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\sin(\theta - \beta) \\ \frac{1}{h} v \sin(\psi - \beta)\sin(\theta - \psi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{6.4}$$

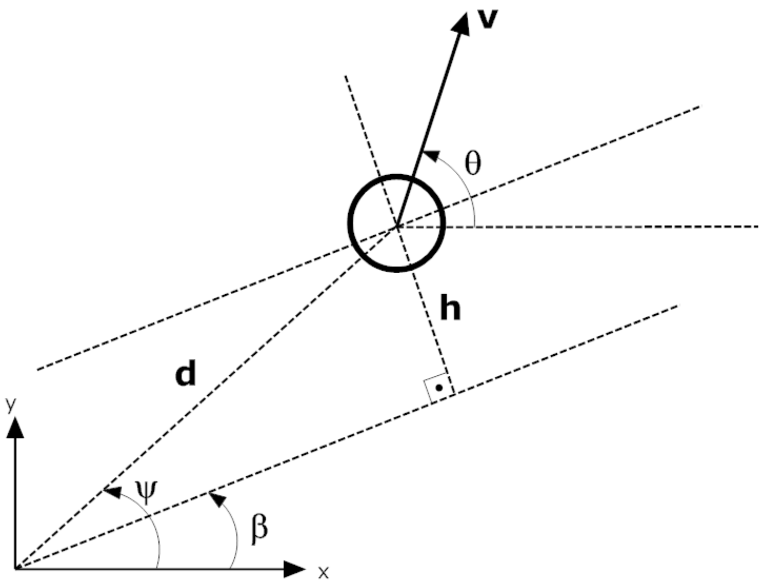


Figure 6.1: Relative Coordinate Frame. Origin is target. Circle represents the vehicle. The line to be followed is the one with the slope β.

which may be described as

$$\dot{x} = f(x) + g(x)u \tag{6.5}$$

where $x = [h, \psi, \theta]^T$ is the state vector.

The equilibrium of the states at the line following behavior can be described by the equations

$$\begin{aligned} \dot{d} &= v \quad OR \quad \dot{h} = 0 \Rightarrow \psi = \theta = \beta && \text{(6.6a)} \\ \dot{\psi} &= 0 \Rightarrow \psi = \beta && \text{(6.6b)} \\ \dot{\theta} &= 0 \Rightarrow u = 0 && \text{(6.6c)} \end{aligned}$$

To achieve this equilibrium, we will use the feedback linearization method to linearize the system and then design linear system controllers.

The output of the system is selected as

$$y = k(x) = h \tag{6.7}$$

Then the lie derivatives along the nonlinear system functions $f(x)$ and $g(x)$ are

$$\begin{aligned} L_g k(x) &= \frac{\partial k}{\partial x} g(x) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0 \qquad (6.8a) \\ L_f k(x) &= \frac{\partial k}{\partial x} f(x) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v\sin(\theta-\beta) \\ \frac{1}{h} v\sin(\psi-\beta)\sin(\theta-\psi) \\ 0 \end{bmatrix} = v\sin(\theta-\beta) \qquad (6.8b) \\ L_g L_f k(x) &= \frac{\partial L_f k(x)}{\partial x} g(x) = \begin{bmatrix} 0 & 0 & v\cos(\theta-\beta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = v\cos(\theta-\beta) \qquad (6.8c) \end{aligned}$$

At equilibrium ($\theta = \beta$)

$$L_g L_f k(x)|_{\theta=\beta} = v\cos(0) = v \neq 0 \qquad (6.9)$$

which is not equal to 0. Therefore, the relative degree is 2.

To calculate the linearizing feedback input we need the second order Lie derivative of $k(x)$ along $f(x)$

$$L_f L_f k(x) = \frac{\partial L_f k(x)}{\partial x} f(x) = \begin{bmatrix} 0 & 0 & v\cos(\theta-\beta) \end{bmatrix} \begin{bmatrix} v\sin(\theta-\beta) \\ \frac{1}{h} v\sin(\psi-\beta)\sin(\theta-\psi) \\ 0 \end{bmatrix} = 0 \qquad (6.10)$$

then the linearizing input becomes

$$\begin{aligned} u &= \frac{1}{L_g L_f k(x)} \left[-L_f L_f k(x) + w \right] \\ &= \frac{1}{v\cos(\theta-\beta)} [0 + w] \\ &= \frac{w}{v\cos(\theta-\beta)} \qquad (6.11) \end{aligned}$$

The normal form of the system dynamics becomes

$$\begin{aligned} \xi_1 &= k(x) = h \qquad (6.12a) \\ \xi_2 &= L_f k(x) = v\sin(\theta-\beta) \qquad (6.12b) \end{aligned}$$

then,

$$\begin{aligned}
\dot{\xi}_1 &= v\sin(\theta-\beta) && (6.13)\\
\dot{\xi}_2 &= v\dot{\theta}\cos(\theta-\beta) &&\\
&= v\frac{w}{v\cos(\theta-\beta)}\cos(\theta-\beta) &&\\
&= w && (6.14)
\end{aligned}$$

The new transformed dynamics of the system becomes

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 && (6.15)\\
\dot{\xi}_2 &= w && (6.16)
\end{aligned}$$

Since the relative degree is less than the number of states we will select a the new state η independent of input i.e. orthogonal to the function $g(x)$.

$$\frac{\partial\eta}{\partial x}g(x) = 0 \Rightarrow \begin{bmatrix} \frac{\partial\eta}{\partial h} & \frac{\partial\eta}{\partial\psi} & \frac{\partial\eta}{\partial\theta} \end{bmatrix}\begin{bmatrix} 0\\ 0\\ 1 \end{bmatrix} = \frac{\partial\eta}{\partial\theta} = 0 \tag{6.17}$$

which means η should be independent of θ. The selected function is

$$\eta = d = \frac{h}{\sin(\psi-\beta)} \tag{6.18}$$

The time derivative of this state is

$$\begin{aligned}
\dot{\eta} &= \frac{\dot{h}}{sin(\psi-\beta)} - \frac{h\dot{\psi}\cos(\psi-\beta)}{\sin^2(\psi-\beta)}\\
&= \frac{v\sin(\theta-\beta)}{sin(\psi-\beta)} - \frac{\cancel{h}\cos(\psi-\beta)}{\sin^{\cancel{2}}(\psi-\beta)}\frac{1}{\cancel{h}}v\cancel{\sin(\psi-\beta)}\sin(\theta-\psi)\\
&= \frac{v\sin(\theta-\beta)}{sin(\psi-\beta)} - \frac{v\sin(\theta-\psi)\cos(\psi-\beta)}{\sin(\psi-\beta)}\\
&= \frac{v}{sin(\psi-\beta)}\left[\sin(\theta-\beta) - \sin(\theta-\psi)\cos(\psi-\beta)\right]\\
&= \frac{v}{sin(\psi-\beta)}\left[\sin(\theta-\beta) - \sin(\theta-\beta-(\psi-\beta))\cos(\psi-\beta)\right]\\
&= \frac{v}{sin(\psi-\beta)}\left[\sin(\theta-\beta) - (\sin(\theta-\beta)\cos(\psi-\beta) - \cos(\theta-\beta)\sin(\psi-\beta))\cos(\psi-\beta)\right]
\end{aligned}$$

(6.19)

substituting the equalities $h = \xi_1$, $\sin(\theta-\beta) = \frac{\xi_2}{v}$, $\cos(\theta-\beta) = \sqrt{1-\frac{\xi_2^2}{v^2}}$, $\sin(\psi-\beta) = \frac{\xi_1}{\eta}$, $\cos(\psi-\beta) = \sqrt{1-\frac{\xi_1^2}{\eta^2}}$

$$\dot{\eta} = \frac{v\eta}{\xi_1}\left[\frac{\xi_2}{v} - \left(\frac{\xi_2}{v}\sqrt{1-\frac{\xi_1^2}{\eta^2}} - \sqrt{1-\frac{\xi_2^2}{v^2}}\frac{\xi_1}{\eta}\right)\sqrt{1-\frac{\xi_1^2}{\eta^2}}\right] \tag{6.20}$$

The zero dynamics of the system is calculated at $\xi_1 = \xi_2 = 0$. Then the value of η at this equilibrium becomes

$$\begin{aligned}
\dot{\eta}|_{\xi_1=\xi_2=0} &= \lim_{\xi_1=\xi_2=0} \frac{v\eta}{\xi_1}\left[\frac{\xi_2}{v} - \left(\frac{\xi_2}{v}\sqrt{1-\frac{\xi_1^2}{\eta^2}} - \sqrt{1-\frac{\xi_2^2}{v^2}}\frac{\xi_1}{\eta}\right)\sqrt{1-\frac{\xi_1^2}{\eta^2}}\right] \\
&= \lim_{\xi_1=\xi_2=0} \frac{v\eta}{\xi_1}\left[\frac{\xi_2}{v} - \left(\frac{\xi_2}{v} - \frac{\xi_1}{\eta}\right)\right] \\
&= \lim_{\xi_1=\xi_2=0} \frac{v\eta}{\xi_1}\left[\frac{\xi_1}{\eta}\right] \\
&= v \qquad (6.21)
\end{aligned}$$

The new state η has its time derivative at $\dot{\eta} = v$ which is consistent with the desired steady state value of the original states, $\dot{d} = \dot{\eta} = v$.

Now, the stabilizing controller input w can be designed by many linear system controller design methods. Here we will just derive the sufficient conditions on the input function parameters and present some results.

The jacobian matrix of the transformed model becomes

$$J = \begin{bmatrix} 0 & 1 \\ \frac{\partial w}{\partial \xi_1} & \frac{\partial w}{\partial \xi_2} \end{bmatrix} \qquad (6.22)$$

The characteristic polynomial of the Jacobian matrix is

$$\begin{aligned}
\begin{vmatrix} \lambda & -1 \\ -\frac{\partial w}{\partial \xi_1} & \lambda - \frac{\partial w}{\partial \xi_2} \end{vmatrix} &= \lambda\left(\lambda - \frac{\partial w}{\partial \xi_2}\right) - \frac{\partial w}{\partial \xi_1} \\
&= \lambda^2 - \frac{\partial w}{\partial \xi_2}\lambda - \frac{\partial w}{\partial \xi_1} = 0 \qquad (6.23)
\end{aligned}$$

Then the eigenvalues are

$$\lambda_{1,2} = \frac{\frac{\partial w}{\partial \xi_2} \pm \sqrt{\left[\frac{\partial w}{\partial \xi_2}\right]^2 + 4\frac{\partial w}{\partial \xi_1}}}{2} \qquad (6.24)$$

For a stable system

$$\frac{\partial w}{\partial \xi_2} < 0 \ \ \& \ \ \frac{\partial w}{\partial \xi_1} < 0 \qquad (6.25)$$

For an underdamped response

$$\begin{aligned}
\left[\frac{\partial w}{\partial \xi_2}\right]^2 + 4\frac{\partial w}{\partial \xi_1} < 0 &\Rightarrow \\
\frac{\partial w}{\partial \xi_1} < -\frac{\left[\frac{\partial w}{\partial \xi_2}\right]^2}{4} & \qquad (6.26)
\end{aligned}$$

For an overdamped response

$$0 < \left[\frac{\partial w}{\partial \xi_2}\right]^2 + 4\frac{\partial w}{\partial \xi_1} < \left[\frac{\partial w}{\partial \xi_2}\right]^2 \Rightarrow$$

$$-\frac{\left[\frac{\partial w}{\partial \xi_2}\right]^2}{4} < \frac{\partial w}{\partial \xi_1} < 0 \qquad (6.27)$$

Let w be a linear state feedback

$$w = K_1\xi_1 + K_2\xi_2 \qquad (6.28)$$

where $K_1 = \frac{\partial w}{\partial \xi_1}$ and $K_2 = \frac{\partial w}{\partial \xi_2}$. Now selecting the parameters K_1 and K_2 according to the conditions above one can achieve circling behaviors.

6.2.2 Line Following - SISO - Stationary Target - Simulation Results

In this section we present the results of the developed line following controller in the previous section. As mentioned above we may select underdamped or overdamped controller parameters. The first example is the overdamped behavior. For this behavior we select the parameters as $K_2 = -8 < 0$ and $-|K_2|^2/4 < K_1 = -15 < 0$. In the lower part of the Figure 6.2 the change of the distance between robot and line (h) is presented. As seen the robot approaches the line with an overdamped response and stays steady at $h = 0$. The upper part of the figure shows the distance between robot and target (d). Robot gets far away from the target. The change of this distance becomes steady at $\dot{d} = v = 100[units/s]$. In Figure 6.3 the change of angles ψ and θ are presented. As seen both angles converge to the slope $\beta = 120^o$.

The path of the vehicle is presented in Figure 6.4. As seen from the figure, the vehicle reaches the line with an overdamped response and continues to follow the line with slope $\beta = 120^o$.

The underdamped response is obtained by the controller parameters $K_2 = -4 < 0$ and $K_1 = -12 < -|K_2|^2/4$. In Figure 6.5 the change of h and d are presented. As seen the response is underdamped. In the next Figure 6.6 the change of ψ and θ may be observed. The Figure 6.7 shows the path of the vehicle.

Note that, in the simulation results above the vehicle's initial orientation is 80^o and converges to 120^o as time passes. However, if it starts with the orientation $\theta = -30^o$, then it converges to the line but with the opposite slope $\theta = \beta + \pi$. The path of the vehicle for this case is shown in Figure 6.8. This is a beneficial property of the controller since the vehicle converges to

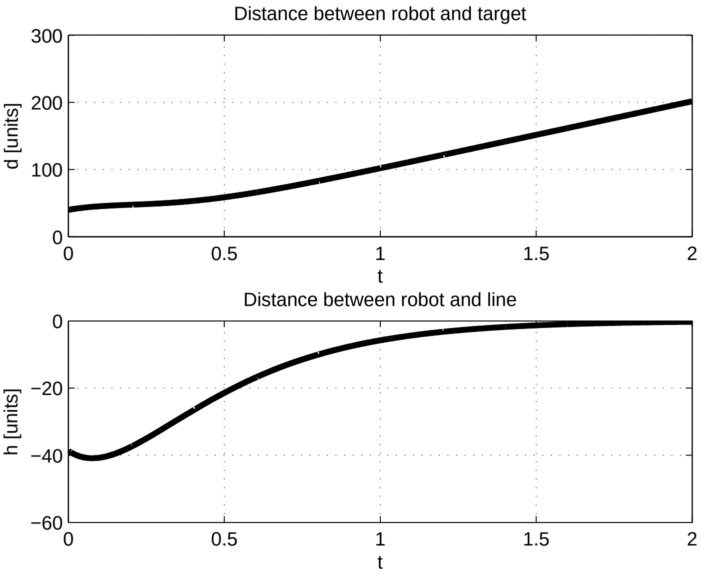


Figure 6.2: The change of the distance between robot and target (*d*), and robot and line (*h*) with respect to time. Overdamped response for line following behavior.

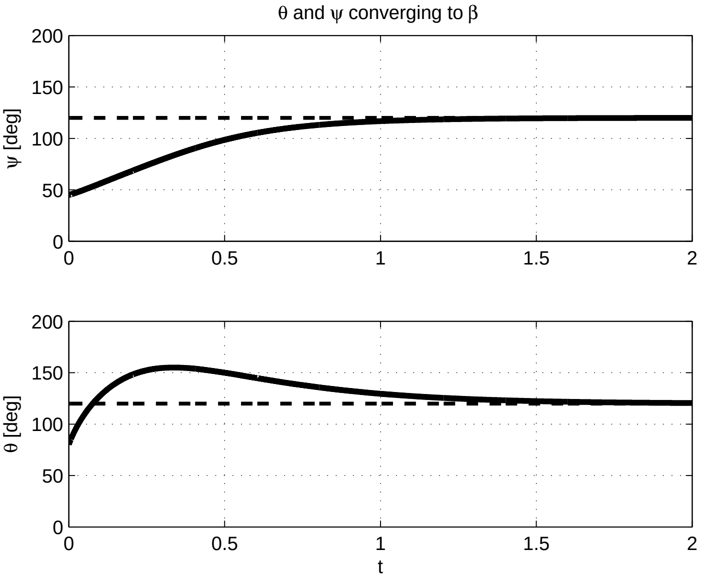


Figure 6.3: The change of the angles θ and ψ with respect to time. Overdamped response for line following behavior.

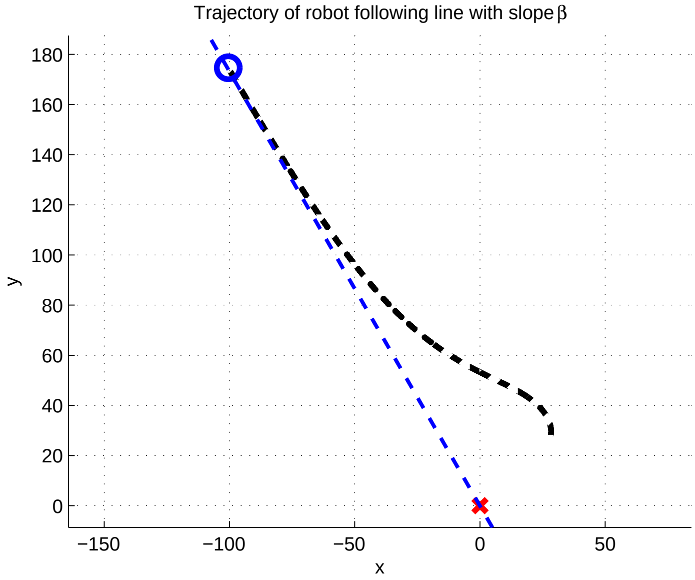


Figure 6.4: The path of the robot. Overdamped response for line following behavior.

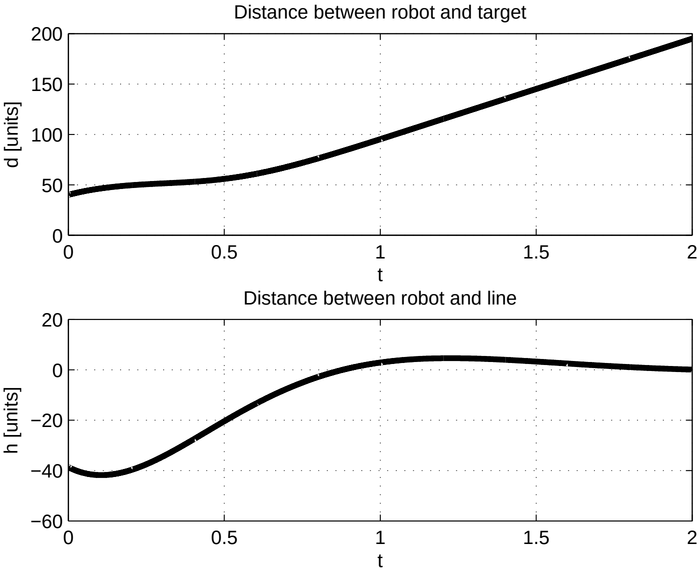


Figure 6.5: The change of the distance between robot and target (d), and robot and line (h) with respect to time. Underdamped response for line following behavior.

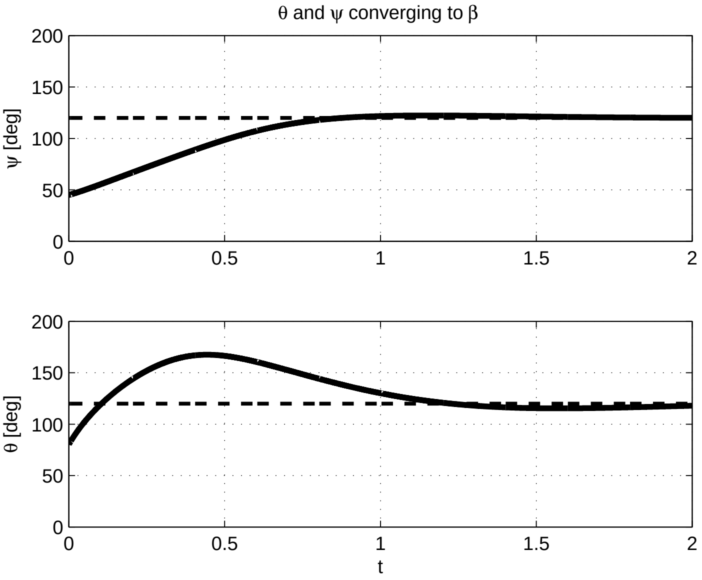


Figure 6.6: The change of the angles θ and ψ with respect to time. Underdamped response for line following behavior.

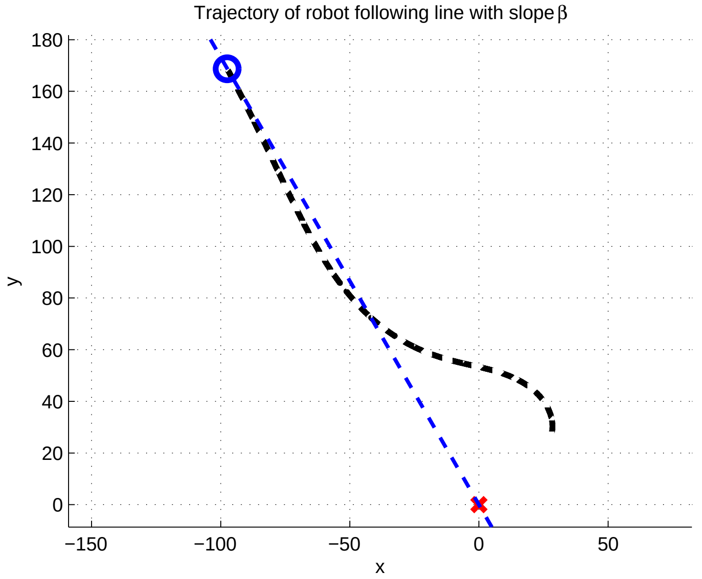


Figure 6.7: The path of the robot. Underdamped response for line following behavior.

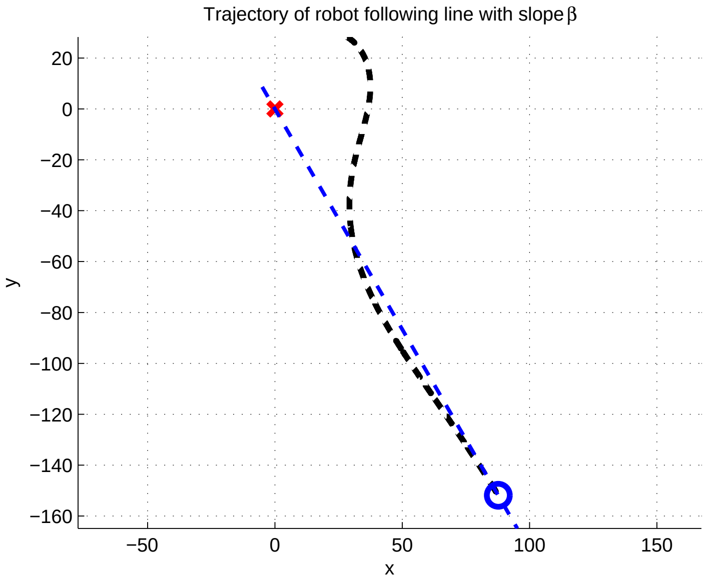


Figure 6.8: The path of the robot following the slope $\beta + \pi$.

the line by using the shortest path. Recall that the aim of the controller is just achieving the convergence of vehicle to the line passing through the target.

6.2.3 Line Following - MIMO - Stationary Target

In this section we will develop the feedback linearization of line following dynamics by taking speed v and angular velocity $\dot{\theta}$ as the inputs. Lets call $v = u_1$ and $\dot{\theta} = u_2$. Then the system dynamics becomes

$$\dot{d} = u_1 \cos(\theta - \psi) \tag{6.29a}$$

$$\dot{\psi} = \frac{1}{d} u_1 \sin(\theta - \psi) \tag{6.29b}$$

$$\dot{\theta} = u_2 \tag{6.29c}$$

In the line following behavior, the vehicle travels at a specified speed $\dot{d} = v_0$ on the line passing through the target with the specified angle β. Since the controller is going to adjust the speed we need one more integrator for the distance variable d.

$$\dot{d} = z_1 \tag{6.30a}$$

$$\dot{z}_1 = \dot{u}_1 \cos(\theta - \psi) - u_1(\dot{\theta} - \dot{\psi}) \sin(\theta - \psi) \tag{6.30b}$$

for input u_1 we define a new integrator such that $u_1 = z_2$, then $\dot{z_2} = w_1$.

$$\begin{aligned} \dot{d} &= z_1 && (6.31a) \\ \dot{z_1} &= w_1 \cos(\theta - \psi) - z_2(\dot{\theta} - \dot{\psi}) \sin(\theta - \psi) && (6.31b) \\ \dot{z_2} &= w_1 && (6.31c) \end{aligned}$$

For the convenience of inputs let us call $u_2 = w_2$. Then, substituting the other states we get

$$\begin{aligned} \dot{d} &= z_1 && (6.32a) \\ \dot{z_1} &= w_1 \cos(\theta - \psi) - w_2 z_2 \sin(\theta - \psi) + \frac{1}{d} z_2^2 \sin^2(\theta - \psi)) && (6.32b) \\ \dot{z_2} &= w_1 && (6.32c) \\ \dot{\psi} &= \frac{1}{d} z_2 \sin(\theta - \psi) && (6.32d) \\ \dot{\theta} &= w_2 && (6.32e) \end{aligned}$$

or in matrix form

$$\begin{bmatrix} \dot{d} \\ \dot{z_1} \\ \dot{z_2} \\ \dot{\psi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} z_1 \\ \frac{1}{d} z_2^2 \sin^2(\theta - \psi)) \\ 0 \\ \frac{1}{d} z_2 \sin(\theta - \psi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \cos(\theta - \psi) \\ 1 \\ 0 \\ 0 \end{bmatrix} w_1 + \begin{bmatrix} 0 \\ -z_2 \sin(\theta - \psi) \\ 0 \\ 0 \\ 1 \end{bmatrix} w_2 \qquad (6.33)$$

which can be described as

$$\dot{x} = f(x) + g_1(x) w_1 + g_2(x) w_2 \qquad (6.34)$$

where $x = [d, z_1, z_2, \psi, \theta]^T$ is the state vector. The equilibrium for the line following behavior is at $z_1 = z_2 = v_0$ and $\theta = \psi = \beta$. Therefore, we select the outputs as

$$y_1 = k_1(x) = \psi \qquad (6.35a)$$

$$y_2 = k_2(x) = z_1 \qquad (6.35b)$$

so that in the resulting linearized system the state z_1 will be forced to converge v_0 and the state ψ will be forced to converge to β.

