

SELF-ORGANIZED FLOCKING WITH A MOBILE ROBOT SWARM

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

BY

ALİ EMRE TURGUT

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

MARCH 2008

Approval of the thesis:

SELF-ORGANIZED FLOCKING WITH A MOBILE ROBOT SWARM

submitted by **ALİ EMRE TURGUT** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mechanical Engineering, Middle East Technical University** by,

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

Prof. Dr. Kemal İder
Head of Department, **Mechanical Engineering**

Assist. Prof. Dr. Buğra Koku
Supervisor, **Mechanical Engineering Dept., METU**

Assist. Prof. Dr. Erol Şahin
Co-supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Reşit Soylu
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Buğra Koku
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Göktürk Üçoluk
Computer Engineering Dept., METU

Assist. Prof. Dr. Melik Dölen
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Veysel Gazi
Electrical and Electronics Engineering Dept., ETU

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 : ALİ EMRE TURGUT

Signature :

ABSTRACT

SELF-ORGANIZED FLOCKING WITH A MOBILE ROBOT SWARM

TURGUT, ALI EMRE

Ph.D., Department of Mechanical Engineering

Supervisor: Assist. Prof. Dr. Buğra Koku

Co-Supervisor: Assist. Prof. Dr. Erol Şahin

March 2008, 111 pages

In this thesis, we study self-organized flocking using a swarm of mobile robots. We first present a mobile robot platform having two novel sensing systems developed specifically for swarm robotic studies. We describe its infrared-based short-range sensing system, capable of measuring the range to obstacles and detecting kin robots. In particular, we describe a novel sensing system called the virtual heading sensor (VHS), which combines a digital compass and a wireless communication module to form a scalable method for sensing the relative headings of neighboring robots.

We propose a behavior based on heading alignment and proximal control and show that it is capable of generating self-organized flocking in a group of seven robots. Then, we propose a number of metrics to evaluate the quality of flocking and use them to evaluate four main variants of this behavior. We characterize and model the sensing abilities of the robots and develop a physics-based simulator that is verified against the physical robots for flocking in open environments. After showing in simulation that we can achieve flocking in a group of up to 1000 robots in an open environment, we perform experiments to determine the performance of flocking under different controller parameters and characteristics of VHS using the predefined metrics. In the experiments, we vary the three main characteristics of VHS, namely: (1) The amount and nature of noise in heading measurement, (2) The number of neighboring robots that can be "heard", and (3) the range of wireless

communication. Our results show that range of communication is the main factor that determines the scale of flocking, and that the behavior is highly robust against the other two characteristics.

We extend an existing particle-based model to determine the phase transition characteristics of flocking under different VHS characteristics. An analytical treatment of the model is also presented and verified against the results obtained from experiments in a physics-based simulator.

Keywords: Swarm robotics, Flocking, Self-organization, Phase Transition, Particle-based Models

ÖZ

BİR GEZER ROBOT OĞULUNUN KENDİ-KENDİNE SÜRÜ HALİNDE HAREKET ETMESİ

TURGUT, ALİ EMRE

Doktora, Makina Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Buğra Koku

Ortak Tez Yöneticisi: Yrd. Doç. Dr. Erol Şahin

Mart 2008, 111 sayfa

Bu tezde, bir gezer robot oğulunun, sürü halinde hareket etmesi (*ing.* flocking) incelenecektir. Başlangıçta, oğul robot sistemleri çalışmalarında kullanılmak üzere geliştirilmiş ve iki yeni algılama sistemi bulunduran gezer robot sistemi anlatılacaktır. Bunlardan birincisi, robotların yakın mesafedeki robotlar ile engelleri ayırmalarını sağlayan ve de bu cisimlerin robota göre mesafelerini ölçebilen kızıl-berisi algılama sistemidir. İkincisi ise, robotların diğer robotların yönünü sanal olarak ölçmesini sağlayan ve bir sayısal pusula ile iletişim biriminden oluşan sanal yön algılayıcısıdır (SYA). SYA, birçok farklı oğul-robot sistemlerinde kullanılabilir bir algılama sistemi olarak ortaya çıkmaktadır.

Sürü halinde hareket davranışı, yön ayarlama ve mesafe koruma davranışları kullanılarak oluşturulmuştur. Bu davranış ilk başta yedi robot üzerinde sınanmıştır. Daha sonra davranışın içerisindeki denetim parametreleri kullanılarak dört farklı davranış biçimi elde edilmiş ve oluşturulan davranışlar tanımlanmış olan ölçütler kullanılarak değerlendirilmiştir. Davranışı yüksek sayıda robot kullanarak sınamak için robotun ve algılama sisteminin modelleri fizik-tabanlı bir benzetimci üzerinde kullanılmak üzere çıkartılmış ve bu benzetimci üzerinde engelsiz ortamda modellenmiştir. Benzetimcinin sonuçlarının robotlardan elde edilmiş olanlarla uyumu gösterildikten sonra benzetimci içinde 1000 robot

kullanılarak deneyler yapılmış ve önerilen davranışın ölçeklenebilir olduğu gösterilmiştir. Daha sonra denetim parameterlerin değerleri ve SYA'nın algılama özellikleri değişimine karşı sürü halinde hareket davranışındaki değişimler önerilmiş olan ölçütler kullanılarak araştırılmıştır. Bu deneylerden çıkan en önemli sonuçlardan bir tanesi akın davranışını gerçekleştiren oğulun grup büyüklüğünün SYA'nın iletişim mesafesine bağlı olmasıdır.

Son olarak da sürü halinde hareket davranışında gözlemlenen faz değişimi, parçacık bazlı bir model kullanılarak modellenmiş ve model başarı ile değişik SYA algılama özellikleri karşısında davranışın tepkisini tahmin edebilmiştir. Model, analitik olarak çözülmüş ve modelin ön gördüğü sonuçlar fiziksel benzetimcide elde edilen sonuçlarla karşılaştırılmıştır.

Anahtar Kelimeler: Oğul robot bilimi, Sürü halinde hareket, Kendi-kendine örgütlenme, Parçacık-bazlı modeller, Faz değişimi

To Özgül Turgut

ACKNOWLEDGMENTS

First of all, I would like to thank to my gratitude advisor, Erol Şahin, who opened a door into a dark room for me, let me wander around and advised me whenever I got lost and made *me* out of me. This is especially for him:

When the night has come,
And the land is dark.
And the moon is the only light we see.
Well, I won't be afraid.
No, I won't be afraid.
As long as you stand by me.

— Ben E. King

I am also grateful to my other supervisor, Buğra Koku, for being such a nice person and giving me the chance to work with him. Without him, it would have definitely been impossible.

My sincere thanks go to Cristián Huepe, the physicist and musician, for his great papers, advice and guidance he has given me from a distance. And for his work on the analytical solution of the model. Things would have been really boring without him.

Special thanks go to Beth Doğan whom I met in the last days of this study. She did a great job in proof reading the thesis. The thesis is undoubtedly better with her efforts.

I would also like to thank Emre Uğur for being such a nice, hardworking and, most importantly, caring person. He is the second "Erol Şahin" in my life, who has criticized me for giving up and not working hard enough. He always encouraged me to do it.

This thesis would be a dream without Hande Çelikkanat and Fatih Gökçe. They did their best, most of the time or I should say all of the time better than me. They are forever friends that I would like to be with throughout my life.

I would also like to thank Levent Bayındır and Kovan alumni, Erkin Bahçeci, Onur Soysal, Mehmet Remzi Dođar and Maya akmak for their support and for creating such a good atmosphere.

I am very thankful to my friend and my brother Ahmet Yozgatlıgil who believed in me and showed me the right way throughout my life.

Kutluk Bilge Arıkan and Serkan Grođlu: not much to say about them. They are my brothers and forever friends.

İbrahim Sarı, Mjde -yenge- Sarı, Burak Sarı and Perihan Sarı shortly the Sarı family, one of the most important families in my life through their friendship, advice, encouragement, hospitality and never-ending coffee breaks.

I am thankful to Segah zdemir for her friendship, support and golden heart.

Ozan akmakçı, my dearest friend from my childhood. Years have passed, but things have not changed. We are still kids and friends and will be for the rest of our lives.

Yaman akmakçı is brother of Ozan, great thanks to him also. He is a great man and I am sure that we will do lots of good things together.

Special thanks go to Gll Kızıldaş, for her support throughout my life. She is now far away, but our hearts are together and will stay like that forever.

The person that you should be scared of. The person whom I forgot to thank in my master's thesis and have been reminded about it many times. I should have written her name on the top of the list not to forget. Thanks to my dear friend Nilgn Fesiođlu nver, for being on my side and helping me to shape my life.

I would like to thank to Nazik Selvi, a girl with a golden heart in Switzerland. She is a very hard-working person and is a figure-of-success to be followed.

I would like to express my gratitude to Mehmet Durna whom we started working from the days of my master's thesis. He, with his brilliant ideas, always directed me to the right way.

My sincere thanks go to Korkmazıđit brothers, Levent and Oktay. They are like their surnames. They have the wisdom and power to do anything in their minds. I asked them to do the things in my mind, and they did. It was just a piece of cake for them. Thanks for that great robot, sorry the Kobot!

I am indebted to Hsamettin Kazancı for his wisdom and his guidance throughout my life. I really love the never-ending discussions with him.

I do not know if they are put out with each other for the time being, but anyhow, I thank them all. Thanks to Muharrem Dođan and Turan Kalender for their support, chat

and great teas for so many years.

I would also like to thank to my master's thesis advisor Prof. Dr. Abdülkadir Erden. He first introduced me the subject of mechatronics and gave me enough freedom to play with the things and supported me throughout my master's study.

When it comes to acknowledgments, I should not forget Orkun Mungan, a forever-friend from middle school years who has a great talent in literature and mathematics. Wish you happiness with your wife and Kuzey.

My gratitude thanks go to Özgür Baykal and Uğur Baygın who always supported me throughout this study and kept telling me that I could do it. Things would be much difficult without them.

It is very exciting and fun to find your 15-year-old classmates after so long years. I started my education with them and had the chance to meet them again in the last days of this study. I love you all class of 3-C and will never forget you.

I would also like to thank to the Commodore people for this wonderful machine, the best-selling personal computer of all time and Peter Fraser, an Englishman in Ankara, who has given me the chance to use a Commodore 64 at this early age, in simulating the Gaussian distribution. I still can't believe how we did that. Thank you, I will never forget you. You lit the fire, and I will always try to keep it burning.

Words are not enough for him. It is said about J. S. Bach that he is half of the music in the World. If you take him out, then you would lose the half. The same is true for Erkan Oğur. He is half, probably even much more than that, of me. With the help of his great pieces and wisdom, I had the chance to hear the music inside me and know myself much better. I would be deaf and mostly blind without him.

I would also thank to my brother, Hasan Turgut, for pushing me into the subject and willing me to do it deep in his heart. And of course the little Turgut's. Elif, Laçın and Şamil. I love you all.

It is hard to thank to your mother and father, as you owe everything to them, even your existence. They are always the ones in your life, in your heart, in your mind and in every breath you take. A great thanks to them for giving me the chance to be in the World and for all their support.

The work done in this thesis has been mainly supported by TUBITAK through the "KARIYER: Kontrol Edilebilir Robot Oğulları" project with number 104E066.

No matter how long you write, there is an end. I hope it will not be an end, but the beginning of a new era in my life. Anyhow, half of me is happy and the other is sad, just

what Chang felt when Tintin was about to return Moulinsart.

"There is a rainbow in my heart. I weep because Tintin is going, but the sun shines because I have a new mother and father."

— Chang (Tintin in Tibet)

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	vi
DEDICATON	viii
ACKNOWLEDGMENTS	ix
TABLE OF CONTENTS	xiii
LIST OF TABLES	xvi
LIST OF FIGURES	xvii
CHAPTER	
1 INTRODUCTION	1
1.1 Problem	3
1.2 Contributions	3
2 LITERATURE SURVEY ON FLOCKING	5
2.1 Statistical Physics	7
2.2 Control Theory	10
2.3 Biology	13
2.4 Robotics	14
3 REVIEW OF ROBOTIC SYSTEMS IN SWARM ROBOTICS RESEARCH	17
3.1 Requirements of Swarm Robotic Systems	17
3.2 Review of Existing Mobile Robots	19
4 THE DESIGN OF THE KOBOT ROBOTIC SYSTEM	24
4.1 A Brief History	25
4.1.1 Version 1	25
4.1.2 Version 2	25

4.1.3	Version 3	27
4.1.4	Version 4	27
4.2	The Short-range Sensing Sub-system	31
4.3	The Heading Sensing Sub-system	34
4.4	The Communication Sub-system	36
4.5	The Control Sub-system	40
4.6	The Locomotion Sub-system	44
4.7	The Power Sub-system	45
4.8	The Structural Sub-system	47
5	CHARACTERISTICS AND MODELING OF KOBOT	49
5.1	The Characteristics of the Short-range Sensing Sub-system	49
5.2	The Characteristics of the Virtual Heading Sensor	50
5.2.1	Number of VHS Neighbors	50
5.2.2	Noise	53
5.3	The Co-Swarm Simulator	55
6	THE FLOCKING BEHAVIOR	57
6.1	The Flocking Behavior	57
6.1.1	Heading Alignment	58
6.1.2	Proximal Control	59
6.1.3	Motion Control	60
6.2	Metrics of Flocking Behavior	62
6.3	Full-fledged Flocking in Constrained Space	64
6.4	Full-fledged Flocking in Open Space	66
6.5	Full-fledged Flocking with a Large Group in Open Space	67
7	ANALYSIS OF VARIANT BEHAVIORS AND OPTIMIZATION OF THE CON- TROLLER PARAMETERS	69
7.1	Analysis of Variant Behaviors	69
7.1.1	The Advance-Phase	70
7.1.2	The Alignment-Phase	72
7.1.3	The Avoidance-Phase	74
7.2	Optimization of the Controller Parameters	75
7.2.1	Weight of Proximal-Control Behavior	76

7.2.2	Maximum Forward Speed	78
7.2.3	Proportional Gain	79
8	ANALYSIS OF FLOCKING BEHAVIOR UNDER DIFFERENT VHS CHARACTERISTICS	83
8.1	Heading Noise	83
8.1.1	Magnitude of Noise	84
8.1.2	Nature of Noise	84
8.2	Number of VHS Neighbors	85
8.2.1	Large Group	85
8.2.2	Small Group	85
8.3	Range	87
9	MODELING PHASE TRANSITION IN FLOCKING	90
9.1	Phase Transition in Physical Systems	91
9.2	Vectorial Network Model	93
9.3	Stiff Vectorial Network Model	94
9.4	Analytical Treatment of S-VNM	96
9.5	Results of S-VNM	99
10	CONCLUSION	101
	REFERENCES	103
	VITA	109

LIST OF TABLES

TABLES

3.1	Comparison of mobile robots for swarm robotics research	22
4.1	Power consumption of sub-systems	46
4.2	Comparison of mobile robots for swarm robotics research with Kobot	48
6.1	The default settings for the controller and VHS parameters.	65

LIST OF FIGURES

FIGURES

2.1	Three basic behaviors of Reynolds' flocking model. (a) Separation behavior. (b) Alignment behavior. (c) Cohesion behavior. Arrows indicate the steering direction of the individual due to each behavior. Reproduced from [1]	6
2.2	The velocities and trajectories of particles in Vicsek model using 300 particles after a certain time. The velocities are indicated by an arrow and the trajectories by a continuous curve. (a) Initial time. (b) Low density and noise. (c) High density and noise. (d) High density and low noise. Reproduced from [2]	8
4.1	(a) Kobot prototype version 1.0. (b) Kobot prototype version 1.1. (c) Kobot prototype version 1.2 (d) Kobot prototype version 1.3. . . .	26
4.2	(a) Kobot prototype version 1.4. (b) Kobot prototype version 1.5. . . .	26
4.3	(a) Kobot prototype version 2.0. (b) Kobot prototype version 2.0. . . .	27
4.4	(a) Kobot prototype 3.0 front view. (b) Kobot prototype 3.0 exploded view.	28
4.5	Close-up view of the short-range sensing sub-system.	29
4.6	(a) Kobot version 4.0 with the virtual heading sensor. (b) Seven Kobots with the virtual heading sensor.	29
4.7	Block Diagram of Kobot.	30
4.8	(a) Schematic diagram of the sensor board. The arrow indicates the forward direction of Kobot. (b) A photograph of the sensor board. . .	32
4.9	Block diagram of an individual sensor.	33
4.10	Timing diagram of a sensor detecting an obstacle.	34

4.11	(a) Photo of the HMC-6352 digital compass module. (b) Schematic of HMC-6352 placed in a magnetic field. Reproduced from [3].	35
4.12	Illustration of the hard-iron effect on the VHS.	36
4.13	(a) XBee-PC interface card. (b) XBee placed on Kobot having an address of 10h.	39
4.14	Duration of programming. XBee is interfaced at 57600 baud. Program size is 5.6 kWord.	43
4.15	Main control card.	44
4.16	Voltage conversion module.	46
5.1	(a) Experimental setup used in the obstacle-proximity measurement and robot-proximity experiments. (b) Results of the obstacle-proximity experiment. (c) Rate of success of robot detection. (d) Results of the robot-proximity experiment.	51
5.2	The communication range experiment. (a) Topology of nodes. Circles indicate various communication ranges of the central node. (b) Average number of VHS neighbors of the central node. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median points, respectively.	52
5.3	(a) The group-size experiment. Average number of VHS neighbors for five different group sizes. (b) Neighbor-selection experiment. The ends of the boxes and the horizontal line in it correspond to second and third quartiles and the median points, respectively.	54
5.4	Vectorial noise model. The heading measurement vector ($e^{i\theta'_j}$) is denoted by a continuous arrow. The noise vector ($e^{i\xi_j}$) is shown by a dashed arrow and the resultant vector (\tilde{h}_j) is denoted by a bold continuous arrow. (a) $\eta = 1$. (b) $\eta = 1.5$. Adapted from [4].	55
5.5	Histogram of simulated noisy measurements of VHS for $\theta'_j = 0^\circ$. The simulations are performed for 10000 steps for three different η values.	55
5.6	(a) A photo of seven robots in CoSS. (b) A snapshot of seven robots.	56
6.1	The heading alignment. The central robot aligns its heading to the average of the three randomly selected VHS neighbors drawn in black. The other robots are drawn in gray. The communication range is set to R and \mathcal{N}_R is set to 3. (a) $t = 0$. (b) $t = 1$	58

6.2	The virtual force (f_k) is plotted as a function of $d_k \in \{0, 1, \dots, 7\}$, where $d_k = 1$ for a distant and $d_k = 7$ for a nearby object.	59
6.3	The reference frame is fixed to the center of the robot where the x -axis coincides with the rotation axis of the wheels. The forward velocity (u) is parallel to the y -axis. ω denotes the angular velocity of the robot. v_R and v_L denote the velocity of the right and left motors, respectively. The y -axis of the body-fixed reference frame makes an angle of θ (current heading) with the <i>sensed North</i> direction (n_s) at the instant the figure is drawn. l is the distance between the wheels.	60
6.4	Modulation of the forward velocity (u) when $\gamma = 1$. (a) The robot makes rotation and translation. $\alpha \approx 0^\circ \Rightarrow u \approx u_{max}$. (b) The robot makes mostly rotation. $\alpha \approx 90^\circ \Rightarrow u \approx 0$. (c) The robot only makes rotation. $\alpha > 0^\circ \Rightarrow u = 0$	61
6.5	Entropy values for four different configurations of seven robots. Entropy decreases as the configuration loses its positional order.	64
6.6	Self-organized flocking with seven Kobots. Starting from a connected but unaligned state, Kobots negotiate a common heading and move as a group in free space, bouncing off a wall without losing their cohesion.	66
6.7	Self-organized flocking in open space with seven Kobots and seven robots in CoSS. (a) Plot of order. (b) Plot of entropy. Five simulations are performed and the mean values are plotted together with error-bars indicating \pm one standard deviation of the value from the mean.	66
6.8	Full-fledged flocking with a large group. (a) The screenshot at the beginning of the simulation. (b) The screenshot when $t = 2000$ s. The total displacement of the flock is approximately 110 m.	68
7.1	<i>Advance</i> -phase experiments. (a) Plot of order. (b) Plot of entropy. (c) Plot of normalized average forward velocity. (d) Plot of normalized average angular velocity. (e) Snapshots of the initial and final configurations of Kobots. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by a plus sign.	73