Now, we examine the first derivatives of the outputs

$$\begin{aligned} \dot{y_1} &= \dot{\psi} = \frac{1}{d} z_2 \sin(\theta - \psi) && (6.36a) \\ \dot{y_2} &= \dot{z_1} = w_1 \cos(\theta - \psi) - w_2 z_2 \sin(\theta - \psi) + \frac{1}{d} z_2^2 \sin^2(\theta - \psi)) && (6.36b) \end{aligned}$$

which may be written in matrix form as

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_2} \end{bmatrix} = \begin{bmatrix} \frac{1}{d} z_2 \sin(\theta - \psi) \\ \frac{1}{d} z_2^2 \sin^2(\theta - \psi)) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \cos(\theta - \psi) & -z_2 \sin(\theta - \psi) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{6.37}$$

which can also be described as

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_2} \end{bmatrix} = T_1(x) + A_1 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{6.38}$$

To be able to linearize the system the coefficient matrix A_1 should have full rank. However, it has rank 1. Note, that the inputs do not appear in the first input's time derivative. Therefore, we take the second time derivative of the first input y_1

$$\begin{aligned} \ddot{y_1} &= -\frac{\dot{d}}{d^2} z_2 \sin(\theta - \psi) + \frac{1}{d} \dot{z_2} \sin(\theta - \psi) + \frac{1}{d} z_2 (\dot{\theta} - \dot{\psi}) \cos(\theta - \psi) \\ &= -\frac{z_1 z_2}{d^2} \sin(\theta - \psi) + \frac{1}{d} w_1 \sin(\theta - \psi) + \frac{1}{d} z_2 w_2 \cos(\theta - \psi) - \frac{1}{d^2} z_2^2 \sin(\theta - \psi) \cos(\theta - \psi) \end{aligned} \tag{6.39}$$

then

$$\begin{bmatrix} \ddot{y_1} \\ \dot{y_2} \end{bmatrix} = \begin{bmatrix} -\frac{z_1 z_2}{d^2} \sin(\theta - \psi) - \frac{1}{d^2} z_2^2 \sin(\theta - \psi) \cos(\theta - \psi) \\ \frac{1}{d} z_2^2 \sin^2(\theta - \psi)) \end{bmatrix} + \begin{bmatrix} \frac{1}{d} \sin(\theta - \psi) & \frac{1}{d} z_2 \cos(\theta - \psi) \\ \cos(\theta - \psi) & -z_2 \sin(\theta - \psi) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

which can be summarized as

$$\begin{bmatrix} \ddot{y_1} \\ \dot{y_2} \end{bmatrix} = T_2(x) + A_2 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

The determinant of A_2 is $Det(A_2) = -z_2/d$. At the equilibrium $z_2 = v_0 \neq 0$ the determinant of A_2 is bounded away from zero. Therefore, matrix A_2 has full rank at steady state. Now, we can derive the normal states as

$$\xi_1^1 = k_1(x) = \psi \tag{6.40a}$$

$$\xi_2^1 = L_f k_1(x) \tag{6.40b}$$

$$\xi_1^2 = k_2(x) = z_1 \tag{6.40c}$$

where

$$
\begin{aligned}
\xi_2^1 &= L_f k_1(x) = \frac{\partial k_1(x)}{\partial x} f(x) \\
&= [0\ 0\ 0\ 1\ 0] \begin{bmatrix} z_1 \\ \frac{1}{d} z_2^2 \sin^2(\theta - \psi)) \\ 0 \\ \frac{1}{d} z_2 \sin(\theta - \psi) \\ 0 \end{bmatrix} \\
&= \frac{1}{d} z_2 \sin(\theta - \psi) = \dot{\xi}_1^1 \qquad (6.41)
\end{aligned}
$$

The time derivative of the second state is

$$
\begin{aligned}
\dot{\xi}_2^1 &= -\frac{z_1 z_2}{d^2} \sin(\theta - \psi) + \frac{1}{d} w_1 \sin(\theta - \psi) \\
&\quad + \frac{1}{d} z_2 w_2 \cos(\theta - \psi) - \frac{1}{d^2} z_2^2 \sin(\theta - \psi) \cos(\theta - \psi) \qquad (6.42)
\end{aligned}
$$

The time derivative of the third state is

$$\dot{\xi}_1^2 = w_1 \cos(\theta - \psi) - w_2 z_2 \sin(\theta - \psi) + \frac{1}{d} z_2^2 \sin^2(\theta - \psi)) \qquad (6.43)$$

Then the feedback linearizing inputs are calculated as

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = -{A_2}^{-1} T_2(x) + {A_2}^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \qquad (6.44)$$

The calculations are performed by a symbolic equation solver. The results are

$$
\begin{aligned}
w_1 &= p_2 - 2p_2 \sin^2\left(\frac{\psi - \theta}{2}\right) - dp_1 \sin(\psi - \theta) + \frac{1}{d} z_1 z_2 \sin^2(\psi - \theta) \qquad (6.45) \\
w_2 &= \frac{1}{dz_2} \left(z_2^2 \sin(\theta - \psi) - \frac{1}{2} z_1 z_2 \sin(2\psi - 2\theta) \right) \\
&\quad + \frac{1}{z_2} p_2 sin(\psi - \theta) + \frac{1}{z_2} dp_1 \cos(\psi - \theta) \qquad (6.46)
\end{aligned}
$$

Substituting the linearizing inputs we get the normal states as $\dot{\xi}_2^1 = p_1$ and $\dot{\xi}_1^2 = p_2$. Note that the normal states are decoupled.

$$
\begin{aligned}
\dot{\xi}_1^1 &= \xi_2^1 \qquad (6.47a) \\
\dot{\xi}_2^1 &= p_1 \qquad (6.47b)
\end{aligned}
$$

$$\dot{\xi}_1^2 = p_2 \qquad (6.48)$$

As in the SISO case, we again shift the states as $\xi_1^1 = \psi - \beta$ and $\xi_1^2 = z_1 - v_0$. Now, we can use any linear system controller design method to have stable systems at $\dot{d} = v_0$ and $\psi = \theta = \beta$. Consider the inputs p_1 and p_2 are linearly dependent on the normal states. Then the normal state space becomes

$$\begin{bmatrix} \dot{\xi}_1^1 \\ \dot{\xi}_2^1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ K_1^1 & K_1^2 \end{bmatrix} \begin{bmatrix} \xi_1^1 \\ \xi_2^1 \end{bmatrix} \tag{6.49}$$

and

$$\dot{\xi}_1^2 = K_2^1 \xi_1^2 \tag{6.50}$$

Selecting the proper values for the controller parameters, one can obtain stable responses. Note that the minimal dynamics should also be investigated. In the simulations all the states are observed to be stable. The stability of the unobservable states at the desired behavior or basically the zero dynamics is an issue which needs to be investigated.

Since the relative degree (3) is less than the number of states (5) we will select a the new states η_1 and η_2 independent of input i.e. orthogonal to the function $g(x)$.

$$\frac{\partial \eta}{\partial x} g(x) = 0 \Rightarrow \begin{bmatrix} \frac{\partial \eta_1}{\partial d} & \frac{\partial \eta_1}{\partial z_1} & \frac{\partial \eta_1}{\partial \psi} & \frac{\partial \eta_1}{\partial z_2} & \frac{\partial \eta_1}{\partial \theta} \\ \frac{\partial \eta_2}{\partial d} & \frac{\partial \eta_2}{\partial z_1} & \frac{\partial \eta_2}{\partial \psi} & \frac{\partial \eta_2}{\partial z_2} & \frac{\partial \eta_2}{\partial \theta} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \cos(\theta - \psi) & -z2 \sin(\theta - \psi) \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.51}$$

$$\frac{\partial \eta_1}{\partial z_2} + \frac{\partial \eta_1}{\partial z_1} \cos(\theta - \psi) = 0 \tag{6.52}$$

$$\frac{\partial \eta_2}{\partial \theta} + \frac{\partial \eta_2}{\partial z_1} z_2 \sin(\theta - \psi) = 0 \tag{6.53}$$

which means η_1 should be independent of z_1 and z_2 in addition to ξ states, and η_2 should be independent of θ and z_1. We select $\eta_1 = \theta - \beta$ and $\eta_2 = d$. Let us examine the zero dynamics of these variables.

The time derivative of η_1 is

$$\begin{aligned} \dot{\eta_1} &= w_2 \\ &= \frac{1}{dz_2}\left(z_2^2 \sin(\theta - \psi) - \frac{1}{2} z_1 z_2 \sin(2\psi - 2\theta)\right) && (6.54) \\ &\quad + \frac{1}{z_2} p_2 sin(\psi - \theta) + \frac{1}{z_2} d p_1 \cos(\psi - \theta) && (6.55) \end{aligned}$$

We first substitute the equations in (6.45) and (6.47) into this equation. And then we do the following replacement of parameters to obtain equations in reduced system domain

$$\psi = \xi_1^1 + \beta \tag{6.56}$$

$$z_1 = \xi_1^2 + v_0 \tag{6.57}$$

$$d = \eta_2 \tag{6.58}$$

$$\theta = \eta_1 + \beta \tag{6.59}$$

Now we use the zero dynamics of the system at $\xi_1^1 = \xi_2^1 = \xi_1^2 = 0$, then $\dot{\eta}_1$ becomes [1]

$$\dot{\eta}_1 = -\frac{\sin(2\eta_1)(K_2^1\eta_2 - v_0)}{2\eta_2} \tag{6.60}$$

Using the small angle assumption for linearization we get

$$\dot{\eta}_1 = -\eta_1 \frac{(K_2^1\eta_2 - v_0)}{\eta_2} \tag{6.61}$$

Here note that $\eta_2 = d \geq 0$, since it is the physical distance between agent and target. Therefore, the minimal dynamics of η_1 is locally asymptotically stable satisfying $K_2^1 d > v_0$.

The time derivative of second selected state $\eta_2 = d$ becomes

$$\begin{aligned} \dot{\eta}_2 &= z_1 \\ &= \xi_1^2 + v_0 \end{aligned} \tag{6.62}$$

The zero dynamics becomes

$$\dot{\eta}_2 = v_0$$

Here note that this state is not converging to a constant, instead it is increasing with a velocity of v_0. The behavior is indeed a tracking problem of the state. The variable d tracks the constant speed increasing behavior at the zero dynamics. This is completely consistent with the controlled dynamics of the system.

The last state we should examine is the integrator state of the agent z_2. This is directly related to the orientations and rotational speeds as in the following equation

$$z_2 = \frac{1}{d}\left[z_1 \sin(\theta - \psi)\right] \tag{6.63}$$

---

[1] The intermediate steps are performed with symbolic equation solver toolbox of Matlab, and not represented here due to the space requirements.

At the steady state ($z_1 = v_0, \theta - \psi = 0$) it becomes

$$z_2|_{eq} = 0 \tag{6.64}$$

which is again consistent with the controlled dynamics of the system.

6.2.4 Line Following - MIMO - Stationary Target - Simulation Results

In this section we present the results of the developed line following controller in the previous section. As mentioned above we may select underdamped or overdamped controller parameters. The first example is the overdamped behavior. For this behavior we select the parameters as $K_1^2 = -9 < 0$ and $-\left|K_1^2\right|^2/4 < K_1^1 = -15 < 0$ and $K_2^1 = -10 < 0$. In the lower part of the Figure 6.9 the change of the distance between robot and line (h) is presented. As seen the robot approaches the line with an overdamped response and stays steady at $h = 0$. The upper part of the figure shows the distance between robot and target (d). Robot gets far away from the target. The change of this distance becomes steady at $\dot{d} = v_0 = 10[units/s]$. In Figure 6.10 the change of angles ψ and θ are presented. As seen both angles converge to the slope $\beta = 120^o$.

The path of the vehicle is presented in Figure 6.11. As seen from the figure, the vehicle reaches the line with an overdamped response and continues to follow the line with slope $\beta = 120^o$.

The underdamped response is obtained by the controller parameters $K_2 = -5 < 0$ and $K_1 = -15 < -|K_2|^2/4$ and $K_2^1 = -10 < 0$. In Figure 6.12 the change of h and d are presented. As seen the response is underdamped. In the next Figure 6.13 the change of ψ and θ may be observed. The Figure 6.14 shows the path of the vehicle.

Note that, in the simulation results above the vehicle's initial orientation is 100^o and converges to $\beta = 120^o$ as time passes. The vehicle converges to $\beta = 120^o$ sloped line for any initial orientation. If the slope is given opposite $\beta = 120 + 180 = 300^o$ then the vehicle follows the path shown in Figure 6.8. The vehicle does not follow the shortest path, instead it always rotates in the CCW direction to approach the line. Therefore, the future work for this controller is to add strategies that make the vehicle follow the shortest path similar to the SISO controller case.

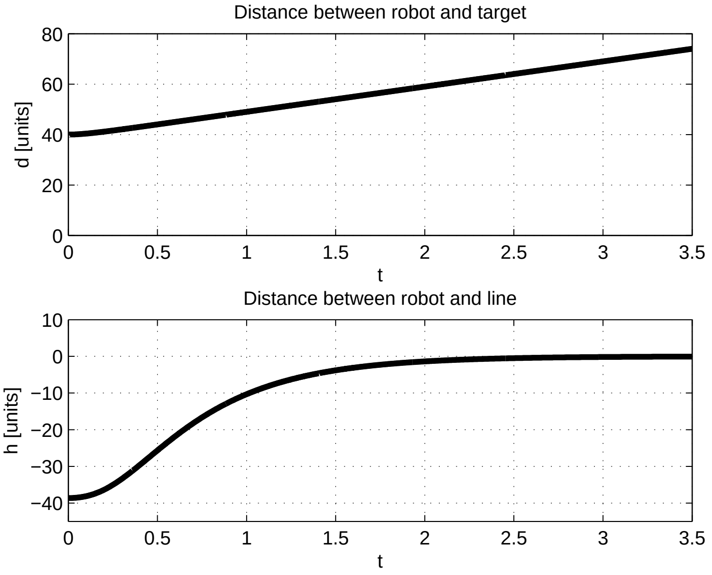


Figure 6.9: The change of the distance between robot and target (*d*), and robot and line (*h*) with respect to time. Overdamped response for line following behavior.

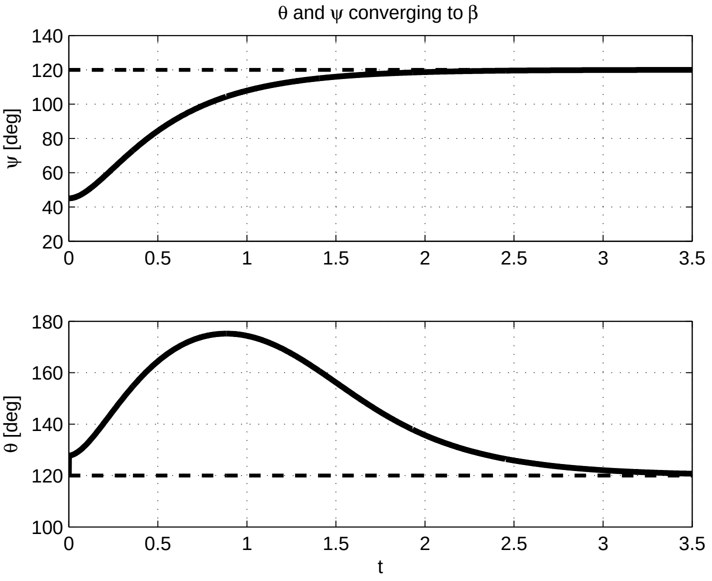


Figure 6.10: The change of the angles θ and ψ with respect to time. Overdamped response for line following behavior.

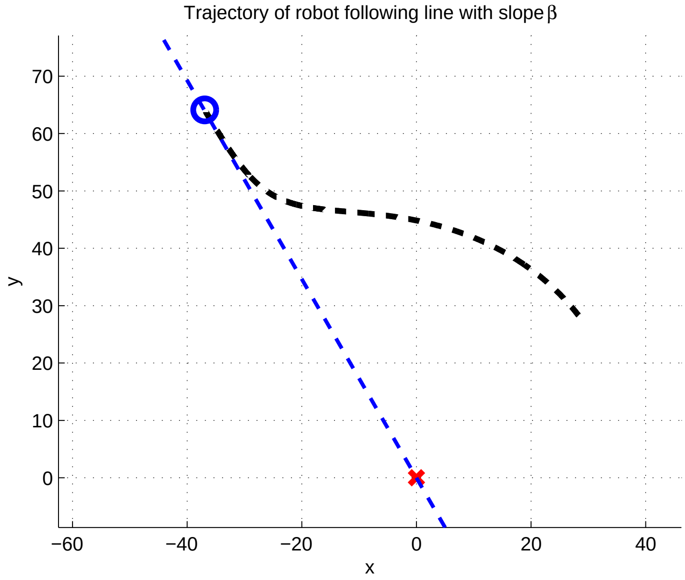


Figure 6.11: The path of the robot. Overdamped response for line following behavior.

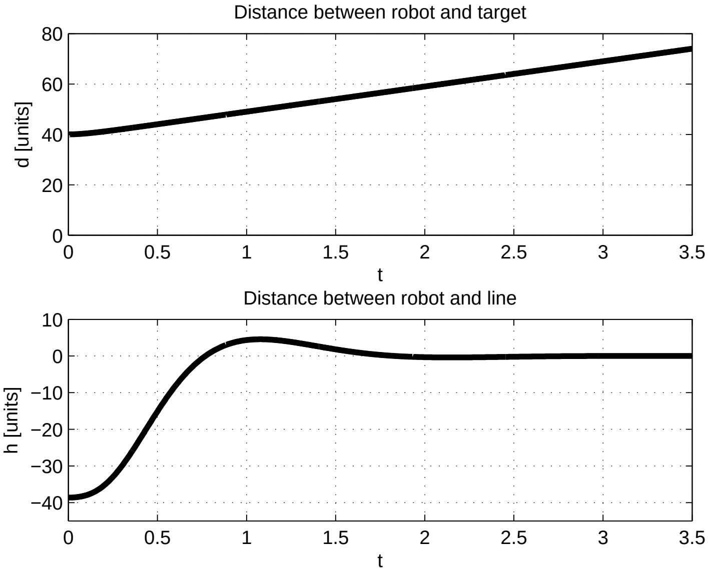


Figure 6.12: The change of the distance between robot and target (d), and robot and line (h) with respect to time. Underdamped response for line following behavior.

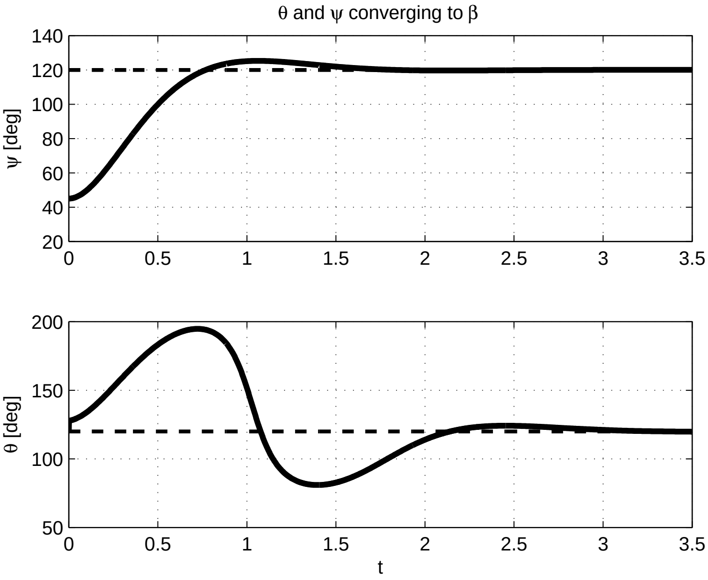


Figure 6.13: The change of the angles θ and ψ with respect to time. Underdamped response for line following behavior.

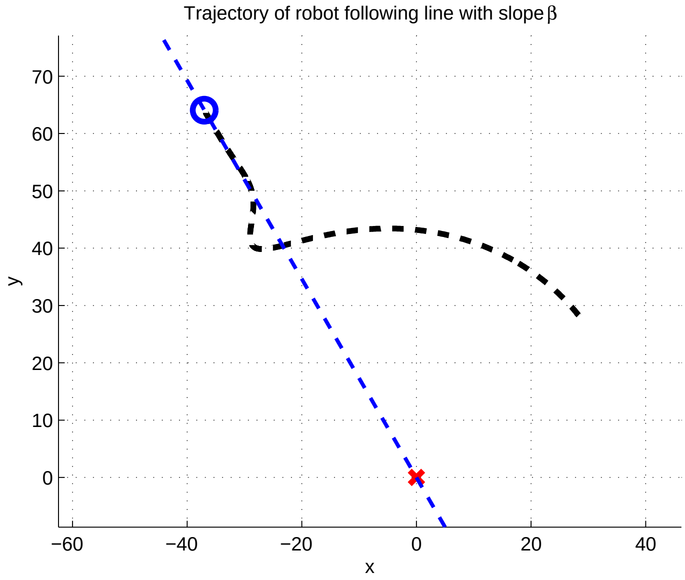


Figure 6.14: The path of the robot. Underdamped response for line following behavior.

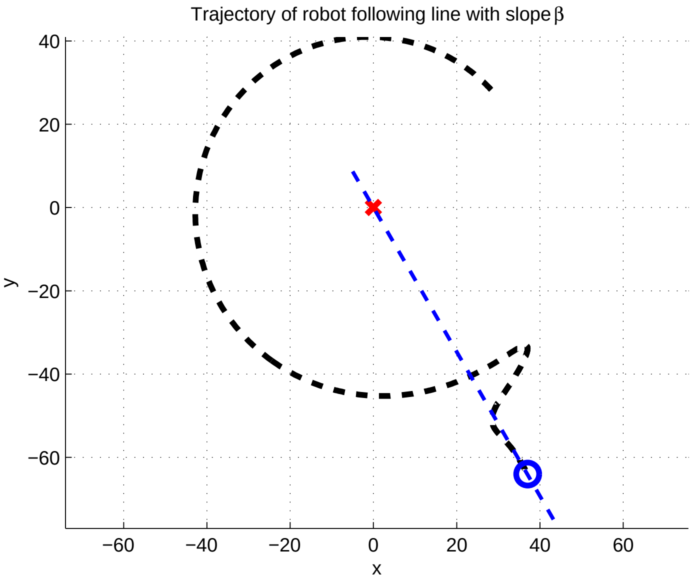


Figure 6.15: The path of the robot following the slope $\beta + \pi$.

6.2.5 Line Following - MIMO - Non-Stationary Target Case

In this section we will develop the feedback linearization of line following dynamics around a non-stationary target by taking speed v and angular velocity $\dot{\theta}$ as the inputs. Lets call $v = u_1$ and $\dot{\theta} = u_2$. Then the system dynamics becomes

$$\begin{aligned} \dot{d} &= u_1 \cos(\theta - \psi) - v_c \cos(\theta_c - \psi) && (6.65a) \\ \dot{\psi} &= \frac{1}{d}\left[u_1 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] && (6.65b) \\ \dot{\theta} &= u_2 && (6.65c) \end{aligned}$$

In the line following behavior, the vehicle travels at a specified speed $\dot{d} = v_0$ on the line passing through the target with the specified angle β. Since the controller is going to adjust the speed we need one more integrator for the distance variable d.

$$\begin{aligned} \dot{d} &= z_1 && (6.66a) \\ \dot{z_1} &= \dot{u}_1 \cos(\theta - \psi) - u_1(\dot{\theta} - \dot{\psi}) \sin(\theta - \psi) \\ &\quad - a_c \cos(\theta_c - \psi) + v_c(\dot{\theta}_c - \dot{\psi}) \sin(\theta_c - \psi) && (6.66b) \end{aligned}$$

For input u_1 we define a new integrator such that $u_1 = z_2$, then $\dot{z_2} = w_1$.

$$\begin{aligned}
\dot{d} &= z_1 && (6.67a)\\
\dot{z_1} &= w_1\cos(\theta-\psi) - z_2(\dot{\theta}-\dot{\psi})\sin(\theta-\psi) - a_c\cos(\theta_c-\psi) + v_c(\omega_c-\dot{\psi})\sin(\theta_c-\psi) && (6.67b)\\
\dot{z_2} &= w_1 && (6.67c)
\end{aligned}$$

For the convenience of inputs let us call $u_2 = w_2$. Then, substituting the other states we get

$$\begin{aligned}
\dot{d} &= z_1 && (6.68a)\\
\dot{z_1} &= w_1\cos(\theta-\psi) - w_2 z_2\sin(\theta-\psi) + z_2\sin(\theta-\psi)\frac{1}{d}\left[z_2\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right]\\
&\quad -a_c\cos(\theta_c-\psi) + v_c\sin(\theta_c-\psi)\omega_c\\
&\quad -v_c\sin(\theta_c-\psi)\frac{1}{d}\left[z_2\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right] && (6.68b)\\
\dot{\psi} &= \frac{1}{d}\left[z_2\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right] && (6.68c)\\
\dot{z_2} &= w_1 && (6.68d)\\
\dot{\theta} &= w_2 && (6.68e)
\end{aligned}$$

or in matrix form

$$\begin{bmatrix}\dot{d}\\ \dot{z_1}\\ \dot{\psi}\\ \dot{z_2}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix} z_1\\ f_{12}\\ \frac{1}{d}\left[z_2\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right]\\ 0\\ 0\end{bmatrix} + \begin{bmatrix}0\\ \cos(\theta-\psi)\\ 0\\ 1\\ 0\end{bmatrix} w_1 + \begin{bmatrix}0\\ -z_2\sin(\theta-\psi)\\ 0\\ 0\\ 1\end{bmatrix} w_2 \qquad (6.69)$$

where

$$\begin{aligned}
f_{12} &= z_2\sin(\theta-\psi)\frac{1}{d}\left[z_2\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right]\\
&\quad -a_c\cos(\theta_c-\psi) + v_c\sin(\theta_c-\psi)\omega_c\\
&\quad -v_c\sin(\theta_c-\psi)\frac{1}{d}\left[z_2\sin(\theta-\psi) - v_c\sin(\theta_c-\psi)\right] && (6.70a)
\end{aligned}$$

Defining functions of x we write

$$\dot{x} = f(x) + g_1(x)w_1 + g_2(x)w_2 \qquad (6.71)$$

where $x = [d, z_1, \psi, z_2, \theta]^T$ is the state vector.

The equilibrium for the line following behavior is at $z_1 = v_0$ and $\theta = \psi = \beta$. Therefore, we select the outputs as

$$y_1 = k_1(x) = \psi \tag{6.72a}$$

$$y_2 = k_2(x) = z_1 \tag{6.72b}$$

so that in the resulting linearized system the state z_1 will be forced to converge v_0 and the state ψ will be forced to converge to β.

Now, we examine the first derivatives of the outputs

$$\begin{aligned} \dot{y_1} = \dot{\psi} &= \frac{1}{d}\left[z_2 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] \qquad (6.73a) \\ \dot{y_2} = \dot{z_1} &= f_{12} + w_1 \cos(\theta - \psi) - w_2 z_2 \sin(\theta - \psi) \end{aligned}$$

which may be written in matrix form as

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_2} \end{bmatrix} = \begin{bmatrix} \dot{\psi} \quad = \quad \frac{1}{d}\left[z_2 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] \\ f_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \cos(\theta - \psi) & -z_2 \sin(\theta - \psi) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{6.74}$$

which can also be described as

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_2} \end{bmatrix} = T_1(x) + A_1 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{6.75}$$

To be able to linearize the system the coefficient matrix A_1 should have full rank. However, it has rank 1. Note, that the inputs do not appear in the first input's time derivative. Therefore, we take the second time derivative of the first input y_1

$$
\begin{aligned}
\ddot{y}_1 &= -\frac{\dot{d}}{d^2}\left[z_2\sin(\theta-\psi)-v_c\sin(\theta_c-\psi)\right] \\
&\quad +\frac{1}{d}\dot{z}_2\sin(\theta-\psi)+\frac{1}{d}z_2(\dot{\theta}-\dot{\psi})\cos(\theta-\psi) \\
&\quad -\frac{1}{d}a_c\sin(\theta_c-\psi) \\
&\quad -\frac{1}{d}v_c(\omega_c-\dot{\psi})\cos(\theta_c-\psi) \\
&= -\frac{z_1}{d^2}\left[z_2\sin(\theta-\psi)-v_c\sin(\theta_c-\psi)\right] \\
&\quad +\frac{1}{d}w_1\sin(\theta-\psi)+\frac{1}{d}z_2w_2\cos(\theta-\psi) \\
&\quad -\frac{1}{d^2}z_2\cos(\theta-\psi)\left[z_2\sin(\theta-\psi)-v_c\sin(\theta_c-\psi)\right] \\
&\quad -\frac{1}{d}a_c\sin(\theta_c-\psi) \\
&\quad -\frac{1}{d}v_c\omega_c\cos(\theta_c-\psi) \\
&\quad +\frac{1}{d^2}v_c\cos(\theta_c-\psi)\left[z_2\sin(\theta-\psi)-v_c\sin(\theta_c-\psi)\right]
\end{aligned}
\tag{6.76}
$$

then

$$
\begin{bmatrix}\ddot{y}_1\\ \dot{y}_2\end{bmatrix}=\begin{bmatrix}T_2^1\\ \frac{1}{d}z_2^2\sin^2(\theta-\psi))\end{bmatrix}+\begin{bmatrix}\frac{1}{d}\sin(\theta-\psi) & \frac{1}{d}z_2\cos(\theta-\psi)\\ \cos(\theta-\psi) & -z_2\sin(\theta-\psi)\end{bmatrix}\begin{bmatrix}w_1\\ w_2\end{bmatrix}
$$

which can be summarized as

$$
\begin{bmatrix}\ddot{y}_1\\ \dot{y}_2\end{bmatrix}=T_2(x)+A_2\begin{bmatrix}w_1\\ w_2\end{bmatrix}
$$

where

$$
\begin{aligned}
T_2^1 &= -\frac{z_1}{d^2}\left[z_2\sin(\theta-\psi)-v_c\sin(\theta_c-\psi)\right] \\
&\quad -\frac{1}{d^2}z_2\cos(\theta-\psi)\left[z_2\sin(\theta-\psi)-v_c\sin(\theta_c-\psi)\right] \\
&\quad -\frac{1}{d}a_c\sin(\theta_c-\psi) \\
&\quad -\frac{1}{d}v_c\omega_c\cos(\theta_c-\psi) \\
&\quad +\frac{1}{d^2}v_c\cos(\theta_c-\psi)\left[z_2\sin(\theta-\psi)-v_c\sin(\theta_c-\psi)\right]
\end{aligned}
\tag{6.77}
$$

The determinant of A_2 is $Det(A_2)=-z_2/d$. Note that at the equilibrium

$$
z_2^{eq} = \left.\frac{\dot{d}+v_c\cos(\theta_c-\psi)}{\cos(\theta-\psi)}\right|_{\dot{d}=v_0\ ,\ \theta=\psi=\beta} \tag{6.78}
$$

$$
= \frac{v_0+v_c\cos(\theta_c-\beta)}{\cos(\beta-\beta)} \tag{6.79}
$$

$$
= v_0+v_c\cos(\theta_c-\beta) \tag{6.80}
$$

Satisfying both $\theta_c \neq \beta + \pi$ and $v_c \neq v_0$, the determinant $Det(A_2) = -z_2/d$ is bounded away from zero. Therefore, matrix A_2 has full rank at steady state for the given conditions. Now, we can derive the normal states as

$$\xi_1^1 = k_1(x) = \psi \tag{6.81a}$$

$$\xi_2^1 = L_f k_1(x) \tag{6.81b}$$

$$\xi_1^2 = k_2(x) = z_1 \tag{6.81c}$$

where

$$\begin{aligned}
\xi_2^1 &= L_f k_1(x) = \frac{\partial k_1(x)}{\partial x} f(x) \\
&= [0\ 0\ 1\ 0\ 0] \begin{bmatrix} z_1 \\ f_{12} \\ \frac{1}{d}\left[z_2 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] \\ 0 \\ 0 \end{bmatrix} \\
&= \frac{1}{d}\left[z_2 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] = \dot{\xi}_1^1
\end{aligned} \tag{6.82}$$

The time derivative of the second state is

$$\begin{aligned}
\dot{\xi}_2^1 = & -\frac{z_1}{d^2}\left[z_2 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] \\
& + \frac{1}{d} w_1 \sin(\theta - \psi) + \frac{1}{d} z_2 w_2 \cos(\theta - \psi) \\
& - \frac{1}{d^2} z_2 \cos(\theta - \psi)\left[z_2 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right] \\
& - \frac{1}{d} a_c \sin(\theta_c - \psi) \\
& - \frac{1}{d} v_c \omega_c \cos(\theta_c - \psi) \\
& + \frac{1}{d^2} v_c \cos(\theta_c - \psi)\left[z_2 \sin(\theta - \psi) - v_c \sin(\theta_c - \psi)\right]
\end{aligned} \tag{6.83}$$

The time derivative of the third state is

$$\dot{\xi}_1^2 = f_{12} + w_1 \cos(\theta - \psi) - w_2 z_2 \sin(\theta - \psi) \tag{6.84}$$

Then the feedback linearizing inputs are calculated as

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = -A_2^{-1} T_2(x) + A_2^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \tag{6.85}$$

The calculations are performed by a symbolic equation solver. The results are

$$\begin{aligned}
w_1 \;=\;& p_2 \cos(\psi - \theta) \\
& - \cos(\psi - \theta) \left[\frac{1}{d} \left(v_c^2 \sin(\theta_c - \psi)^2 - 2 v_c z_2 \sin(\theta - \psi) \sin(\theta_c - \psi) + z_2^2 \sin(\theta - \psi)^2 \right) \right. \\
& \left. - a_c \cos(\theta_c - \psi) + \omega_c v_c \sin(\theta_c - \psi) \right] \\
& - d \sin(\psi - \theta) \left\{ \frac{1}{d^2} \left[(v_c \cos(\theta_c - \psi) - z_2 \cos(\theta - \psi)) (v_c \sin(\theta_c - \psi) - z_2 \sin(\theta - \psi)) \right] \right. \\
& + \frac{1}{d} (a_c \sin(\theta_c - \psi)) - \frac{1}{d^2} \left[z_1 (v_c \sin(\theta_c - \psi) - z_2 \sin(\theta - \psi)) \right] \\
& \left. + \frac{1}{d} (\omega_c v_c \cos(\theta_c - \psi)) \right\} \\
& - d p_1 \sin(\psi - \theta)
\end{aligned} \tag{6.86}$$

$$\begin{aligned}
w_2 \;=\;& \frac{1}{2 d z_2} \left\{ 2 d p_2 \sin(\psi - \theta) - 2 z_2^2 \sin(\psi - \theta) - v_c^2 \sin(\psi + \theta - 2\theta_c) \right. \\
& - 2 a_c d \sin(\theta - \theta_c) - v_c^2 \sin(\psi - \theta) + 3 v_c z_2 \sin(\psi - \theta_c) + v_c z_1 \sin(\theta - \theta_c) \\
& + 2 d^2 p_1 \cos(\psi - \theta) + v_c z_1 \sin(2\psi - \theta - \theta_c) - z_1 z_2 \sin(2\psi - 2\theta) \\
& \left. + v_c z_2 \sin(\psi - 2\theta + \theta_c) + 2 domega_c v_c \cos(\theta - \theta_c) \right\}
\end{aligned} \tag{6.87}$$

Substituting the linearizing inputs we get the normal states as $\dot{\xi}_2^1 = p_1$ and $\dot{\xi}_1^2 = p_2$. Note that the normal states are decoupled.

$$\dot{\xi}_1^1 = \xi_2^1 \tag{6.88a}$$

$$\dot{\xi}_2^1 = p_1 \tag{6.88b}$$

$$\dot{\xi}_1^2 = p_2 \tag{6.89}$$

As in the SISO case, we again shift the states as $\xi_1^1 = \psi - \beta$ and $\xi_1^2 = z_1 - v_0$. Now, we can use any linear system controller design method to have stable systems at $\dot{d} = v_0$ and $\psi = \theta = \beta$. Consider the inputs p_1 and p_2 are linearly dependent on the normal states. Then the normal state space becomes

$$\begin{bmatrix} \dot{\xi}_1^1 \\ \dot{\xi}_2^1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ K_1^1 & K_1^2 \end{bmatrix} \begin{bmatrix} \xi_1^1 \\ \xi_2^1 \end{bmatrix} \tag{6.90}$$

and

$$\dot{\xi}_1^2 = K_2^1 \xi_1^2 \tag{6.91}$$

Selecting the proper values for the controller parameters, one can obtain stable responses.

The minimal dynamics should also be investigated. In the simulations all the states are observed to be stable. Since the relative degree (3) is less than the number of states (5) we will select a the new states η_1 and η_2 independent of input i.e. orthogonal to the function $g(x)$.

$$\frac{\partial \eta}{\partial x} g(x) = 0 \Rightarrow \begin{bmatrix} \frac{\partial \eta_1}{\partial d} & \frac{\partial \eta_1}{\partial z_1} & \frac{\partial \eta_1}{\partial \psi} & \frac{\partial \eta_1}{\partial z_2} & \frac{\partial \eta_1}{\partial \theta} \\ \frac{\partial \eta_2}{\partial d} & \frac{\partial \eta_2}{\partial z_1} & \frac{\partial \eta_2}{\partial \psi} & \frac{\partial \eta_2}{\partial z_2} & \frac{\partial \eta_2}{\partial \theta} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \cos(\theta - \psi) & -z2\sin(\theta - \psi) \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.92}$$

$$\frac{\partial \eta_1}{\partial z_2} + \frac{\partial \eta_1}{\partial z_1} \cos(\theta - \psi) = 0 \tag{6.93}$$

$$\frac{\partial \eta_2}{\partial \theta} + \frac{\partial \eta_2}{\partial z_1} z_2 \sin(\theta - \psi) = 0 \tag{6.94}$$

which means in addition to ξ_1^1, ξ_2^1, and ξ_1^2, the state η_2 should be independent of z_1 and z_2. Similarly η_1 should be independent of θ and z_1 or it should satisfy the partial differential equation in (6.93) . We select $\eta_2 = d$ and $\eta_1 = z_2 \cos(\theta - \psi) - z_1$. Let us examine the zero dynamics of these variables.

The time derivative of η_2 is

$$\dot{\eta}_2 = z_1 \tag{6.95}$$

We first substitute the equations in (6.87) and (6.88) into this equation. And then we do the following replacement of parameters to obtain equations in reduced system domain

$$\psi = \xi_1^1 + \beta \tag{6.96}$$

$$z_1 = \xi_1^2 + v_0 \tag{6.97}$$

$$d = \eta_2 \tag{6.98}$$

Now we use the zero dynamics of the system at $\xi_1^1 = \xi_2^1 = \xi_1^2 = 0$, then $\dot{\eta}_2$ becomes [2]

$$\dot{\eta}_1|_{eq} = v_0 \tag{6.99}$$

Here note that this state is not converging to a constant, instead it is increasing with a speed of v_0 and diverging as $t \to \infty$. The behavior is indeed a tracking problem of the state. The

---

[2] The intermediate steps are performed with symbolic equation solver toolbox of Matlab

distance variable d tracks the constant speed travel behavior at the zero dynamics. This is completely consistent with the controlled dynamics of the system. Opposite to the stationary target case, in the dynamic targeting system θ does not converge to constant β. The change of θ with respect to time depends on the motion of the target. Furthermore, the velocity of the agent is not along the line passing through the target if target is not stationary. The velocity of the agent converges to a vector consisting normal and tangential components. The normal component of agent velocity is equal to normal component of the target velocity plus $\dot{d} = v_0$ (normal direction is along ψ direction). The tangential component of the agent velocity is equal to the tangential component of the target velocity. Therefore, we can write

$$z_2 \sin(\theta - \psi) = v_c \sin(\theta_c - \psi) \tag{6.100}$$

and

$$z_2 \cos(\theta - \psi) = v_c \cos(\theta_c - \psi) + z_1 \tag{6.101}$$

If we consider the relation between $\eta_1 = z_2 \cos(\theta - \psi) - z_1$ and the equations above we can simply say that η_2 should converge to $v_c \cos(\theta_c - \psi)$ at the steady state. In the analysis of the zero dynamics of η_1 it is hard to solve for this convergence. Therefore, here we use a simplifying method and examine the dynamics of a new variable say $\bar{\eta}_1 = \eta_1 - v_c \cos(\theta_c - \psi)$. The time derivative of this variable is

$$\begin{aligned}
\dot{\bar{\eta}}_1 \quad = \quad & (v_c \sin(\theta_c - \psi) - z_2 \sin(\theta - \psi))^2/d - v_c \sin(\theta_c - \psi) \\
& (\omega_c + (v_c \sin(\theta_c - \psi) - z_2 \sin(\theta - \psi))/d) \\
& + z_2 \sin(\theta - \psi)(w_2 + (v_c \sin(\theta_c - \psi) - z_2 \sin(\theta - \psi))/d) \\
& + \omega_c v_c \sin(\theta_c - \psi) - w_2 z_2 \sin(\theta - \psi)
\end{aligned}$$

Performing the substitutions done for η_1 again in the above equation we would get

$$\dot{\bar{\eta}}_2 = 0$$

which means $\dot{\eta}_2$ equals to $v_c \cos(\theta_c - \psi)$. Here we have two tracking variables. The results are all consistent with the controlled line following behavior of the agents.

6.2.6 Line Following - MIMO - Non-Stationary Target - Simulation Results

In this section we will present the results of the developed controller for non-stationary targets. For better understanding of the performance of non-stationary target following controller, we presented the results of the same simulation for the controller developed in the previous section for stationary targets. The controller parameters are the same for both controllers. We select underdamped controller parameters for both which are $K_2^1 = -5 < 0$ and $-\left|K_2^1\right|^2/4 > K_1^1 = -10$ and $K_1^2 = -1 < 0$. The target is stationary at the beginning of the simulation to show that the controllers are successfully in convergence to the reference inputs, after a while the target starts a complex motion and stops towards the end of the simulation. The target motion is modeled by particle kinematics similar to the agents.

$$\begin{aligned} \dot{x}_c &= v_c \cos(\theta_c) && (6.102a) \\ \dot{y}_c &= v_c \sin(\theta_c) && (6.102b) \\ \dot{v}_c &= a_c && (6.102c) \\ \dot{\theta}_c &= \omega_c && (6.102d) \end{aligned}$$

where for this simulation the acceleration and angular velocity of the target are selected to be partial functions as

$$a_c = \begin{cases} 0 & t \leq 5 \;\&\; t \geq 26 \\ 5\sin(0.3(t-5)) & 5 < t < 26 \end{cases} \tag{6.103}$$

$$\omega_c = \begin{cases} 0 & t \leq 5 \;\&\; t \geq 26 \\ 1.5\sin(0.8(t-5)) & 5 < t < 26 \end{cases} \tag{6.104}$$

Using the above dynamic target model, the simulation is performed for 30 seconds. The reference inputs are $v_0 = 5$ and $\beta = 35^o$. The agents with different controllers (stationary target and non-stationary target controllers) start with the same initial conditions for better comparison. In Figure 6.16 h, the distance between the agents and the line passing through agent and the target are plotted. In Figure 6.17 $e_\psi = \beta - \psi$, the errors of the desired input are plotted. At the begining of the simulation both agents have the same values of these states since the target is stationary. When target starts to move the controller for stationary target cannot keep the desired distance $h = 0$ and desired vector orientation, $e_\psi \neq 0$. When the target stops both agents again converge to the same desired values. The velocities of the agents can be compared in Figure 6.18. As seen the non-stationary controller utilizes more speed in inputs. The paths of both controllers are presented in the Figure 6.19.

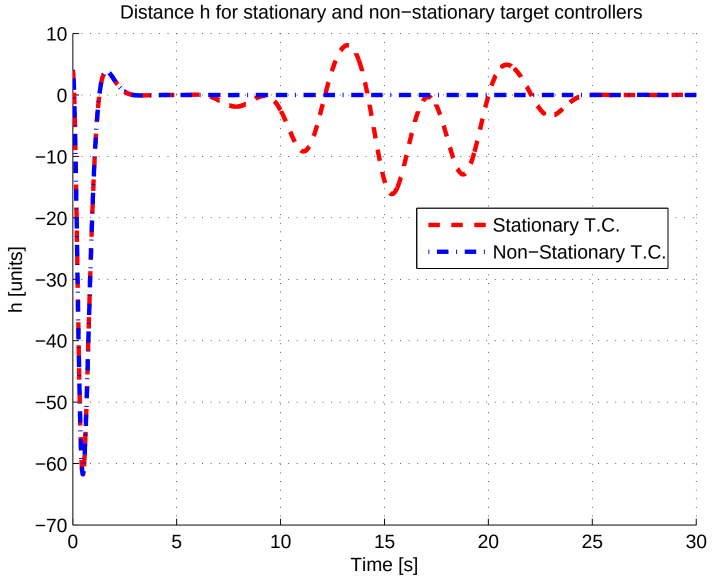


Figure 6.16: The distance between agents and target d. The responses of non-stationary and stationary line following controllers.

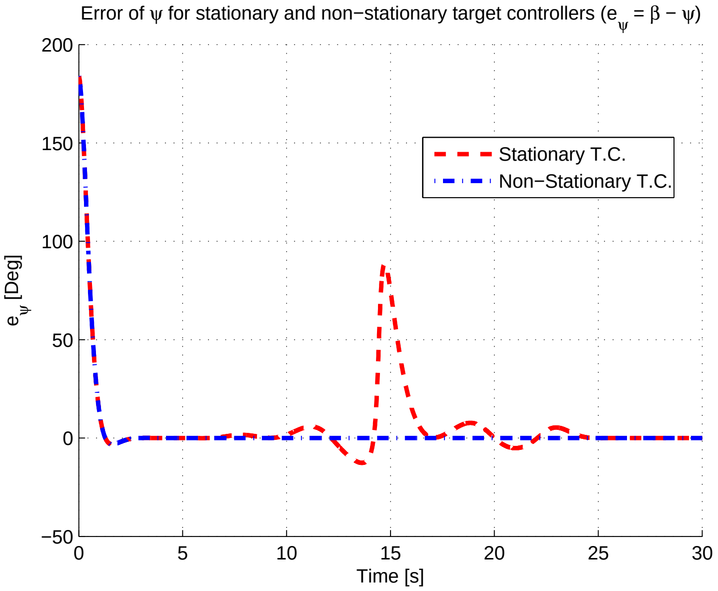


Figure 6.17: Error of ψ ($e_\psi = \beta - \psi$) for non-stationary and stationary line following controllers.

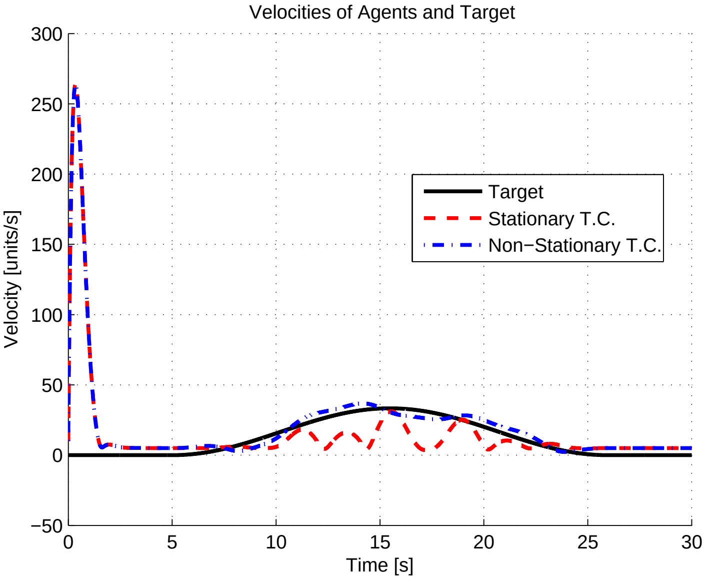


Figure 6.18: Velocities of target and agents with non-stationary and stationary line following controllers.

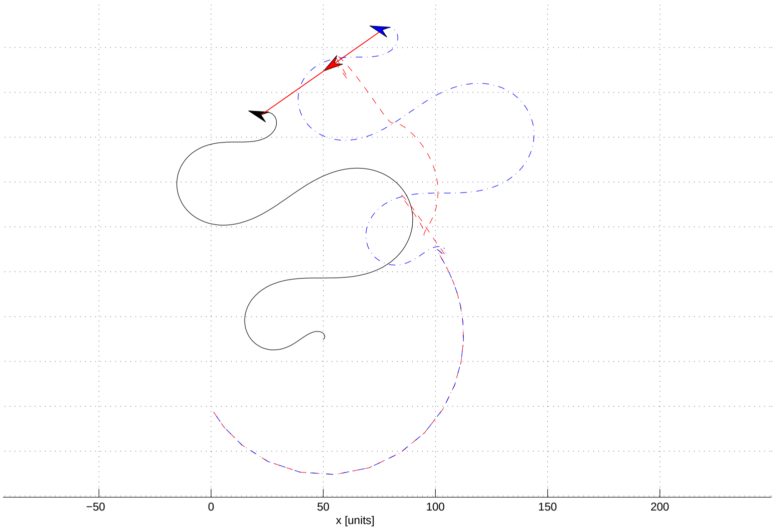


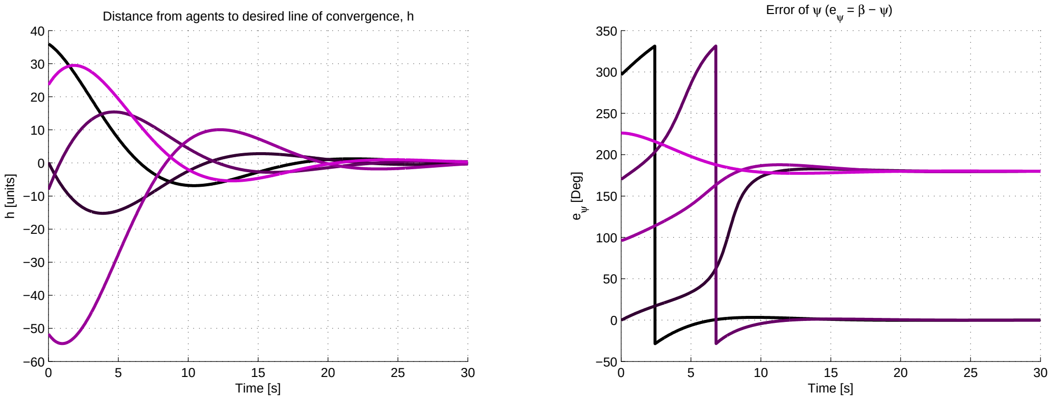
Figure 6.19: Path of the agents and target. Non-stationary controller path: dash-dotted blue, stationary controller path: dashed red, target path: solid black.

6.2.7 Application of Controllers to Multi-agent Systems

In this section, the controllers developed for line following behavior are applied to multi-agent systems. The developed controllers are designed for a single agent to follow a line passing through a a specified target with a specified slope. If the objective is to make all agents of a swarm to follow a line passing through a specified target with a specified slope independently, then the controllers are successful in doing so. However, in many multi-agent applications the aim is to force agents to perform a specified task independent of external inputs like leader, target, centralized controllers etc.. Therefore, in this part of the study we will develop methods for agents to perform the desired motion utilizing the centroid of the swarm. We will start with the SISO controller developed for a stationary target.

6.2.7.1 Line Formation of Multi-Agent System with SISO - Stationary Target Controller

In this section we present the simulation results of the line following behavior of multi agent systems where they use the center of the swarm as the target point. The simulations are run for 5 agents. The desired orientation for the agents to converge is $\beta = 35^o$. The controller parameters in equation (6.28) are set to $K_1 = -0.1$ and $K_2 = -0.3$ which results in underdamped response. In Figure 6.20 the change of distance from agent to the line of convergence, h (a) and change of orientation error ($e_\psi = \beta - \psi$) (b) are represented. The h variable converges to zero or in other words agents settle on the line passing through center of swarm with the desired slope. The transient response of h is underdamped. The left hand side plot, (b) shows that 2 of the agents are following $\psi = \beta$ and the remaining 3 are $\psi = \beta + \pi$. As mentioned before, the SISO controller is able to control just the distance h; therefore, the heading of the agents depend on the initial conditions. In several simulations with random initial conditions there occur the behaviors: All 5 agents heading $\psi = \beta$ or just the reverse ($\psi = \beta + \pi$) or one of the agents is heading in the opposite direction of the other 4, and the ones similar to the one represented here. In Figure 6.21 the paths of the agents are represented. The arrow head like geometries and the tail like dashed curves represents the agents and paths respectively. The cross shape is the center of the swarm. As seen from the figure the agents are converging to the desired line passing through the center of the swarm with the desired slope.