7.2	The results of the <i>alignment</i> -phase experiment. (a) Time evolution of order. (b) Time evolution of entropy. (c) Snapshots of the initial and final configurations of Kobots.	74
7.3	The <i>avoidance</i> -phase experiment. (a) Plot of order. (b) Plot of entropy. (c) Snapshots of the initial and final configurations of Kobots.	76
7.4	Plot of order in β experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars are not shown for clarity.	77
7.5	Plot of rate of change of entropy in β experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars indicate ± 1 standard deviation from the mean. Error-bars in (b) are not shown for clarity. . . .	78
7.6	Plot of average forward velocity in β experiments (a) CoSS experiments. (b) Kobot experiments. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs. . .	79
7.7	Plot of order for u_{max} experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars are not shown for clarity.	79
7.8	Plot of rate of change of entropy for u_{max} experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars indicate ± 1 standard deviation from the mean. Error-bars in (b) are not shown for clarity. .	80
7.9	Plot of average forward velocity for u_{max} experiments. (a) CoSS experiments. (b) Kobot experiments. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs. 81	81
7.10	Plot of order for K_p experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars are not shown for clarity.	81
7.11	Plot of rate of change of entropy for K_p experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars indicate ± 1 standard deviation from the mean. Error-bars in (b) are not shown for clarity. .	82
7.12	Plot of average forward velocity for K_p experiments. (a) CoSS experiments. (b) Kobot experiments. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs. .	82

8.1	Heading noise experiments. (a) Plot of order for different levels of noise (η) when ξ is uniformly distributed in $[-\pi, \pi]$. (b) Plot of the steady-state value of order for Gaussian noise $N(0, \pi/2)$, $N(0, \pi)$ and uniformly distributed noise in $[-\pi, \pi]$	84
8.2	Number of VHS neighbors experiment with 100 robots in CoSS. (a) Plot of order. (b) Plot of the steady-state value of order. Uniformly distributed noise is assumed with $\eta = 3$ to accentuate the results.	86
8.3	Number of VHS neighbors experiment for 100 robots in CoSS. (a) Plot of rate of change of entropy. (b) Plot of average forward velocity. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. The outliers are denoted by '+' signs.	86
8.4	Number of VHS neighbors experiments with seven Kobots. (a) Plot of order. (b) Plot of rate of change of entropy. (c) Plot of average forward velocity. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. The outliers are indicated by '+' signs.	87
8.5	Number of VHS neighbors experiment with seven robots in CoSS. (a) Plot of order. (b) Plot of rate of change of entropy. (c) Plot of average forward velocity. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. The outliers are indicated by '+' signs.	87
8.6	Range experiments. (a) Plot of order. (b) Plot of rate of change of entropy.	88
8.7	Plot of the size of the largest cluster. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs.	89
9.1	Plot of magnetization as a function of scaled temperature for five different magnetic materials. Reproduced from [5].	91

9.2	Plot of order. Experiments are conducted in CoSS using 100 robots for various number of VHS neighbors. Each experiment is repeated 10 times. Steady-state value of order is calculated by averaging its value for the last 5000 time-steps. Error-bars indicate ∓ 1 standard deviation from the mean.	92
9.3	Neighbor selection in the vectorial network model. Three possible neighbor configurations for $p = 7/8$ when the number of neighbors (N) is set to be 8. Reproduced from [6].	94
9.4	Phase transition diagram predicted by the VNM with $p = 0$ (solid line) and $p = 0.99$ (dashed line). The simulations are carried out using 2×10^4 particles with 15 neighbors. η denotes the magnitude of noise, and γ is the order parameter of the group. Adapted from [6].	95
9.5	The effect of κ and λ coefficients on $\bar{\psi}$. Plot of $\bar{\psi}$ as a function of η . (a) $\kappa \in \{1, 5, 10, 20\}$ and $\lambda = 10$. (b) $\lambda \in \{1, 5, 10, 15\}$ and $\kappa = 1$. Error-bars indicate \pm standard deviation from the mean.	96
9.6	The phase transition diagram obtained using simulation of S-VNM. In simulations, κ is set to 1.5 and λ is set to 22.	99
9.7	Critical noise values for various number of neighbor values.	100

CHAPTER 1

INTRODUCTION

Swarm robotics [7][8] is a new approach to the coordination of large numbers of robots that takes its inspiration from the impressive coordination abilities of social insects such as bees, ants and termites [9]. It studies how a large number of robots can interact to create collectively intelligent systems without any centralized coordination and achieve *robustness, flexibility* and *scalability* at the system level.

Robustness is the insensitivity of a system to disturbances, a desirable property required in any engineering system. Robustness is a consequence of several attributes in swarm robotics. First is the redundancy of the system, i.e., the loss of individuals up to a certain level is compensated by the others, and system performance does not deteriorate considerably. Second is the decentralized control of the system. In swarm robotics, control is embodied in individuals, rather than having a centralized controller. Third is the simplicity of the individuals. Individuals in a swarm robotic system are much simpler than in a single robotic system, which makes the system more robust.

Flexibility is the ability of the system to generate different solutions to different tasks it faces during its life-cycle. It can be argued that *flexibility* is the opposite of specialization, meaning the utilization of an individual in various tasks rather than utilizing a specific individual in a specific task. As can be observed in ant colonies, depending on the situation, an individual can take responsibilities in a wide span of tasks such as foraging, prey retrieval or chain formation.

Scalability is the insensitivity of performance to changes in number of individuals. Scalability in swarm robotics is facilitated by several factors. First, coordination strategies are designed to be decentralized, so that an increase in the number of individuals in the system does not increase its complexity, and hence, deteriorate the performance of a centralized controller. Second, the sensing systems utilized in swarm robotics are decentralized, so

that a centralized sensor, such as an overhead camera, does not exist in the system. Hence, each individual has the same sensing capabilities, regardless of the size of the group.

Having these promising properties, it is desirable to utilize swarm robotics in real-world applications. In this respect, the research spans two different directions. First is the development of robots capable yet simple enough to be utilized in batches. Swarm robotic platforms should facilitate study rather than interfering with it, and allow researchers to concentrate on the problems of coordination. Second is the development of flexible and scalable coordination methods to operate on swarm robotic systems.

We will develop a new robotics platform designed specifically for swarm robotics research, having two novel sensing systems for kin-detection, having the longest operation time in its class, and specifically designed for our intended coordination problem, yet flexible enough to be utilized in other problems, too.

Many coordination problems are studied in swarm robotics: *Pattern formation, aggregation, self-assembly, coordinated movement, foraging* and *self-deployment*. *Pattern formation* is the formation of certain desired patterns globally using only local interactions. *Aggregation* is the gathering of randomly distributed robots to form the largest possible cluster via local rules. *Self-assembly* denotes the connection of individual robots physically to solve complex tasks which are impossible to accomplish as single entities. *Coordinated movement* denotes the movement of robots coherently in confined environments, where the flocking of birds and herds of quadrupeds are a few of the examples seen in nature. *Foraging* is a task performed by natural swarms in which they retrieve prey back to their nests. Robots performing *foraging* mimic this behavior. *Self-deployment*, also known as dispersion, denotes the maximum coverage of an area by a swarm of robots [10].

In this thesis, we study flocking as a coordinated movement problem. Flocking is a fundamental behavior that is desirable in most swarm robotic systems in real-world applications. When a swarm of mobile robots is to be utilized in a surveillance or rescue task in a certain place, it should first move to that place and then perform its tasks. As noted previously, the most efficient and advantageous way to move from one place to another in animal societies is flocking, which is almost the same for robotics applications. Flocking has been studied for almost a decade, and yet it remains as an "open" problem in swarm robotics. It is a challenging task due to the sensing needs and inevitable interference problems caused by the operation of many robots in close proximity.

1.1 Problem

In this thesis, we study the self-organized flocking of a swarm of mobile robots. By self-organized flocking, we mean that a group of mobile robots, initially placed within local proximity of each other but not necessarily aligned, being able to move in free space cohesively, avoiding any obstacles in the environment, as if it were a "super-organism".

1.2 Contributions

Towards this end, we have developed a new mobile robotic platform designed from scratch for swarm robotics applications, focusing mainly on the requirements of flocking. The robotic platform consists of two novel sensing systems. First is the infrared (IR) short-range sensing system, capable of kin-detection and proximity measurements at close distances with minimal interference from the environment and other robots. Second is the virtual heading sensor, which utilizes digital compass and wireless communication modules to "virtually" sense the headings of the robots.

We also proposed a behavior for self-organized flocking in a constrained space. Our approach is significant in several aspects. First, it is the first study in literature to achieve self-organized flocking without any static or elected leader. Second, we performed extensive experiments with robots and a physics-based simulator to systematically analyze the proposed flocking behavior against the controller parameters and the virtual heading sensor characteristics. The results of these experiments are presented and analyzed using the proposed metrics such as order and entropy. Third, a simple particle based-model is introduced to model the steady-state characteristics of flocking against virtual heading sensor characteristics. The model is also treated analytically, and its predictions are verified against the simulation results.

In the next chapter, we present a literature survey regarding flocking in different fields. Then, in chapter three, we review the existing robotic platforms against the requirements of swarm robotic systems, focusing on flocking behavior. In the fourth chapter, the design of the robotic platform is described and the details of each sub-system are explained thoroughly. In chapter five, we model the characteristics of the short-range and heading-sensing systems of the robotic platform to utilize them in a physics-based simulator developed to achieve the flocking behavior using a large group of simulated robots. In the sixth chapter, we develop a flocking behavior based on heading alignment and proximal

control. In chapters seven and eight, we analyze flocking behavior against changes in controller parameters and virtual heading sensor characteristics, respectively. In chapter nine, we propose a simple particle-based model, called the stiff vectorial network model, to model flocking behavior and predict conditions for stability against sensing noise. The thesis ends with conclusions.

CHAPTER 2

LITERATURE SURVEY ON FLOCKING

"They circle; now dense like a polished roof, now disseminated like the meshes of some vast all-heaven-sweeping net wheeling, rending, darting a madness in the sky."

— Edmund Selous

Flocking is a widely-observed phenomenon in nature in which a group of animals such as fish or birds [11] move and maneuver as if they were a single creature. Flocks are observed to be highly scalable, so that a group size of tens to several thousands are not uncommon in nature. Flocking is of vital importance for the survival of many species, besides its other advantages [12][13]. The survival rate increases due to two factors. First, the chance of being eaten while in a large group is less than the chance of being eaten while alone [14]. Second, being together increases the overall size of the group and threatens predators. A school of small fish is much larger and threatening than a humble predator, which may well persuade the predator from attacking the school. Another advantage of flocking is the emergence of "collective intelligence" due to the interaction of the individuals in the flock [15]. It can easily be argued that having many brains and eyes in charge of searching for food is superior to having a single brain and a couple of eyes [16]. Flocking enables energy efficiency in the movement of animals, birds being a good example. Birds utilize the streamlines formed by their frontal neighbors for an up thrust which decreases the energy consumption necessary to fly. This is especially beneficial in the case of long journeys such as migration [16][12].

The subject of flocking did not attract much attention from fields other than biology until Reynolds' seminal work [17]. Reynolds, inspired by flocking in natural systems, performed pioneering work and proposed for the first time that flocking in birds can be artificially generated utilizing a combination of simple behaviors based on local sensing rules.

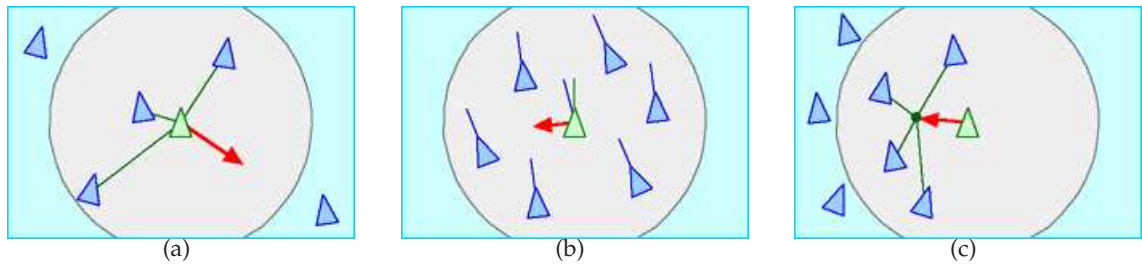


Figure 2.1: Three basic behaviors of Reynolds' flocking model. (a) Separation behavior. (b) Alignment behavior. (c) Cohesion behavior. Arrows indicate the steering direction of the individual due to each behavior. Reproduced from [1]

He then successfully used this principle and achieved a realistic computer animation of flocking.

In the study, a flock is formally defined as:

"... a group of objects that exhibits a general class of polarized, non-colliding, aggregate motion."

He suggested that a realistic-looking simulation of flocking could be made possible only through local interactions of the individuals rather than a centralized controller in charge of calculating the paths of each individual. He proposed three simple behaviors constituting the basic flocking model as:

- *Separation*: Individuals move so that they avoid collisions with their neighbors as shown in Figure 2.1(a).
- *Alignment*: Individuals match their velocities to the average of their neighbors. Velocity is regarded as a vectorial quantity composed of heading and speed components as shown in Figure 2.1(b).
- *Cohesion*: Individuals move towards the geometric center of their neighbors as shown in Figure 2.1(c).

The neighbors of an individual are local as in case of fish and are determined by a distance from the center of the individual and an angle in the direction of motion of the individual. Obstacle-avoidance and goal-seeking behaviors are also included in the model to enable realistic flocking behavior in the presence of obstacles.

Reynolds' algorithm is both scalable and, most importantly depends on only three simple rules of motion to achieve coherent flocking action, which makes it a favorable solution

to achieve flocking in any domain. However, there is a major leakage related to sensing where real-world applications are concerned. In order to apply Reynolds' algorithm to the real world, one should ensure that each individual has a knowledge of the velocity and position of its neighbors on an individual basis, free of any uncertainty or noise. This assumption is quite unrealistic, and is hardly possible with the current state of technology.

Reynolds' seminal work fostered studies in many different fields, statistical physics, control, and robotics being the major ones. In the rest of this chapter, we will review them.

2.1 Statistical Physics

In statistical physics, the emergence of collective behavior observed in a wide range of fields, from biological systems to fluid mechanics, is studied using simple particle-based models. Particle-based models consist of interacting particles under the effect of actuation or sensing noise. Tools of statistical physics are utilized to analyze the resulting dynamics.

Vicsek et al. [2] proposed a simple model, called self-driven particles (SDP), to study the emergence of the self-aligned motion of particles due to local interactions observed in biological systems [18]. The SDP model can be regarded as a subset of Reynolds' algorithm, and is able to model collective motion in different systems such as fish, birds, quadrupeds and some bacteria [19]. In the SDP model, particles move at a constant speed in a square region with periodic boundary conditions. The heading of each particle is updated to the average of its neighbors' at a local range with the inclusion of a noise term (scalar noise model). It is observed that, the particles having random orientations initially as shown in Figure 2.2(a), tend to move form small clusters moving in different random directions at low density and noise values depicted in Figure 2.2(b). The particles move in random directions with some correlation at high density and noise values as shown in Figure 2.2(c). The particles move in a common random direction at high density and low noise values as illustrated in Figure 2.2(d).

Systematic Monte-Carlo simulations of the model were conducted with periodic boundary conditions, varying the noise and the density of particles. The results of these simulations revealed that particles undergo a phase transition¹ from an unaligned state to an aligned state above a certain density value or below a certain noise value which are called the critical noise density and critical noise, respectively [19]. The phase transition is observed to be of second-order.

¹http://en.wikipedia.org/wiki/Phase_transition. Last visited: April 2008.

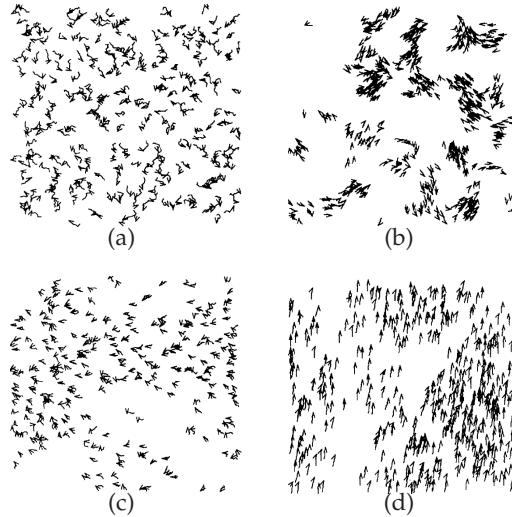


Figure 2.2: The velocities and trajectories of particles in Vicsek model using 300 particles after a certain time. The velocities are indicated by an arrow and the trajectories by a continuous curve. (a) Initial time. (b) Low density and noise. (c) High density and noise. (d) High density and low noise. Reproduced from [2]

Systems possessing second-order phase transition characteristics have a continuous phase diagram in which the system transforms continuously from one state to another. Phase transition is said to occur at a special point called the critical point. In the analysis of these systems, an order parameter is defined, representing the state of the system against an independent variable. In the case of the SDP model, the average alignment of the particles is defined as the order and the independent variable is taken as the actuation noise or density of the particles.

The conclusions of the SDP model contradict the Mermin-Wagner theorem [20], which states that ordered phase (aligned state in the SDP model) cannot be observed in one or two-dimensional systems having local interactions at non-zero temperatures (noise in the SDP model) unless long-range interactions exist [20]. In order to justify the predictions of the SDP model analytically, Czirok et al. [21] and [22] proposed a continuum model. In their model, particles are allowed to move in 1-D, having their speed (speed is variable, unlike in the SDP model) and headings set to the average of their neighbors' within close proximity, at each time-step. The predictions of the continuum model are in accordance with the SDP model, indicating that a second-order phase transition from disordered to ordered state exists in the presence of noise. This result clarifies the ambiguity between the Mermin-Wagner theorem and the predictions of the SDP model. In the SDP model, although the particles interact with their neighbors in close proximity, long-range inter-

actions still exist due to diffusion and relative displacement of the particles among each other [23]. It is this effect that causes aligned motion in the SDP model.

Huepe et al. [24] studied the intermittency characteristic of the SDP model which is defined as the unintended random motion of particles observed in non-equilibrium systems. It is observed that, although the particles have aligned motion, they frequently become unaligned and move randomly in the case of high noise and low density conditions. The intermittency condition continues until the noise level is decreased or the density is increased.

Gregoire et al. [25] extended the SDP model by adding an attraction/repulsion term based on the local bearing and range measurement of neighboring particles, which enables cohesive motion in open-space. The heading of each particle is updated to the weighted sum of the heading-adjustment and attraction/repulsion terms, having the noise added vectorially (vectorial noise model) to each heading measurement. Simulations are performed by changing the relative weights of the two terms, and unaligned-to-aligned transition is observed to be of first order. Coherently moving clusters in open space are achieved by the proper selection of the two coefficients.

Huepe et al. [26] in a recent study investigated the underlying dynamics of the SDP and Gregoire-Chate models [25]. They found that the SDP model creates high and unrealistic local density values when compared to the Gregoire-Chate model during the unaligned-to-aligned state phase transition. The unrealistic local density values make it hard for the SDP model to be used in modeling natural or robotic swarms, since local density values are quite low in both cases [12].

Gregoire et al. [27], in another study, claimed that the transition of unaligned to aligned motion of particles is of first order, even for the SDP model, regardless of the model of noise utilized, the second-order phase transition being a numerical consequence of the finite system-size in the simulations. In contrast, Nagry et al. [23] in a recent study showed by simulation that in the small-velocity regime where displacement of particles is much smaller than the measurement range -the regime where the SDP model is intended to be used- the transition is still of second order with the SDP model [2]. In another study, Aldana et al. [28] reached a similar conclusion utilizing a simple particle-based model. They showed that the type of transition depends only on the type of noise model utilized. When the scalar-noise model is utilized, as in the case of the SDP model, the transition is of second order, and when the vectorial-noise model is utilized, as in the Gregoire-Chate model, the transition is of first order.

Aldana et al. [6] proposed a simple model called the vectorial network model (VNM). Unlike the SDP model, in the VNM, the particles are kept stationary while their headings are updated to the average of their neighbors with an addition of a scalar noise term, as in the SDP model. However, unlike the SDP model, the neighbors of a particle are not only picked up randomly from the local neighborhood but also randomly from the entire group. It is observed that the system undergoes a second-order phase transition from an unaligned to an aligned state when there is at least one random neighbor in the neighboring set of particles and the noise is also below the critical value. In the case of totally local neighbors, the system stays in an unaligned state unless noise is set to zero. The reason behind this is the lack of long-range interactions due to having only local neighbors, which is in accordance with the predictions of the Mermin-Wagner theorem [20]. The predictions of the VNM also help us to understand the existence of phase transition observed in the SDP model. The VNM in the all-random neighbor case can be treated analytically to predict the phase transition characteristics of the system and the critical noise value.

The common point of the particle-based models is the utilization of either actuation or sensing noise in calculating the heading of particles. One of the major drawbacks of these models is the lack of analytical treatment due to the non-equilibrium characteristics of the models, the 1-D SDP model [22] and the VNM [6] being two exceptions that can be treated analytically to predict the conditions of phase transition. Despite a few implementations in biological systems [18][12], the particle-based models are yet to be implemented in other fields, such as robotics.

2.2 Control Theory

In control theory, flocking algorithms are proposed based on local heading, bearing, and range information on an agent basis. The stability of these algorithms is ensured analytically. Actuation and sensing noise are disregarded in all studies with the exception of [29].