(a) (b)

Figure 6.20: Distance h (a) and error of e_ψ (b) of 5 agents with line following SISO controller

6.2.7.2 Line Formation of Multi-Agent System with MIMO - Stationary Target Controller

In this section the MIMO controller developed for the stationary targets is applied for the Multi-agent systems. The agents utilize the center of the swarm, but not exactly the location of the center as the target. Note that, the MIMO controller developed for stationary targets aims to go away with a specified speed $\dot{d} = z_1 = v_0$ (Eqn. 6.30) from the target. However, since all of the agents move to get far away from the center, the center also moves with the agent flock. Therefore, controllers of agents increase the speed and so does the center which results in divergence of the speed of the swarm. To avoid from this problem we will modify the target point deriving a simple relation with the center of the swarm. Note that, the developed controller aims the agents to converge to a line passing through the target with a specified slope. Therefore, we define a new point on the same line which does not move along the line. The center of the swarm changes location as the agents move. If we define such a point it will move just along the orthogonal direction of the flock. That point may be selected as the tangent point of circle centered at a pre-specified fixed point and the line passing through the center of the flock with the given orientation (see Fig. 6.22)

Using the geometry in Fig. 6.22 we can derive the relations between center of swarm (x_c, y_c) and the target point (x_t, y_t). For example selecting the fixed point as the origin of the coordi-

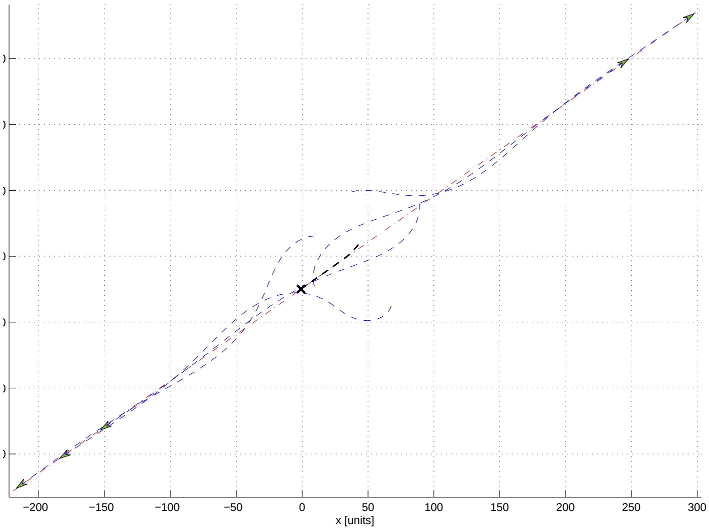


Figure 6.21: Path of 5 agents with line following SISO controller. Arrow heads: agents, Cross: center of swarm.

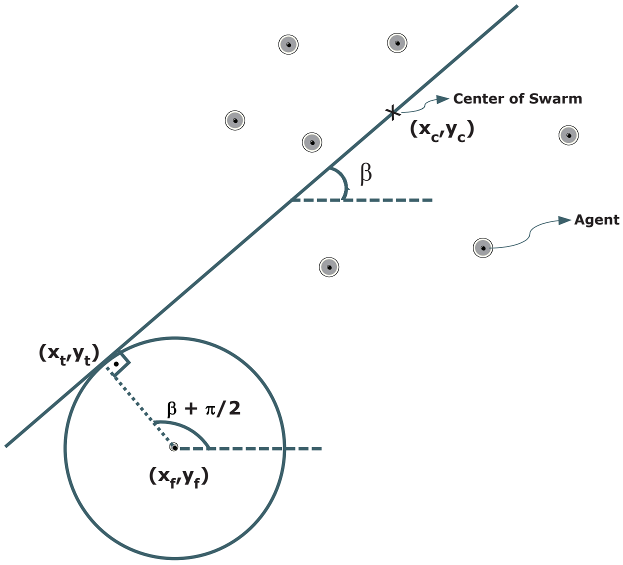


Figure 6.22: The relation between target and center of swarm.

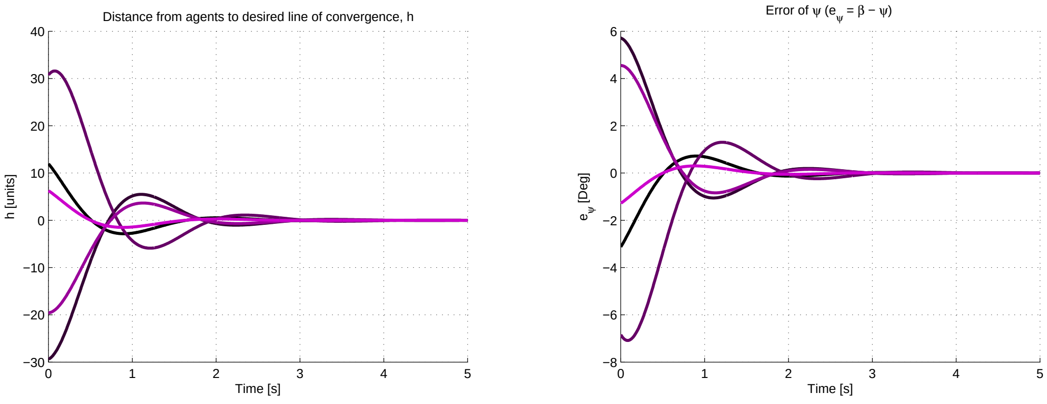
nate frame ($x_f = 0, y_f = 0$) the target position is generated as:

$$x_t = \frac{y_c - x_c \tan(\beta)}{\tan(\beta + \pi/2) - \tan(\beta)} \tag{6.105a}$$

$$y_t = x_t \tan(\beta + \pi/2) \tag{6.105b}$$

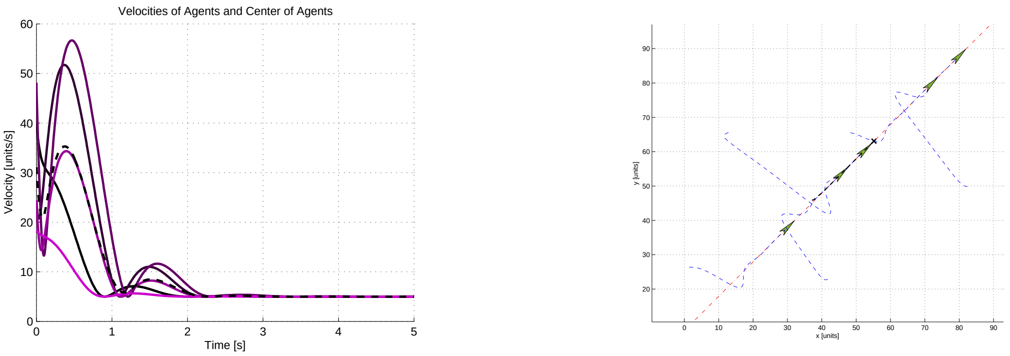
Using the new target derived from the center of agents some simulations are performed to observe the success of controllers. Here, we present the simulations for 5 agents. The controller parameters are selected as $K_1^1 = -10$, $K_2^1 = -3$, and $K_1^2 = -10$ to obtain an underdamped system. The reference input for orientation of line is $\beta = 35^o$, and reference input for speed of agents is $v_0 = 5 units/s$. The initial conditions are randomly generated to distribute the agents in a $200x200$ square area.

In Fig. 6.23(a) the change of the distances between agents and the line of convergence (h) for each agent are presented. As seen from the figure all h values converge to zero with damped oscillations. In 6.23(b) the change of $e_\psi = \beta - \psi$ values of agents are presented. The values converges to $e_\psi = 0$ with decaying exponential oscillations. In Fig. 6.24(a) the velocities of agents and the center of agents are plotted. The velocities are converging to the reference input $v_0 = 5 units/s$. On the right hand side the path of the agents are plotted. The arrow heads represents the agents and the cross sign is the center of the swarm. The paths are the tailing dashed curves.



(a) (b)

Figure 6.23: Distance h (a) and error e_ψ (b) of 5 agents with line following MIMO controller for stationary target.



(a) (b)

Figure 6.24: Velocities of agents and center of swarm (a), and path of 5 agents (b) with line following MIMO controller for stationary target.

6.2.7.3 Line Formation of Multi-Agent System with MIMO - Non-Stationary Target Controller

In this section the MIMO controller developed for non-stationary targets is modified for the multi-agent systems. The center of the swarm is taken to be the target of the agents (Again shifted backwards as explained in the previous section). Therefore, the target dynamics are coupled to the agent dynamics such that

$$\overrightarrow{v}_c = \frac{1}{M}\sum_{j=1}^{M}\overrightarrow{v}_j \tag{6.106a}$$

$$\overrightarrow{a}_c = \frac{1}{M}\sum_{j=1}^{M}\overrightarrow{a}_j \tag{6.106b}$$

where v_c and a_c are the velocity and acceleration vectors of the center of swarm, and v_j and a_j are the velocity and acceleration vectors of the jth agent. M is the number of agents in the swarm. And note that the velocity component along the line passing through the target (center of swarm) and the agent itself becomes

$$v_c\cos(\theta_c - \psi_i) = \frac{1}{M}\sum_{j=1}^{M}v_j\cos(\theta_j - \psi_i) \tag{6.107a}$$

$$v_c\sin(\theta_c - \psi_i) = \frac{1}{M}\sum_{j=1}^{M}v_j\sin(\theta_j - \psi_i) \tag{6.107b}$$

Substituting these relations into the mathematical model in (6.65) we obtain the following set of equations for each agent *i*

$$\begin{aligned}
\dot{d}_i &= v_i \cos(\theta_i - \psi_i) - \frac{1}{M}\sum_{j=1}^{M} v_j \cos(\theta_j - \psi_i) && \text{(6.108a)}\\
\dot{\psi}_i &= \frac{1}{d}\left[v_i \sin(\theta_i - \psi_i) - \frac{1}{M}\sum_{j=1}^{M} v_j \sin(\theta_j - \psi_i)\right] && \text{(6.108b)}\\
\dot{\theta}_i &= u_i && \text{(6.108c)}
\end{aligned}$$

If we extract the velocity components belonging to agent *i* in the equations we get

$$\begin{aligned}
\dot{d}_i &= v_i \cos(\theta_i - \psi_i) - \frac{1}{M}\left[v_i \cos(\theta_i - \psi_i) + \sum_{j=1,j\neq i}^{M} v_j \cos(\theta_j - \psi_i)\right] && \text{(6.109a)}\\
\dot{\psi}_i &= \frac{1}{d}\left\{v_i \sin(\theta_i - \psi_i) - \frac{1}{M}\left[v_i \sin(\theta_i - \psi_i) + \sum_{j=1,j\neq i}^{M} v_j \sin(\theta_j - \psi_i)\right]\right\} && \text{(6.109b)}\\
\dot{\theta}_i &= u_i && \text{(6.109c)}
\end{aligned}$$

Rearranging the terms

$$\begin{aligned}
\dot{d}_i &= v_i\left(1 - \frac{1}{M}\right)\cos(\theta_i - \psi_i) - \frac{1}{M}\sum_{j=1,j\neq i}^{M} v_j \cos(\theta_j - \psi_i) && \text{(6.110a)}\\
\dot{\psi}_i &= \frac{1}{d}\left[v_i\left(1 - \frac{1}{M}\right)\sin(\theta_i - \psi_i) - \frac{1}{M}\sum_{j=1,j\neq i}^{M} v_j \sin(\theta_j - \psi_i)\right] && \text{(6.110b)}\\
\dot{\theta}_i &= u_i && \text{(6.110c)}
\end{aligned}$$

In this new model the terms in the summation are independent of the velocity of *i*th agent. Therefore, lets simplify the equation by the following relations

$$\begin{aligned}
\vec{v}_c^{\,i} &= \frac{1}{M}\sum_{j=1,j\neq i}^{M} \vec{v}_j && \text{(6.111a)}\\
\theta_c^i &= angle(\vec{v}_c^{\,i}) && \text{(6.111b)}
\end{aligned}$$

where $\overrightarrow{v}_c^i$ is the vectorial summation of the velocity of all agents except ith agent, divided by the number of agents, M. The variable θ_c^i is the orientation of vector $\vec{v}_c^i$. Let us define $v_c^i = norm(\vec{v}_c^i)$. Note that these variables are independent of the states of ith agent.

The summation of the components of the velocity vectors along and orthogonal to the line passing through center of swarm and the ith agent becomes

$$v_c^i \cos(\theta_c^i - \psi_i) = \frac{1}{M} \sum_{j=1, j\neq i}^{M} v_j \cos(\theta_j - \psi_i) \tag{6.112a}$$

$$v_c^i \sin(\theta_c^i - \psi_i) = \frac{1}{M} \sum_{j=1, j\neq i}^{M} v_j \sin(\theta_j - \psi_i) \tag{6.112b}$$

Substituting these into the system model

$$\dot{d}_i = v_i \left(1 - \frac{1}{M}\right) \cos(\theta_i - \psi_i) - v_c^i \cos(\theta_c^i - \psi_i) \tag{6.113a}$$

$$\dot{\psi}_i = \frac{1}{d}\left[v_i \left(1 - \frac{1}{M}\right) \sin(\theta_i - \psi_i) - v_c^i \sin(\theta_c^i - \psi_i)\right] \tag{6.113b}$$

$$\dot{\theta}_i = u_i \tag{6.113c}$$

For the line formation of agents, the equilibrium occurs at equal speeds and orientations of the agents on the same line with given orientation. In other words $v_i = v_0$, $\theta_i = \theta_0$, and $\psi_i = \beta$ for $i = 1, 2, ..., M$. Then the equilibrium states become

$$\begin{aligned} \dot{d}_i^{eq} &= v_0 \left(1 - \frac{1}{M}\right) \cos(\theta_0 - \beta) - \frac{M-1}{M} v_0 \cos(\theta_0 - \beta) \\ &= 0 \end{aligned} \tag{6.114a}$$

$$\begin{aligned} \dot{\psi}_i^{eq} &= \frac{1}{d}\left[v_0 \left(1 - \frac{1}{M}\right) \sin(\theta_0 - \beta) - \frac{M-1}{M} v_0 \sin(\theta_0 - \beta)\right] \\ &= 0 \end{aligned} \tag{6.114b}$$

$$\dot{\theta}_i = 0 \tag{6.114c}$$

Note that, the equilibrium is satisfied without regarding the value of θ_0 and v_0. If the agents are located on the line passing through the center and moving along parallel paths oriented with $\theta_i = \theta_0$ then the above dynamics are at equilibrium (see Fig. 6.25). The values of θ_0 and v_0 are emergent values depending on the initial conditions and controller parameters.

The aim of the controller in this section is not only to settle the agents on the line in Fig. 6.25 but also to make their path coincident with the line, ($\theta_i = \psi_i = \beta$). We will define a new

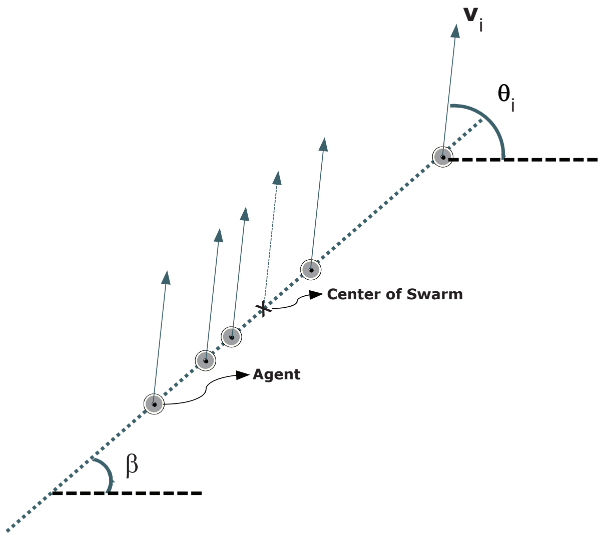


Figure 6.25: Line formation of agents at equilibrium.

virtual reference frame where the agents are targeting not the exact center of flock, instead they target a slower center of flock. The center of flock still depends on the positions of all agents, however the velocity and acceleration of the new center is smaller than the exact one. Therefore, we are going to define a new center of flock velocity by weighting the exact center of flock velocity such that $v_{cw} = Kv_c$ $(0 < K < 1)$. Note here also that the agents will converge to states where $\dot{d} = 0$ in real coordinate frames. But for this new virtual coordinate frame the convergence occurs at $\dot{d} = v_0(1 - K)$.

For the virtual slower center of flock the states become

$$\dot{d}_i = v_i \cos(\theta_i - \psi_i) - K\frac{1}{M}\sum_{j=1}^{M} v_j \cos(\theta_j - \psi_i) \tag{6.115a}$$

$$\dot{\psi}_i = \frac{1}{d}\left[v_i \sin(\theta_i - \psi_i) - K\frac{1}{M}\sum_{j=1}^{M} v_j \sin(\theta_j - \psi_i)\right] \tag{6.115b}$$

$$\dot{\theta}_i = u_i \tag{6.115c}$$

and rearranging the states

$$\dot{d}_i = v_i\left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i) - Kv_c^i \cos(\theta_c^i - \psi_i) \tag{6.116a}$$

$$\dot{\psi}_i = \frac{1}{d}\left[v_i\left(1 - \frac{K}{M}\right)\sin(\theta_i - \psi_i) - Kv_c^i \sin(\theta_c^i - \psi_i)\right] \tag{6.116b}$$

$$\dot{\theta}_i = u_i \tag{6.116c}$$

In the line following behavior, the vehicle travels at a specified speed $\dot{d} = v_0(1-K)$ on the line passing through the target with the specified angle β. Since the controller is going to adjust the speed $v_i = u_1^i$ we need one more integrator for the distance variable d.

$$
\begin{aligned}
\dot{d}_i &= z_1^i && (6.117a)\\
\dot{z}_1^i &= \dot{u}_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - u_1^i\left(1-\frac{K}{M}\right)(\dot{\theta}_i-\dot{\psi}_i)\sin(\theta_i-\psi_i)\\
&\quad -Ka_c^i\cos(\theta_c^i-\psi_i) + Kv_c^i(\dot{\theta}_c^i-\dot{\psi}_i)\sin(\theta_c^i-\psi_i) && (6.117b)
\end{aligned}
$$

for input u_1^i we define a new integrator such that $u_1^i = z_2^i$, then $\dot{z}_2^i = w_1^i$.

$$
\begin{aligned}
\dot{d}_i &= z_1^i && (6.118a)\\
\dot{z}_1^i &= w_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - z_2^i\left(1-\frac{K}{M}\right)(\dot{\theta}_i-\dot{\psi}_i)\sin(\theta_i-\psi_i)\\
&\quad -Ka_c^i\cos(\theta_c^i-\psi_i) + Kv_c^i(\omega_c^i-\dot{\psi}_i)\sin(\theta_c^i-\psi_i) && (6.118b)\\
\dot{z}_2^i &= w_1^i && (6.118c)
\end{aligned}
$$

For the convenience of inputs let us call $u_2^i = w_2^i$. Then, substituting the other states we get

$$
\begin{aligned}
\dot{d}_i &= z_1^i && (6.119a)\\
\dot{z}_1^i &= w_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - w_2^i z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)\\
&\quad +z_2^i\sin(\theta_i-\psi_i)\frac{1}{d_i}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi^i)\right]\\
&\quad -Ka_c^i\cos(\theta_c^i-\psi_i) + Kv_c^i\sin(\theta_c^i-\psi_i)\omega_c^i\\
&\quad -Kv_c^i\sin(\theta_c^i-\psi_i)\frac{1}{d_i}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] && (6.119b)\\
\dot{\psi}_i &= \frac{1}{d_i}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi^i)\right] && (6.119c)\\
\dot{z}_2^i &= w_1^i && (6.119d)\\
\dot{\theta}_i &= w_2^i && (6.119e)
\end{aligned}
$$

or in matrix form

$$\begin{bmatrix} \dot{d}_i \\ \dot{z}_1^i \\ \dot{\psi}_i \\ \dot{z}_2^i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} z_1^i \\ f_{12}^1 \\ \frac{1}{d_i}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c\sin(\theta_c^i-\psi_i)\right] \\ 0 \\ 0 \end{bmatrix}$$

$$+\begin{bmatrix} 0 \\ \left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) \\ 0 \\ 1 \\ 0 \end{bmatrix} w_1 + \begin{bmatrix} 0 \\ -z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) \\ 0 \\ 0 \\ 1 \end{bmatrix} w_2 \qquad (6.120)$$

where

$$\begin{aligned} f_{12}^1 = {} & z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)\frac{1}{d_i}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] \\ & -Ka_c^i\cos(\theta_c^i-\psi_i) + Kv_c^i\sin(\theta_c^i-\psi_i)\omega_c^i \\ & -Kv_c^i\sin(\theta_c^i-\psi_i)\frac{1}{d}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] \qquad (6.121a) \end{aligned}$$

Defining functions of x_i we write

$$\dot{x}_i = f(x_i) + g_1^1(x_i)w_1^i + g_2^1(x_i)w_2^i \qquad (6.122)$$

where $x = [d_i, z_1^1, \psi_i, z_2^1, \theta_i]^T$ is the state vector.

The equilibrium for the line following behavior is at $z_1^i = v_0(1-K)$ and $\theta_i = \psi_i = \beta$. Therefore, we select the outputs as

$$y_1 = k_1(x) = \psi_i \qquad (6.123a)$$

$$y_2 = k_2(x) = z_1^i \qquad (6.123b)$$

so that in the resulting linearized system the state z_1^i will be forced to converge $v_0(1-K)$ and the state ψ_i will be forced to converge to β.

Now, we examine the first derivatives of the outputs

$$\dot{y}_1^i = \dot{\psi}_i = \frac{1}{d_i}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] \qquad (6.124a)$$

$$\dot{y}_2^i = \dot{z}_1^i = f_{12} + w_1^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) - w_2^i z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) \qquad (6.124b)$$

which may be written in matrix form as

$$\begin{aligned}\begin{bmatrix}\dot{y}_1^i \\ \dot{y}_2^i\end{bmatrix} &= \begin{bmatrix}\dot{\psi}_i & = & \frac{1}{d}\left[z_2\left(1-\frac{K}{M}\right)\sin(\theta-\psi) - Kv_c\sin(\theta_c-\psi)\right] \\ f_{12}^i & & \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & 0 \\ \cos(\theta_i-\psi_i) & -z_2^i\sin(\theta_i-\psi_i)\end{bmatrix}\begin{bmatrix}w_1^i \\ w_2^i\end{bmatrix} \end{aligned} \tag{6.125}$$

which can also be described as

$$\begin{bmatrix}\dot{y}_1^i \\ \dot{y}_2^i\end{bmatrix} = T_1^i(x) + A_1^i\begin{bmatrix}w_1^i \\ w_2^i\end{bmatrix} \tag{6.126}$$

To be able to linearize the system the coefficient matrix A_1^i should have full rank. However, it has rank 1. Note, that the inputs do not appear in the first input's time derivative. Therefore, we take the second time derivative of the first input y_1^i

$$\begin{aligned}\ddot{y}_1^i &= -\frac{\dot{d}_i}{d_i^2}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] \\ &\quad +\frac{1}{d_i}\dot{z}_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) + \frac{1}{d_i}z_2^i\left(1-\frac{K}{M}\right)(\dot{\theta}_i-\dot{\psi}_i)\cos(\theta_i-\psi_i) \\ &\quad -\frac{1}{d_i}Ka_c^i\sin(\theta_c^i-\psi_i) \\ &\quad -\frac{1}{d_i}Kv_c^i(\omega_c^i-\dot{\psi}_i)\cos(\theta_c^i-\psi_i) \\ &= -\frac{z_1}{d_i^2}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] \\ &\quad +\frac{1}{d_i}w_1\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) + \frac{1}{d_i}z_2^iw_2\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) \\ &\quad -\frac{1}{d_i^2}z_2^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i)\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right] \\ &\quad -\frac{1}{d_i}Ka_c^i\sin(\theta_c^i-\psi_i) \\ &\quad -\frac{1}{d_i}Kv_c^i\omega_c^i\cos(\theta_c^i-\psi_i) \\ &\quad +\frac{1}{d_i^2}Kv_c^i\cos(\theta_c^i-\psi_i)\left[z_2^i\sin(\theta_i-\psi_i) - Kv_c^i\sin(\theta_c^i-\psi_i)\right]\end{aligned} \tag{6.127}$$

then

$$\begin{bmatrix}\ddot{y}_1^i \\ \dot{y}_2^i\end{bmatrix} = \begin{bmatrix} T_2^1 \\ \frac{1}{d_i}(z_2^i)^2\sin^2(\theta_i-\psi_i))\end{bmatrix} + \begin{bmatrix}\frac{1}{d_i}\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) & \frac{1}{d_i}z_2^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) \\ \left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) & -z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)\end{bmatrix}\begin{bmatrix}w_1 \\ w_2\end{bmatrix}$$

which can be summarized as

$$\begin{bmatrix} \ddot{y}^i_1 \\ \dot{y}^i_2 \end{bmatrix} = T^i_2(x) + A^i_2 \begin{bmatrix} w^i_1 \\ w^i_2 \end{bmatrix}$$

where

$$\begin{aligned}
T^1_2 &= -\frac{z_1}{d_i^2}\left[z^i_2\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv^i_c\sin(\theta^i_c-\psi_i)\right] \\
&\quad -\frac{1}{d_i^2}z^i_2\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i)\left[z^i_2\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv^i_c\sin(\theta^i_c-\psi_i)\right] \\
&\quad -\frac{1}{d_i}Ka^i_c\sin(\theta^i_c-\psi_i) \\
&\quad -\frac{1}{d_i}Kv^i_c\omega^i_c\cos(\theta^i_c-\psi_i) \\
&\quad +\frac{1}{d_i^2}Kv^i_c\cos(\theta^i_c-\psi_i)\left[z^i_2\sin(\theta_i-\psi_i) - Kv^i_c\sin(\theta^i_c-\psi_i)\right] \qquad (6.128)
\end{aligned}$$

The determinant of A^i_2 is $Det(A^i_2) = -z^i_2/d_i$. Note that at the equilibrium

$$\begin{aligned}
z^i_2 &= \left.\frac{\dot{d}_i + Kv^i_c\cos(\theta^i_c-\psi_i)}{\cos(\theta_i-\psi_i)}\right|_{\dot{d}_i=v_0(1-K)\ ,\ \theta_i=\psi_i=\beta} \qquad (6.129) \\
&= \frac{v_0(1-K) + Kv^i_c\cos(\theta^i_c-\beta)}{\cos(\beta-\beta)} \qquad (6.130) \\
&= v_0(1-K) + Kv^i_c\cos(\theta^i_c-\beta) \qquad (6.131)
\end{aligned}$$

The determinant $Det(A_2) = -z^i_2/d_i$ is bounded away from zero. Therefore, matrix A_2 has full rank at steady state and during transient response since $z^i_2 \neq 0$, for the given conditions. Now, we can derive the normal states as

$$\begin{aligned}
\xi_1^{1^i} &= k^i_1(x_i) = \psi_i \qquad (6.132a) \\
\xi_2^{1^i} &= L_f k^i_1(x_i) \qquad (6.132b) \\
\xi_1^{2^i} &= k^i_2(x_i) = z^i_1 \qquad (6.132c)
\end{aligned}$$

where

$$\begin{aligned}
\xi_2^{1^i} &= L_f k^i_1(x_i) = \frac{\partial k^i_1(x_i)}{\partial x_i} f(x_i) \\
&= [0\ 0\ 1\ 0\ 0]\begin{bmatrix} z^i_1 \\ f^i_{12} \\ \frac{1}{d_i}\left[z^i_2\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv^i_c\sin(\theta^i_c-\psi_i)\right] \\ 0 \\ 0 \end{bmatrix} \\
&= \frac{1}{d_i}\left[z^i_2\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) - Kv^i_c\sin(\theta^i_c-\psi_i)\right] = \dot{\xi}^1_1 \qquad (6.133)
\end{aligned}$$

The time derivative of the second state is

$$\begin{aligned}
\dot{\xi}^{1\,i}_{2} \;=\; & -\frac{z_1}{d_i^2}\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)-Kv_c^i\sin(\theta_c^i-\psi_i)\right] \\
& +\frac{1}{d_i}w_1\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)+\frac{1}{d_i}z_2^i w_2\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) \\
& -\frac{1}{d_i^2}z_2^i\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i)\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)-Kv_c^i\sin(\theta_c^i-\psi_i)\right] \\
& -\frac{1}{d_i}Ka_c^i\sin(\theta_c^i-\psi_i) \\
& -\frac{1}{d_i}Kv_c^i\omega_c^i\cos(\theta_c^i-\psi_i) \\
& +\frac{1}{d_i^2}Kv_c^i\cos(\theta_c^i-\psi_i)\left[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)-Kv_c^i\sin(\theta_c^i-\psi_i)\right] \qquad (6.134)
\end{aligned}$$

The time derivative of the third state is

$$\dot{\xi}^{2\,i}_{1} = f_{12}^i + w_1^i\cos(\theta_i-\psi_i) - w_2^i z_2^i\sin(\theta_i-\psi_i) \qquad (6.135)$$

Then the feedback linearizing inputs are calculated as

$$\begin{bmatrix} w_1^i \\ w_2^i \end{bmatrix} = -{A_2^i}^{-1}T_2^i(x) + {A_2^i}^{-1}\begin{bmatrix} p_1^i \\ p_2^i \end{bmatrix} \qquad (6.136)$$

The calculations are performed by a symbolic equation solver. The results are

$$\begin{aligned}
w_1^i \;=\; & (M\cos(\psi_i-\theta_i)(Kv_c^i\sin(\theta_c^i-\psi_i)(\omega_c^i+(Kv_c^i\sin(\theta_c^i-\psi_i) \\
& +z_2^i\sin(\theta_i-\psi_i)(K/M-1))/d_i) \\
& -Ka_c^i\cos(\theta_c^i-\psi_i)+(z_2^i\sin(\theta_i-\psi_i)(Kv_c^i\sin(\theta_c^i-\psi_i)+z_2^i\sin(\theta_i-\psi_i) \\
& (K/M-1))(K/M-1))/d_i))/(K-M)-(Mp_2^i\cos(\psi_i-\theta_i))/(K-M) \\
& +(Md_ip_1^i\sin(\psi_i-\theta_i))/(K-M) \\
& +(Md_i\sin(\psi_i-\theta_i)(((Kv_c^i\cos(\theta_c^i-\psi_i)+z_2^i\cos(\theta_i-\psi_i)(K/M-1)) \\
& (Kv_c^i\sin(\theta_c^i-\psi_i)+z_2^i\sin(\theta_i-\psi_i)(K/M-1)))/d_i^2-(z_1^i(Kv_c^i\sin(\theta_c^i-\psi_i) \\
& +z_2^i\sin(\theta_i-\psi_i)(K/M-1)))/d_i^2+(Ka_c^i\sin(\theta_c^i-\psi_i))/d_i \\
& +(K\omega_c^iv_c^i\cos(\theta_c^i-\psi_i))/d_i))/(K-M) \qquad (6.137)
\end{aligned}$$

$$
\begin{aligned}
w_2^i \quad = \quad & (M\sin(\psi_i-\theta_i)(Kv_c^i\sin(\theta_c^i-\psi_i)(\omega_c^i+(Kv_c^i\sin(\theta_c^i-\psi_i) \\
& +z_2^i\sin(\theta_i-\psi_i)(K/M-1))/d_i)-Ka_c^i\cos(\theta_c^i-\psi_i) \\
& +(z_2^i\sin(\theta_i-\psi_i)(Kv_c^i\sin(\theta_c^i-\psi_i) \\
& +z_2^i\sin(\theta_i-\psi_i)(K/M-1))(K/M-1))/d_i))/(z_2^i(K-M)) \\
& -(Mp_2^i\sin(\psi_i-\theta_i))/(z_2^i(K-M)) \\
& -(Md_ip_1^i\cos(\psi_i-th))/(z_2^i(K-M))-(Md_i\cos(\psi_i-\theta_i)(((Kv_c^i\cos(\theta_c^i-\psi_i) \\
& +z_2^i\cos(\theta_i-\psi_i)(K/M-1))(Kv_c^i\sin(\theta_c^i-\psi_i)+z_2^i\sin(\theta_i-\psi_i)(K/M-1)))/d_i^2 \\
& -(z_1^i(Kv_c^i\sin(\theta_c^i-\psi_i)+z_2^i\sin(\theta_i-\psi_i)(K/M-1)))/d_i^2+(Ka_c^i\sin(\theta_c^i-\psi_i))/d_i \\
& +(K\omega_c^iv_c^i\cos(\theta_c^i-\psi_i))/d_i))/(z_2^i(K-M)) \qquad (6.138)
\end{aligned}
$$

Substituting the linearizing inputs we get the normal states as ${\dot{\xi}_2^1}{}^i = p_1$ and ${\dot{\xi}_1^2}{}^i = p_2^i$. Note that the normal states are decoupled.

$$
\begin{aligned}
{\dot{\xi}_1^1}{}^i &= {\xi_2^1}^i && (6.139a) \\
{\dot{\xi}_2^1}{}^i &= p_1^i && (6.139b)
\end{aligned}
$$

$$
{\dot{\xi}_1^2}{}^i = p_2^i \qquad (6.140)
$$

Similar to the previous solutions, we again shift the states as ${\xi_1^1}^i = \psi_i - \beta$ and ${\xi_1^2}^i = z_1 - v_0(1 - K)$. Now, we can use any linear system controller design method to have stable systems at $\dot{d}_i = v_0$ and $\psi_i = \theta_i = \beta$. Consider the inputs p_1^i and p_2^i are linearly dependent on the normal states. Then the normal state space becomes

$$
\begin{bmatrix} {\dot{\xi}_1^1}{}^i \\ {\dot{\xi}_2^1}{}^i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ K_1^1 & K_1^2 \end{bmatrix} \begin{bmatrix} {\xi_1^1}^i \\ {\xi_2^1}^i \end{bmatrix} \qquad (6.141)
$$

and

$$
{\dot{\xi}_1^2}{}^i = K_2^1 {\xi_1^2}^i \qquad (6.142)
$$

Selecting the proper values for the controller parameters, one would obtain stable responses. Considering the state $\dot{\psi}_i = {\xi_2^1}^i$ the equilibrium is satisfied only when $\theta_i = \psi_i = \beta$ or $\theta_i = \beta + \pi$.

Note that, since the controller is using the velocity of the virtual center of flock (the slowed down center of flock) the equilibrium does not occur for the other values of θ_i. Speaking mathematically, at equilibrium the equation in (6.133) becomes

$$\dot{\psi}_i = \xi_2^{1^i} = \frac{1}{d_i}\left[z_2^i\left(1 - \frac{K}{M}\right)\sin(\theta_i - \beta) - Kv_c^i\sin(\theta_c^i - \beta)\right] = 0 \tag{6.143}$$

Consider the agents are moving in parallel directions with a speed of $z_2^i = \bar{v}$ satisfying $\dot{d}_i = \bar{v}(1 - K)$, as in Figure 6.25. Then mean speed of agents out of ith becomes $v_c^i = \frac{M-1}{M}\bar{v}$ and the orientation of this speed becomes $\omega_c^i = \theta_i$. Substituting these into equation

$$\begin{aligned} 0 &= \bar{v}\left(1 - \frac{K}{M}\right)\sin(\theta_i - \beta) - K\frac{M-1}{M}\bar{v}\sin(\theta_i - \beta) && (6.144)\\ &= \bar{v}\left(1 - \frac{K}{M} - K\frac{M-1}{M}\right)\sin(\theta_i - \beta) && (6.145)\\ &= \bar{v}\,(1 - K)\sin(\theta_i - \beta) && (6.146)\end{aligned}$$

which is satisfied only when $\theta_i = \beta$ or $\theta_i = \beta + \pi$. The first condition ($\theta_i = \beta$) means the agent is in the same direction of the vector from the shifted center of flock to the agent itself, and it is getting far away ($\dot{d} > 0$) from the virtual center of flock . When $\theta_i = \beta + \pi$, the agent is getting closer to the center of flock. However the decoupled state $\xi_1^2 = z_1^i$ is controlled to converge to a positive value $v_0(1 - K)$ which means the agent is heading to get away from the virtual center of flock. Therefore, the state θ_i converges to $\theta_i = \psi_i = \beta$.

Utilizing the above states let us examine the motion of the shifted center of the swarm. If we call the real distance between shifted center of swarm and each agent as $\bar{d}$ we would write the following set of equations

$$\begin{aligned} x_i &= x_c + \bar{d}_i\cos(\psi_i) && (6.147)\\ y_i &= y_c + \bar{d}_i\sin(\psi_i) && (6.148)\end{aligned}$$

Summing positions of all agents

$$\begin{aligned} \sum_{i=1}^{M} x_i &= \sum_{i=1}^{M} x_c + \sum_{i=1}^{M}\bar{d}_i\cos(\psi_i) = Mx_c + \sum_{i=1}^{M}\bar{d}_i\cos(\psi_i) && (6.149)\\ \sum_{i=1}^{M} y_i &= \sum_{i=1}^{M} y_c + \sum_{i=1}^{M}\bar{d}_i\sin(\psi_i) = My_c + \sum_{i=1}^{M}\bar{d}_i\sin(\psi_i) && (6.150)\end{aligned}$$

using $\sum_{i=1}^{M} x_i = Mx_c$ and $\sum_{i=1}^{M} y_i = My_c$

$$\sum_{i=1}^{M} \bar{d}_i \cos(\psi_i) = 0 \tag{6.151}$$

$$\sum_{i=1}^{M} \bar{d}_i \sin(\psi_i) = 0 \tag{6.152}$$

The time derivative of these equations are

$$\sum_{i=1}^{M} \dot{\bar{d}}_i \cos(\psi_i) - \sum_{i=1}^{M} \bar{d}_i \dot{\psi}_i \sin(\psi_i) = 0 \tag{6.153}$$

$$\sum_{i=1}^{M} \dot{\bar{d}}_i \sin(\psi_i) + \sum_{i=1}^{M} \bar{d}_i \dot{\psi}_i \cos(\psi_i) = 0 \tag{6.154}$$

At the equilibrium of ξ states $\psi_i = \beta$, and $\dot{\psi}_i = 0$ for each agent. At the equilibrium the above equations become

$$\sum_{i=1}^{M} \dot{\bar{d}}_i \cos(\beta) = 0 \tag{6.155}$$

$$\sum_{i=1}^{M} \dot{\bar{d}}_i \sin(\beta) = 0 \tag{6.156}$$

results in

$$\sum_{i=1}^{M} \dot{\bar{d}}_i = 0 \tag{6.157}$$

Recall the equilibrium of $\dot{d}_i = v_0(1-K)$ for each agent. This is the condition when the center of flock has a virtual speed of $v_{cw} = Kv_c$. If we take $K = 1$, in other words use the exact speed of the center of swarm, the time derivative of real distance becomes $\dot{\bar{d}}_i = 0$. This result satisfies the above equation. Therefore, at the equilibrium the agents and the center of flock are on the same line heading towards the same direction ($\theta_i = \psi_i = \beta$).

The velocity of the center of swarm can be expressed as

$$\dot{x}_c = \sum_{i=1}^{M} z_2^i \cos(\theta_i) \tag{6.158}$$

$$\dot{y}_c = \sum_{i=1}^{M} z_2^i \sin(\theta_i) \tag{6.159}$$

at the steady state $\theta_i = \psi_i = \beta$ and $z_2^i = v_0$. Then, substituting these into the equations

$$\dot{x}_c = \sum_{i=1}^{M} v_0 \cos(\beta) \tag{6.160}$$

$$\dot{y}_c = \sum_{i=1}^{M} v_0 \sin(\beta) \tag{6.161}$$

and rearranging

$$\dot{x}_c = v_0 \sum_{i=1}^{M} \sin(\beta) \tag{6.162}$$

$$\dot{y}_c = v_0 \sum_{i=1}^{M} \cos(\beta) \tag{6.163}$$

which means the center is moving with constant speed v_0 in the direction of β. Furthermore, the velocity and orientation components in equation (6.111) can be derived at the steady state

$$v_c^i = \frac{v_0(M-1)}{M} \tag{6.164}$$

$$\theta_c^i = \beta \tag{6.165}$$

Therefore, at steady state the mean velocity of the other agents (different from ith agent) is equal to $v_0(M-1)/M$ in the same direction of ith agents velocity.

The minimal dynamics should also be investigated. Since the relative degree (3) is less than the number of states (5) we will select the new states η_1^i and η_2^i independent of input i.e. orthogonal to the function $g^i(x)$.

$$\frac{\partial \eta^i}{\partial x} g^i(x) = 0 \Rightarrow$$

$$\begin{bmatrix} \frac{\partial \eta_1^i}{\partial d_i} & \frac{\partial \eta_1^i}{\partial z_1^i} & \frac{\partial \eta_1^i}{\partial \psi_i} & \frac{\partial \eta_1^i}{\partial z_2^i} & \frac{\partial \eta_1^i}{\partial \theta_i} \\ \frac{\partial \eta_2^i}{\partial d_i} & \frac{\partial \eta_2^i}{\partial z_1^i} & \frac{\partial \eta_2^i}{\partial \psi_i} & \frac{\partial \eta_2^i}{\partial z_2^i} & \frac{\partial \eta_2^i}{\partial \theta_i} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) & -z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.166}$$

$$\frac{\partial \eta_1^i}{\partial z_2^i} + \frac{\partial \eta_1^i}{\partial z_1^i}\left(1-\frac{K}{M}\right)\cos(\theta_i-\psi_i) = 0 \tag{6.167}$$

$$\frac{\partial \eta_2^i}{\partial \theta_i} + \frac{\partial \eta_2^i}{\partial z_1^i} z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i) = 0 \tag{6.168}$$

which means in addition to ${\xi_1^1}^i$, ${\xi_2^1}^i$, and ${\xi_1^2}^i$, the state η_1^i should be independent of z_1^i and z_2^i. Similarly η_2^i should be independent of θ_i and z_1^i. We select $\eta_1^i = \theta_i - \beta$ and $\eta_2^i = d_i$. Let us examine the zero dynamics of these variables.

The time derivative of η_1^i is

$$\dot{\eta}_1^i = \dot{\theta}_i = w_2^i \tag{6.169}$$

We first substitute the equations in (6.139) and (6.141) into this equation. And then we do the following replacement of parameters to obtain equations in reduced system domain

$$\begin{aligned}
\psi_i &= {\xi_1^1}^i + \beta && (6.170)\\
z_1^i &= {\xi_1^2}^i + v_0(1-K) && (6.171)\\
\theta_i &= \eta_1^i + \beta && (6.172)\\
d_i &= \eta_2^i && (6.173)
\end{aligned}$$

Now we use the zero dynamics of the system at ${\xi_1^1}^i = {\xi_2^1}^i = {\xi_1^2}^i = 0$. In addition we use the results in (6.164). Then $\dot{\eta}_1^i$ becomes

$$\begin{aligned}
\dot{\eta}_1^i &= \left(\frac{\cos(\eta_1^i)\sin(\eta_1^i)(K + M - K\cos(\eta_1^i) + M\cos(\eta_1^i) - 2KM) - \sin(\eta_1^i)^3(K-M)}{M\eta_2^i}\right)v_0\\
&\quad - \frac{K_2^1 \sin(2\eta_1^i)}{2} && (6.174)
\end{aligned}$$

The stability of η_1^i may be examined by linearization around the equilibrium $\eta_1^i = 0$ and $\eta_2^i = d_i = c > 0$. The derivative of $\dot{\eta}_1^i$ with respect to η_1^i at $\eta_1^i = 0$ is

$$\begin{aligned}
\frac{\partial \dot{\eta}_1^i}{\partial \eta_1^i}\Big|_{\eta_1^i=0,\eta_2^i=c} &= -K_2^1 + \frac{2v_0(1-K)}{c}\\
&= 0 && (6.175)
\end{aligned}$$

The minimal dynamics of η_1^i is independent of η_2^i. The value in the first equation is the eigenvalue of the minimal dynamics of η_1^i, and it is definitely negative when $K_2^1 > \frac{2v_0(1-K)}{c}$. Satisfying this condition, the state is locally asymptotically stable. Physically this means the system is converging to $\theta_i = \beta$ that is the motion of each agent along the same line in the direction β.

The time derivative of second selected state $\eta_2 = d$ becomes

$$\begin{aligned}
\dot{\eta}_2^i &= z_1^i\\
&= {xi_1^2}^i + v_0(1-K) && (6.176)
\end{aligned}$$

The zero dynamics becomes

$$\dot{\eta}_2^i = v_0(1 - K) \tag{6.177}$$

As in the previous controllers here note that this state is not converging to a constant, instead it is increasing with a velocity of $v_0(1 - K)$ and diverging as $t \to \infty$. The behavior is indeed a tracking problem of the state. The distance variable d tracks the constant speed travel behavior at the zero dynamics. This is completely consistent with the controlled dynamics of the system.

The last state we should examine is the speed of the agent z_2^i

$$z_2^i = \frac{z_1^i + K v_c^i \cos(\theta_c^i - \psi_i)}{\left(1 - \frac{K}{M}\right)\cos(\theta_i - \psi_i)} \tag{6.178}$$

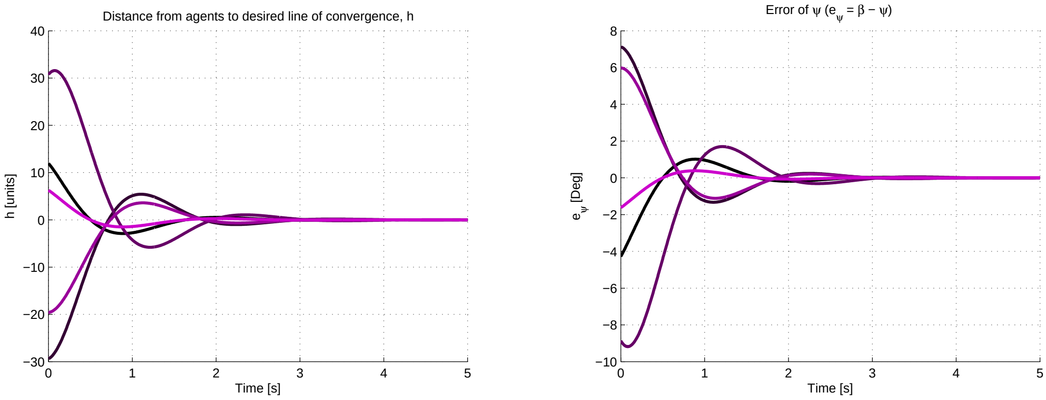
At the steady state ($z_1^i = v_0(1 - K)$, $v_c^i = v_0(M - 1)/M$, $\theta_i = \psi_i = \theta_c^i = \beta$) it becomes

$$z_1^i = v_0 \tag{6.179}$$

which is again consistent with the controlled dynamics of the system.

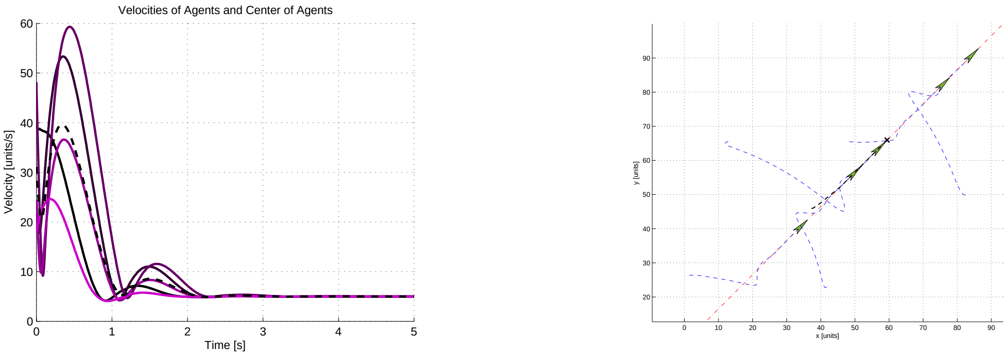
6.2.7.4 Multi-Agent System with MIMO - Non-Stationary Target Following Controller - Simulations

The MIMO controller developed in the previous section for multi-agent systems is applied to a 5 agent swarm in this section. The controller parameters are selected to be $K_1^1 = -10$, $K_1^2 = -3$, and $K_2^1 = -3$ as in the non-stationary targeting controller section. The response will be underdamped for states ψ_i and z_1^i. The speed ratio for the virtual center of flock is $K = 0.8$. The reference inputs are $\beta = 45^o$ and $v_0 = 5\ units/s$. The initial conditions are same with the non-stationary targeting controller for better comparison. In Fig. 6.26(a) the change of the distances between agents and the line of convergence (h) for each agent are presented. As seen from the figure all h values converge to zero with damped oscillations. In 6.26(b) the change of $e_\psi = \beta - \psi$ values of agents are presented. The values converges to $e_\psi = 0$ with decaying exponential oscillations. In Fig. 6.27(a) the velocities of agents and the center of agents are plotted. The velocities are converging to the reference input $v_0 = 5 units/s$. On the right hand side the path of the agents are plotted. The arrow heads represents the agents and the cross sign is the center of the swarm. The paths are the tailing dashed curves. Comparing with the results, we observe that the non-stationary targeting controller has a faster convergence than the stationary targeting controller.



(a) (b)

Figure 6.26: Distance h (a) and error e_ψ (b) of 5 agents with line following MIMO controller for stationary target.



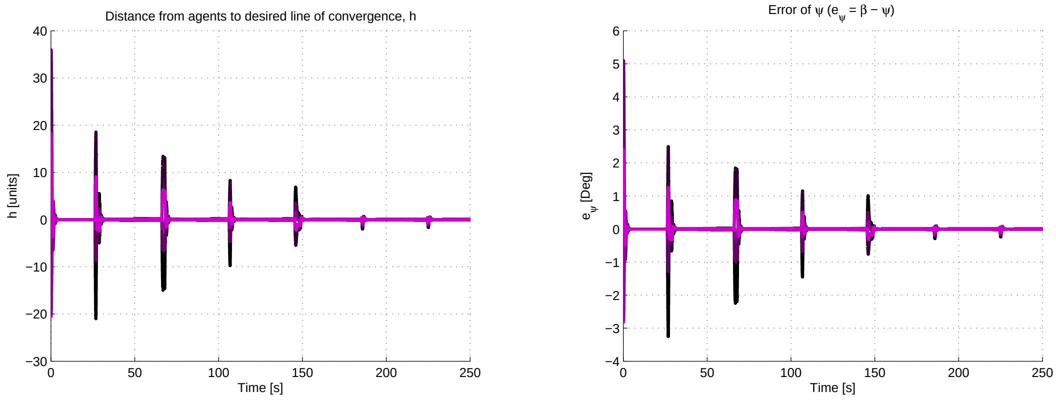
(a) (b)

Figure 6.27: Velocities of agents and center of swarm (a), and path of 5 agents (b) with line following MIMO controller for non-stationary target.

6.2.8 Discussion and Future Directions

The line following and circling controllers developed in the last two chapters are working fine in the simulations. In this section we will discuss some more simulation results of the controllers with some modifications. Meanwhile we will direct some future studies.