Tanner et al. [30][31] proposed a stable control law for flocking in free space based on range, bearing and velocity information of neighbors of a robot in close proximity. Two cases are considered in their study. One is the fixed-topology [30] case, where neighbors are assumed to be fixed regardless of the sensing range, resulting in smooth control laws; and the other is the dynamic-topology [31] case, where neighbors are subject to change with sensing range, resulting in non-smooth control laws. In the dynamic topology case,

it is assumed that the change of neighbors is such that a neighboring graph² of the system always remains connected. The proposed control law, in both cases, includes an attraction/repulsion term depending on local distance measurement, and an alignment term depending on local velocity measurement. The attraction/repulsion force is implemented using a continuous version of the potential function in [25]. Stability is proved for the fixed-topology [30] and dynamic-topology [31] cases using Graph Theory and Lyapunov's stability theorem for non-smooth systems, respectively.

The fixed-topology case is successfully applied to the control of two complex robots and a virtual leader [32]. The robots have high-resolution encoders for displacement and velocity measurements, and an on-board computer with a high-speed communication facility which is used to broadcast its position and velocity to the other robot. In this way, each robot has the exact absolute position and velocity information of the other robot and the virtual leader. Although this study seems to be an implementation of flocking in real robots, the robots are far from practical, and are too complex to be utilized in batches.

Jadbabaie et al. [33][34] investigated the stability conditions of the aligned motion of particles in the SDP model [2], neglecting the effect of noise on the heading calculation. They showed that stability is ensured when the neighboring graph remains connected within a finite time interval, with time divided into infinitely many irregular intervals. A more relaxed condition is also proposed, indicating stable motion even if none of the neighboring graphs is connected but the union remains connected within a finite time interval.

Olfati-Saber [35] considered the case of flocking without a leader in free space and environments with obstacles based on local-proximity and velocity measurements. The algorithm consists of a gradient-based attraction/repulsion term and a velocity-matching term. It was shown that the algorithm boils down to Reynolds' flocking algorithm [17] and is not stable for agent counts larger than ten. The addition of a group objective term is shown to be essential for stable flocking behavior in obstacle-free environments. Environments with obstacles are handled by introducing virtual agents to the algorithm, which are assumed to move in the periphery of obstacles.

Cezayirli et al. [36] proposed an algorithm using a fixed leader to enable flocking in free space. Each agent is assumed to have range and bearing information about its neigh-

²A neighboring graph is a simple and undirected graph whose vertices denote the agents. An edge exists between two vertices if and only if the related agents are neighbors. The neighboring graph is said to be connected if one can start from a node and reach any other node by passing through the edges of the graph.

bors and moves to keep its proximity within a predefined range, while maintaining the connectivity of the neighboring graph. The fixed leader has *a priori* knowledge about the coordinates of the target points, and simulations have shown that initially connected agents preserve their connectivity and move to the target points coherently.

Lindhe et al. [37] proposed a flocking algorithm based on Voronoi partitions. In their algorithm, the obstacle-free Voronoi partition³ of each agent is calculated using the local range and heading information of each agent at each time-step. The agents are then moved toward the centroid of this region having guaranteed that the distance to the goal point is decreasing, keeping the formation in a desired condition, if feasible. If these conditions are not realizable, agents do not move at the current time-step. This algorithm ensures stable and collision-free flocking in environments with complex obstacles.

Hanada et al. [38] introduced an algorithm for flocking in environments with obstacles. In this algorithm, the agents are assumed to measure the range and bearing information of their neighbors and the obstacles. It is also assumed that the goal direction is known *a priori* by all of the agents. Based on this information, each agent selects two neighbors and moves in such a way that an isosceles triangle is formed among the neighbors. Meanwhile, agents keep their heading toward the goal direction and avoid obstacles. With the proposed algorithm, simulations have shown that the agents are able to split and rejoin in the presence of obstacles, and that they form equilateral triangles in the long-run as well.

Moshtagh et al. [29] proposed an algorithm to align the headings of the agents without explicit measurement of heading information. This algorithm relies on computer vision techniques in which agents are assumed to measure relative bearing, optical flow of neighboring agents and time-to-collision between the neighboring agents to deduce heading information. Simulations revealed that the algorithm works successfully when the neighboring graph is connected, even in the case of noisy measurements.

Gervasi et al. [39] proposed an algorithm which enables a group of agents to flock and to keep their desired formation. Agents are assumed to measure the range and bearing of the others and distinguish the leader, which is driven externally independent of the rest of the flock. Agents, at each time-step, compare the current formation of the group, using the leader as the common reference, with the desired formation and move to a target location to transform the current formation into the desired one. Some variants of this algorithm

³The Voronoi partition of a point (p) in a set (S) of N points is a planar region in which any point in the region has the smallest distance to the point p rather than to any other point in set S . http://en.wikipedia.org/wiki/Voronoi_partition. Last visited: April 2008.

are also proposed to increase performance and stability.

In all of the proposed flocking algorithms, range and bearing information are utilized on an agent basis to enable stable flocking in free space [30][31][34][35][36][39][29][37] and in environments with obstacles [37][38]. In all of the studies except [29], measurement noise is neglected and under this assumption, the stability of the algorithms is ensured analytically, except [39]. The major flaw in control-centric approach is the difficulty in realizing the algorithms in the real world since it is almost impossible to satisfy the sensing requirements with the current state of technology.

2.3 Biology

Flocking and coordinated movement are widely-observed and -studied phenomena in biological systems. Studies in this perspective span different tracks. In one track, the motivations of species to perform flocking are analyzed; in the other track, micro-models, i.e., behavioral models of individuals, are proposed to determine how flocking is achieved by individuals. The final track consists of experimental and theoretical studies related to the factors constituting flocking.

The determining factors in the flocking behavior of desert locusts are studied in [18]. It is proposed that there is a critical density above which locusts form clusters and move in ordered fashion. Preventing locusts from reaching this critical density value will prevent their devastating effect on agricultural goods. The critical noise is determined by experiments using locusts. The process is also modeled using the 1-D SDP model, and the predictions of the model are in agreement with the experiments [21][22].

In a recent study, Ballerini et al. [12] studied the neighboring characteristics of flocking birds, and showed that birds have a fixed number of neighbors which are determined by topological means rather than the metric distance between the birds. This makes the flock robust to segmentation and predator attacks. The findings were also verified using the SDP model [2], which revealed that individuals utilizing topological distance can easily separate and rejoin under a predator attack, which is not observed in the case of metric distance.

The motivations and factors constituting the marching behavior of cannibal crickets are analyzed in [13]. It is indicated that cannibalism is the main motivation in this behavior. Crickets try to move towards other crickets to eat them, while running away from others in order not to be eaten, which together results in an aligned motion of crickets.

A micro-model of the behavior of an army ant (*Eciton burchelli*) is developed in [40], and includes obstacle-avoidance and pheromone-tracking terms. The results of the model revealed that ants tend to move in a common direction to avoid collisions with other ants while moving back and forth in the environment.

In biological studies, it is seen that flocking increases the probability of survival of a species against environmental factors and predator attacks, though in some cases, flocking might not be so attractive on an individual basis, such as where cannibalism exists [13]. Flocking in biological systems is also robust to noise [18] and changing sensing conditions, as seen in birds [12].

2.4 Robotics

In most of the real-world tasks to be accomplished in distant places, robots should first move to this place and then perform the required task. In the case of swarm robotic systems, a group of robots should move together as a group, which can be accomplished by flocking. Therefore, flocking is of crucial importance and has been studied for more than a decade in swarm robotics, but it still remains as an "open" problem.

In one of the earliest attempts towards obtaining flocking in a group of robots, Mataric [41] combined a set of "basis behaviors": namely, safe-wandering, aggregation, dispersion and homing. In this study, the robots were able to sense the obstacles in the environment, localize themselves with respect to a set of stationary beacons and broadcast their position information to the other robots in the group. The flocking behavior developed can be seen as collective homing, where a homing direction is known and the robots try to stay within the sensing range of each other while moving. Through the use of safe-wandering behavior, the robots were also able to avoid obstacles in their path towards the home direction.

In [42], Kelly and Keating used a group of 10 robots which were able to sense the obstacles around them through ultrasound sensors, and the relative range and bearing of neighboring robots through the use of a custom-made active infrared (IR) system. The robots used an off-the-shelf radio frequency (RF) system to elect one of them as leader when none of them declares itself as leader. The leader would then wander in the environment, and the others would follow. The IR system was used to generate attractive forces towards other robots, whereas the ultrasound sensors acted as a repulsive force against other robots and obstacles.

Hayes et al. [43][44] proposed a "leaderless distributed flocking algorithm that is more

conducive to implementation on embodied agents" than the ones being used in computer animation. The flocking behavior consisted of two simpler behaviors; namely collision avoidance and velocity matching flock centering. It was assumed that the robots were able to sense the range and bearing of their neighbors within a predefined sensing range. Using this information, each robot would compute the center-of-mass (CoM) based on the relative placement of its neighbors and the heading towards a pre-defined goal area. The CoM was then used to implement flock cohesion, whereas the change in CoM between consequent sensory cycles was used to align the robot within the group. The authors implemented the proposed algorithm on the Webots simulator and optimized the parameters of the algorithm, which were then verified on a 10-robot group. In the experiments with the physical robots, however, the authors had to "emulate the range and bearing sensor signals" by tracking the robots using an overhead camera system and broadcasting these readings to the robots.

Spears et al. [45] proposed a framework called artificial physics to enable lattice formation in robots using attraction/repulsion and viscous forces based on local range and bearing measurements. Two infrared sensing modules were utilized for range and bearing measurement that were rotated by a motor to have a 360° field of view. Although lattice formation was of main concern in this study, flocking was also considered as a potential application. The experiments were performed using seven robots. In the experiments, robots first formed a regular hexagonal lattice, then moved together towards a light source.

Holland et al. [46] proposed a flocking algorithm for unmanned ground vehicles (UAV) similar to Reynolds' algorithm, based on separation, cohesion and alignment behaviors. Each UAV had a wireless communication module and an on-board camera system facing the ground. A UAV captured an image of the ground and transmitted it to a host, which extracted and transmitted the range, bearing and velocity information of the UAV's neighbors back to the UAV. This "emulated" sensing information was used to perform flocking.

Vaughan et al. [47] focused on controlling the behavior of a flock of ducks using a robotic sheepdog. The requirement was to move the ducks to a predefined goal point with the robotic sheepdog. A simple control algorithm was implemented on the robotic sheepdog based on the range and bearing information of the flock. In the algorithm, the robotic sheepdog was attracted by the flock and repelled by the goal point. The range and bearing measurement of the flock were "emulated" using an overhead camera which tracked the position of the ducks and broadcasted it to the robotic sheepdog. Simulations and real-life experiments were performed with 12 ducks and a robotic sheepdog. In the simulations,

each duck was assumed to be attracted by the distant ducks and repelled by the nearby ones. The ducks were also assumed to be repelled by the robotic sheepdog and the setup walls.

The few experimental studies in robotics reviewed above used either a virtual or explicit leader [42] to lead a group of individuals, or assumed that a goal heading (or area) was sensed by the whole group [41][43]. Moreover, in some of the studies [43][32], the authors had to resort to using "emulated" sensors. Studies that propose to use vision to control flocking [29], although promising, still remain to be implemented and evaluated on physical robots. Hence, swarm robotic systems that can maneuver in an environment as a super-organism and avoid obstacles in their path as a flock do not exist yet.

The main reason behind this non-existence, as partially discussed above, is that the flocking behaviors proposed and studied in other domains, such as computer graphics, statistical physics and control theory, assume that individuals can sense the range to the center of their neighbors and that there is one range reading per neighbor. Such sensing abilities still do not exist on most available robot platforms, with the exception of Kelly and Keatings' [42] custom active IR system. The proximity sensors on most mobile robots (such as ultrasound and IR-based systems) can sense only the range to the closest point of a neighboring robot, and multiple-range readings can be returned from a close neighboring robot. Furthermore, the sensing of bearing, velocity and orientation of neighboring robots is still difficult with the off-the-shelf sensors available on robots. Hence, there exists a major gap between the studies that propose flocking behaviors and robotics.

CHAPTER 3

REVIEW OF ROBOTIC SYSTEMS IN SWARM ROBOTICS RESEARCH

The requirements of a mobile robot used as part of a swarm robotic system differ from these of a mobile robot used stand-alone. Creating a swarm robotic system takes more than gathering a number of copies of any robot platform and making them work together, since the operation of many robots in coordination and in close proximity poses additional constraints.

In this chapter, we will first discuss the additional requirements needed by robots used in swarm robotic systems. Then, we will introduce the existing robots in the literature and discuss their capabilities in the scope of these requirements, focusing on their application to flocking behavior, which is of primary concern in the thesis.

3.1 Requirements of Swarm Robotic Systems

In this section, we investigate in detail the extra requirements expected from the use of robots in swarm robotic systems when compared to their usage as a stand-alone system. These requirements can be categorized as:

- **Sensing and Signaling:** The main emphasis in swarm robotics is the interaction among the robots as well as the interaction of the robots with their environment, putting extra constraints on the robots to be used.
 - **Interference among robots:** The interference among the sensing systems of the robots should be minimal. In most stand-alone robotic systems, proximity sensors are not designed to handle interference that may result from other robots operating in the same environment.

- **Interference from environmental factors:** The interference of environmental factors with the sensing system of the robots should be minimal. In existing stand-alone robotic systems, the effect of environmental factors, such as, daylight on the infrared proximity sensors, is accepted as part of the world, and is not dealt with. In swarm robotics research, however, the major emphasis lies on the use of self-organizing coordination methods, and environmental non-uniformities may bias the experiments unacceptably.
- **Kin-detection:** The robots should be able to distinguish other kin-robots. In most stand-alone robotic systems, such an ability is usually regarded as high-level and is usually handled through visual processing. In swarm robotics systems, however, such a sensing ability is vital to study the coordination mechanisms involved in tasks such as flocking or pattern formation. Therefore, it is preferred to make kin-detection as easy as proximity sensing.
- **Stigmergic sensing and signaling:** The robots should be able to leave "marks" in the environment and be able to sense them. This is rather a difficult and challenging task, even for stand-alone robotic systems. Although these abilities can be considered as an extra for a stand-alone robotic system, they are actually essential in swarm robotic systems, since stigmergic communication is known to be heavily used by social insects for coordinating their behavior.
- **Generic sensing:** The robots should also provide some form of generic sensing to allow the researcher to test novel sensing strategies which may not be included in the existing sensing abilities implemented by the fixed hardware design.
- **Communication:** Unlike stand-alone robotic systems, communication by plugging cables into the robots is no longer feasible when working with a swarm robotic system, and therefore, the robots must support some form of wireless communication.
 - **Wireless communication between the robots and a console:** The robots should support wireless communication with a console to allow easier monitoring and the debugging of algorithms on individual robots.
 - **Wireless communication among robots:** The robots should have inter-robot wireless communication. Such an ability would allow the robots to be used in the formation of *ad-hoc* networks.

- **Wireless programming:** The robots should support wireless downloading of control algorithms from a console.
- **Parallel programming:** The robots should be programmable in parallel. In swarm robotics research, the robots usually share the same control algorithms, and programming the swarm as a whole in one shot would be a big time-saver.
- **Physical interaction:** The robots should be able to physically interact with each other and the environment. Self-assembly or self-organized building of objects into larger structures remain interesting research topics in swarm robotics.
- **Power:** The robots should have a long battery-life. In swarm robotics research, the swarm may need to remain mobile for a period that is long enough for the collective behavior to emerge and the goal to be reached. Low battery-life would imply a situation where many robots run on batteries that need to be recharged and replaced frequently, creating inconvenience during the experiments.
- **Size:** Size does matter in swarm robotic systems. The robots should be small enough not to increase the size of test arena when experimenting with the system, yet big enough not to limit the expandability of the robot or increase the cost of the swarm robots due to miniaturization of components.
- **Cost:** The robots should be low-cost, since unlike stand-alone robots, they will be used in groups of at least ten.
- **Simulation:** Swarm robotic systems require realistic simulators which are essential to speed up the development of new control algorithms. Such simulators need to model the interactions between the robots as well as the interactions of the robots with their environment in a realistic way that can also be verified in the physical robots.

3.2 Review of Existing Mobile Robots

In this section, we review the existing mobile robot platforms that have been developed for or are suitable for conducting swarm robotics research, and evaluate them according to the requirements listed in the previous section, focusing mainly on the requirements of flocking behavior.

e-puck [48] has a circular shape with a diameter of 70 mm and is made of plastic. Two stepper motors are used for locomotion and there is a speaker for audible feedback. An

accelerometer, eight IR sensors (for obstacle and ambient light detection), a camera, three omni-directional microphones and a Bluetooth module are utilized in the robots, where vision, color LED communication and ZigBee communication modules can be added on demand. The robots can be programmed via the Bluetooth module. A three-hour of autonomy is reported using a 5 Wh Li-Ion battery.

Alice [49] has a rectangular prism body, having dimensions of 21x21x21 mm and is made of plastic and PCB. Two high-efficiency watch motors are used for locomotion. Alice has many optional sensory modules, such as four IR proximity sensors for obstacle detection, a short-range robot-to-robot communication module, an IR receiving module, a linear camera module, a wireless communication module and an ANT extension module. In addition, there is an optional gripper module and various locomotion modules to be utilized on different terrains. Ten hours of autonomy is reported with two button batteries and twenty hours of autonomy is achieved with an additional Li-Poly battery.

Jasmine [50] is another micro-robot platform which has a rectangular prism body, having a 26x26x26 mm measurement and being made of aluminum and PCB. Two small gear-head motors are used for locomotion. Jasmine has six IR sensors (LED+receiver) for proximity measurement and communication over short ranges. There is one powerful IR LED for detailed analysis of an object of interest, and a modulated IR module for communication with the host. Jasmine III has a modular design in which different sensing modules, such as an ambient light sensor, a color sensor and different locomotion modules, can be utilized. Two hours of autonomy is reported with Li-Poly batteries.

s-bot [51] has a circular shape, having a diameter of 116 mm and a height of 100 mm. s-bots have a patented locomotion sub-system consisting of both wheels and tracks, which are driven by two DC gear-head motors. s-bots have many sensors of different modalities. These sensors are fifteen IR proximity sensors for obstacle detection, a torque sensor on the wheels, a force sensor between the base and the wheels, a 3-axis accelerometer, an omni-directional camera and eight RGB LEDs for messaging between each s-bot. There are two grippers for holding/lifting other s-bots and objects. There is a Wi-Fi module for wireless communication. Li-Ion batteries are utilized in the robots, but the operation time is not mentioned in the study.

Swarmbot [52] is a square-shaped robot having dimensions of 130 mm. It has four wheels on each side driven by two DC gear-head motors. There are bump sensors, light sensors, drive wheel encoders and a patented IR robot-to-robot communication system in a Swarmbot. Additional modules are a linear CCD, magnetic food and swarm-cam emitters

which can be utilized on-demand. There is an RF communication unit for debugging and programming purposes. Battery-life is not reported but automatic charging stations are utilized to enable recharging by the robots themselves.

Pherobot [53] has a circular shape with a diameter of 110 mm. There are two DC motors in the locomotion sub-system. There are eight modulated IR transmitters and receivers in each robot for robot-to-robot communication. An interesting user interface is utilized in the robots, in which users wear virtual-reality goggles to interact with the robots visually. No information is available for the operation time of the robots.

Khepera II [54] has a circular shape with a diameter of 70 mm. It has two DC brushed servo motors with incremental encoders and eight infra-red proximity and ambient light sensors. There are extension modules such as different modules (b/w or color, directly used or combined with an omni-directional mirror), a wireless video module, a linear vision module, a matrix vision module, a radio communication module, a gripper and a general I/O module which can be utilized on demand. One hour of autonomy is reported with Ni-MH batteries.

Flockbots [55] are circular-shaped robots with a diameter of 180 mm. Two modified RC servomotors are used for locomotion. There are five IR range finders, one bump sensor and a CMUCAM-2 vision module. A gripper is also designed to grip objects. Additional wireless communication modules are planned to be added to the robots. A Bluetooth module is used for debugging and downloading programs. Two hours of autonomy is reported with NiMH batteries.