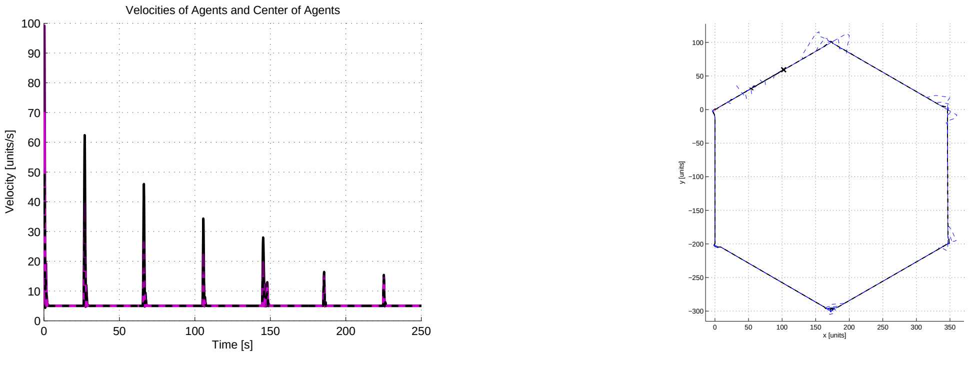
The first simulation study examines the servo characteristics and targeting performance of the line following controller. Recall that the controllers developed for multiagent systems just use the center of swarm as the target. Therefore, the convergence of center of swarm is emergent - depends on the initial conditions. However, if we modify the controllers to utilize a fixed target in addition to the center of swarm, we observe that the center of swarm converges to the line passing through that target. We simulated this modified controller for swarm to follow a hexagonal path. In figures 6.28 and 6.29 the simulation results of 5 agents traveling on a hexagon are presented. The corners and the slope of the sides of the hexagon are given as fixed targets. The controller switches to the new target position and orientation when the center of swarm approaches that target up to a prespecified distance (taken as 30 in these simulations).



(a)

(b)

Figure 6.28: Distance h (a) and error e_{ψ} (b) of 5 agents traveling on a hexagon with line following controller.

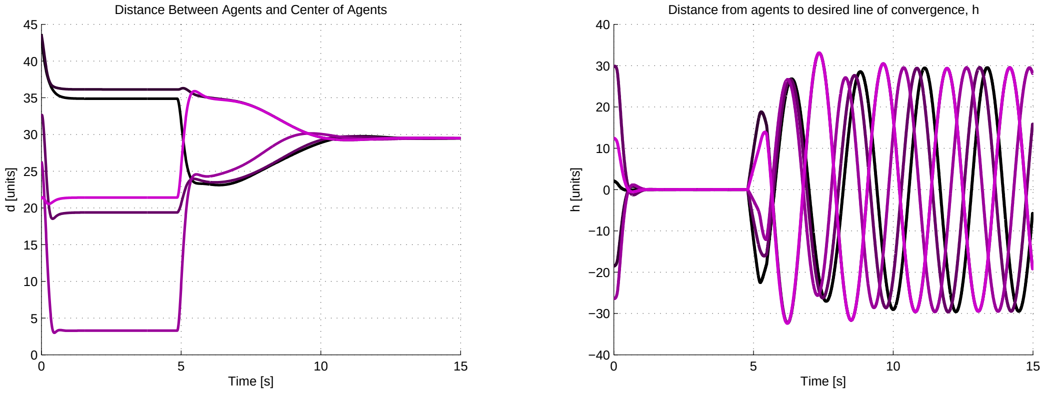


(a)

(b)

Figure 6.29: Velocities of agents and center of swarm (a), and path of 5 agents (b) 5 agents traveling on a hexagon with line following controller.

In Figure 6.28(a) the distances between agents and the line to be followed are given. The distances deviate from zero when the target points and orientations change at the corners. The controllers always push the state to zero with underdamped responses. And note that as the swarm passes a corner the deviations are getting lower since the agents are getting closer. The controllers are not designed or modified for controlling the inter agent distances. This one of the future directions of this thesis study. In Figure 6.29(b) the error $e_\psi = \beta - \psi$ is presented. The responses are again underdamped due to the controller parameters. The deviations at corners are getting lower since the agents are getting closer. The figures 6.29(a) and (b) are the plots of velocity and the path of agents. One may observe the deviations from the hexagon and their shrink at the corners.



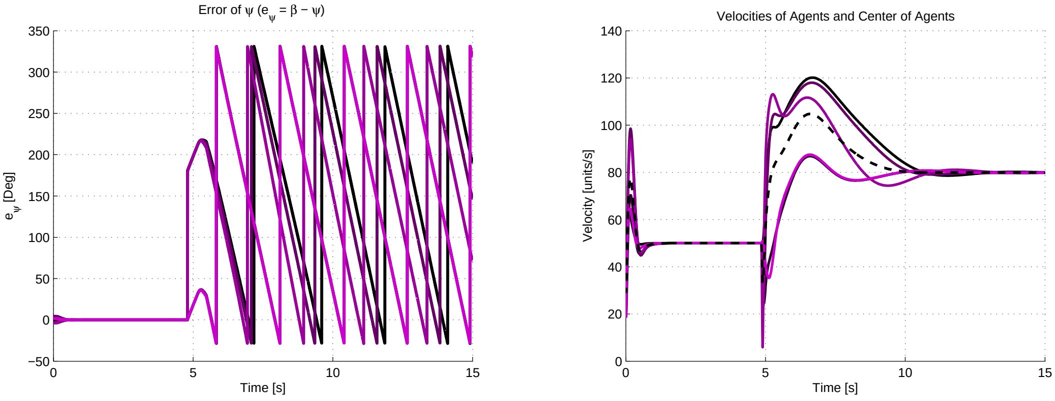
(a)

(b)

Figure 6.30: Distance d (a) and distance h (b) of 5 agents switching between line following and circling behaviors.

The next simulation is performed for examining the switching performance between the line following and circling behaviors. In addition controllers for the circling behavior are modified for targeting around a given point. As in the line following controllers in the above simulations, the circling controllers are utilizing fixed points in addition to the center of swarm. The figures 6.30, and 6.31, and 6.32 present the results of simulations for 5 agents, first following a line and then switching to circling around a fixed point behavior. In Figure 6.30(a) the distance between agents and the center of swarm is presented. During the line following behavior the distances are not converging to a common reference value but when the controller switches to the circling behavior they do converge to the reference value $d_0 = 30$. On the contrary the distance h represented in Figure 6.30(b) is convergent during the line following behavior and oscillating during the circling behavior. The error $e_\psi = \beta - \psi$ in Figure 6.31(a) also show a similar characteristic as h. The velocities of agents are consistent with the reference inputs ($v_0 = 50$ for line following, and $v = \alpha d_0 = 4\ x\ 20 = 80$ for circling behavior).

The paths of the agents are presented in Figure 6.32.



(a) (b)

Figure 6.31: Error e_ψ (a) and velocities of agents and center of swarm (b), of 5 agents switching between line following and circling behaviors.

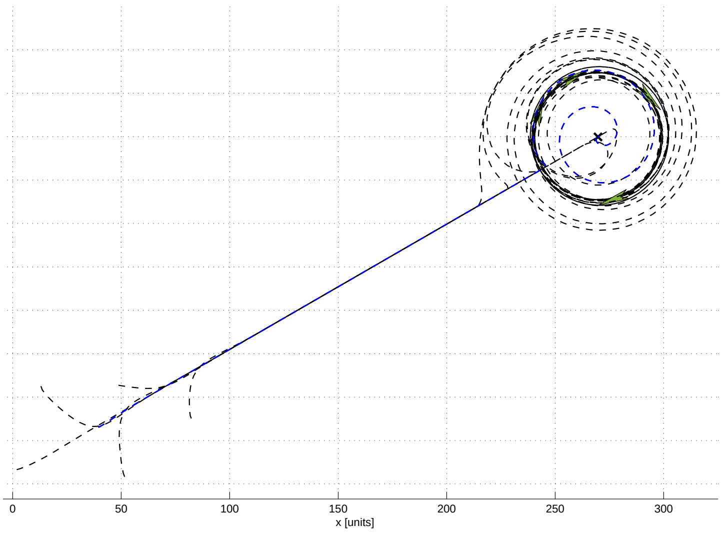


Figure 6.32: Path of 5 agents switching between line following and circling behaviors.

The line following controllers of multi agent systems are designed for agents to follow a prespecifed path without utilizing tracking points on the path. The controllers are fine in achieving this. However, in many swarm applications (UGV, UAV etc.) the surveillance missions require agents to rank side by side and move on parallel paths while each agent is gathering information in its neighborhood. The line following controllers are modified for this purpose. The equilibriums of ξ states in equation (6.132) are changed as

$$\begin{aligned}
\xi_1^1 &= \psi_i - \beta && \text{(6.180a)}\\
\xi_2^1 &= \frac{1}{d}\Big[z_2^i\left(1-\frac{K}{M}\right)\sin(\theta_i-\psi_i)\\
&\quad -Kv_c^i sin(\theta_c^i-\psi_i) - v_0\left(1-\frac{K}{M}\right)(sin(\theta_0-\beta) - K\sin(\theta_0-\pi/2-\beta))\Big] && \text{(6.180b)}\\
\xi_1^2 &= z_1^i - v_0\left(1-\frac{K}{M}\right)(\cos(\theta_0-\beta) - K\cos(\theta_0-\pi/2-\beta)); && \text{(6.180c)}
\end{aligned}$$

where θ_0 is the reference input for the orientations of agents. Using the above reference inputs, the original states are converging to $\psi_i = \beta$, $\theta_i = \theta_0$, and $z_2^i = v_0$. The simulation results of this new controller are presented in Figures 6.33, 6.34, and 6.35. The controller reference inputs are switching at certain time instances. The speed reference is taken contant at $v_0 = 3\ units/s$ and the other inputs are set as

$$\beta = 45^o \;\; \theta_0 = 90 + \beta \;\; for \;\; 0 \le t < 10$$

$$\beta = 45^o \;\; \theta_0 = 60 + \beta \;\; for \;\; 10 \le t < 20$$

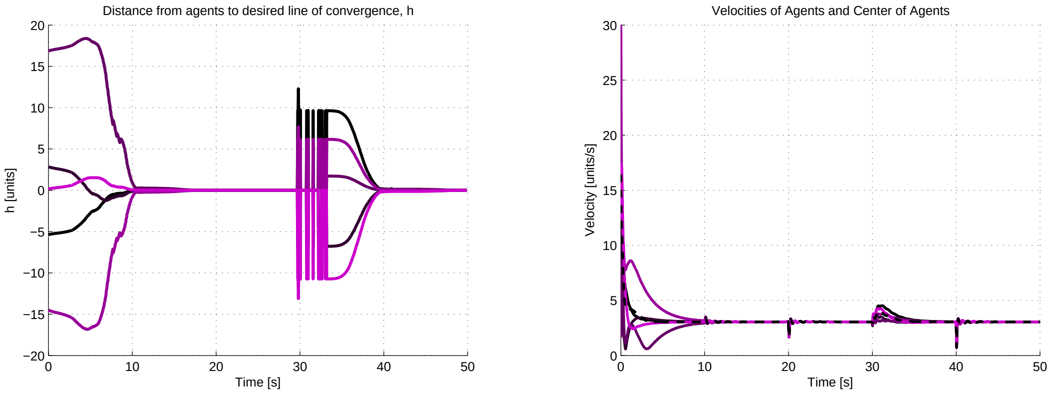
$$\beta = 45^o \;\; \theta_0 = 120 + \beta \;\; for \;\; 20 \le t < 30$$

$$\beta = 70^o \;\; \theta_0 = 0 + \beta \;\; for \;\; 30 \le t < 40$$

$$\beta = 70^o \;\; \theta_0 = 90 + \beta \;\; for \;\; 40 \le t < 50$$

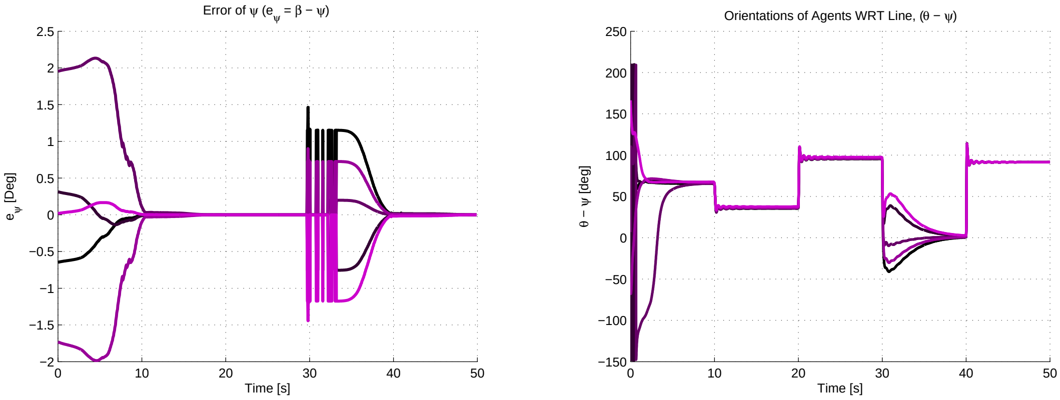
In Figure 6.33(a) the distance variable h is plotted. The values converge to zero at the beginning of the simulation. They do not change when θ_0 reference input switches to a new value, but they deviate from zero as the reference β changes. The controller is successful in achieving the convergence of h during the simulation. The next Figure 6.33(b) shows the velocities of the agents. The values deviate from the reference input $v_0 = 3$ at each switch. The daviations are more significant when β input changes. Figure 6.34(a) presents the error $e_\psi = \beta - \psi$. The change in the values are similar with the ones in h. In the Figure 6.34(b) the difference $\theta - \psi$ is plotted. This is the deviation of agent orientation from the orientation of line of ranking. The values are deviating at each switch of reference θ_0 and β. The last Figure 6.35 includes the path of the agents and the center of swarm.

The above simulation examples are given to show some future directions of this thesis study. In addition to the above results the future studies may include improving the existent controllers by considering (i) preserving inter-agent distances, (ii) neighborhood size effect and obstacle avoidance, (iii) loss of agents in the swarm, (iv) asynchronism of the agents, (v)



(a) (b)

Figure 6.33: Distance h (a) and velocities (b) of 5 agents traveling on parallel paths.



(a) (b)

Figure 6.34: Error e_ψ (a) and orientation θ (b), of 5 agents traveling on parallel paths..

discrete time models (vi) robustness to parameter uncertainties and disturbances, (vii) analytical examinations of fixed targeting modifications in controllers, (viii) application to practical multi-agent systems i.e UAV, UGV swarms, (ix) kinetics of vehicle dynamics, (x) extension to 3-D Space .

6.2.9 Concluding Remarks

In the literature most of the studies utilize a pre-generated path for tracking points, straight lines, circular or randomly generated paths by mobile robots. In this Section the problem solved is the behavior of the robot that converges to a line passing through a target. The slope

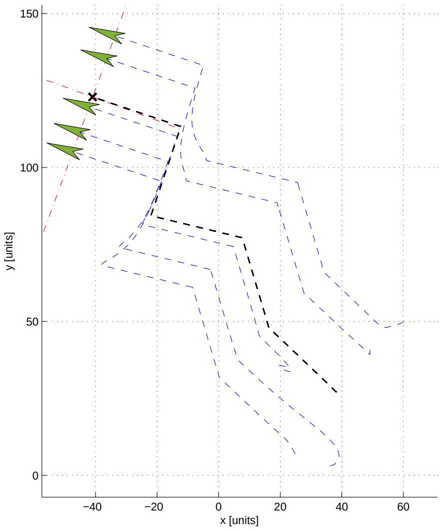


Figure 6.35: Path of 5 agents traveling on parallel paths..

of the line is specified as a parameter in the controller. The agent approaches the line and travels on it with the given constant linear speed. We propose different controllers for two different cases. In the first one, the only controllable input is the angular velocity of the robot. The translational speed of the robot is taken as constant. In the second case in addition to the angular velocity input the translational speed is taken as an input. We use static feedback linearization methods in the first case and dynamics feedback linearization in the second one.

The controllers developed in this study may be utilized in UGV, UAV, and UWV applications dedicated to tracking of targets. Furthermore, the controllers are adapted for the swarm of these vehicles traveling in parallel formation.

CHAPTER 7

Conclusion

In this thesis it is aimed to develop decentralized coordination and control strategies for robotic swarms especially consisting of unmanned air, space, land and underwater vehicles. One main problem in robotic swarms is generating geometric formations for specific purposes. The very demanding purposes are the area exploration and surveillance of the swarm members, agents. The controllers developed to achieve these missions would better bring up the most powerful properties of multi-agent systems which are robustness, scalability and flexibility. Therefore, the controllers should be decentralized and independent of agent identity (common for each agent). In this way, the swarm becomes robust, i.e. they may re-organize the individuals to negate the absence of lost individual and complete the mission. Additionally, the number of the agents in the swarm may change in the run time from two to hundreds of agents (scalability property). Furthermore, the swarm becomes flexible bringing the possibility of re-distributing the tasks among them in order to adapt different situations and/or missions or negate the absence of some individuals. Another property of the swarms is the emergence. The initial conditions, number of agents in the swarm, uncertainties, under modeled dynamics, interactions between agents etc. may be unpredictable, so that especially the transient and steady state responses may be unpredictable or simply called as emergent. The controllers developed for multi-agent systems should be appropriate with the emergence property. More specifically the controllers running individually on each agent should be robust to emergent results of the behaviors. One of the best ways of achieving this is not utilizing prespecified or re-specified parameters during the run time of the multi-agent systems. However, considering the geometric formation of agents, in most of the swarm applications the agents utilize a trajectory generation scheme updating continuously. The continuous replanning of the trajectories requires absolute positions of agents, methods for trajectory and next set-point calculation of each agent, and one of the various trajectory tracking techniques.

The main difference of the controllers developed in this thesis study is their path/trajectory free motion control methods. All of the paths of each agent are emergent. In most of the controllers we developed the relative position and orientation variables are utilized instead of the absolute position, and orientation variables and their derivatives. So that there is no need for a global or common coordinate frame. This is a better way of controller design due to hardware limitations. In robotic applications the global position and orientation of agents are hard to obtain variables. Because the technologies like odomery, IMU, GPS are still not much capable in obtaining accurate, precise, and repeatable global position info due to integrating errors and less resolution. Instead the relative position and orientation info are more available considering the proximity sensors. On the other hand the relative parameter info would also be gathered by each individual agent without the need to broadcast this info to other agents.

Furthermore, for each agent the reference inputs of states are the same; where in most of other controllers developed in the literature using regeneration of paths, they differ in between agents. Another resulting powerful property of the controllers developed here, is their success in following dynamic targets. There is no need of the replanning of paths of each agent due to the target motion; instead only the relative position of the agents or at most the absolute position of the target are the required info. The mathematical models are derived for relative polar coordinates in 2-D space and only the kinematics of the agents are considered for the proof of concept of the controllers. The developed controllers have the capability of achieving

- Orientation agreement behavior (Chapter 3),
- Circling around static and dynamic target behaviors (Chapters 4, 5),
- Following line passing through specified point and with a specified orientation behavior (Chapters 4, 6),
- Area surveillance with parallel motion behavior, (Chapter 6)
- Switching between the above controller strategies (Chapter 6),

The development of the controllers are performed by using both linear and nonlinear control theories. Especially, we examined the stability of the controlled systems by linearization around the equilibrium points and using Lyapunov stability criteria in Chapters 4, 5, and 6. The circling and line following behavior strategies are developed by state feedback linearization techniques in Chapters 5, and 6. The aim of the studies in these chapters were to develop controllers for multi-agent systems. However, since there is no study in the literature for even single agents to circle or follow a line without path planning, we first developed the path free controllers for single agents and then adapted them for the multi-agent systems. The literature surveys showed that there are no similar studies and alternative controllers to be compared with the ones we developed here.

Another focus of the study is on the asynchronism property of the multi-agent systems. The agents in the swarms run asynchronously since they do not use a common clock. Furthermore, they face with delays in the sensing, communication, computation and manipulation states. These delays would most probably differ in between agents. Therefore, all controllers should be validated under these time delayed asynchronous working conditions. In this thesis study, we validated the cyclic pursuit controller under these conditions (Chapter 2). We examined the stability of synchronous system by analytical methods and extended the results for asynchronous model. We also validated the results by simulations and experiments. The experiments are performed on a set-up designed and manufactured for mobile robotic swarm applications. The detailed architecture (E-puck robots, camera, software etc.) of the set-up is given in the Appendix A. The set-up arena is observed by a high quality USB camera connected to a computer. The image processing works are done in Matlab and the results (positions and orientations) are fed back to the behavior algorithms governing the swarm dynamics. The time delays due to image processing would be decreased by improvement in the algorithms running on Matlab or a different compilation environment (i.e. OpenCV, AForge) may be utilized to run those image processing algorithms faster.

In Chapter 3, we examined the orientation agreement problem of swarms. We analyzed the following properties: (i) the multi-agent systems are synchronous or asynchronous, (ii) they travel in bounded or unbounded regions and (iii) the mobile agents have various amount of turning speed restrictions. For orientation agreement problem, we developed three different control strategies and compared the performance of the strategies via some metrics we proposed and discussed the results both numerically and analytically. The results showed that agents exhibit best performance for orientation agreement in the case in which they are synchronous, holonomic, and using Strategy 1, and in bounded region. Strategy 1 requires the absolute orientation values while Strategy 2 and 3 require just the relative orientations of agents. Therefore, in practical applications we suggest the second and third strategies to be utilized.

We considered the unicycle mobile robot kinematics and developed simple proportional controllers for the speed and angular velocity of the robot for achieving approaching and circling behavior around a static target, and settling on the line passing through a target with a pre-specified orientation in Chapter 4. The controllers are designed for pre-specified or re-specified path/trajectory free control of agents. One of the controllers developed for single-agent systems achieved the behavior which first approaches a target and then start to circle around that target without any external switching input or external decision mechanism. This work is going to be extended for multi-agent systems as a future work. The future directions also include the redesign of the controllers for dynamic targets and utilization of PID strategies in addition to just proportional controllers.

Similarly, in Chapters 5 and 6, we considered the unicycle mobile robot kinematics in relative polar coordinates. We derived SISO (only angular speed manipulation) and MIMO (both angular speed and translational speed are manipulated) feedback linearization inputs for the controllers and achieved circling around target and line following behaviors without pre-specified or run time specified paths. For the circling behavior, the controllers are developed to force the states to converge to periodical trajectories. The resulting controllers are validated by simulations. These controllers may be utilized in exploration and surveillance operations of UGV, UAV, and UWV by circling around static or dynamic targets.

The line following controllers developed in Chapter 6 takes the target position and the slope of the line as reference inputs. The static and dynamic linearization techniques resulted in controller parameters requiring either relative or absolute position and orientation parameters of agents. The MIMO controllers utilizing target position and its time derivatives showed best performance in the simulations. However, it requires the global parameter values. Therefore, we propose the MIMO controllers with static target controllers in practical application where only the relative position and orientation variables are available. Furthermore, in this Chapter we extended the controller strategies for multi-agent systems to circle around the center of swarm which is forced to converge to a pre-specified point by simply adding fixed virtual targets at those pre-specified points. Similarly, we utilized the addition of virtual fixed target method in line following behavior and presented the simulation results. Also, we investigated the servo characteristic of the controllers by changing the reference target position inputs in the run time and obtained successful results. The swarm tracked a hexagonal path as a proof of concept for multi-agent applications that is supposed to follow some pre-specified paths. Note here that the controllers does not need to calculate any next set-point in following the given paths. They just need to regulate motion states like distance between agent and center of swarm etc. Furthermore, we investigated the switching between circling and line following behaviors and obtained successful results. Lastly we extended the line following controller by changing the equilibrium points and obtained a very demanding formation of agents which is parallel motion of the agents. The controller is successful in achieving the formation of agents on a line with a prespecified orientation and travel through any given orientation. This behavior is very beneficial in surveillance and exploration missions of multi-agent systems. The future directions of this thesis study includes the application of these new methods to real robotic swarms. In addition, we plan to study on the following topics to extend the results of this thesis in the very near future:

- Preserving inter-agent distances - avoiding collisions among themselves
- Neighborhood size effect and obstacle avoidance
- Agent loss in the run time
- Effect of asynchronism on controller performance
- Examining the discrete time models
- Robustness to parameter uncertainties and disturbances
- Analytical examinations of fixed virtual targeting modifications in controllers
- Application to practical multi-agent systems i.e UAV, UGV swarms,
- Kinetics of vehicle dynamics
- Extension to 3-D Space

APPENDIX A

Experimental Set-up For Multi-Robot Applications

A.1 Introduction

In this part of the thesis study we are motivated by the needs on realistic applications of designed and simulated swarm behaviors. There are many studies on swarm robotics that are simulation based and/or performed analytically. However, additional realistic experiments would contribute new insights to these works. Therefore, we designed an experimental set-up to observe the realistic behaviors of robot swarms. This set-up would also be useful for undergraduate and postgraduate educational studies on control systems and robotics.

Many robotic swarm applications typically reject any dependency on a global system such as global positioning. However, if available the global positions and orientations of the robots can be used for development, debugging, and monitoring of swarm robot applications. On the other hand, the local information that a robot may get by its own sensors can be simulated in this set-up, i.e. the relative positions and orientations of robots in a neighborhood of a robot can be derived from the global information and sent to the robots. Therefore, the swarm applications utilizing only local information of robots can also be studied experimentally by just deriving the local information from the global information. Furthermore, the collective robotic studies which may require global information can utilize this set-up for experimental validations. Even further, the information of the positions and orientations of robots can be recorded for later analysis of the swarm/collective behaviors. One common method of gathering this information is using the odometer of the robots if it is present (in most of the relatively simple mobile robots there is no odometer). However, odometry itself is not a reliable method and odometry errors tend to accumulate over time. Our system which utilizes an overhead camera to determine the positions and orientations of the robots, provides a fast development environment of swarm coordination and control algorithms since it relieves the designer from dealing with low-level odometric estimation and correction. Since we should deal with more than one robot, we also had to develop identification methods to find out which position and orientation belongs to which robot.

There are some studies on the observation of the arena of the swarm by an overhead camera for behavior analysis [163, 164, 165, 166]. However, these studies use overhead cameras or marker technology for only observing, visualizing, identification, and/or recording of the behavior/activity of the system. They do not feedback information to the robots.

On the other hand, there are some studies that utilize the overhead camera for position feedback to the members of swarm. Hayes and Dormiani-Tabatabaei used an overhead camera tracking system, combined with a radio LAN among the robots and an external workstation in [167]. They logged position data during the trials, reposition the robots between trials, and emulated the range and bearing sensor signals. Another experimental set-up for robot swarm applications is described in [168]. The authors develop a middleware solution called DISC-World and describe a prototype system where the precise location information of the robots are extracted by using an overhead camera.

The objective of this study is to build a low cost set-up that can track multiple robots at the same time. In order to allow near-real-time operation, we setup the system such that position and orientation estimation process time is kept as short as possible. The built setup is independent of robots used; hence, enabling researchers using different robots may adopt this framework. We also employed an easy to use software environment (Matlab) to facilitate the use of the proposed setup by various researchers with a rapid learning curve. Matlab is a tool that engineering students already learn in other courses and it has specialized functions for image processing and controller development. Therefore, the set-up is easy to use in senior undergraduate and graduate courses as well.