Table 3.1: Comparison of mobile robots for swarm robotics research

	e-puck	Alice	Jasmine	s-bot	Swarmbot	Pherobot	Khepera II	Flockbot
Interference robots	N/A	YES	YES	N/A	NO	NO	YES	YES
Interference environment	NO	NO	NO	YES	NO	NO	NO	NO
Kin detection	YES	YES	YES	YES	YES	YES	YES	NO
Stigmergic sensing	NO	NO	NO	NO	NO	NO	NO	NO
Generic sensing	vision	vision	NO	vision	vision	NO	vision	vision
Wireless communication	Bluetooth 802.15.4 ZigBee	RF mo- dem	IR	Wi-Fi	RF mo- dem	NO	RF mo- dem	Bluetooth
Wireless prog.	YES	NO	NO	YES	YES	NO	YES	YES
Parallel prog.	NO	NO	NO	NO	NO	NO	NO	NO
Wireless robot-robot	YES	YES	YES	YES	YES	YES	YES	YES
Physical Int.	NO	gripper	gripper	gripper	NO	NO	gripper	gripper
Battery-Life	medium	high	medium	N/A	N/A	N/A	short	medium
Size (cm)	dia.= ϕ 7	2.1 \times 2.1 \times 2.1	2.6 \times 2.6 \times 2.6	dia.= ϕ 12 ht.=15	13 \times 13 \times 13	dia.= ϕ 11	dia.= ϕ 7 ht.=3	dia.= ϕ 18
Cost	low	low	low	N/A	N/A	N/A	high	medium
Simulator	Webots	Webots	Breeve	Swarmbot 3D	N/A	N/A	Webots	MASON and Breeve
Computation	30F6014A	16F877	ATMEGA 18	XScale	40 MHz ARM	PALM V	25MHz Motorola 68331	PXA255

Satisfying all of these constraints in a single design is a difficult, if not impossible, challenge. Hence, we will mainly focus on the requirements related to flocking behavior in evaluating the existing robotic platforms. In flocking, robots should perform two basic behaviors. One is to assume a common heading, and the other is to avoid obstacles and maintain a desired distance from other robots in close proximity. The former requires sensing of heading, and the latter requires proximity measurement of nearby kin-robots and obstacles. The measurements are to be performed in a short time and interference in the measurements should be minimized.

None of the existing robotic platforms satisfy these requirements. The inconvenience arises in the following issues: (1) Kin-robot detection is only available in a few robotic platforms utilizing imaging systems. (2) Sensing systems are open to interference due to environment and kin-robots, imaging systems being an exception. (3) Heading measurement is not possible in the existing platforms. (4) Operation time of the robots is not long enough. (5) Cost of the robots is too high to utilize them in batches. Therefore, we designed a novel robotic platform from scratch to satisfy the aforementioned needs related to flocking and be able to perform self-organized flocking without the need of "emulated" sensors or any leaders. Yet, it had to be simple and cost-effective so as to be utilized in batches.

CHAPTER 4

THE DESIGN OF THE KOBOT ROBOTIC SYSTEM

"You start with wide nets. You gather a bunch of ideas. And you finally settle on the elements that are most promising."

— David Hill

In this chapter, we present a novel mobile robot, called Kobot, that was designed from scratch to meet most of the requirements of swarm robotic studies focusing mainly on flocking behavior. Kobot was designed to be light, modular and power-efficient, yet relatively cheap in order to be utilized in batches conveniently.

The most distinguishing properties of Kobots are their ability to measure range and bearing of kin-robots in close proximity through the short-range sensing system and heading measurement through the virtual heading sensor. These properties are the two distinguishing properties that enable Kobots to be utilized in flocking studies.

Kobot is composed of several sub-systems that can be listed as: sensory, control, locomotion, structural and power sub-systems. The sensory sub-system can further be divided into communication, heading sensing and short-range sensing sub-systems. The short-range sensing sub-system senses the distance from kin-robots and obstacles in close proximity and distinguishes robots from obstacles. The heading sensing sub-system virtually measures the headings of neighboring robots in the group through digital compass and XBee modules. Information from the sensors is fed to the control subsystem, which controls the behavior of the robot through the control algorithm. The communication sub-system is utilized to program Kobots wirelessly and to enable an *ad-hoc* communication channel between Kobots. The power sub-system supplies the necessary operating voltage

to all other sub-systems of the Kobot. The structural sub-system is the body of the Kobot. It holds and protects all the other sub-systems while maintaining the aesthetics as well.

In this chapter, we will first present the evolution of these subsystems and then go into the details of the design of each subsystem.

4.1 A Brief History

Prior to reaching the final design of Kobot, many iterations were performed. We will discuss these to demonstrate the development process of Kobot and the ideas behind it.

4.1.1 Version 1

The earliest version of Kobot was version 1, as shown in Figures 4.1 and 4.2. The body had a layered structure where the bottom layer contained the locomotion, power and control sub-systems. The middle layer had the short-range sensing sub-system and the third layer had the imaging sub-system. The structural sub-system in this version was composed of two concentric circular parts manufactured from Delrin, each having a diameter of 120mm. The two circular layers were connected to each other via a circular Delrin rod at the center, which together resulted in a height of 70mm. Two standard hacked RC servo-motors were utilized with the wheels placed inside the robot body and directly coupled to the motors. And in versions 1.1 to 1.3 as shown in Figures 4.1(b), 4.1(c) and 4.1(d), Sharp GP2D12 IR-sensors were utilized to detect kin-robots. These sensors were abandoned in versions 1.4 and 1.5 due to their low-speed operation. In these versions, an embedded camera board (CMUcam-2) and a spherical mirror were utilized to detect the range and heading of kin-robots. Four AA-size NiMH rechargeable batteries were utilized as the power source. The weight of version 1 happened to be greater than expected. The placement of the controller and the batteries were also inconvenient and not ergonomic to handle. Hence, the structural sub-system was redesigned from the ground up in the next version.

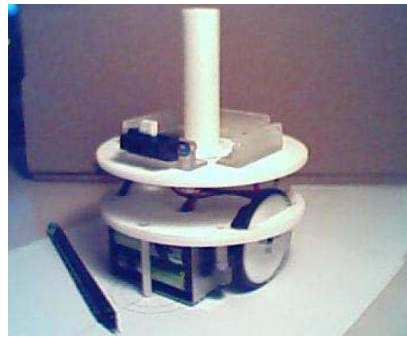
4.1.2 Version 2

In this version, there was a major leap ahead in the type of body material and the way it was manufactured, as shown in Figure 4.3. The body was composed of top and bottom parts which were casted from polyurethanes (PU)⁴.

⁴PU is an elastomer having favorable properties such as low density (specific gravity is 0.2), aesthetics, and durability. It is widely utilized in many industries such as the automotive and construction industries.



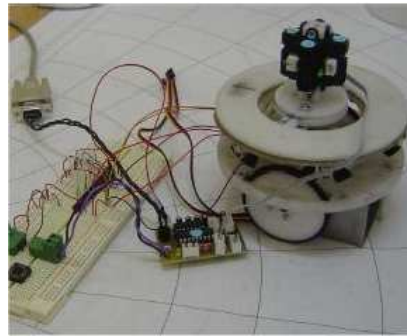
(a)



(b)



(c)



(d)

Figure 4.1: (a) Kobot prototype version 1.0. (b) Kobot prototype version 1.1. (c) Kobot prototype version 1.2 (d) Kobot prototype version 1.3.



(a)



(b)

Figure 4.2: (a) Kobot prototype version 1.4. (b) Kobot prototype version 1.5.

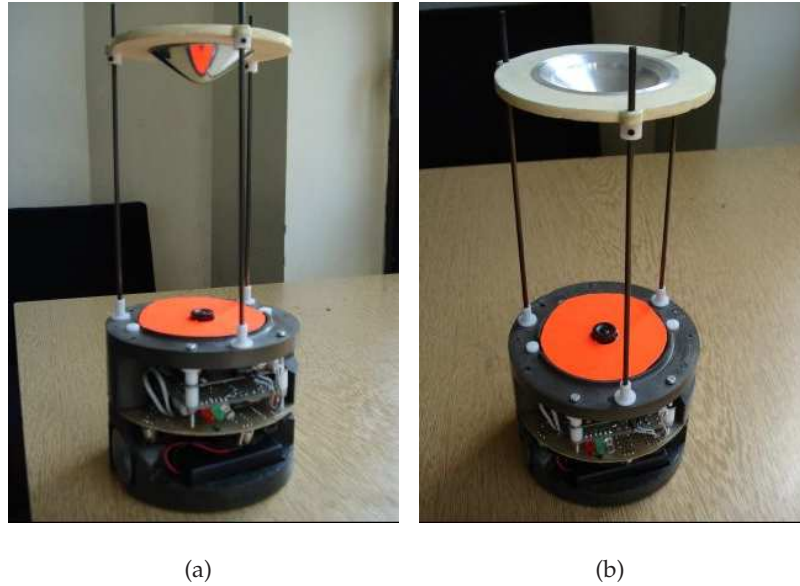


Figure 4.3: (a) Kobot prototype version 2.0. (b) Kobot prototype version 2.0.

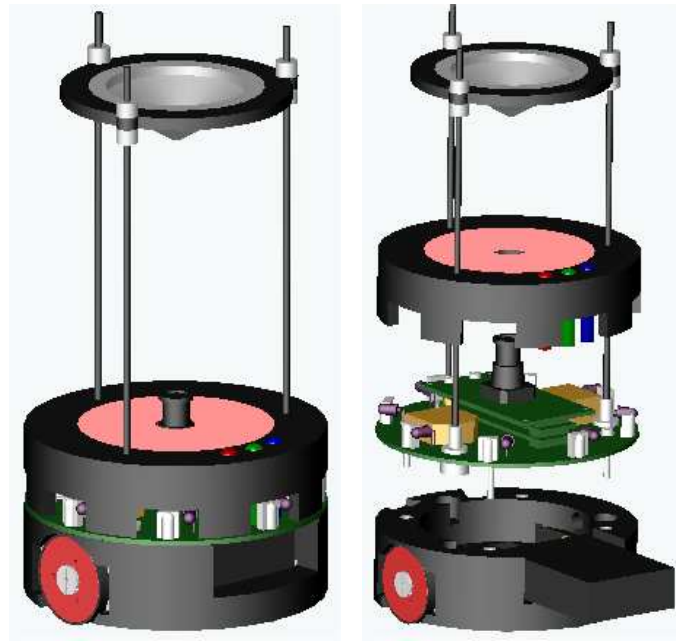
Two high-quality gearhead DC motors, the battery and the short-range sensors were mounted on the bottom part. The camera and controller board were placed on the top part. A linear mirror was placed on the top part via three carbon-fiber rods. The frontal side of the two parts were left open for easy access to the battery and the electronics. Three of these prototypes were manufactured. Their performance was tested by conducting simple aggregation experiments, and found acceptable. Power consumption was the major flaw in this version. The embedded camera board (CMUcam-2) consumed too much power decreasing the operation time of the robot considerably. Therefore, we decided to replace CMUcam-2 with a different sensing system in the next version of Kobot.

4.1.3 Version 3

The body and the electronics were redesigned. A low-power imaging system instead of the CMUcam-2 was planned to be designed and utilized in this version, shown in Figures 4.4(a) and 4.4(b). However, due to time constraints, we had to give up this idea and leave it as a future project. It remains an ongoing project and will be utilized in future studies.

4.1.4 Version 4

Version 4 had exactly the same mechanical and electronic design as version 3 except for the utilization of a virtual heading sensor and the abandonment of the imaging sub-system. The virtual heading sensor was designed to enable measurement of the headings of neigh-



(a)

(b)

Figure 4.4: (a) Kobot prototype 3.0 front view. (b) Kobot prototype 3.0 exploded view.

boring robots in the group. It is a combination of a digital compass and wireless communication modules. A wireless communication module was also used in robot-console communication for programming and debugging purposes. The short-range sensing subsystem was redesigned to enable analog proximity measurement over short distances. It was able to distinguish kin-robots from obstacles as well. Interference due to nearby robots was also minimized through the utilization of a CSMA-CA (carrier sense multiple access with collision avoidance) like algorithm. All the electronics and mechanics were reshaped from the ground up.

The body was composed of top and bottom parts. All of the electronic cards were mounted on the bottom part and the top part was only utilized to protect the cards and for aesthetic purposes. The digital compass was mounted on top of the robot using a circular rod to decrease the effect of electromagnetic noise due to the electronic cards. In Figure 4.4(b), the short-range sensing system is shown in between the bottom and top parts. All the electronic cards (communication module, power module and main control board) and components were stacked on the short-range sensing card using a small form factor connector as shown in Figure 4.5. This design had several advantages: (1) The cabling problem we faced in prototype 2.x was totally solved. (2) Cards were connected both electrically and mechanically in a reliable fashion. (3) Modularity was facilitated in design.

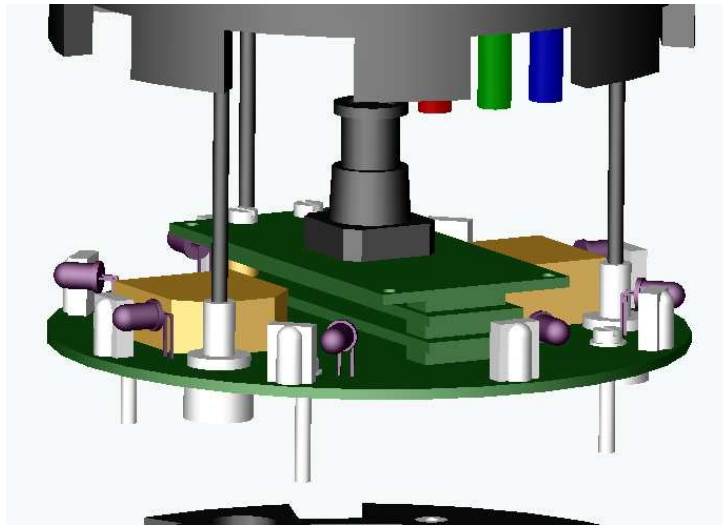


Figure 4.5: Close-up view of the short-range sensing sub-system.



(a)

(b)

Figure 4.6: (a) Kobot version 4.0 with the virtual heading sensor. (b) Seven Kobots with the virtual heading sensor.

A suitably designed card such as a camera board and a controller could be integrated into the system conveniently. In Figure 4.5, a hypothetical camera card is placed on top of the card stack. Removal of a card from the stack is also as easy as placing a new one.

Old batteries could easily be replaced with new ones by removing the battery case from the front-opening of the bottom part. In order to prevent usage of cables in the top part, three debug LEDs were also mounted on the short-range sensing card and the lights of these LEDs were transmitted to the top part using acrylic light-guides. A photo of version 4 is shown in Figures 4.6(a) and 4.6(b). Ten of these prototypes were manufactured and utilized in the study.

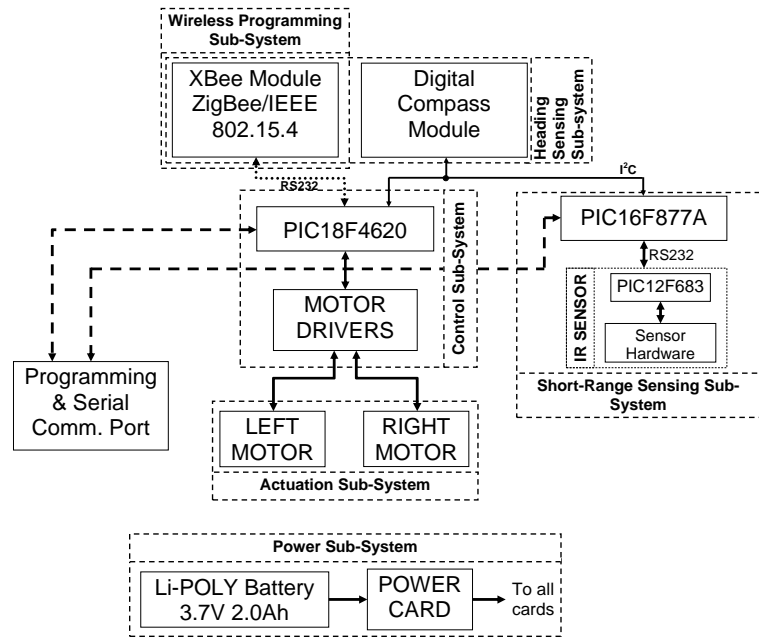


Figure 4.7: Block Diagram of Kobot.

Specifications

The final version of Kobot is an extendable, power-efficient and relatively cheap robot platform which weighs only 350 gr with batteries. It is differentially driven by two high quality gearhead motors. The robot has eight IR sensors for kin and obstacle detection and a digital compass for heading measurement. An IEEE 802.15.4/ZigBee compliant wireless communication module with a range of approximately 20 m indoors is utilized for communication between robots as well as between the robots and a console. The robot hosts a 20 MHz PIC184620A microcontroller, which can be programmed through the wireless communication channel to control its behavior. The low-power design of its systems gives the robot an operation time of 10 hours with a 2000 mAh Li-Poly battery.

The block diagram of Kobot is shown in Figure 4.7. At the heart of the Kobot is the control sub-system. All of the information from the communication, short-range sensing and heading sensing sub-systems is fed into the control sub-system, which runs the control algorithm on a PIC18F4620 microcontroller. The behaviors are controlled via driving the motors, changing the control inputs to the motor drivers using pulse width modulation (PWM) which together form the locomotion sub-system. The control sub-system also controls the three debug LEDs and a buzzer to indicate the state of the Kobot.

A comparison of existing robot platforms and the final version of Kobot is shown in Table 4.2.

4.2 The Short-range Sensing Sub-system

The short-range sensing sub-system is designed to measure the range and bearing of kin-robots and other objects in close proximity. The issues related to the short-range sensing sub-system can be listed as: generic sensing, minimal interference from the environment, minimal interference from other robots, analog proximity measurement and low-power.

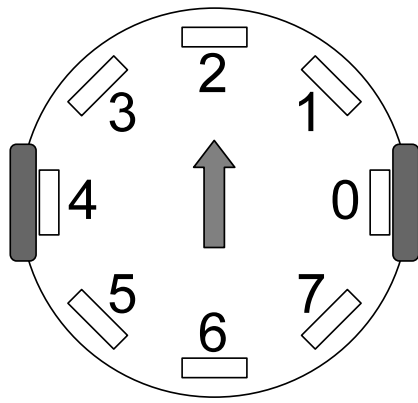
The system should satisfy generic sensing principles, which enable change of the sensing characteristics using software. The proximity measurement is required to be performed in analog fashion with minimal power consumption. The most important of all is the elimination of interference in the sensing system from other robots and the environment. Interference is inevitable due to the operating conditions of the robots in flocking. Many approaches exist to eliminate environmental noise; however, elimination of interference from other robots still remains an unsolved problem.

Design

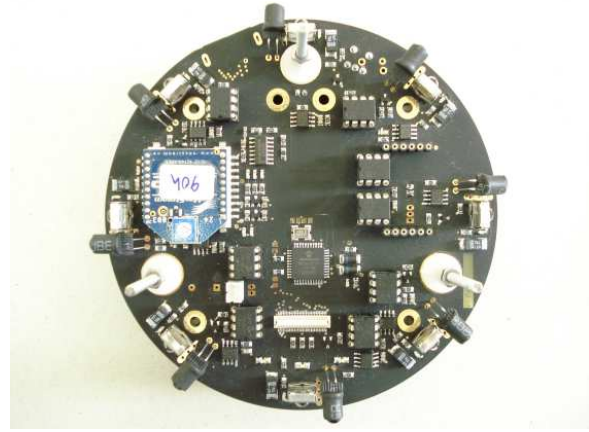
A novel sensing system was designed to satisfy the short-range sensing requirements of the robotic system. The early versions of the system operated in digital fashion and lacked kin-detection ability. In the most recent design, however, kin-detection was enabled and the sensing system operates in analog fashion where proximity measurements are performed at eight discrete levels with a range of approximately 21 cm.

The short-range sensing sub-system is composed of eight IR sensors placed uniformly at 45° intervals, as shown in Figures 4.8(a) and 4.8(b). The system is a low-power and high-speed (18 Hz) object detection unit which has the ability to distinguish robots from obstacles/walls in the surrounding area. The range of operation is approximately 21 cm (for round objects), and the output of the sensor is presented at eight discrete levels having equal spacing. 38kHz modulation is also utilized in IR LEDs to eliminate environmental noise.

Each IR sensor is composed of an IR LED, an IR receiver, a PIC12F683 microcontroller and an IR LED driving circuit with an op-amp, as shown in Figure 4.9. The PIC12F683 both controls the operation of the IR sensor and transmits the measurement result to the main sensor controller, the PIC16F877A, when requested. The IR LED is driven to seven predefined voltage levels by low-pass filtering the pulse width modulation (PWM) output of the PIC12F683. The voltage value that the LED is driven to is measured using the analog-to-digital (A/D) converter channel of the PIC12F683. When the desired value is attained,



(a)



(b)

Figure 4.8: (a) Schematic diagram of the sensor board. The arrow indicates the forward direction of Kobot. (b) A photograph of the sensor board.

the IR LED is turned on and a 38kHz modulated IR signal is emitted. The IR signal is reflected back if there is an object nearby, and detected by the IR receiver. The IR receiver multiplies, filters and demodulates the detected signal and outputs the result to the digital port of the PIC12F683.

The sensors operate in three states: *kin-detection*, *proximity-sensing* and *data-transmission*, which are controlled by the main sensor microcontroller. The sensor starts in *kin-detection* state initially. In this state, the IR receiver is active and the IR LED is off. The sensor "listens" to the environment to detect any IR irradiation, which, when detected, is an indication of a nearby kin-robot. The next state is the *proximity-sensing* state. In this state, the sensor first looks for any incoming IR irradiation; if detected, it waits for approximately 6.7ms which is the maximum possible time spent by a sensor to detect an object, for the nearby robot to finish its measurement. After that, it performs proximity sensing by varying the power of the IR LED to determine the minimum level at which the radiated signal reflects back from the object. Proximity measurement result in one of eight discrete levels, which is then transmitted to the main sensor microcontroller in the *data-transmission* state. The sensor then returns back to the *kin-detection* state.

This process is illustrated in Figure 4.10. The sensor first entered the *kin-detection* state upon receiving a C1 command from the main sensor controller. The sensor, in this state, tried to detect any IR emission from neighboring robots. The sensor in the figure was not able to detect any irradiation which means that there was no nearby kin-robot. The main sensor controller then puts the sensor into the *proximity-detection* state by sending a C2 com-

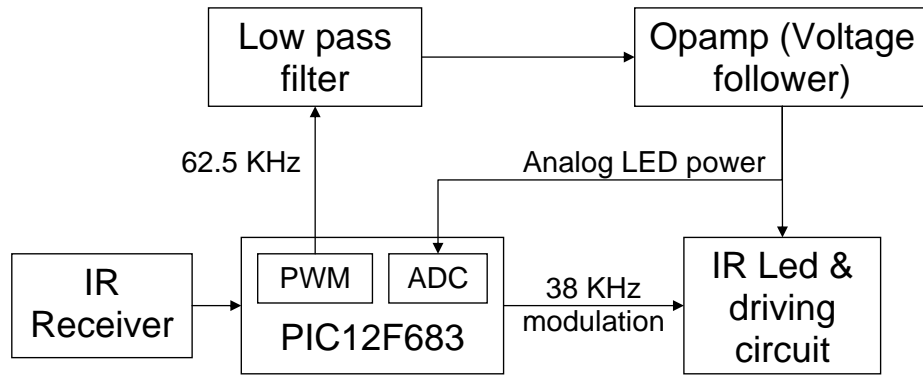


Figure 4.9: Block diagram of an individual sensor.

mand. The sensor, in this state, detected an IR signal, which means that a nearby sensor was making a measurement, and the sensor waited for approximately 6.7 ms to guarantee that the neighboring sensor had finished its measurement. The sensor then started measurement from *Level1* and detected an obstacle in *Level3*. The proximity reading was then transmitted to the main sensor controller when the sensor was put into *data-transmission* state by the C3 command.

The main sensor controller, PIC16F877A, coordinates the states of the sensors. First, the even-numbered sensors ($I_0, I_2, I_4,$ and I_6) are put into *kin-detection* state. Then, after a predefined amount of time (2 ms), the odd-numbered sensors ($I_1, I_3, I_5,$ and I_7) are put into *kin-detection* state to prevent accidental reception of any reflected IR signal from a neighboring sensor. The main sensor controller ensures that the sensors stay in *kin-detection* state for a random period of time (11-15ms) to prevent synchronization of the sensors of two neighboring robots. The sensors are eventually put into *proximity-sensing* and *data-transmission* states.

The k^{th} sensor returns an integer pair (d_k, r_k) to the main sensor controller via a software-implemented serial port. $d_k \in \{0, 1, \dots, 7\}$ denotes the distance from the object being sensed. $d_k = 1$ and $d_k = 7$ indicate a distant and a nearby object, respectively. $d_k = 0$ stands for no detection. $r_k \in \{0, 1\}$ shows whether the detected object is a kin-robot or not. The main sensor controller having received the measurement results from the eight sensors, transmit these to the main controller via I²C protocol.

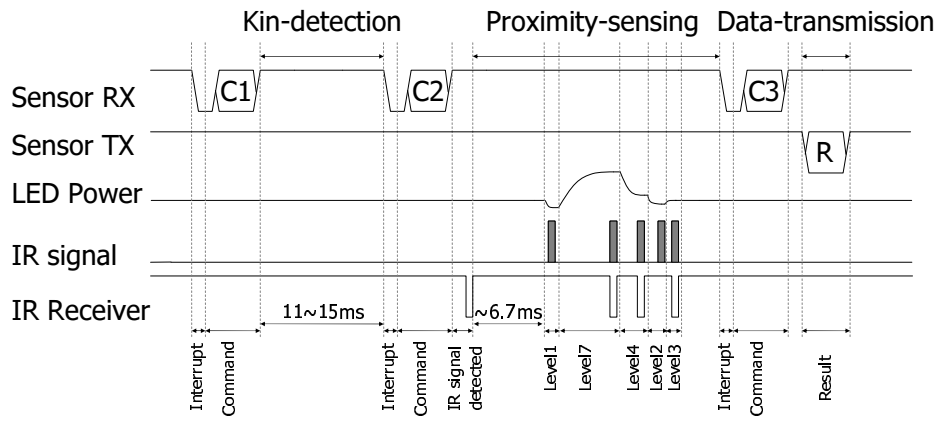


Figure 4.10: Timing diagram of a sensor detecting an obstacle.

4.3 The Heading Sensing Sub-system

Heading information is vital to achieve stable flocking in open space, and an off-the-shelf sensor does not exist for this purpose. Therefore, we designed a heading sensor from the ground up to measure the headings of the robots. The issues related to the design of the heading sensing system are: generic sensing, indoor operation, minimal interference from robots, minimal interference from the environment, low-weight and low-power.

The sensing system should be able to operate indoors since our main concern is indoor operation. Noise and interference due to neighboring robots and the environment should be eliminated or at least minimized in the design. If these are inevitable, they should be handled by software through the generic sensing capability and kept at acceptable values. The weight and power consumption of the system should also be kept at the minimum possible values.

Design

It is almost impossible to design a simple sensing system that can be utilized to sense headings. We therefore approached the problem in a different manner. We utilized the wireless communication module available in each robot (Section 4.4) and added a digital compass module to constitute a novel sensing system called the virtual heading sensor (VHS). Honeywell's small-sized, low-cost and low-power HMC-6352 digital compass module is used as the compass and is shown in Figure 4.11(a). The HMC-6352 has two highly-sensitive magneto-resistive sensors placed at 90° from each other as shown in Figure 4.11(b). It can detect its orientation with respect to the North pole of the Earth with high accuracy and repeatability when operated in outdoor environments.

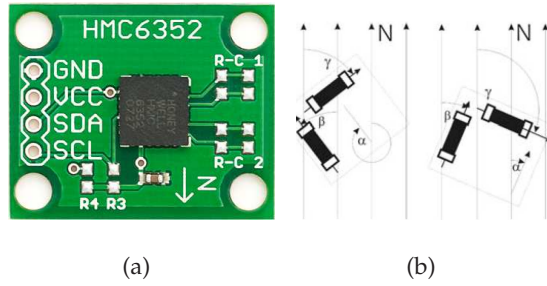


Figure 4.11: (a) Photo of the HMC-6352 digital compass module. (b) Schematic of HMC-6352 placed in a magnetic field. Reproduced from [3].

The VHS operates in two states. One is the *active* state and the other is the *passive* state. In the *active* state, the VHS, through its digital compass, measures the heading of the robot with respect to *sensed North* in degrees with a precision of $\pm 0.5^\circ$. The heading is then converted to radians (θ) and broadcasted to the neighboring robots within the communication range. The *active* state lasts for approximately 3 ms and is repeated at each control step (110 ms). In the *passive* state, the VHS listens to the environment for *heading messages* from robots for approximately 110 ms, and then the system returns back to the active state again. This cycle continues throughout the operation of the robot.

The received heading value (θ_j) from the j^{th} robot is then converted to the body-fixed reference frame of the robot having a heading⁵ of θ shown in Figure 6.3 using the following equation:

$$\theta'_j = \frac{\pi}{2} - (\theta_j - \theta) \quad (4.1)$$

where θ'_j is the heading of the j^{th} VHS neighbor with respect to the body-fixed reference frame of the robot.

The assumption behind the operation of the VHS is that the *sensed North* direction is considered to be approximately the same within the communication range (R) of robots; hence, the values conveyed by the *heading messages* are considered to be with respect to the same reference, the *sensed North*, and Equation 4.1 holds. This situation is illustrated in Figure 4.12. Three nearby robots on the left-hand side of the figure have approximately the same *sensed North* direction (indicated by dashed arrows) (n_{s1} , n_{s2} and n_{s3}); hence, it is logical to assume *sensed North* as the common reference among these robots.

The major flaw behind this assumption is the hard-iron effect of nearby ferrous met-

⁵The heading of the robot is the angle between the *sensed North* and the y-axis of its body-fixed reference frame measured in clockwise direction.

als in the environment, which may cause the *sensed North* direction deviate considerably from the North pole. This is illustrated in Figure 4.12. The robot on the right-hand side is close to a ferrous metal (shown as a U-magnet) and its *sensed North* direction (n_{s4}) is greatly deviated from the North pole (indicated by the bold arrow, \mathbf{N} , in the y -axis) and different from the other robots' (n_{s1} , n_{s2} and n_{s3}). In this case, the common reference assumption among the robots is not realistic. This is a fact, but since the average of the heading values are utilized in the flocking behavior, averaging acts as a low-pass filter and cancels the counter-effects of noise. And in experiments related to the VHS characteristics given in Section 8.1, we observed that this type of sensing is quite robust, even in indoor environments where metal objects are abundant.

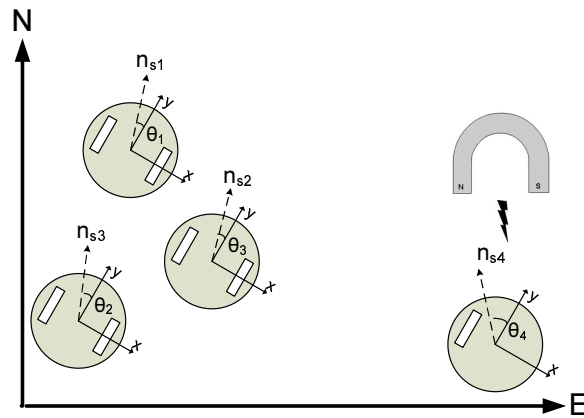


Figure 4.12: Illustration of the hard-iron effect on the VHS.

4.4 The Communication Sub-system

The control algorithm or certain parameters are frequently changed during the development of the robotic system. It is also necessary to debug the control algorithms during run-time. It is almost impossible to accomplish these tasks utilizing a cable connection. Hence, a means of wireless communication should be utilized in robots to enable both debugging and programming during run-time. It is convenient to perform programming in batches in order to decrease the programming time, especially when large groups are concerned. The design issues related to the communication sub-system are: power consumption, communication speed, scalability, real-time debugging support and cost.

In the design of the communication sub-system, the minimization of power consumption is the most important factor. There exists a compromise between power consumption and rate of communication which directly affects the programming time. An appropriate communication speed should be determined by keeping the programming time within acceptable limits. The sub-system should also enable a wireless link between the robots and a console for real-time data acquisition from the robots, both for debugging and testing purposes.

One of the drawbacks of using wireless communication is its adverse effect on scalability, which is an important issue in swarm robotic systems as explained in Chapter 1. In order to preserve scalability, range of communication should be limited or preferably adjusted depending on the mode of operation. The range of communication can be maximized when the robots are being programmed and minimized when they are communicating with each other. In this way, scalability and low-power operation can be facilitated in the communication sub-system.

Related Work

Wireless programming is a common need in multi-agent systems including swarm robotics and sensor networks. It should be power-efficient, transparent to the user and rapid. Almost all of the swarm robotic systems in the literature support wireless programming, but none of them has a wireless parallel programming feature. Robots are programmed one-by-one instead. Alice [49] and Jasmine [50] are micro-robot platforms equipped with a frequency-modulated IR receiver module which is used for wireless programming. *s-bot* [51] and *Flockbot* [55] are more advanced robotic platforms designed for swarm studies, and they both use the IEEE802.11 wireless standard for communication and wireless programming. *Swarmbot* [52] is another swarm robotic platform which has an RF modem for wireless communication and programming. *e-puck* [48] is one of the most recent robotic platforms, and has the ability of wireless programming using an on-board IEEE802.15.4/ZigBee wireless adapter.

The constraints of wireless programming in swarm robotic systems are quite different from those in sensor networks. The first difference is that the number of individuals, called motes in a sensor network, is much greater than the number of robots in a swarm robotic system, the former measured in thousands and the latter probably in tens. The other difference is in power-efficiency. Wireless programming should be extremely efficient in sensor

networks to save battery-life since it is very hard, if not impossible, to recharge thousands of batteries.

Several different wireless-programming algorithms are proposed for sensor networks. Trickle [56] uses an algorithm called "polite gossip". In this algorithm, nodes broadcast a code summary periodically, and stop broadcasting when they receive a summary identical to their own. If an older version of the code is heard, each node continuously broadcasts its own code summary to trigger its neighbors to broadcast the newer version. This algorithm is power-efficient since only code summaries are broadcasted in normal operation and the code itself is only broadcasted when needed. New code also propagates rapidly within the network.

Another approach to wireless programming in sensor networks is to implement a virtual machine in motes to decrease the code size to be programmed. Mate [57] is an example of this approach. Mate is a byte code interpreter which works on the TinyOS [58] operating system. It decreases the code size of very complex programs so that the amount of code to be broadcasted wirelessly decreases considerably. Code is encapsulated into small packages and only updated packages are broadcasted. When a mote receives an updated code segment, it updates itself and immediately forwards the new version to its neighbors. In this way, the new code segment propagates through the network effectively.

Design

Hardware

A PIC 16F877A⁶ microcontroller (A PIC18F4620 is utilized in the current version) is utilized in the main control board. The 16F877A is a widely-used microcontroller having the ability to write to its own program memory; it has one hardware-implemented serial port. Many examples of open-source wireless programming software are also available for the PIC16F877A. Among these, Screamer⁷ by Sparkfun Inc. was selected as the most appropriate alternative due to its extensive use and good support.

MaxStream's wireless communication module, called XBee, supports the IEEE802.15.4/ZigBee protocol and is utilized as the wireless communication module in Kobots. The IEEE802.15.4/ZigBee provides low-power networking capability that can support point-to-point, point-to-multipoint and peer-to-peer communication facilities.⁸

⁶<http://www.microchip.com>. Last visited: April 2008.

⁷<http://www.sparkfun.com>. Last visited: April 2008.

⁸<http://www.maxstream.net>. Last visited: April 2008.

This protocol was preferred to the IEEE802.11 because of its power efficiency, and to Bluetooth due to its ability to address 65536 modules instead of seven as supported by Bluetooth. Through this system, *ad-hoc* communication among the robots and communication with a console are supported. Also, using the broadcasting capability of this protocol, we were able to implement wireless parallel programming of multiple robots. The XBee modules from Maxstream, as shown in Figure 4.13(b), communicate with the host via a serial interface at up to 115.2kbps and provide a 250kbps RF data transmission rate for wireless communication. We have utilized XBee modules at a data rate of 57.6kbps.

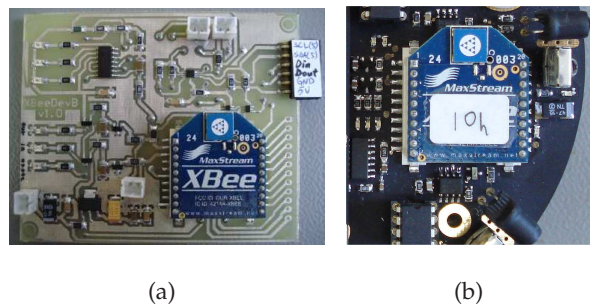


Figure 4.13: (a) XBee-PC interface card. (b) XBee placed on Kobot having an address of 10h.

Software

The wireless programming software is composed of two parts. One is the host program, running on a console, and the other is the client, running on Kobots. The host takes care of the user interface, communicates with the Kobots, and downloads and verifies the desired program. The client consists of programmer software which programs and verifies the program desired by the host, and a bootloader which either runs the programmer software or the control algorithm, depending on the request of the host.

In a typical scenario (Algorithm 1), the user first stops the currently operating Kobots, then marks the Kobots to be programmed and selects the application program to be downloaded. The selected program is divided into 4-to-8-word-long blocks. The blocks are then fetched in sequence, and their lengths, starting memory addresses and checksums are determined. These values, together with the program words, are broadcasted and verified on each robot. Robots reporting checksum errors are put on the CandidateList and a predefined number of programming retries are performed on these robots. Robots still not able

to be programmed are discarded. The same procedure continues until all of the program is downloaded to the robots.

The Kobots are initially kept in *wait* state, as indicated in Algorithm 2. They change their state to *receive* state when the console starts to broadcast program words. On receiving the last byte of the program block, the checksum is calculated. If it is correct, the received block is written to the program memory and TRUE is sent as a reply to the console. Otherwise, the write operation is not performed and FALSE is sent. In this case, the host program re-broadcasts the same block. This procedure continues until all the blocks are successfully broadcasted to the robots.

Performance of the Wireless Parallel Programming System

Several experiments were conducted to test the performance of the wireless parallel programming system. In these experiments, a custom-designed XBee-PC interface card was utilized to connect the console with XBee module, as shown in Figure 4.13(a). An XBee module was also interfaced to each Kobot, as shown in Figure 4.13(b). The host program was run on the console and the client program was run on the Kobots.

The results of these experiments are shown in Figure 4.14. During the experiments, the XBee wireless module was interfaced at 57600 baud, which was the fastest baudrate at which we faced the least amount of trouble. The program size in the experiments was 5.6 kWord, which is approximately 70% of the program memory of the PIC16F877A. It takes approximately 33 s to program one robot and each robot adds approximately 10 s to the programming time. The host program spends 10 s to verify after broadcasting a block of program. A total of 130 s was spent to program 10 robots via parallel programming, which would have been approximately 330 s ($=33 \times 10$) if the robots had been programmed one by one in serial fashion.

4.5 The Control Sub-system

The control sub-system runs the main control algorithm and controls the behavior of the robots. Distributed architecture, processing power and low-power operation are the design issues related to this sub-system.

The architecture of the control sub-system is required to be distributed, meaning that each sub-system must have its own dedicated controller. In this way, modularity is assured in the design of the robot and new sub-systems can easily be introduced into the design.

Algorithm 1 The host program

```
1: {User marks the Kobots to be programmed}
2: marked Kobots are inserted in CandidateList
3: {User selects the program to be downloaded}
4: load the program to be downloaded
5: for Robot in CandidateList do
6:   send ENQ to Robot
7:   if not timeout and reply equal to ACK then
8:     change robot's tag to present
9:   else
10:    change robot's tag to not present
11:   end if
12: end for
13: for block in program do
14:   for Robot in CandidateList do
15:     if there exists RetryTaggedRobot then
16:       put all RetryTaggedRobot in ProgramList
17:       change block to previous
18:     else
19:       put all PresentTaggedRobot in ProgramList
20:     end if
21:   end for
22:   for Robot in ProgramList do
23:     send ENQ to robot to request checksum result
24:     wait for ACK from robot
25:     if no reply or unexpected character then
26:       change tag to notpresent
27:     else
28:       if reply FALSE and retry less than 3 then
29:         change tag to retry
30:       else
31:         change tag to present
32:       end if
33:     end if
34:   end for
35:   broadcast START character
36:   broadcast block length
37:   broadcast starting memory address
38:   broadcast checksum
39:   broadcast program block
40: end for
41: broadcast END character
```

Algorithm 2 The client program

```
1: if not timeout and ENQ received then
2:   send ACK to console
3: else
4:   run current program
5: end if
6: while TRUE do
7:   wait for ENQ
8:   if checksum false then
9:     reply FALSE
10:  else
11:    reply TRUE
12:  end if
13:  wait for START character
14:  receive block length
15:  if block length equal to END then
16:    start downloaded program
17:  end if
18:  receive memory address
19:  receive checksum
20:  receive program block
21:  calculate checksum
22:  if checksum = 0 then
23:    write program block to memory
24:  end if
25: end while
```

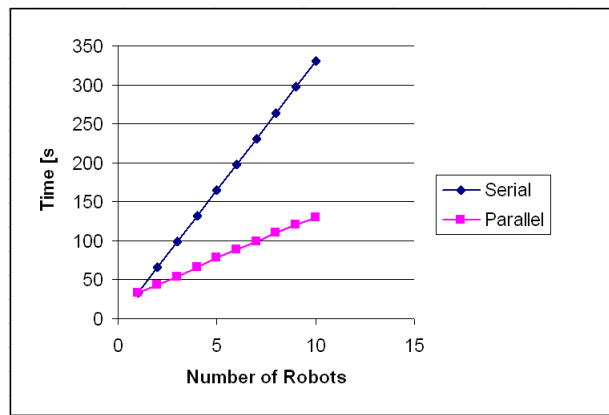


Figure 4.14: Duration of programming. XBee is interfaced at 57600 baud. Program size is 5.6 kWord.

Processing power and power consumption are again the two competing factors in the design. The operating frequencies of microcontrollers are kept at minimum levels to decrease power consumption, and hence, save battery-life.