SwarmCam is a single system consisting of 120x180 cm experimental area, 6 E-puck robots with bluetooth interface, logitech USB camera and Matlab as the main image processing (and possibly control) development platform. The positions and orientations of the robots are determined by a labelling system consisting of three small colored dots on the robots. In addition their ID's are determined by a binary coding system consisting of black colored small dots placed on the top of the robots. The system constitutes a very useful platform for hardware in the loop simulations.

A.2 The Set-Up Structure

The multi-robot experimental set-up is composed of 6 mobile robots (although higher number is also possible), a high quality USB webcam, a high speed computer and an arena (see Figure A.1).

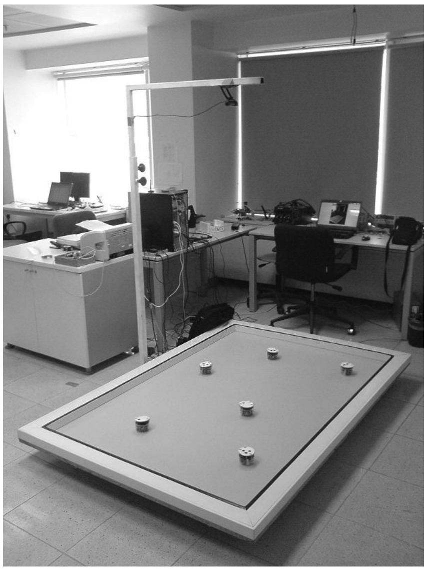


Figure A.1: Experimental setup consisting of an arena, robots, PC and overhead camera.

The mobile robots in this set-up should be small enough such that high number of robots may be utilized simultaneously in the experiments. They must have wireless communication modules like bluetooth, wifi, or zigbee for information exchange with the computer and each other. The existence of proximity sensors (IR, US etc.) is preferred for more realistic experiments. In some of our experiments we utilized the E-puck Robot [169]. The E-puck robot is a small (7.0 cm diameter) mobile robot that has powerful microcontroller dsPIC30 (Microchip, PIC microcontroller), 2 stepper motors for differential drive, 8 infrared proximity sensors, bluetooth communication module, and some other sensory units. The robots are programmed such that they set their motor speeds according to the commands supplied by the computer via the bluetooth interface. Another option is to program the robots so that they receive their global position (and/or possibly the relative positions of the neighboring or all the other robots) and have their own internal decision making and control. In addition the abject avoidance logic may be integrated on the robots. If the robots have enough proximity sensors (like ultrasonic, infrared sensors) around their body they may utilize the proximity information of objects (other robots or walls etc.) to avoid collisions.

The overhead camera placed 156 cm above the arena is directly connected to the computer via USB. An image resolution of 640×480 is sufficient for this set-up considering the arena and the robot sizes. The frames grabbed per second (fps) is not a main criteria in the selection of the camera since the image processing unit cannot process more than 3-4 frames per second (the actual robot detection time is 340 ms for the time being). Therefore, 15 or 30 fps of a camera is suitable for this set-up. The optical distortions on the vertices effect the system considerably. Therefore, a camera with high quality lens is essential. We used the webcam QuickCam Pro9000 (Logitech Europe S.A., European Headquarters Moulin du Choc CH - 1122 Romanel-sur-Morges) for grabbing the images of the arena.

The mobile robots move in a bounded arena of size 120×180 *cm*. The aspect ratio of the arena is designed to be appropriate with the aspect ratio of the camera images which is 4:3. The color of the arena is selected as light gray to be able to distinguish the robot hats from the arena easily. The arena size should be increased with the same aspect ratio for bigger robots or larger area applications.

The image processing, agent behavior algorithms and communication are all performed by Matlab (Mathworks Inc., Natick, MA, USA). Matlab is preferred due to its build in image acquisition and processing toolboxes. Moreover, Matlab is a very common, easy to use, rapid prototyping environment for engineering applications and many scientists and students are familiar with Matlab. However, it is computationally inefficient and might be inappropriate for applications requiring higher fps rates. Therefore, we are also considering developing a software interface with other tools like C#, C++ to have a faster version of the set-up.

The software of the set-up consists of two main parts, robot tracker and robot controller. In the robot tracker part the frames of the arena are grabbed and processed to determine the position, orientation, and identification of the robots. This information set is supplied to the robot controller part that runs functions of behavior of robots. The robot controller part transmits the control signals of the angular and translational speeds to the robots. The resulting angular and translational speeds of the agents are transferred to the agents via wireless communication modules (bluetooth for E-puck robots, around 10ms is consumed per robot to pass the information). The set-up is designed such that one may utilize only the robot tracker part to obtain and analyze the robot behaviors in the case of the robot controllers are embedded on the robots.

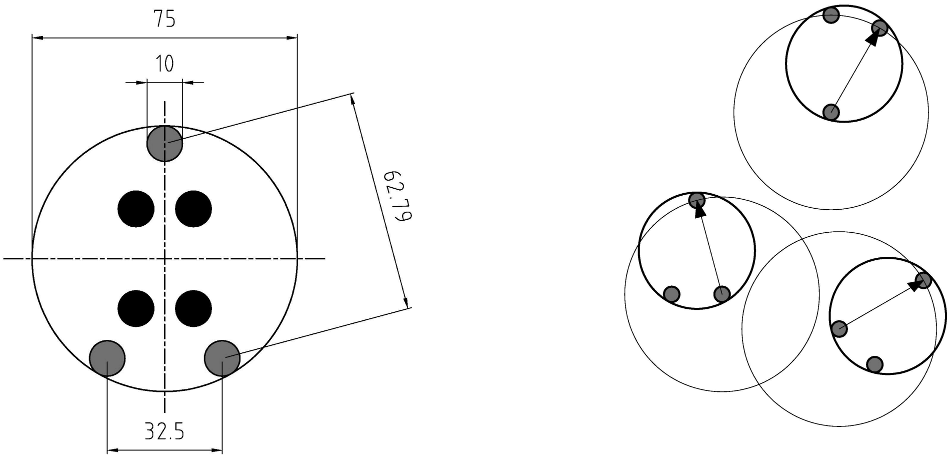
The main delay in the system occurs due to image processing (around 340ms per detection of robot pose, orientation, and identification). Therefore, a computer with enough memory to store the images of the arena and high speed central processing unit would result in better system performances (A double core 64bit CPU at 2.4 GHz with 2GB RAM is utilized in our experiments). As mentioned above another option could be to pass the position and orientation information to the robots and let their internal algorithm to calculate the values of the control inputs. That would better model more decentralized and realistic applications.

A.3 Image Processing Setup/Methods

Depending on the application the image processing system can be used to determine the robot ID's, the global or relative positions, and/or absolute or relative orientations of the robots. Determining the positions of the robots from the overhead images is very simple. However, the problem is to determine which location belongs to which robot. Therefore, additional methods need to be applied to distinguish the robots. In our setup robot hats are designed to find the location, orientations and identification of robots simultaneously. A sample hat is shown in Figure A.2(a). The hat has a diameter of 75mm which is slightly larger than the diameter of the E-puck robots. Three circles all having the same color (bright orange) (one placed at the front and the other two placed symmetrically at the rear) are used to find the locations and orientations of robots and the black circles are utilized for the identification of robots.

A.3.1 Determining Robot Locations

First the colored circles are detected in the bitmap images gathered from the camera. Note that most of the cameras supply compressed form of the images (i.e. jpg). However, Matlab reads/converts the images in bitmap format which includes all three color information (8 bits) of the images in three dimensional matrices. A sample configuration of three robots are shown in Figure A.2(b).



(a) (b)

Figure A.2: (a)A sample robot hat used to find the position, orientation and identity of robot. (Dimensions are in mm). (b) Three robots in the arena. The colored dots and the boundaries that other robots should stay out of are shown.

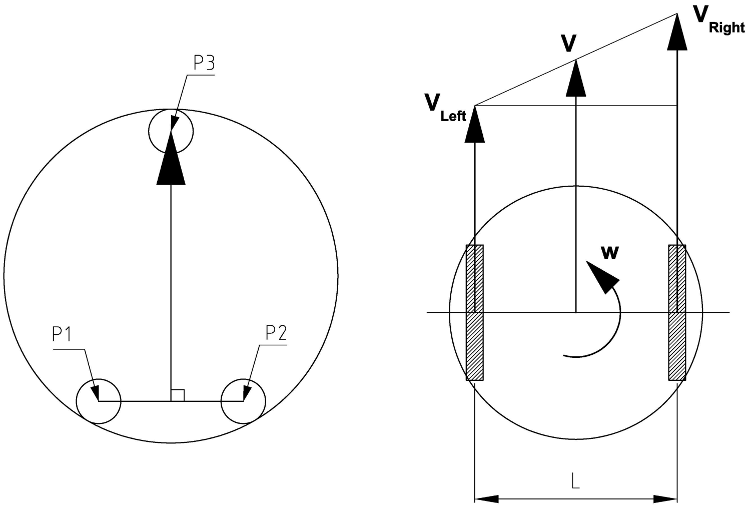
For distinguishing the colored circles from the rest of the objects in the image we simply use intervals of the color values. In some applications, we also utilize the corresponding HSV (Hue-Saturation-Value) images to find the colored circles. For example, bright purple, bright yellow, and bright green are easy to distinguish colors in HSV format. We get a bitwise matrix (image) by logical operations which outputs 1 for the pixel values in the intervals we set for the Hue, Saturation, and Value (or RGB) of the colors we utilized on the hats and 0 for the other color specifications. In equation (A.1) the logical operation is shown

$$A = (\underline{H} < Im(:,:,1) < \overline{H})\&(\underline{S} < Im(:,:,2) < \overline{S})\&(\underline{V} < Im(:,:,3) < \overline{V}) \tag{A.1}$$

where Im is the HSV image of the arena, Im(:,:,i) corresponds to the i^{th} index of the HSV image matrix (i = 1 is for Hue, i = 2 is for Saturation, and i = 3 is for Value) for all columns and rows (":" stands for all of pixel indexes). $\underline{H}$ and $\overline{H}$, $\underline{S}$ and $\overline{S}$, $\underline{V}$ and $\overline{V}$, are the minimum and maximum values of the HSV values of colored circles, respectively. A is the output bitwise matrix that has several objects (let us call the clusters of true valued pixels as objects) on it. Note that there are 8 different logical operations (6 comparison, 2 AND operations) performed on the matrix Im, however in most of the cases we just need three or even two of these operations which needs lower computational efforts that are usually very important in image processing applications. The objects are labelled according to the 8th neighborhood rule with the build in function of Matlab. To eliminate the noise in the binary images some post-processing methods like erosion and dilation may be utilized on the images. The next step is to find the centers of these objects. A simple center of geometry algorithm is run and the positions of these center of geometries is kept in memory. Now, the problem is to find which three points belong to the same robot. For this purpose the distances between the points are utilized. Simply the closest three ones are said to belong to the same robot. This approach is simple and fast but, has one drawback which is when there are robots too close to each other some of the points on these robots may get mixed up and show nonexisting robots. In Figure A.2b, some of the bounding circles of robots in which other robots should not travel are shown. It might be also possible to develop a robot identification method which utilizes the previous positions of the robots to overcome this problem; however, so far we have not considered such an approach. Moreover, not allowing the robots to get too close to each other is also good for collision avoidance. This should be guaranteed by the control algorithm. Following the identification process, the positions of the three colored circles of each robot are determined. After that, the positions of the robots are found simply by averaging these three points of each robot.

A.3.2 Determining the Robot Orientations

Note that the colored circles are placed such that they form an isosceles triangle. The vertex on the intersection of the equal edges is called P_3 and the remaining two points on the other ends of the equal edges are called P_1 and P_2 (see Figure A.3(a)). The hat is placed such that the vector from the midpoint of the line connecting the points P_1 and P_2 to the point P_3 is the heading of the robot. The problem here is to determine which point is P_3 (which is also called the heading point). To determine the identification of points we again utilize the distances between the points. The geometry (isosceles triangle) allows us to state that the farthermost point of these three points is the heading point. The remaining two points P_1 and P_2 are identified depending on whether they are on the left or right of the robot. Finding the heading point is sufficient to compute the orientation of the robot. There is no need to determine whether remaining two points are on the left or right hand side of the robot. The average of the points P_1 and P_2 gives the starting point of the orientation vector and P_3 is the end point of this vector. Lastly we compute the unit orientation vector of each robot and store in the memory.



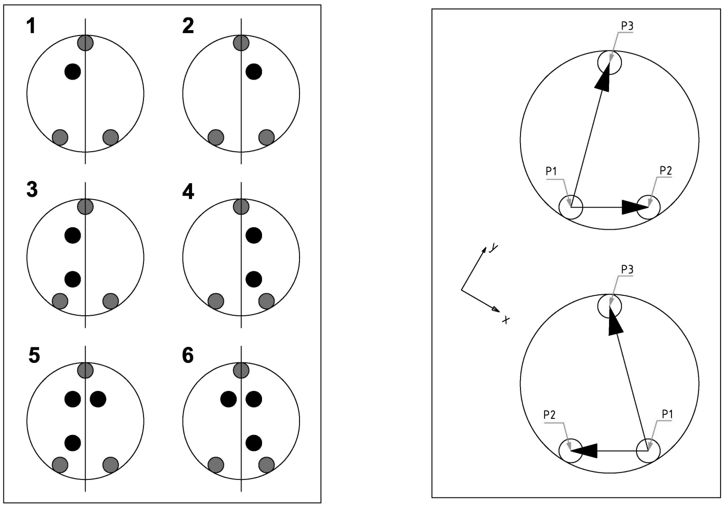
(a) (b)

Figure A.3: (a) The colored dots that are used to find the position and orientation of a robot. (b)Speeds of a differential drive robot.

A.3.3 Determining Robot ID's

As mentioned above for the position and orientation calculations there is no need to find whether P_1 (or P_2) is on the left or right of the robot. However, it is required for the identification of the robots. The regions on the left and right of the robots are utilized to place the black dots in the regions and use an algorithm that checks the number of black dots present in these regions in order to identify the robots. In Figure A.4a the black dot placements are shown for 6 different robot hats.

At most two black dots are used on the left and right sides of the robots. Here note that we used two methods simultaneously for the identification: (i) the location of the black dots (ii) and the number of black dots on these two regions. It would be easier to identify the robots based only on the number of black dots without dividing the area into regions (such as one dot = robot 1, six dots = robot 6 etc). However, in that case one needs to use as many black dots as the number of robots (i.e., 6 dots for 6 robots). This would result in some problem due to the size of the robots and resolution of the camera when the number of robots increases. By adding the method of dividing regions (left and right in our case) allows us to use less number of black dots (at most 3 for 6 robots) and larger black dots which will be represented with more/enough number of pixels in the image. This approach is better when the number of robots is higher. For example by using 3 dots on each left and right of the regions we



(a) (b)

Figure A.4: (a) Robot Identities. (b) Vectors from P_1 to P_2 and P_1 to P_3 for two different cases: P_1 is on the left for the left figure and P_1 is on the right for the right figure.

can identify 16 ($4^2 = 16$) different robots or with 4 dots, 25 ($5^2 = 25$) different robots. On the other hand, increasing the regions of interest (say top and bottom in addition to left and right-in total 4 regions) will allow one to use less number of larger black dots with enough spacing. Note that the larger the black dots and the larger the spacing between them would give better images that one can process. The black dots are determined in the image matrix similar to the colored ones mentioned in Section A.3.1. This time HSV value intervals are set for black. The only problem left is to find the left and right regions of the robots.

For determining the regions on the left and right hand side of robots we need to first find which of the points P_1 and P_2 are on the left (or right). For this purpose we will simply utilize the cross products of the vectors from the point P_1 to P_2 and P_3. If this cross product is negative then P_1 is on the right hand side of the robot (so P_2 is on the left) and if it is positive then P_1 is on the left hand side while P_2 is on the right. The vectors and the global coordinate axes are shown in Figure A.4b. Mathematically speaking

$$P_1 \; is \; on \; the \; left \; if \qquad sign(\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}) > 0 \tag{A.2a}$$

$$P_1 \; is \; on \; the \; right \; if \qquad sign(\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}) < 0 \tag{A.2b}$$

In image processing applications similar to the one discussed here, there are many factors effecting the performance of the system in achieving the goals. One of the main disturbances is the non-unique light source. Some low quality cameras may become very volatile in getting the colored images of the objects. The color values of the objects may change tremendously such that the values may not become stable between the pre-specified threshold values in getting the binary matrices of objects. Therefore, we recommend cameras with auto focus, adjusting luminance of images, and a qualified lens. Otherwise, one may have to calibrate the system (the software) before each experiment. The resolution of the images that we grab from the camera in our setup is 640×480. One would prefer to use higher resolutions which would result in better object detections but with a longer processing time. Image processing time is an important criteria affecting the response time of the system. In fact, the main cause of delays is slow image processing in most of our applications. Therefore, a high capacity/speed memory and processing units of computers and efficient coding (with less memory usage, optimized image processing and appropriate variable types) would result in higher speed.

A.4 Transmitting Position and Orientation Information

The position and orientation information gathered from the overhead images of the arena can be used for determining the new translational and angular speeds of the robots according to the behavioral algorithms/models investigated in the particular application under consideration. The computing unit (i.e., PC, Laptop) should pass these new speed setting information to each robot. The bluetooth interface is utilized in this setup. Each robot is connected to the master processing unit via bluetooth. However, the bluetooth interface can support at most 7 slaves at the same time. Therefore, the set-up may work for at most 7 robots. For higher number of robots alternative communication units can be: Zigbee and wifi.

The robots we utilized are differential drive robots. They have two stepper motors which can be driven at different speeds. Therefore, speed information of each motor are transmitted via bluetooth. The mathematical relationship between the left and right motor speeds and the translational and angular speeds of the robots can be obtained as

$$V = \frac{V_{right} + V_{left}}{2} = \frac{r}{2}(\omega_R + \omega_L) \ \ ; \ \ \omega = \frac{V_{right} - V_{left}}{L} = \frac{r}{L}(\omega_R - \omega_L) \qquad \text{(A.3)}$$

where V and ω are the translational and rotational speeds of the robot, respectively. V_{left} and

V_{right} are the left and right wheel speeds, respectively. L is the distance between the left and right wheels. The speeds are shown in Figure A.3(b).

A.5 Concluding Remarks

The objective of this study is to develop an experimental set-up for swarm robot applications. The small sized relatively simple mobile robots called E-puck are utilized in the set-up for the time being. However, it can be easily used with other robot platforms as well. The arena is observed by a high quality USB camera connected to a high speed PC. We perform the image processing works in Matlab and feedback the positions and orientations of robots to the behavior algorithms governing the swarm dynamics. The main delays are resulted from the image processing computations. Further optimization of the algorithms for faster response of the system is still possible.

We believe that this test bed is a very useful experimental facility which can be used for testing swarm coordination and control algorithms.

REFERENCES

[1] Cao, Y.U., Fukunaga, A.S., Kahng, A.B.: Cooperative mobile robotics: Antecedents and directions. Autonomous Robots **4**(1) (1997) 7–23

[2] Mataric, M.J.: Issues and approaches in the design of collective autonomous agents. Robotics and Autonomous Systems **16**(2/4) (1995) 321–332

[3] Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford Univ. Press, NY (1999)

[4] Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publisher (2001)

[5] Clerc, M., Kennedy, J.: The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. on Evolutionary Computation**6**(1) (2002) 58–73

[6] Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine **22**(3) (2002) 52–67

[7] Liu, Y., Passino, K.M.: Biomimicry of social foraging behavior for distributed optimization: Models, principles, and emergent behaviors. Journal of Optimization Theory and Applications **115**(3) (2002) 603–628

[8] Akyildiz, I.F., Su, W., Sankarasubramniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Commununications Magazine **40**(8) (2002) 102–114

[9] Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. Comp. Graph. **21**(4) (1987) 25–34

[10] Balch, T., Arkin, R.C.: Behavior-based formation control for multirobot teams. IEEE Trans. on Robotics and Automation**14**(6) (1998) 926–939

[11] Breder, C.M.: Equations descriptive of fish schools and other animal aggregations. Ecology **35**(3) (1954) 361–370

[12] Warburton, K., Lazarus, J.: Tendency-distance models of social cohesion in animal groups. Journal of Theoretical Biology**150** (1991) 473–488

[13] Okubo, A.: Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. Advances in Biophysics **22** (1986) 1–94

[14] Grünbaum, D., Okubo, A.: Modeling social animal aggregations. In: Frontiers in Theoretical Biology. Volume 100 of Lecture Notes in Biomathematics. Springer-Verlag, New York (1994) 296–325

[15] Grünbaum, D.: Schooling as a strategy for taxis in a noisy environment. Evolutionary Ecology **12** (1998) 503–522

[16] Parrish, J.K., Viscido, S.V., Grünbaum, D.: Self-organized fish school: An examination of emergent properties. Biol. Bull. **202** (2002) 296–305

[17] Gazi, V., Passino, K.M.: Stability analysis of swarms. IEEE Trans. on Automatic Control**48**(4) (2003) 692–697

[18] Gazi, V., Passino, K.M.: A class of attraction/repulsion functions for stable swarm aggregations. Int. J. Control**77**(18) (2004) 1567–1579

[19] Gazi, V., Passino, K.M.: Stability analysis of social foraging swarms. IEEE Trans. on Systems, Man, and Cybernetics: Part B **34**(1) (2004) 539–557

[20] Liu, Y., Passino, K.M.: Stable social foraging swarms in a noisy environment. IEEE Transactions on Automatic Control **49**(1) (2004) 30–44

[21] Liu, Y., Passino, K.M., Polycarpou, M.M.: Stability analysis of one-dimensional asynchronous swarms. IEEE Trans. on Automatic Control**48**(10) (2003) 1848–1854

[22] Liu, Y., Passino, K.M., Polycarpou, M.M.: Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology. IEEE Trans. on Automatic Control**48**(1) (2003) 76–95

[23] Gazi, V., Passino, K.M.: Stability of a one-dimensional discrete-time asynchronous swarm. IEEE Trans. on Systems, Man, and Cybernetics: Part B **35**(4) (2005) 834–841

[24] Gazi, V.: Swarm aggregations using artificial potentials and sliding mode control. IEEE Trans. on Robotics**21**(6) (2005) 1208–1214

[25] Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-Organization in Biological Systems. Princeton University Press, New Jersey (2001)

[26] Dorigo, M., Sahin, E., eds.: Special Issue on Swarm Robotics. Volume 17. Autonomous Robots (2004)

[27] Kumar, V.J., Leonard, N.E., Morse, A.S., eds.: Cooperative Control: 2003 Block Island Workshop on Cooperative Control. Volume 309 of Lecture Notes in Control and Information Sciences. Springer-Verlag (2005)

[28] Sahin, E., Spears, W.M., eds.: Swarm Robotics, A State of the Art Survey. Lecture Notes in Computer Science 3342. Springer-Verlag, Berlin Heidelberg (2005)

[29] Sahin, E., Spears, W.M., Winfield, A.F.T., eds.: Proceedings of the SAB06 Workshop on Swarm Robotics Swarm Robotics. Lecture Notes in Computer Science (LNCS) 4433. Springer-Verlag, Berlin Heidelberg (2007)

[30] Sahin, E.: Swarm robotics: From sources of inspiration to domains of application. In Sahin, E., Spears, W., eds.: Swarm Robotics: State-of-the-art Survey. Lecture Notes in Computer Science (LNCS 3342). Springer-Verlag, Berlin Heidelberg (2005) 10–20

[31] Gazi, V., Fidan, B.: Coordination and control of multi-agent dynamic systems: Models and approaches. In Sahin, E., Spears, W.M., Winfield, A.F.T., eds.: Proceedings of the SAB06 Workshop on Swarm Robotics Swarm Robotics. Lecture Notes in Computer Science (LNCS) 4433. Springer-Verlag, Berlin Heidelberg (2007) 71–102

[32] Guldner, J., Utkin, V.I.: Sliding mode control for gradient tracking and robot navigation using artificial potential fields. IEEE Trans. on Robotics and Automation**11**(2) (1995) 247–254

[33] Campion, G., Bastin, G., Dandrea-Novel, B.: Structural properties and classification of kinematic and dynamicmodels of wheeled mobile robots. IEEE Tr. on Robotics and Automation **12**(1) (1996) 47–62

[34] Yi, B.J., Kim, W.: The kinematics for redundantly actuated omnidirectional mobile robots. Journal of Robotic Systems **19**(6) (2002) 255–267

[35] Yamaguchi, H.: A cooperative hunting behavior by mobile-robot troops. The International Journal of Robotics Research **18**(8) (1999) 931–940

[36] Tanner, H., Pappas, G.J., Kumar, V.: Leader-to-formation stability. IEEE Trans. on Robotics and Automation **20**(3) (2004) 443–455

[37] Sandeep, S., Fidan, B., Yu, C.: Decentralized cohesive motion control of multi-agent formations. In: Proc. 14th Mediterranean Conference on Control and Automation, Ancona, Italy (2006)

[38] Leonard, N.E., Fiorelli, E.: Virtual leaders, artificial potentials and coordinated control of groups. In: Proc. Conf. Decision Contr., Orlando, FL (2001) 2968–2973

[39] Bachmayer, R., Leonard, N.E.: Vehicle networks for gradient descent in a sampled environment. In: Proc. Conf. Decision Contr., Las Vegas, Nevada (2002) 112–117

[40] Tanner, H.G., Jadbabaie, A., Pappas, G.J.: Stable flocking of mobile agents, part i: Fixed topology. In: Proc. Conf. Decision Contr., Maui, Hawaii (2003) 2010–2015

[41] Tanner, H.G., Jadbabaie, A., Pappas, G.J.: Stable flocking of mobile agents, part ii: Dynamic topology. In: Proc. Conf. Decision Contr., Maui, Hawaii (2003) 2016–2021

[42] Olfati-Saber, R.: Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Trans. on Automatic Control**51**(3) (2006) 401–420

[43] Yao, J., Ordonez, R., Gazi, V.: Swarm tracking using artificial potentials and sliding mode control. In: Proc. Conf. Decision Contr., San Diago, CA, USA (2006)