Design

Robots are designed to have a distributed control architecture, meaning that each sub-system has its own controller to take care of low-level tasks. There is a common communication medium between these controllers, as illustrated in Figure 4.7. The control sub-system consists of the main controller board whose design is based on the PIC18F4620 microcontroller, as shown in Figure 4.15. It communicates with the main sensor controller via I²C communication protocol to receive the sensor readings. Having the sensor readings and the control algorithm, behavior of the Kobots is controlled by commanding the left and the right motors with the main controller. The XBee wireless communication module is also interfaced to the serial port of the main controller to download or debug the control algorithm wirelessly.

The design is scalable and modular in the sense that any additional sub-system, such as an imaging sub-system, can easily be interfaced to the control sub-system via I²C communication protocol. The imaging sub-system will have its own controller for low-level tasks like camera parameter adjustment, data collection, and for high-level tasks like image processing and pattern-recognition algorithms.

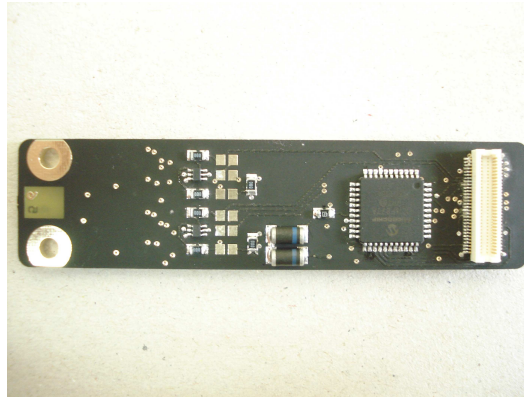


Figure 4.15: Main control card.

4.6 The Locomotion Sub-system

The locomotion sub-system is responsible for the movement of the robot on the $x - y$ plane. Design issues related to the locomotion sub-system are: the driving system, the size of the actuators, modularity, the driving electronics and low-power.

The driving system determines how Kobots are driven. Among various alternatives, the differential drive system is utilized in Kobots for its ease of application and high maneuverability characteristics. DC motors are used as the actuators and they should be power efficient, high-torque and small. Flat motors are preferable to save space inside the Kobot. The locomotion sub-system should also be modular, which means ease of changing motors to more powerful ones or more capable ones for different applications, without making considerable changes in the structural sub-system. The driving electronics should be simple and be able to drive motors efficiently at high PWM frequencies.

Design

The Kobot is a differentially-driven robot which has two powered wheels on its sides and two caster wheels in front and back. Kobots can only move on smooth surfaces without any gradients in indoor environments. 3 V, high-efficiency (to save power), low-profile (to save space) and high torque motors from FTB Inc. of Faulhaber⁹ are used in the locomotion sub-system. The wheels are directly coupled to the motors, and are composed of three parts. The inner part, which is the wheel hub is manufactured from high-grade aluminum. It is specially designed to decrease shaft load on the motors and to eliminate any possible cause of eccentricity in the wheel. The middle part is the rim and is manufactured from Delrin.

⁹<http://www.faulhaber.de>. Last visited: April 2008.

The outer part is the tire, in which an O-ring is utilized to increase the coefficient of friction between the wheel and the ground. The motors are mounted using an aluminum adapter plate so that different motors can be utilized in Kobot with only a change of the adapter plate.

The motors require a high-frequency PWM signal to increase efficiency; hence, the Si9988 motor-driver chip from Vishay¹⁰, is utilized in the motor-driver module. The Si9988 is an efficient motor-driver chip which requires almost no external components. It allows voltage and current measurement of the motors. It is not implemented in the current version of the Kobot, but will be utilized in the future for close-loop speed or torque control of the motors, respectively. The motors are driven in open-loop fashion for the time being, with a frequency of 78.5 kHz, and consume approximately 30 mA in the worst case.

4.7 The Power Sub-system

Autonomous operation of the robots requires batteries to be used as the energy source. Voltage from the batteries is converted to a convenient level by a voltage converter. The design issues related to the power sub-system (battery and voltage converter) are: duration of operation (battery), efficiency (voltage converter), size and weight (battery), and recharging (battery).

Batteries supply energy to the robot. The output of the batteries is converted to a convenient level by a voltage conversion system, which should be efficient to maximize battery-life. The energy content of the batteries depends on their chemistry and their size. Increasing the size of the battery increases its capacity, but this causes problems in the system. Therefore, it is logical to use high-technology batteries with an appropriate size having high energy-density. These batteries should not only be rechargeable, but also allow rapid recharging. An ideal figure for the duration of operation is assumed to be three hours.

Design

The power sub-system consists of rechargeable batteries and a voltage-conversion module as shown in Figure 4.16. The voltage-conversion module is designed to operate with different battery types and sizes. A three cell AAA/AA sized Ni-MH battery or a one-cell Li-Poly battery can be utilized with the voltage-conversion module. The most popular rechargeable battery on the market today is the Ni-MH battery, which can be found in different

¹⁰<http://www.vishay.com>. Last visited: April 2008.

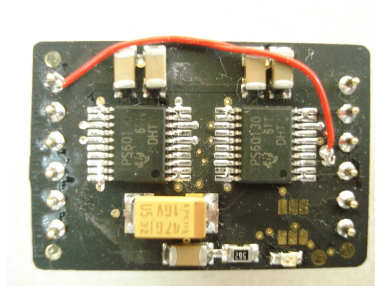


Figure 4.16: Voltage conversion module.

Table 4.1: Power consumption of sub-systems

Sub-system	Current (mA)	Voltage (V)	Power (mW)
Locomotion	60	5	300
Short-range	60	5	300
Control	20	5	100
Communication	50	5	250
Total			950

sizes and shapes. AA and AAA sizes are convenient due to their size and weight. Li-Poly batteries are the most recently-introduced rechargeable batteries. They have an energy density of approximately three times that of Ni-MH batteries. One of the important points one should consider when using Li-Poly batteries is to prevent the battery voltage from dropping below a critical value, or else the batteries will be damaged permanently. The voltage-conversion module consists of a 10F202 microcontroller (yet to be implemented) to automatically stop the robot when the battery voltage drops below the critical value.

The voltage-conversion module regulates the input voltage to a constant 5 V, which is the required TTL voltage level for the robot, except for the communication sub-system, which has its own regulator that converts 5 V to 3.3 V for the operation of the XBee module. There is a red LED in the voltage-conversion module to indicate low-battery voltage.

The measured power consumption values are tabulated in Table 4.1. Endurance tests were performed with Kobot, having all of its sub-systems operating with three-cell AA-size Ni-MH or one-cell Li-Poly batteries. The results of the experiment were quite impressive, and exceeded our expectations. 7.5 hours and 10.5 hours of autonomy were observed with Ni-MH and Li-Poly batteries, respectively. These figures put Kobot in the first place in its segment.

4.8 The Structural Sub-system

The structural sub-system forms the body of the robot, that carries and protects the other sub-systems. The issues related to the design of the structural sub-system can be listed as: weight, modularity, durability, aesthetics, size and cost.

The weight of the body should be minimal to decrease the weight of the robot. Two ways exist to minimize the weight. One is to decrease the size and the other is to decrease the density of the material used for the body. The former approach is not feasible, since the size of the components used in the robot limits the minimization of the body considerably. The size of the robot should also be at a certain value for practical reasons. Hence, there is a lower limit to the size of the robot. Having fixed the size to the an appropriate value, a low-density material, polyurethanes was utilized as the body material of Kobot. Modularity was facilitated through two-piece design of the body.

In the next chapter, we will model Kobot and its sensing systems in order to utilize them in a physics-based simulator. The simulator is used to perform systematic experiments with a large group of robots.

Table 4.2: Comparison of mobile robots for swarm robotics research with Kobot

	e-puck	Alice	Jasmine	s-bot	Swarmbot	Pherobot	Khepera II	Flockbot	Kobot
Interference robots	N/A	YES	YES	N/A	NO	NO	YES	YES	NO
Interference environment	NO	NO	NO	YES	NO	NO	NO	NO	NO
Kin detection	YES	YES	YES	YES	YES	YES	YES	NO	YES
Stigmergic sensing	NO	NO	NO	NO	NO	NO	NO	NO	NO
Generic sensing	vision	vision	NO	vision	vision	NO	vision	vision	vision+IR
Wireless communication	Bluetooth 802.15.4 ZigBee	RF mo- dem	IR	Wi-Fi	RF mo- dem	NO	RF mo- dem	Bluetooth	802.15.4 ZigBee
Wireless prog.	YES	NO	NO	YES	YES	NO	YES	YES	YES
Parallel prog.	NO	NO	NO	NO	NO	NO	NO	NO	YES
Wireless robot-robot	YES	YES	YES	YES	YES	YES	YES	YES	YES
Physical Int.	NO	gripper	gripper	gripper	NO	NO	gripper	gripper	NO
Battery Life	medium	high	medium	N/A	N/A	N/A	short	medium	high
Size (cm)	dia.= ϕ 7	2.1*2.1*2.1	2.6*2.6*2.6	dia.= ϕ 12 ht.=15	13*13*13	dia.= ϕ 11	dia.= ϕ 7 ht.=3	dia.= ϕ 18	dia.= ϕ 12 ht.=7
Cost	low	low	low	N/A	N/A	N/A	high	medium	low
Simulator	Webots	Webots	Breeve	Swarmbot 3D	N/A	N/A	Webots	MASON and Breeve	CoSS
Computation	30F6014A	16F877	ATMEGA 18	XScale	40 MHz ARM	PALM V	25MHz Motorola 68331	PXA255	18F4620

CHAPTER 5

CHARACTERISTICS AND MODELING OF KOBOT

In this chapter, we determine the Kobot's characteristics of the short-range sensing system and virtual heading sensor using systematic experiments. These characteristics are then utilized in CoSS, a physics-based realistic simulator, to analyze the flocking behavior under different controller parameters and virtual heading sensor characteristics for large groups.

5.1 The Characteristics of the Short-range Sensing Sub-system

Two different experiments were conducted to determine the characteristics of the short-range sensing sub-system. One of these is the obstacle-proximity experiment as shown in Figure 5.1(b) and the other is the robot-proximity experiment as depicted in Figure 5.1(d). In both of the experiments, the robot for which proximity data was collected was kept stationary (the bottom robot in Figure 5.1(b)) and the other robot (the top robot in Figure 5.1(b)) was moved in the first quadrant of the $x - y$ plane at 1cm intervals, as shown in Figure 5.1(a). 200 measurements were taken at each point. The stationary robot had its frontal sensor (sensor I_0 in Figure 4.8(a)) turned on in both of the experiments whereas the moving robot's sensors are turned on only in the robot-proximity experiment.

The results of the obstacle-proximity experiment are shown in Figure 5.1(b). In the figure, the dark points denote low proximity values whereas the light ones denote high proximity ones. When this graph is compared with the graph of the robot-proximity experiment, shown in Figure 5.1(d), it can easily be seen that they are in accordance with each other, which reveals that the sensing system has almost the same characteristics in obstacle- and robot-proximity measurements. The experiments also revealed that the sensor has a range of 21 cm in lateral and 10 cm in transverse directions (lateral direction lies

on the x -axis and transverse direction lies on the y -axis, as shown in Figure 5.1(a)). The results of these experiments are arranged as a look-up table and utilized in the simulator to be explained in Section 5.3.

The percentage of success of robot detection is plotted in Figure 5.1(c). The dark points show a higher success rate, and the lighter ones show a lower success rate. It can be observed that for most of the operating range, the sensing system is 100% successful. The outlying points, which are beyond the range of the sensor might be interpreted as out-of-range values but are a proof of existence of a neighboring robot. These occur because sensors may receive IR irradiation from very distant kin-robots, even up to 1 m away but they cannot make a proximity measurement since the kin-robot is beyond the range of the sensor. The output of the sensor reveals the detection of a kin-robot beyond the 21 cm range. This information is quite useful in dispersion or aggregation tasks. It is also seen that the rate of success decreases considerably for nearby points, which might be due to the high intensity of IR irradiation and the possibility that the sensor cannot operate properly within such proximity.

5.2 The Characteristics of the Virtual Heading Sensor

Two characteristics of the virtual heading sensor are of main concern. One is the number of VHS neighbors that a VHS node can "hear" for different group sizes and communication ranges, and the other is noise in the heading measurement.

5.2.1 Number of VHS Neighbors

Three sets of experiments were performed to determine the average number of VHS neighbors that a VHS node (hereafter be called node) can communicate with during a predefined amount (equal to the duration of the control loop). A VHS neighbor of a robot is not necessarily one of the robots in close-proximity. It can, rather, be any robot within the communication range of robot's VHS. Thus, in that sense, neighbor relations in Kobots can be regarded as *topological* one rather than a *metric*, as in [12].

The experiments were performed in Prowler [59], an event-driven probabilistic wireless network simulator, since the experiments required more nodes than were available physically. In all of the experiments, the nodes were placed on a hexagonal grid with a 25 cm distance and kept stationary.

The first experiment aimed to test the effect of the communication range on the number

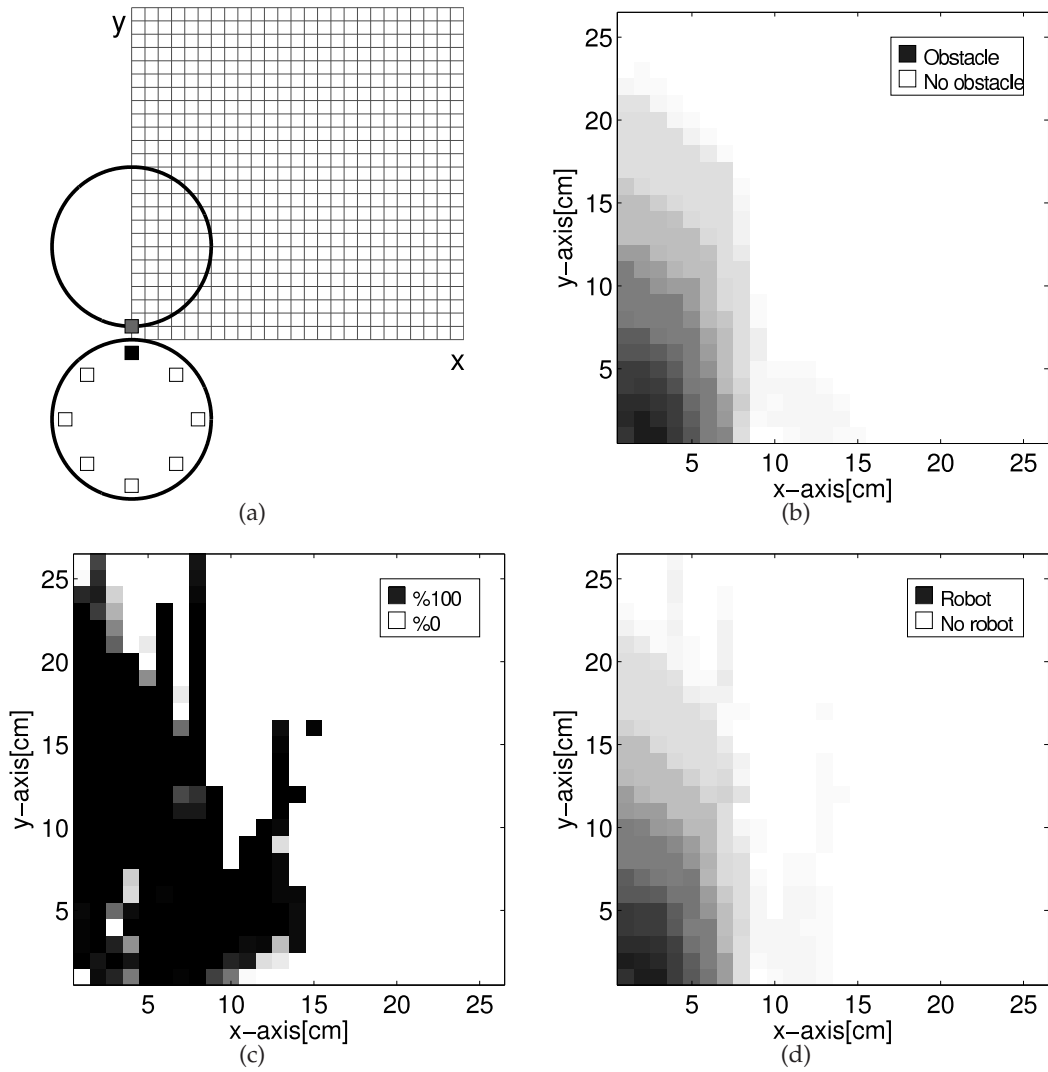


Figure 5.1: (a) Experimental setup used in the obstacle-proximity measurement and robot-proximity experiments. (b) Results of the obstacle-proximity experiment. (c) Rate of success of robot detection. (d) Results of the robot-proximity experiment.

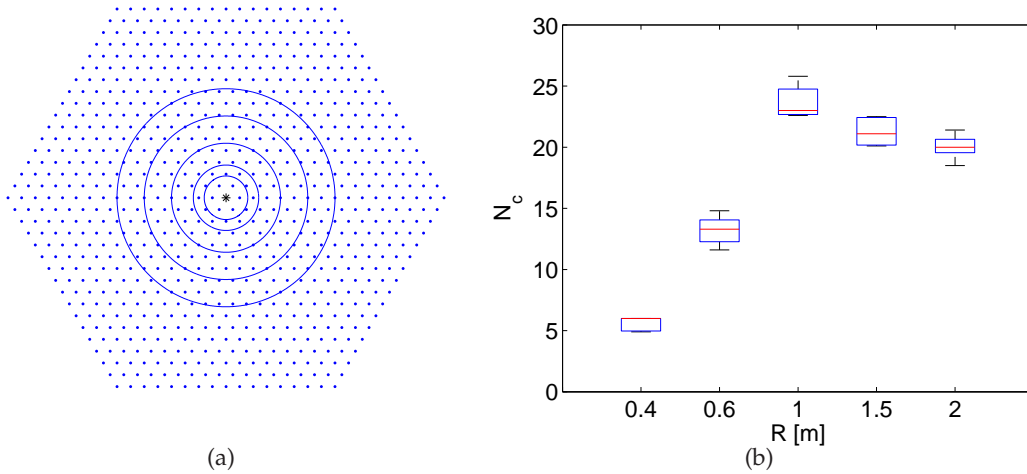


Figure 5.2: The communication range experiment. (a) Topology of nodes. Circles indicate various communication ranges of the central node. (b) Average number of VHS neighbors of the central node. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median points, respectively.

of VHS neighbors. In order to test this, the communication range was varied in such a way that there existed robots beyond the communication range of the central node. The experiment was performed with 817 nodes placed in a regular hexagonal grid having 25 cm spacing. The communication range of the VHS was changed to five different values, indicated by the concentric circles in Figure 5.2(a). When the communication range was set to its largest value (200 cm), indicated by the outermost circle, many robots were still out of the communication range, which indicates the locality of communication. For each range, five different simulations were performed with random initial conditions for 10 time-steps. The distribution of the average number of VHS neighbors for the central node are plotted in Figure 5.2(b).

It was observed that the average number of VHS neighbors increased with an increase in the communication range up to a certain value (~ 25), then decreased and settled at 20 with a further increase in the communication range. This was due to the fact that an increase in the communication range increases the number of collisions in the heading messages, preventing a further increase in the number of VHS neighbors.

In the second experiment, the effect of group size on the number of VHS neighbors was investigated. 91 nodes were utilized in a hexagonal formation having 25 cm spacing. The communication range was set to its maximum value (20 m) so that all of the nodes were within communication range of the other nodes; in other words, the communication range

was global. The number of nodes was set to five different values and five simulations were carried out for each case with random initial conditions. The simulations were performed for 10 time-steps and the average number of VHS neighbors of a node was calculated for each simulation. The results are plotted in Figure 5.3(a).

This experiment reveals that an increase in the number of nodes increased the number of VHS neighbors up to 72, and then it decreases with a further increase in the group size and settles at 20. The decrease in number of VHS neighbors is due to an increase in the number of collisions of the heading messages, which consequently limited the number of VHS neighbors to a smaller value.

The third experiment was conducted to measure the degree of randomness in the neighbor selection of a node, i.e. to determine any bias or anisotropy in neighbor selection process. In this experiment, we place 91 nodes on a hexagonal grid with 25 cm spacing. The communication range of the nodes was again set to its maximum value (20 m). The experiment was run for 100 time-steps and repeated 5 times. The hexagonal grid was hypothetically divided into six equal regions and the region of the VHS neighbors of the central node was recorded at each time-step. The results are plotted in Figure 5.3(b). It can be seen that VHS neighbors are distributed evenly to each region. This reveals that there is no directional preference in the selection of VHS neighbors.

The last two experiments revealed two important facts related to the VHS. (1) The number of VHS neighbors of a node is almost constant and depends only on group size. (2) Nodes pick up their neighbors randomly and without any directional preference. These two points were utilized extensively in simulations and modeling of the flocking behavior. It was also these two facts that enabled the possibility of an analytical treatment of the model.

5.2.2 Noise

The VHS is composed of a digital compass and wireless communication modules. Hence, its noise characteristic is a combination of these two modules. The digital compass module has a noise of $\pm 0.5^\circ$ in heading measurement which is practically negligible. The wireless communication module's operation is also free of noise, due to the parity check¹¹ feature on the reception side and the CSMA-CA¹² protocol (Carrier Sense Multiple Access with

¹¹http://en.wikipedia.org/wiki/Parity_bit. Last visited: April 2008.

¹²CSMA-CA is a widely utilized method to minimize communication errors especially in wireless communication protocols. http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_

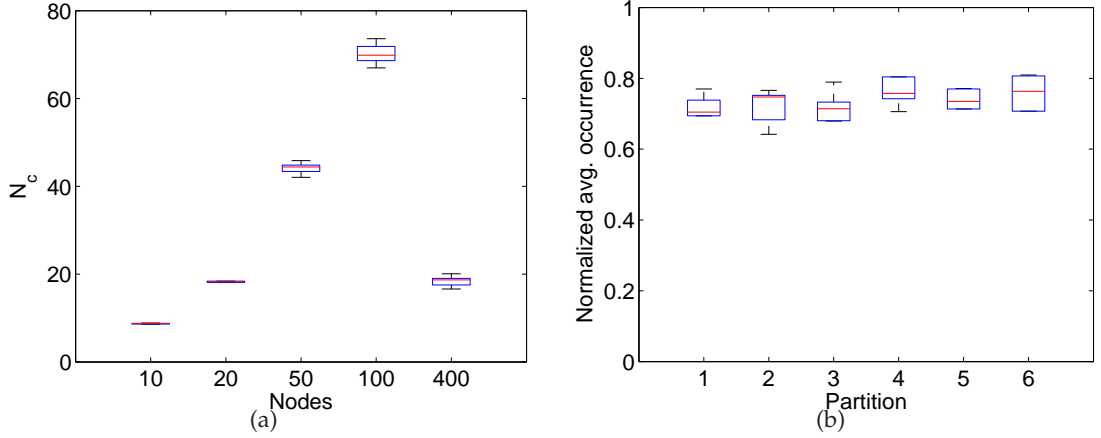


Figure 5.3: (a) The group-size experiment. Average number of VHS neighbors for five different group sizes. (b) Neighbor-selection experiment. The ends of the boxes and the horizontal line in it correspond to second and third quartiles and the median points, respectively.

collision avoidance) present in the XBee modules, which avoids interference from other robots when broadcasting heading messages.

In this respect, the VHS could be regarded as a perfect sensing system which is extremely accurate and free of noise. However, this is not the case. The noise characteristics of the digital compass module depend highly on unpredictable factors, such as the presence of nearby ferrous metals and external magnetic fields in indoor environments. Therefore, noise is regarded as an unknown parameter of the flocking behavior and is treated accordingly.

Noise is modeled using the vectorial noise model introduced by Gregoire et al. [25], in which a noise vector having a random direction with a variable magnitude is added to the measurement as shown in Figures 5.4(a) and 5.4(b). The resultant heading vector (\tilde{h}_j) is calculated as:

$$\tilde{h}_j = e^{i\theta'_j} + \eta e^{i\xi_j} \quad (5.1)$$

θ'_j is the heading measurement from the j^{th} VHS neighbor with respect to the body fixed-reference frame of the robot given in Equation 4.1. η is the magnitude of the noise vector, and ξ_j is the direction of the noise vector determined by either a Gaussian or uniform distribution.

In the Gaussian distribution case, ξ is a normally-distributed variable having a mean of [collision_detection](#). Last visited: April 2008.

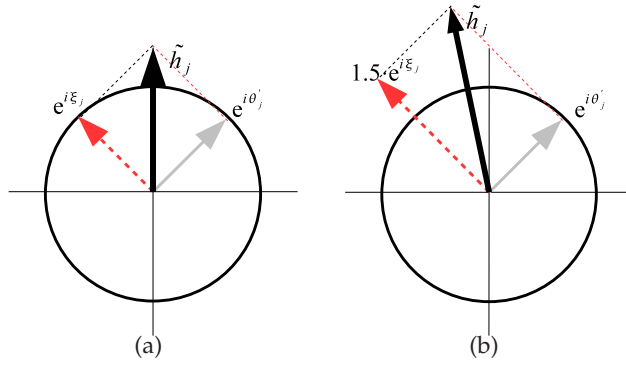


Figure 5.4: Vectorial noise model. The heading measurement vector ($e^{i\theta'_j}$) is denoted by a continuous arrow. The noise vector ($e^{i\xi_j}$) is shown by a dashed arrow and the resultant vector (\tilde{h}_j) is denoted by a bold continuous arrow. (a) $\eta = 1$. (b) $\eta = 1.5$. Adapted from [4].

θ'_j and a standard deviation of either $\pi/2$ or π , whereas, in the uniform distribution case, ξ is randomly selected from $[-\pi, \pi]$. The two cases are denoted by $N(\theta'_j, \pi/2)$ and $N(\theta'_j, \pi)$, respectively. The case of $N(0, \pi/2)$ is plotted in Figure 5.5.

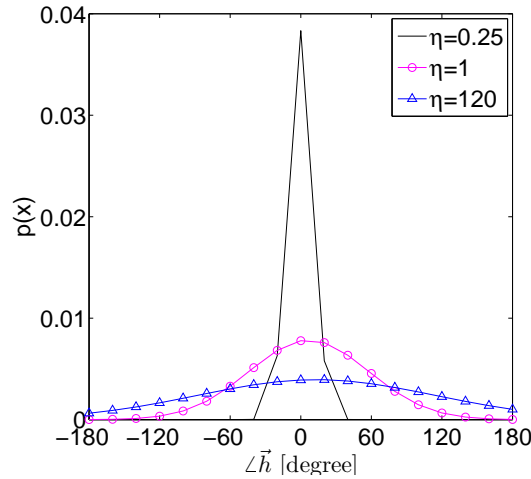


Figure 5.5: Histogram of simulated noisy measurements of VHS for $\theta'_j = 0^\circ$. The simulations are performed for 10000 steps for three different η values.

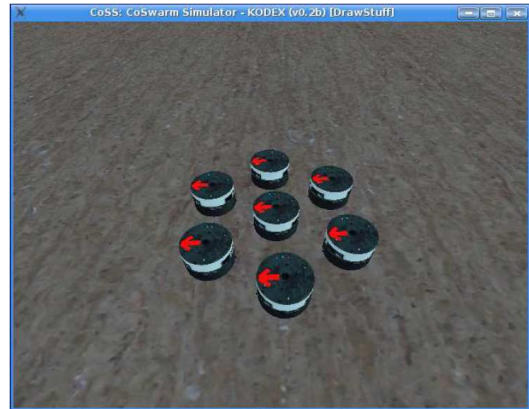
5.3 The Co-Swarm Simulator

A physics-based simulator, called the Co-Swarm Simulator (CoSS), is used in computer simulations. The CoSS is built on top of the Open Dynamics Engine (ODE) which enables

modeling of complex physical interactions like the collision of bodies and slippage in the wheels.



(a)



(b)

Figure 5.6: (a) A photo of seven robots in CoSS. (b) A snapshot of seven robots.

The main body and wheels of the robot are modeled using basic cylindrical collision geometries. DC motors are simulated using virtual motorized hinge joints, and the virtual weights of the components are adjusted to obtain a similar movement pattern with similar motor torques. The IR short-range sensing system and the VHS are modeled utilizing the results of systematic experiments. Since the characteristics of the short-range sensing system are determined using kin-robots (in passive/active state), they are not valid for objects like walls or irregularly-shaped obstacles in the environment. Therefore, for the time being, CoSS is only utilized to model the flocking behavior in free space. A photo of seven robots and its CoSS counterpart is shown in Figure 5.6

CHAPTER 6

THE FLOCKING BEHAVIOR

"... thousands of tree swallows gathering for flight: an order held in constant change: a congregation rich with entropy: nevertheless, separable, noticeable as one event, not chaos ..."

— A. R. Ammons

In this chapter, we propose a self-organized flocking behavior based on two basic behaviors. By self-organized flocking, we mean that a group of robots, initially connected but not necessarily aligned, should be able to wander in an environment by moving as a coherent group in free space and avoiding any obstacles, as if it were a "super-organism". Different from other studies, the flocking behavior is regarded as a fully decentralized and scalable coordination method which does not utilize any designated or elected leader within the group.

We then define several metrics such as order and entropy to measure the performance of flocking qualitatively. Following that, we perform full-fledged flocking in a constrained space using nine Kobots to demonstrate how the flocking behavior works. We then perform flocking in open space using seven robots in CoSS and seven Kobots, and verify the CoSS simulations against the Kobot experiments using order and entropy metrics. Finally, we utilize 1000 robots in CoSS to evaluate the flocking behavior with a large group in open space.

6.1 The Flocking Behavior

The flocking behavior consists of heading-alignment and proximal-control behaviors, combined in a weighted vector sum:

$$\mathbf{a} = \alpha \mathbf{h} + \beta \mathbf{p} \quad (6.1)$$

where \mathbf{h} is the normalized heading-alignment vector having a weight of α , \mathbf{p} is the normalized proximal-control vector having a weight of β , and \mathbf{a} is the desired acceleration vector.

6.1.1 Heading Alignment

Heading alignment behavior aims to align the heading of a robot to the average heading of its VHS neighbors as illustrated in Figure 6.1. The virtual heading sensor is utilized in receiving the heading values of the VHS neighbors. The alignment vector (\tilde{h}) is calculated by vectorially summing up Equation 5.1 for all the heading values of the VHS neighbors of a robot as:

$$\tilde{h} = \sum_{j \in \mathcal{N}_R} \tilde{h}_j \quad (6.2)$$

where \mathcal{N}_R denotes the VHS neighbors of a robot when the communication range of the VHS is set to R .

The normalized heading-alignment vector (\mathbf{h}) is calculated as:

$$\mathbf{h} = \frac{\tilde{h}}{\|\tilde{h}\|}$$

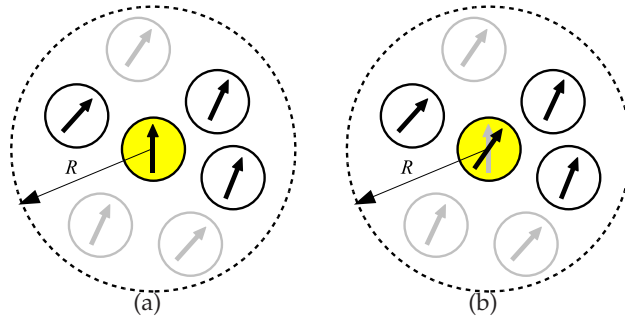


Figure 6.1: The heading alignment. The central robot aligns its heading to the average of the three randomly selected VHS neighbors drawn in black. The other robots are drawn in gray. The communication range is set to R and \mathcal{N}_R is set to 3. (a) $t = 0$. (b) $t = 1$.

6.1.2 Proximal Control

Proximal-control behavior uses readings obtained from the IR sensing system to (1) avoid collisions with other robots and obstacles and (2) maintain cohesion between the robots.

For each IR sensor, a virtual force proportional to the square of the deviation of the measured distance from the desired distance is assumed. The desired distance is taken as an appropriate finite value for kin-robots, and ∞ for obstacles, which forces the robot to maintain a desired distance from kin-robots while moving away from obstacles. The virtual force from the k^{th} sensor is calculated as:

$$f_k = \begin{cases} -\frac{(d_k - d_{des})^2}{C} & \text{if } d_k \geq d_{des} \\ \frac{(d_k - d_{des})^2}{C} & \text{otherwise} \end{cases}$$

where d_{des} , the sensor measurement corresponding to the desired distance, is half of the sensor range ($d_{des} = 3$) for robots and 0 for obstacles, since IR sensors return a reading of 0 for objects at infinity. C is a scaling constant equal to 10 for robots and 35 for obstacles, and is used to normalize f_k . f_k is less than 0 in the case of a repulsive force, and greater than 0 in the case of an attractive one. The sense of the forces is defined with respect to the robot.

The value of the force, f_k , is plotted in Figure 6.2 for both robot- and obstacle-detection cases. The d_{des} value is also indicated for both cases.

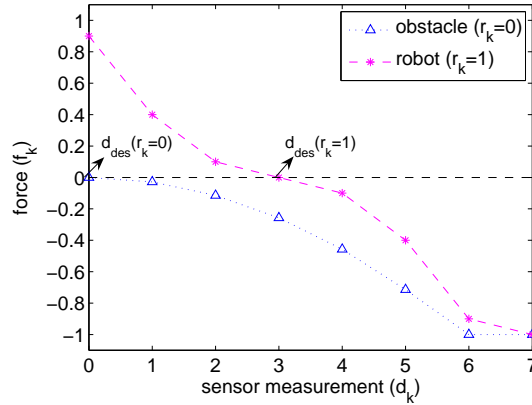


Figure 6.2: The virtual force (f_k) is plotted as a function of $d_k \in \{0, 1, \dots, 7\}$, where $d_k = 1$ for a distant and $d_k = 7$ for a nearby object.

The calculation of the normalized proximal control vector, \mathbf{p} , is as follows:

$$\mathbf{p} = \frac{1}{8} \sum_{k=0}^7 f_k e^{i\phi_k}$$

where k denotes the k^{th} sensor having an angle of $\phi_k = \frac{\pi}{4}k$ with the x -axis shown in Figure 4.8(a) of the body-fixed reference frame shown in Figure 6.3. $\frac{1}{8}$ is the normalization constant.

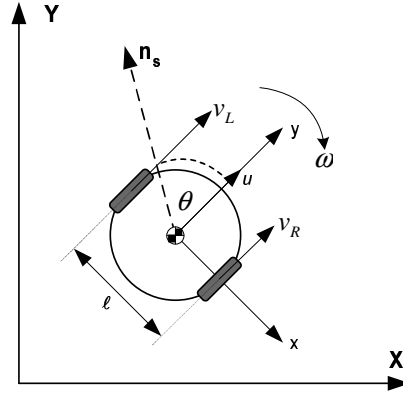


Figure 6.3: The reference frame is fixed to the center of the robot where the x -axis coincides with the rotation axis of the wheels. The forward velocity (u) is parallel to the y -axis. ω denotes the angular velocity of the robot. v_R and v_L denote the velocity of the right and left motors, respectively. The y -axis of the body-fixed reference frame makes an angle of θ (current heading) with the sensed North direction (n_s) at the instant the figure is drawn. l is the distance between the wheels.

6.1.3 Motion Control

The acceleration vector (\mathbf{a}) of the robot is mapped to forward velocity (u) and angular velocity (ω). u is calculated as:

$$u = \begin{cases} (\frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \hat{\mathbf{a}}_c)^\gamma u_{max} & \text{if } \frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \hat{\mathbf{a}}_c \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.3)$$

where $\hat{\mathbf{a}}_c$ is the current direction of the robot that is coincident with the y -axis of the body-fixed reference frame, and equal to \mathbf{j} . γ is the modulation parameter.

When the modulation parameter γ is set to 1, the forward velocity of the robot (u) is modulated depending on the "urge"-to-turn condition as shown in Figure 6.4. u is computed by calculating the dot product of the desired direction (\mathbf{a}) and the current direction

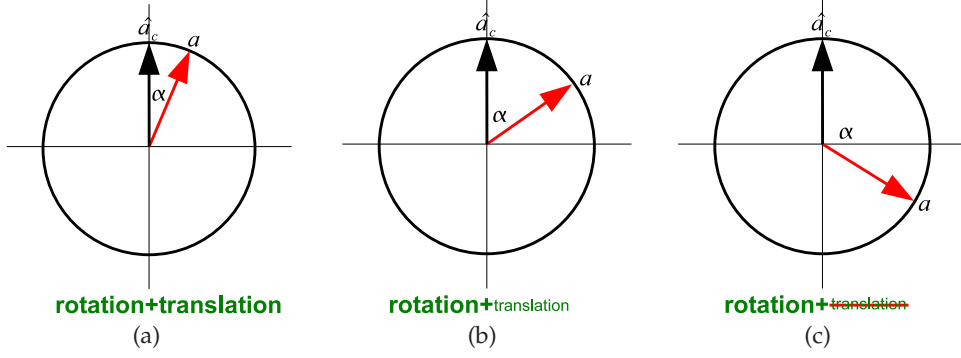


Figure 6.4: Modulation of the forward velocity (u) when $\gamma = 1$. (a) The robot makes rotation and translation. $\alpha \approx 0^\circ \Rightarrow u \approx u_{max}$. (b) The robot makes mostly rotation. $\alpha \approx 90^\circ \Rightarrow u \approx 0$. (c) The robot only makes rotation. $\alpha > 90^\circ \Rightarrow u = 0$.

(\hat{a}_c) vectors as in Equation 6.3. When the urge to turn is low meaning that the robot is already moving in the desired direction, the forward velocity is allowed to achieve its maximum value (u_{max}) as shown in Figure 6.4(a). Conversely, when the urge to turn is high, u decreases, converging to 0 in the extreme case, where the robot only rotates with respect to its center of mass as illustrated in Figure 6.4(b). And when the dot product of the desired direction of motion (a) and the current direction (\hat{a}_c) of the robot is negative, this indicates that the angle between the two vectors is greater than 90° in absolute value as shown in Figure 6.4(c). By setting $u = 0$ in this case, we constrain the robot's motion to rotation only, instead of assigning a negative velocity. Failure to do so would result in robots moving backwards, a situation that would complicate the behavior and its analysis.

The angular velocity (ω) of the robot is determined using a proportional controller by calculating the deviation of the desired direction (a) from the current direction (\hat{a}_c) of the robot.

$$\omega = (\angle a - \angle \hat{a}_c) K_p \quad (6.4)$$

where K_p is the proportional gain of the controller. $\angle(\cdot)$ computes the argument of the related vectors and $\angle \hat{a}_c = \frac{\pi}{2}$.

The rotational speeds of the right (N_R) and the left (N_L) motors as shown in Figure 6.3 are eventually calculated as follows:

$$N_R = \left(u - \frac{\omega l}{2} \right) \frac{60}{2\pi r}$$

$$N_L = \left(u + \frac{\omega}{2}l\right) \frac{60}{2\pi r}$$

where N_R and N_L are the rotational speeds [rpm] of the right and left motors, respectively, l is the distance between the wheels of the robot [m], u is the forward velocity [m/s] and w is the angular velocity [rad/s] of the robot.

6.2 Metrics of Flocking Behavior

In *desirable* flocking behavior, individuals having an ordered formation are supposed to move cohesively in a common direction with minimal energy consumption. It is required to define some measures to quantify *desirable* flocking behavior in mathematical terms. Hence, we define a number of metrics which will help us in two ways. Firstly, they will serve as objective measures and quantify the performance of the flocking behavior. Secondly, they will be utilized in comparing the performance of different behaviors achieved through setting controller parameters or sensing characteristics to different values than the default ones.

Order, entropy and average angular velocity metrics are defined to measure the alignment, positional order and energy consumption of the group, respectively. The average forward velocity metric is also utilized as a secondary measure of the energy consumption, and is more convenient to use in some cases. The success rate metric is defined to indicate the frequency of successful experiments.

Order (ψ) measures the angular order of the robots [2].

$$\psi(t) = \frac{1}{M} \left| \sum_{k=1}^M e^{i\theta_k} \right| \quad (6.5)$$

where M is the number of robots in the group and θ_k is the heading of the k^{th} robot at time t .

Order can have a value between 0 and 1 and is calculated by collecting the heading value broadcasted by the robots at each time-step. When the group is aligned, and hence, the system is in an *ordered* state, the order parameter approaches to 1. When the group is unaligned, the system is in a *disordered* state and the order parameter is close to 0. In *desirable* flocking behavior, robots are supposed to be in an *ordered* state, having a high-order value.

The steady-state value of order denoted by $\bar{\psi}$ is calculated by taking the average of Equation 6.5 as:

$$\bar{\psi} = \langle \psi(t) \rangle_t \quad (6.6)$$

Entropy (S) measures the positional disorder of the group [60]. This metric is calculated by finding every possible cluster combination, finding Shannon's information entropy of these clusters and then sum them up. Different clusters are found out by changing the maximum distance (h) between individuals from 0 to ∞ to consider them to be in the same cluster. Two robots i and j are considered to be in the same cluster, if and only if $\|\vec{r}_i - \vec{r}_j\| \leq h$, where \vec{r}_i and \vec{r}_j denote the position vectors of the i^{th} and j^{th} robots, respectively, and $\|\cdot\|$ calculates the Euclidean norm. Shannon's information entropy ($H(h)$) of a cluster having a maximum distance of h is calculated as:

$$H(h) = \sum_{k=1}^K p_k \log_2(p_k) \quad (6.7)$$

where p_k is the proportion of individuals in the k^{th} cluster, and K is the number of clusters for a given h .