[44] Egerstedt, M., Hu, X.: Formation constrained multi-agent control. IEEE Trans. on Robotics and Automation**17**(6) (2001) 947–951

[45] Desai, J.P., Ostrowski, J., Kumar, V.: Modeling and control of formations of nonholonomic mobile robots. IEEE Trans. on Robotics and Automation**17**(6) (2001) 905–908

[46] Lin, Z., Francis, B., Maggiore, M.: Necessary and sufficient graphial conditions for formation control of unicycles. IEEE Trans. on Automatic Control**50**(1) (2005) 121–127

[47] Marshall, J., Broucke, M., Francis, B.: Formations of vehicles in cyclic pursuit. IEEE Trans. on Automatic Control **49**(11) (2004) 1963–1974

[48] Tanner, H.G., Jadbabaie, A., Pappas, G.J.: Flocking in teams of nonholonomic agents. In V.J. Kumar, N.L., Morse, A., eds.: Cooperative Control. Volume 309 of Lecture Notes in Control and Information Sciences., Springer-Verlag (2005) 229–239

[49] Lawton, J.R.T., Beard, R.W., Young, B.J.: A decentralized approach to formation maneuvers. IEEE Trans. on Robotics and Automation**19**(6) (2003) 933–941

[50] Gazi, V.: Stability Analysis of Swarms. PhD thesis, The Ohio State University (2002)

[51] Sepulchre, R., Palay, D., Leonard, N.E.: Collective motion and oscillator synchronization. In Kumar, V.J., Leonard, N.E., Morse, A.S., eds.: Cooperative Control: 2003 Block Island Workshop on Cooperative Control. Volume 309 of Lecture Notes in Control and Information Sciences., Springer-Verlag (2005)

[52] Dubins, L.: On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. American Journal of Mathematics **79** (1957) 497–516

[53] Savla, K., Bullo, F., Frazzoli, E.: On traveling salesperson problems for Dubins' vehicle: stochastic and dynamic environments. In: Proc. 44th IEEE Conference on Decision and Control and the European Control Conference 2005. (2005) 4530–4535

[54] Tomlin, C., Mitchell, I., Ghosh, R.: Safety verification of conflict resolution manoeuvres. IEEE Tr. on Intelligent Transportation Systems **2**(2) (2001) 110–120

[55] Boscain, U., Piccoli, B.: Optimal Syntheses for Control Systems on 2-D Manifolds. Springer Verlag, New York, NY (2004)

[56] Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. Physical Review Letters **75**(6) (1995) 1226–1229

[57] Czirók, A., Stanley, H.E., Vicsek, T.: Spontaneously ordered motion of self-propelled particles. Journal of Physics A Mathematical General **30** (1997) 1375–1385

[58] Czirok, A., Vicsek, T.: Collective behavior of interacting self-propelled particles. Physica A Statistical Mechanics and its Applications **281** (2000) 17–29

[59] Czirók, A., Ben-Jacob, E., Cohen, I., Vicsek, T.: Formation of complex bacterial colonies via self-generated vortices. Physical Review E **54** (1996) 1791–1801

[60] Czirók, A., Barabási, A.L., Vicsek, T.: Collective motion of self-propelled particles: Kinetic phase transition in one dimension. Physical Review Letters **82**(1) (1999) 209–212

[61] Vicsek, T.: Application of statistical mechanics to collective motion in biology. Physica A Statistical Mechanics and its Applications **274** (1999) 182–189

[62] Savkin, A.V.: Coordinated collective motion of groups of autonomous mobile robots: Analysis of vicsek's model. IEEE Transactions on Automatic Control **49**(6) (2004) 981–983

[63] Jadbabaie, A., Lin, J., Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control **48**(6) (2003) 988–1001

[64] Moreau, L.: Stability of multiagent systems with time-dependent communication links. IEEE Transactions on Automatic Control **50**(2) (2005) 169–182

[65] Ren, W., Beard, R.W.: Consensus seeking in multi-agent systems under dynamically changing interaction topologies. IEEE Trans. on Automatic Control**50**(5) (2005) 655–661

[66] Levine, H., Rappel, W.J., Cohen, I.: Self-organization in systems of self-propelled particles. Physical Review E **63**(17101) (2000) 1–4

[67] Gazi, V., Köksal and B. Fidan, M.I.: Aggregation in a swarm of non-holonomic agents using artificial potentials and sliding mode control. In: Proc. European Control Conf., Kos, Greece (2007) 1485–1491

[68] Gazi, V., Fidan, B., Hanay, Y.S., Köksal, M.I.: Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques. Turkish Journal of Electrical Engineering and Computer Sciences **15**(2) (2007) 149–168

[69] Grünbaum, D.: Schooling as a strategy for taxis in a noisy environment. In Parrish, J.K., Hamner, W.M., eds.: Animal Groups in Three Dimensions. Cambridge Iniversity Press (1997) 257–281

[70] Tanner, H.G., Jadbabaie, A., Pappas, G.J.: Flocking in teams of nonholonomic agents. S. Morse, N. Leonard and V. Kumar (eds.), Cooperative Control, Lecture Notes in Control and Information Sciences, Springer **309** (2002) 229–239

[71] Beni, G., Liang, P.: Pattern reconfiguration in swarms—convergence of a distributed asynchronous and bounded iterative algorithm. IEEE Trans. on Robotics and Automation**12**(3) (1996) 485–490

[72] Beni, G.: Order by disordered action in swarms. In Sahin, E., Spears, W.M., eds.: Proc. SAB 2004 International Workshop on Swarm Robotics. Lecture Notes in Computer Science (LNCS 3342). Springer Verlag (2004) 153–171

[73] Gordon, N., Wagner, I.A., Bruckstein, A.M.: Gathering multiple robotic a(ge)nts with limited sensing capabilities. Lecture Notes in Computer Science **3172** (2004) 142–153

[74] Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous oblivious robots with limited visibility. Lecture Notes in Computer Science **2010** (2001) 247–258

[75] Soysal, O., Sahin, E.: Probabilistic aggregation strategies in swarm robotic systems. In: Proc. of the IEEE Swarm Intelligence Symposium, Pasadena, California (2005)

[76] Bahceci, E., Sahin, E.: Evolving aggregation behaviors for swarm robotic systems: A systematic case study. In: Proc. of the IEEE Swarm Intelligence Symposium, Pasadena, California (2005)

[77] Lin, Z., Broucke, M., Francis, B.: Local control strategies for groups of mobile autonomous agents. IEEE Trans. on Automatic Control **49**(4) (2004) 622–629

[78] Das, A., Fierro, R., Kumar, V.: Control graphs for robot networks. In Butenko, S., Murphey, R., Pardalos, P., eds.: Cooperative Control: Models, Applications and Algorithms, Kluwer Academic (2003) 55–73

[79] Fierro, R., Song, P., Das, A., Kumar, V.: Cooperative control of robot formations. In Murphey, R., Pardalos, P., eds.: Cooperative Control and Optimization, Kluwer Academic (2002) 73–94

[80] Olfati-Saber, R., Murray, R.M.: Distributed cooperative control of multiple vehicle formations using structural potential functions. In: Proc. IFAC World Congress, Barcelona, Spain (2002)

[81] Anderson, B., Yu, C., Fidan, B., Hendrickx, J.: Control and information architectures for formations. In: Proc. IEEE Conference on Control Applications (Joint CCA/CACSD/ISIC). (2006)

[82] Eren, T., Anderson, B., Morse, A., Whiteley, W., Belhumeur, P.: Operations on rigid formations of autonomous agents. Communications in Information and Systems **3**(4) (2004) 223–258

[83] Yu, C., Fidan, B., Anderson, B.: Persistence acquisition and maintenance for autonomous formations. In: Proc. 2nd Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). (2005) 379 – 384

[84] Yu, C., Fidan, B., Anderson, B.: Principles to control autonomous formation merging. In: Proc. American Control Conference. (2006) 762 – 768

[85] Das, A., Fierro, R., Kumar, V., Ostrowski, J.: A vision-based formation control framework. IEEE Trans. on Robotics and Automation **18**(5) (2002) 813–825

[86] P. Ögren, Fiorelli, E., Leonard, N.E.: Formations with a mission: Stable coordination of vehicle group maneuvers. In: Symposium on Mathematical Theory of Networks and Systems. (2002)

[87] Ren, W., Beard, R.W., Atkins, E.M.: A survey of consensus problems in multi-agent coordination. In: Proc. American Control Conf., Portland, OR, USA (2005) 1859–1864

[88] Lin, J., Morse, A.S., Anderson, B.D.O.: The multi-agent rendezvous problem. In: Proc. Conf. Decision Contr., Maui, Hawaii, USA (2003) 1508–1513

[89] Lin, J., Morse, A.S., Anderson, B.D.O.: The multi-agent rendezvous problem - the asynchronous case. In: Proc. Conf. Decision Contr., Atlantis, Paradise Island, Bahamas (2004) 1926–1931

[90] Gordon, N., Wagner, I.A., Bruckstein, A.M.: Gathering multiple robotic a(ge)nts with limited sensing capabilities. In Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T., eds.: Proceedings of ANTS 2004 – Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence. Volume 3172 of Lecture Notes in Computer Science., Brussels, Belgium, Springer Verlag (2004) 142–153

[91] Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of autonomous mobile robots with limited visibility. In: Proc. 18th International Symposium on Theoretical Aspects of Computer Science (STACS 2001). Volume 2010 of Lecture Notes in Computer Science., Dresden, Germany, Springer Verlag (2001) 247–258

[92] Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific, Belmont, MA (1997)

[93] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research **5**(1) (1986) 90–98

[94] Rimon, E., Koditschek, D.E.: Exact robot navigation using artificial potential functions. IEEE Trans. on Robotics and Automation**8**(5) (1992) 501–518

[95] Reif, J.H., Wang, H.: Social potential fields: A distributed behavioral control for autonomous robots. Robotics and Autonomous Systems**27** (1999) 171–194

[96] Spears, W.M., Gordon, D.F.: Using artificial physics to control agents. In: Proceedings of the IEEE International Conference on Information, Intelligence, and Systems. (1999) 281–288

[97] Zarzhitsky, D., Spears, D.F., Spears, W.M., Thayer, D.R.: A fluid dynamics approach to multi-robot chemical plume tracing. In: AAMAS. (2004) 1476–1477

[98] Zarzhitsky, D., Spears, D.F., Thayer, D.R., Spears, W.M.: Agent-based chemical plume tracing using fluid dynamics. In: FAABS. (2004) 146–160

[99] Utkin, V.I.: Sliding Modes in Control and Optimization. Springer Verlag, Berlin, Heidelberg (1992)

[100] Gazi, V.: Formation control of mobile robots using decentralized nonlinear servomechanism. In: 12'th Meditteranean Conference on Control and Automation, Kusadasi, Turkey (2004)

[101] Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. on Automatic Control**49**(9) (2004) 1520–1533

[102] Ögren, P., Egerstedt, M., Hu, X.: A control Lyapunov function approach to multi-agent coordination. IEEE Trans. on Robotics and Automation**18**(5) (2002) 847–851

[103] P. Ögren, Fiorelli, E., Leonard, N.E.: Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. IEEE Trans. on Automatic Control**49**(8) (2004) 1292–1302

[104] Wu, H., Jagannathan, S.: Adaptive neural network control and wireless sensor network-based localization for UAV formation. In: Proc. 14th Mediterranean Conference on Control and Automation, Ancona, Italy (2006)

[105] Butenko, S., Murphey, R., Pardalos, P., eds.: Cooperative Control: Models, Applications and Algorithms. Kluwer Academic (2003)

[106] Murphey, R., Pardalos, P., eds.: Cooperative Control and Optimization. Kluwer Academic (2002)

[107] Pettersen, K., 'and H. Nijmeijer, J.G., eds.: Group Coordination and Cooperative Control. Volume 336 of Lecture Notes in Control and Information Sciences. Springer-Verlag (2006)

[108] Şamiloglu, A.T., Gazi, V., Koku, A.B.: Asynchronous cyclic pursuit. In et al., S.N., ed.: Proc. of 9'th Conference on Simulation of Adaptive Behavior (SAB06). Lecture Notes in Artificial Intelligence (LNAI) 4095. Springer Verlag (2006) 667–678

[109] Şamiloglu, A.T., Gazi, V., Koku, A.B.: Effects of asynchronism and neighborhood size on clustering in self-propelled particle systems. In Levi, A., et al., eds.: Proc. of International Symposium on Computer and Information Sciences (ISCIS06). Lecture Notes in Computer Science (LNCS) 4263. Springer Verlag (2006) 665–676

[110] Şamiloglu, A.T., Gazi, V., Koku, A.B.: An empirical study on the motion of self-propelled particles with turn angle restrictions. In Şahin et al., E., ed.: Proc. of SAB06 Workshop on Swarm Robotics. Lecture Notes in Computer Science (LNCS). Springer Verlag (2006)

[111] Gueron, S., Levin, S.A.: The dynamics of group formation. Mathematical Biosciences**128** (1995) 243–264

[112] Durrett, R., Levin, S.: The importance of being discrete (and spatial). Theoretical Population Biology**46** (1994) 363–394

[113] Agassounon, W., Martinoli, A., Easton, K.: Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. Autonomous Robots **17**(2-3) (2004) 163–192

[114] Lerman, K., Martinoli, A., Galstyan, A.: A review of probabilistic macroscopic models for swarm robotic systems. In Sahin, E., Spears, W., eds.: Swarm Robotics: State-of-the-art Survey. Lecture Notes in Computer Science (LNCS 3342), Berlin Heidelberg, Springer-Verlag (2005) 143–152

[115] Soysal, O., Sahin, E.: A macroscopic model for probabilistic aggregation in swarm robotic systems. In Şahin et al., E., ed.: Proc. of SAB06 Workshop on Swarm Robotics. Lecture Notes in Computer Science (LNCS). Springer Verlag (2006)

[116] Bernhart, A.: Polygons of Pursuit. Scripta Mathematica (1959)

[117] Klamkin, M.S., Newman, D.J.: Cyclic pursuit or "the three bugs problem". The American Mathematical Monthly **78**(6) (1971) 631–639

[118] Behroozi, F., Gagnon, R.: Cyclic pursuit in a plane. Journal of Mathematical Physics **20**(11) (1979) 2212–2216

[119] Richardson, T.J.: Non-mutual captures in cyclic pursuit. Annals of Mathematics and Artificial Intelligence **31** (2001) 127–146

[120] Bruckstein, A.M., Cohen, N., Efrat, A.: Ants, crickets and frogs in cyclic pursuit. Center Intell. Syst., Technion-Israel Inst. Technol. (1991)

[121] Marshall, J.A., Broucke, M.E., Francis, B.A.: A pursuit strategy for wheeled-vehicle formations. Proceedings of the 42nd IEEE Conference on Decision and Control (2003) 2555–2560

[122] Marshall, J.A., Broucke, M.E., Francis, B.A.: Formations of vehicles in cyclic pursuit. IEEE Transactions on automatic control **49**(11) (2004) 1963–1974

[123] Marshall, J.A., Broucke, M.E., Francis, B.A.: Pursuit formations of unicycles. Automatica **42**(1) (2006) 3–12

[124] Lin, Z., Broucke, M., Francis, B.: Local control strategies for groups of mobile autonomous agents. IEEE Transactions on Automatic Control **49**(4) (2004) 622–629

[125] Ren, W., Beard, R.W.: Consensus seeking in multi-agent systems under dynamically changing interaction topologies. IEEE Trans. on Automatic Control **50**(5) (2005) 655–661

[126] Lin, J., Morse, A.S., Anderson, B.D.O.: The multi-agent rendezvous problem - the asynchronous case. In: Proc. of Conf. Decision and Control, Atlantis, Paradise Island, Bahamas (2004) 1926–1931

[127] Sepulchre, R., Palay, D., Leonard, N.E.: Collective motion and oscillator synchronization. In V.J. Kumar, N.E. Leonard, A.M., ed.: in Proc. of the 2003 Block Island Workshop on Cooperative Control, Springer-Verlag (2003)

[128] HORN, R.A., R.JOHNSON, C.: Matrix Analysis. Cambridge University Press (1992)

[129] Şamiloglu, A.T., Gazi, V., Koku, A.B.: Comparison of three orientation agreement strategies in self-propelled particle systems with turn angle restrictions in synchronous and asynchronous settings. Asian Journal of Control **10**(2) (2008) 212–232

[130] Angeli, D., Bliman, P.A.: Stability of leaderless multi-agent systems. extension of a result by moreau. arXiv:math.OC/0411338 v1 (2004)

[131] Fang, L., Antsaklis, P.J., Tzimas, A.: Asynchronous censensus protocols: Preliminary results, simultionsi and open questions. In: Proc. Conf. Decision Contr.and Proc. European Control Conf., Sevile, Spain (2005) 2194–2199

[132] Blondel, V.D., Hendrickx, J.M., Olshevsky, A., Tsitsiklis, J.: Convergence in multiagent coordination, consensus, and flocking. In: Proc. Conf. Decision Contr.and Proc. European Control Conf., Sevile, Spain (2005) 2996–3000

[133] Cao, M., Morse, A.S., Anderson, B.D.O.: Agreeing asynchronously: Anouncement of results. In: Proc. Conf. Decision Contr., San Diego, CA, USA (2006) 4301–4306

[134] Gazi, V.: Stability of a discrete-time asynchronous swarm with time-dependent communication links. In: IEEE Trans. on Systems, Man, and Cybernetics: Part B. Volume 38. (2008) 267–274 to appear.

[135] Stark, H., Woods, J.W.: Probability, Random Processes, and Estimation Theory for Engineers. Prentice Hall (1994)

[136] Papoulis, A.: Probability, Random Variables and Stochastic Processes. 3 edn. McGraw-Hill Companies (1991)

[137] Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T.: A stable tracking control method for an autonomous mobile robot. Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on (1990) 384–389 vol.1

[138] Aicardi, M. Casalino, G.B.A.B.A.: Closed loop smooth steering of unicycle like vehicles. IEEE CONFERENCE ON DECISION AND CONTROL **3**(33) (1994) 2455

[139] Huang, L., Tang, L., Box, P., Zealand, W.: (Dynamic Target Tracking Control for a Wheeled Mobile Robots Constrained by Limited Inputs)

[140] Iida, S., Yuta, S.: Vehicle command system and trajectory control for autonomous mobile robots. Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on (1991) 212–217 vol.1

[141] Lee, S.O., Cho, Y.J., Hwang-Bo, M., You, B.J., Oh, S.R.: A stable target-tracking control for unicycle mobile robots. Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on **3** (2000) 1822–1827 vol.3

[142] Lee, T.C., Song, K.T., Lee, C.H., Teng, C.C.: Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. Control Systems Technology, IEEE Transactions on **9**(2) (2001) 305–318

[143] Shim, H.S., Sung, Y.G.: Stability and four-posture control for nonholonomic mobile robots. Robotics and Automation, IEEE Transactions on **20**(1) (2004) 148–154

[144] Fukao, T., Nakagawa, H., Adachi, N.: Adaptive tracking control of a nonholonomic mobile robot. Robotics and Automation, IEEE Transactions on **16**(5) (2000) 609–615

[145] Xie, F., Fierro, R.: Stabilization of Nonholonomic Robot Formations: A First-state Contractive Model Predictive Control Approach. Journal of Computing and Information Technology **17**(1) (2009) 37–50

[146] Chwa, D.: Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. Control Systems Technology, IEEE Transactions on **12**(4) (2004) 637–644

[147] Li, T.H., Chang, S.J., Tong, W.: Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. Fuzzy Systems, IEEE Transactions on **12**(4) (2004) 491–501

[148] Koksal, M.I., Gazi, V., Fidan, B., Ordonez, R.: Tracking a maneuvering target with a non-holonomic agent using artificial potentials and sliding mode control. 16th Mediterranean Conference on Control and Automation (2008) 1174–1179

[149] Gazi, V., Ordonez, R.: Target tracking using artificial potentials and sliding mode control. International Journal of Control **80**(10) (2007) 1626–1635

[150] Hung, N., Im, J., Jeong, S., Kim, H., Kim, S.: Design of a sliding mode controller for an automatic guided vehicle and its implementation. International Journal of Control, Automation and Systems **8**(1) (2010) 81–90

[151] Solea, R., Nunes, U.: Trajectory planning and sliding-mode control based trajectory-tracking for cybercars. Integrated Computer-Aided Engineering **13** (2006) 1–15

[152] Solea, R., Cernega, D.: Sliding Mode Control for Trajectory Tracking Problem-Performance Evaluation. (Artificial Neural Networks–ICANN 2009) 865–874

[153] Yang, J., Kim, J.: Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION **15**(3) (1999) 579

[154] Micaelli, A., Samson, C.: Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Rapport de recherche (**2097**)

[155] Oriolo, G., De Luca, A., Vendittelli, M., et al.: WMR control via dynamic feedback linearization: design, implementation, and experimental validation. IEEE Transactions on Control Systems Technology **10**(6) (2002) 835–852

[156] Yang, E., Gu, D., Mita, T., Hu, H.: Nonlinear tracking control of a car-like mobile robot via dynamic feedback linearization. dim (**100**) 0

[157] Huang, H., Tsai, C.: Simultaneous tracking and stabilization of an omnidirectional mobile robot in polar coordinates: a unified control approach. Robotica **27**(03) (2008) 447–458

[158] Park, K., Chung, H., Lee, J.: Point stabilization of mobile robots via state space exact feedback linearization. In: Proceedings of SPIE. Volume 3693. (1999) 21

[159] Noijen, S., Lambrechts, P., Nijmeijer, H.: An observer-controller combination for a unicycle mobile robot. International Journal of Control **78**(2) (2005) 81–87

[160] D'Andrea-Novel, B. Campion, G.B.G.: Control of nonholonomic wheeled mobile robots by state feedback linearization. INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH **14**(6) (1995) 543–559

[161] Soetanto, D., Lapierre, L., Pascoal, A.: Adaptive, non-singular path-following control of dynamic wheeled robots. In: 42nd IEEE Conference on Decision and Control, 2003. Proceedings. Volume 2. (2003)

[162] Teimoori, H., Savkin, A.: Equiangular navigation and guidance of a wheeled mobile robot based on range-only measurements. Robotics and Autonomous Systems (2009)

[163] SwisTrack: A tracking tool for multi-unit robotic and biological research, Beijing, China (2006)

[164] Lucas P. J. J. Noldus, Andrew J. Spink, R.A.J.T.: Computerised video tracking, movement analysis and behaviour recognition in insects. Computers and Electronics in Agriculture **35**(2-3) (2002) 201–227

[165] Trifa, V., Cianci, C.M., Guinard, D.: Dynamic control of a robotic swarm using a service-oriented architecture. In: Proceedings of International Symposium on Artificial Life and Robotics, Beppu, Japan (2008)

[166] Fiala, M.: Artag, a fiducial marker system using digital techniques. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on **2** (2005) 590–596 vol. 2

[167] Hayes, A., Dormiani-Tabatabaei, P.: Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on **4** (2002) 3900–3905 vol.4

[168] Hawick, K.A., James, H.A.: Middleware for context sensitive mobile applications. In: ACSW Frontiers '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2003) 133–141

[169] E-puck Robots: E-puck robot specifications. Available from http://www.e-puck.org (2008)

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ŞAMİLOĞLU, Andaç Töre
Nationality: Turkish (TC)
Date and Place of Birth: 2 December 1979 , Kars
Marital Status: Single
Phone (Work): +90 312 246 66 66 / 1459
Phone (Mobile): +90 530 460 39 46
E-mail: andactore@gmail.com
Address: Makine Muhendisliği, Başkent Üniversitesi, Bağlıca Kampüsü, Etimesgut/ANKARA

EDUCATION

Ph.D. (2012), METU, Mechanical Engineering, Thesis Title: "Decentralized Coordination and Control in Robotic Swarms"

MSc. (2007), METU, Economics, Thesis Title: "Export Dynamics, Size, and Productivity of Firms"

MSc. (2005), METU, Mechanical Engineering, Thesis Title: "Mathematical Modeling and Simulation of Commercial Vehicle Performance"

BSc. (2003), METU, Mechanical Engineering

WORK EXPERIENCE

Instructor	Baskent University	2006-Present
Research Assistant	ETU-TUBITAK	2005-2009
Teaching Assistant	Baskent University	2005-2006
Research Assistant	METU	2003-2005

AWARDS & SCHOLARSHIP

The Turkish Academy of Sciences - TUBA	BDHP Scholarship	2007
METU-Grad. Sch. of Nat. & App. Sc.	Course Performance Award	2005

PUBLICATIONS

1) Şamiloğlu A.T., Gazi V., Koku A.B., “Comparison of three orientation agreement strategies in self-propelled particle systems with turn angle restrictions in synchronous and asynchronous settings,” Asian Journal of Control, vol. 10, No. 2, pp. 212-232, Mar. 2008. (SCI)

2) Cilasun, S. M., **Şamiloğlu A. T.**, “Seleksiyon Mekanizmasının Endüstri Dinamikleri Üzerine Etkisi: Bir Simülasyon Çalışması”, Akdeniz İ.İ.B.F. Dergisi vol. 11, No. 21, pp. 1-16, May. 2011 (ASOS, IBSS)

3) Şamiloğlu A.T., Gazi V., Koku A.B., “Asynchronous Cyclic Pursuit,” S. Nolfi et al. (edt.), SAB06, Lecture Notes in Artificial Intelligence (LNAI) 4095, pp. 667-678, 2006.

4) Şamiloğlu A.T., Gazi V., Koku A.B., “Effects of Asynchronism and Neighborhood Size on Clustering in Self-Propelled Particle Systems,” ISCIS06, Lecture Notes in Computer Science (LNCS) 4263, pp. 665-676, 2006.

5) Hanay Y.S, Hünerli H.V., Köksal M.İ., **Şamiloğlu A.T.**, Gazi V, “Formation Control with Potential Functions and Newton’s Iteration,” European Control Conference, pp. 4584-4590, Kos, Greece, July 2007.

6) Şamiloğlu A.T., Çayırpunar Ö., Gazi V., Koku A.B., “An Experimental Set-up For Multi-Robot Applications,” Workshop Proceedings of SIMPAR 2008, Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, Venice(Italy) 2008 November, pp. 539-550.

7) Şamiloğlu A. T., Gazi V., Koku A.B., “Design of Circling Around a Target Controllers for Mobile Robots by Feedback Linearization”, IFAC International Workshop on Periodic Control Systems (PSYCO 2010), Vol.4, Part 1.