These entropy values are integrated over all possible h 's ranging from 0 to ∞ in order to find the total entropy (S) of the distribution:

$$S = \int_0^{\infty} H(h) dh \quad (6.8)$$

When the absolute value or time evolution of the entropy is of no concern, we utilize the rate of change of the entropy (dS/dt) in our analysis. Figure 6.5 shows four possible configurations that a seven-robot flock can attain. In configuration (a), individuals are placed linearly (the least-desired configuration), having the highest entropy value. Among configurations (b) and (c), the former has a larger entropy value since the latter is more ordered. Configuration (d) has the smallest entropy value since it is the most-ordered configuration for a seven-robot flock. In *desirable* flocking behavior, the flock is required to have a high positional order, and hence, a low entropy (S) value.

Entropy is calculated using relative positions of the robots, without the need for a high-quality stationary overhead camera. A mobile overhead camera is utilized to calculate entropy using the relative positions of the robots. The relative positions are determined by utilizing an off-line tracking algorithm using the OpenCV computer vision library.

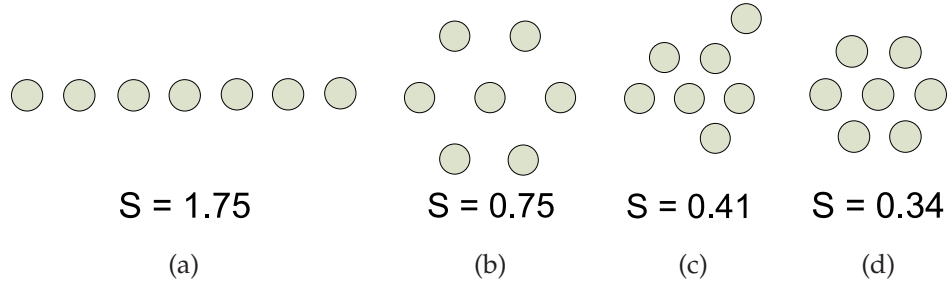


Figure 6.5: Entropy values for four different configurations of seven robots. Entropy decreases as the configuration loses its positional order.

Average angular velocity (ω_{rms}) of the flock is the amount of unnecessary energy spent [41] due to the rotational movement of each individual. It is calculated by taking the average of the root-mean-square (rms) of the angular velocity of each individual over the entire operation time (t).

$$\omega_{rms} = \frac{1}{M} \sum_{i=1}^M \sqrt{\langle \omega_i^2 \rangle_t}$$

where M is the number of robots in the group and ω_i is the angular velocity of robot i .

In the experiments, average angular velocity is calculated using the angular velocity values broadcasted by each robot at each control time-step. ω_{rms} should ideally be 0 in *desirable* flocking behavior to minimize energy consumption.

Average forward velocity (V_G) is the average velocity of the geometric center of the flock, calculated by dividing the distance traveled in the forward direction by the operation time. The average velocity of a flock with non-rotating (having less tendency to rotate) robots is high, meaning that the flock will reach a destination point in a shorter time. In *desirable* flocking behavior, higher values of average forward velocity are preferred.

Success rate (SR) denotes the ratio of successful runs to the total number of experiments. A run is considered a failure when robots collide with each other or get stuck, and the cohesiveness of the group is lost.

6.3 Full-fledged Flocking in Constrained Space

The flocking behavior is implemented on Kobots using the default parameter settings given in Table 6.1. The experiment is conducted with seven Kobots in a constrained environment having an infinite wall in the path of the robots. In almost all of the trials, Kobots successfully adopted a common heading and avoided the wall in their way. It is observed that

Table 6.1: The default settings for the controller and VHS parameters.

	Parameter	Default Value
Controller	weight of heading alignment (α)	0.125
	weight of proximal control (β)	0.5
	modulation parameter for forward velocity (γ)	1
	maximum forward speed of a robot (u_{max})	7 cm/s
	proportionality constant of angular velocity (K_p)	0.5
	optimal IR measurement for kin robots (d_{des})	3
VHS	range (R)	20 m
	number of VHS neighbors (N_c)	20
	nature of noise (ξ_o)	$N(\mu = \theta'_r, \sigma = \pm \frac{\pi}{2})$

flocking behavior results in three distinct phases of operation: (1) The *alignment* phase, in which randomly oriented robots align their headings, keeping proximity with the other robots at the desired value, resulting in a group possessing high positional and angular order. Robots mostly perform rotations and hardly advance in this phase; (2) The *advance* phase, in which an aligned and positionally-ordered group advances. Robots in this phase mainly move forward at the maximum speed, keeping a common alignment and proximity with other robots. (3) The *avoidance* phase, in which an advancing group faces a wall. Robots mainly perform rotations to get past the wall and attain a common heading.

Figure 6.6 shows a sample run, in which seven Kobots positioned in random orientations are allowed to move in an environment having a wall. In the experiment, robots are allowed to communicate with six VHS neighbors and the VHS is only subject to environmental noise; no additional noise is introduced. The following controller parameter set is utilized: $\alpha = 0.125$, $\beta = 1$, $\gamma = 1$, $K_p = 0.5$ and $u_{max} = 7$ cm/s.

The robots begin in the *alignment* phase ($t = 0$ s) in which they try to align their heading with their VHS neighbors'. At the end of this phase, the robots are aligned, and have positional order, which marks the beginning of the *advance* phase ($t = 6$ s). When the flock arrives at the wall, the *avoidance* phase starts ($t = 16$ s). The robots in front sense the presence of the wall, come almost to a halt and rotate to bounce off the wall. The other robots try to avoid the robots in front of them by decreasing their speeds and performing rotations. The robots, after avoiding the wall, once more adopt a common alignment and return to the *advance* phase ($t = 20$ s). The same cycle is repeated and the robots bounce off two additional walls until the end of the experiment.

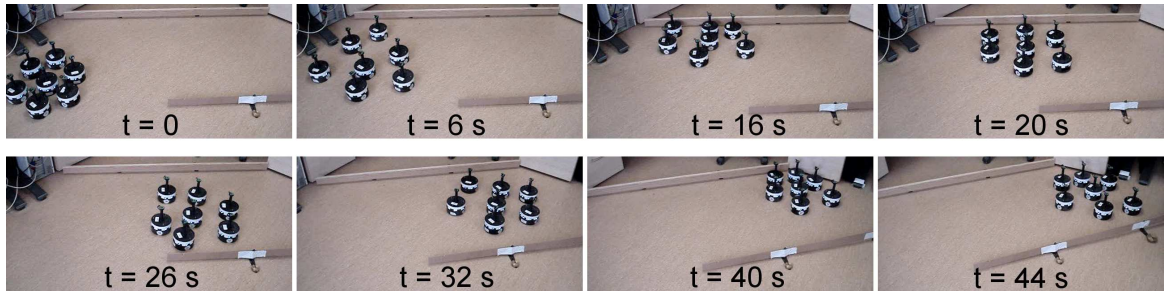


Figure 6.6: Self-organized flocking with seven Kobots. Starting from a connected but unaligned state, Kobots negotiate a common heading and move as a group in free space, bouncing off a wall without losing their cohesion.

6.4 Full-fledged Flocking in Open Space

We performed full-fledged flocking experiments in open space to verify CoSS against Kobots. The experiment was conducted using seven Kobots and seven robots in CoSS, and the robots were placed in a regular hexagonal formation with random orientation. In the experiments, robots first aligned their headings, which increased the order of the group to approximately 1 shown in Figure 6.7(a); then they started to advance. During the experiment, the entropy of the group did not change considerably, which shows that the positional order of the group is maintained throughout the entire operating time as shown in Figure 6.7(b).

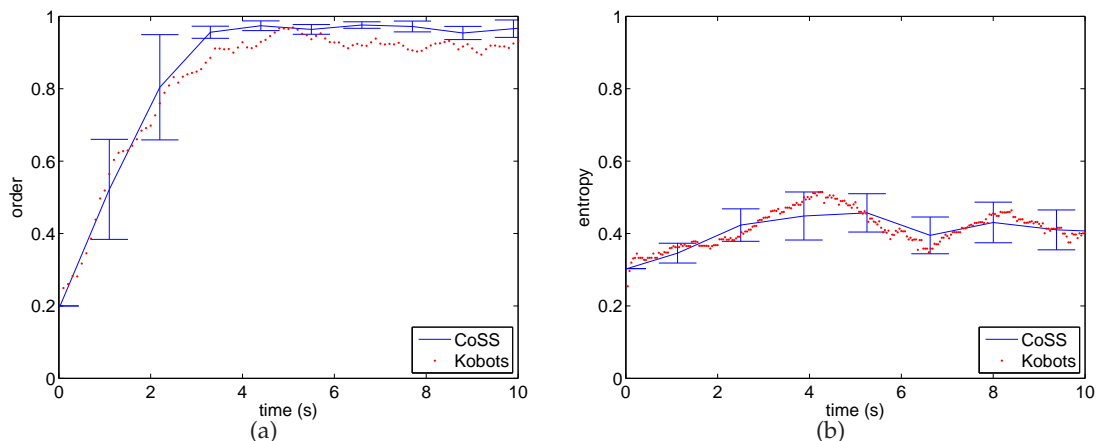


Figure 6.7: Self-organized flocking in open space with seven Kobots and seven robots in CoSS. (a) Plot of order. (b) Plot of entropy. Five simulations are performed and the mean values are plotted together with error-bars indicating \pm one standard deviation of the value from the mean.

We note that the results obtained from Kobots and CoSS are similar for both the transient and steady-state phases of the behavior shown in Figures 6.7(a) and 6.7(b) verifying that CoSS provides a realistic simulation of flocking in open environments for Kobots. Hence, it can be judged that CoSS may safely be utilized to simulate the flocking behavior. CoSS will be utilized especially when experiments are to be performed with a large group of robots.

6.5 Full-fledged Flocking with a Large Group in Open Space

In order to demonstrate the performance of the flocking behavior with a large group, we utilized 1000 robots in CoSS, as shown in Figure 6.8. We set the controller parameters as: $\alpha = 0.125$, $\beta = 2$, $u_{max} = 7$ cm/s and $K_p = 0.5$, and the VHS parameters as: $R = 20$ m, $\eta = 3$ and $N = 10$ in the experiment. These parameters are regarded as the default settings of the flocking behavior and under these settings we obtained robust and scalable flocking of the group. However, in the following chapters, in order to investigate the robustness and the performance of the flocking behavior, we will evaluate it under different controller parameters and VHS characteristics.

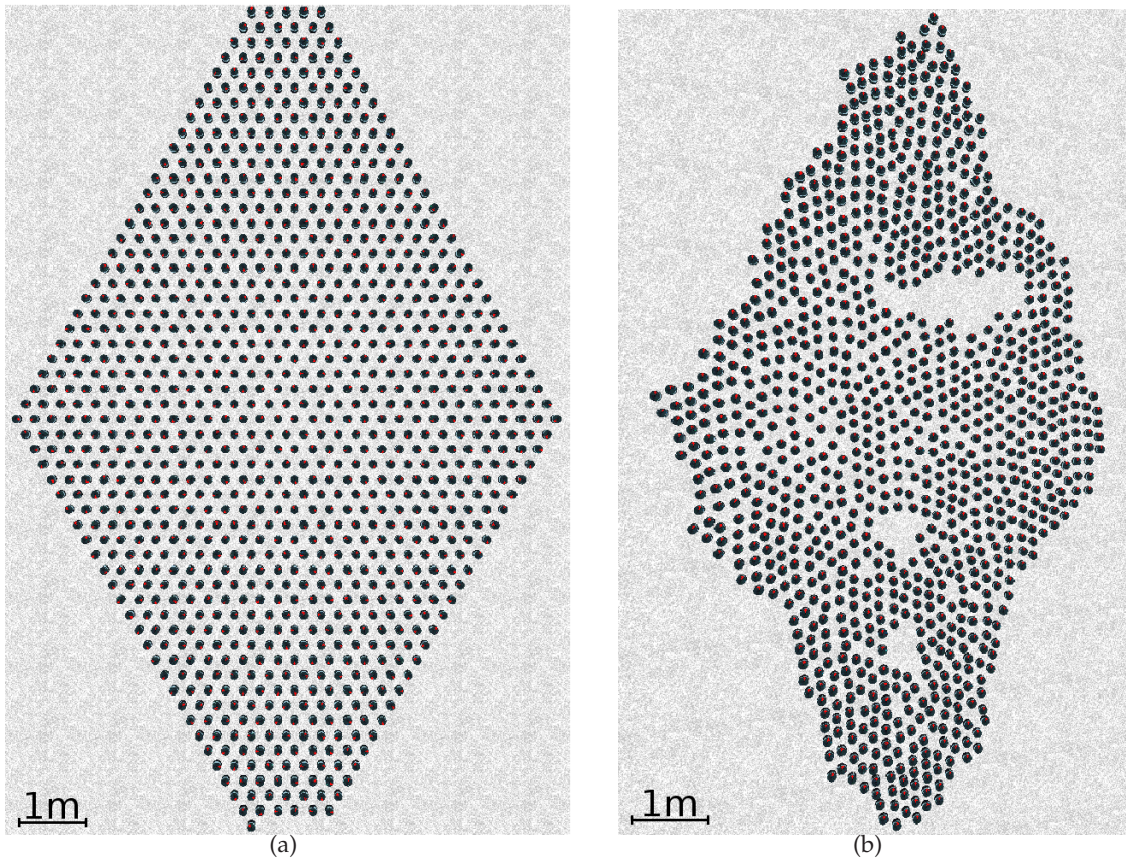


Figure 6.8: Full-fledged flocking with a large group. (a) The screenshot at the beginning of the simulation. (b) The screenshot when $t = 2000$ s. The total displacement of the flock is approximately 110 m.

CHAPTER 7

ANALYSIS OF VARIANT BEHAVIORS AND OPTIMIZATION OF THE CONTROLLER PARAMETERS

In this chapter, we first set the weight of the heading vector (α), weight of the proximal control vector (β) and the modulation parameter (γ) to their extreme values to generate several variants of the flocking behavior and analyze them in *advance*, *alignment* and *avoidance* phases using seven Kobots. We also perform *advance*-phase experiments using CoSS for the sake of comparison.

We then optimize the weight of the proximal control vector, the maximum forward velocity and the proportional for an aligned, cohesive and rapid flocking motion using 100 robots in CoSS and seven Kobots.

7.1 Analysis of Variant Behaviors

Specifically, in order to understand the effects of the different components of flocking, we created four variants of flocking by setting some of the controller parameters to their extreme values. In this way, we gained crude knowledge of the performance of the flocking behavior in extreme cases.

The variants were created by setting the weight of the heading vector (α), the weight of proximal-control vector (β), and the modulation parameter (γ) to their extreme values. In this way, we attained four different variants as listed below:

- *Proximal Control with Constant Forward Velocity (P_{const})*

In this variant, only proximal control is performed and the forward velocity is not

modulated. α is set to 0, β to 1, and γ to 0.

- *Proximal Control with Modulated Forward Velocity (P_{mod})*

In this variant, the forward velocity is modulated with γ set to 1. α is again 0 and β is 1, as in P_{const} .

- *Heading Alignment and Proximal Control with Constant Forward Velocity (HP_{const})* This variant adds a heading alignment term. α is set to 0.125 and β is set to 1. γ is 0 as in P_{const} , which means that the forward velocity is not modulated.

- *Heading Alignment and Proximal Control with Modulated Forward Velocity (HP_{mod})* In this variant, α is set to 0.125, β is set to 1 and the forward velocity is modulated with γ set to 1.

The variants of the flocking behavior were evaluated in three phases. The *advance-* and *alignment-*phase experiments were conducted in free space, while the *avoidance-*phase experiments were conducted in an environment that had a wall blocking the path of the robots. The experiments were conducted using seven Kobots and were repeated ten times. The *advance-*phase experiment was also conducted in CoSS with seven robots and was repeated ten times. In all of the experiments, the robots were initially put into hexagonal formation having 25 cm spacing between their centers to ensure that each robot was in the IR sensing range of at least one other robot. The experiments were conducted with the default parameter set, as shown in Table 6.1, unless otherwise stated.

7.1.1 The Advance-Phase

We started evaluating the variant behaviors in advance-phase, since this is the only phase in which we can compare all of the variant behaviors together. In this phase, the robots were aligned and moving in a common direction at a relatively high speed while maintaining the cohesion of the group.

In the experiments, the robots were initially aligned and then left to move for 15 s. The performance of P_{const} , P_{mod} , HP_{const} and HP_{mod} behaviors were evaluated using order, entropy, average forward velocity and average angular velocity metrics.

The steady-state value (the converged value at $t = 15$ s) of each metric attained in ten experiments are presented as a box-and-whisker plot, showing the results of the CoSS and Kobot experiments next to each other. The order and entropy values are plotted in Figures 7.1(a) and (b). The normalized average forward velocity and normalized average

angular velocity values are depicted in Figures 7.1(c) and (d), respectively. The normalization of the average forward velocity and average angular velocity were performed using the maximum attainable values for the average forward velocity and average angular velocity as 0.07m/s and 0.84rad/s, respectively. The snapshots of sample runs are shown in Figure 7.1(e).

P_{const} behavior results in a disordered state having a low-order value (~ 0.2) as seen in Figure 7.1(a). The reason for this is the lack of heading-alignment behavior, which makes it impossible for the robots to preserve their common heading. The cohesiveness of the group is also not preserved, as is evident from the high entropy values (~ 1) seen in Figure 7.1(b); hence, each robot moves independently unless they collide with each other and get stuck, as seen in Figure 7.1(e). This is due to the fact that the robots cannot modulate their forward speed, which stays at the maximum value (u_{max}) at all times. Since the cohesiveness of the group is lost, it is not logical to say anything about the average forward velocity of the group. The noisy proximal measurements due to the IR sensors cause the robots to perform extensive rotations resulting in high average angular velocity values (~ 1), as shown in Figure 7.1(d).

The second variant, P_{mod} , presents quite a different behavior. The robots, aiming only to control their inter-distance and able to modulate their forward velocity, can indeed remain as a group as shown in Figure 7.1(e), as is evident with a considerably lower entropy value (~ 0.4) than the previous variant, seen in Figure 7.1(b). In order to preserve cohesion, the robots perform so many rotations due to the noisy IR sensors that their forward speeds approach to zero, resulting in a cohesive but stationary group, having approximately zero average forward velocity as shown in Figure 7.1(c) and high average angular velocity values depicted in Figure 7.1(d). However, the robots can still neither preserve their common heading nor adopt a new heading due to the absence of alignment behavior as shown in Figure 7.1(a).

The HP_{const} and HP_{mod} variants perform significantly better in all metrics with the help of the heading-alignment behavior. In both cases, robots are capable of maintaining their alignment perfectly, as indicated by an order value of 1, and keeping their cohesiveness, attaining a low entropy value (~ 0.4), as seen in Figures 7.1(a) and (b), respectively. Alignment behavior balances the effect of proximal-control behavior so that robots perform rotations less frequently, resulting in low average angular velocity values (~ 0.4), as depicted in Figure 7.1(d), and quite high average forward velocity values (~ 0.9) as shown in Figure 7.1(c). It can also be observed in Figure 7.1(e) that the two variants performed

successfully in the *advance*-phase experiment.

Only the HP_{const} and HP_{mod} behaviors can be regarded as successful in the *advance*-phase, since these are the variants in which the robots were able to move forward maintaining a common heading and keep the cohesiveness of the group. Therefore, we will evaluate only the HP_{const} and HP_{mod} variants for their alignment and avoidance performance to determine the best among the two.

In the above discussions, we have not addressed the results of CoSS and Kobot experiments explicitly, since it is evident from the figures that the results are in close agreement, both qualitatively and quantitatively.

7.1.2 The Alignment-Phase

The *alignment*-phase is the phase in which robots are unaligned and try to assume a common orientation by performing rotations most of the time. Meanwhile, the robots have to avoid collisions with their neighbors in close proximity. This experiment is expected to accentuate the necessity of velocity modulation, existing only in HP_{mod} among the two variants that successfully completed the *advance*-phase experiment.

In the experiments, robots were placed with random orientations and left to operate for 5 s, which is enough time to adopt a common heading, if they could. Order and entropy metrics were utilized to measure the performance of the HP_{const} and HP_{mod} variants.

Figures 7.2(a) and 7.2(b) depict the time evolution of the order and entropy values of two sample runs, respectively. The snapshots of these runs are shown in Figure 7.2(c).

In the HP_{mod} behavior, robots attained a common orientation in approximately 4s resulting in a high-order value (~ 1), as shown in Figure 7.2(a). A sudden increase was observed in entropy, indicating that the flock expanded while the robots were attaining a common heading. As soon as order reached a high value (~ 0.8), entropy decreased and settled to a relatively low value (~ 0.5) as shown in Figure 7.2(b). Out of ten experiments, HP_{mod} behavior attained a success rate of 100%. A snapshot of one of the runs is shown in Figure 7.2(c).

Things changed in the case of the HP_{const} variant. The robots were not able to attain a common orientation, which kept the order low (~ 0.4), as shown in Figure 7.2(a). This was mainly due to the fact that the robots were not able to modulate their forward speed; hence, they collided with each other and got stuck. These colliding robots formed independent stationary or moving clusters, increasing entropy continuously, as shown in Figure 7.2(b),

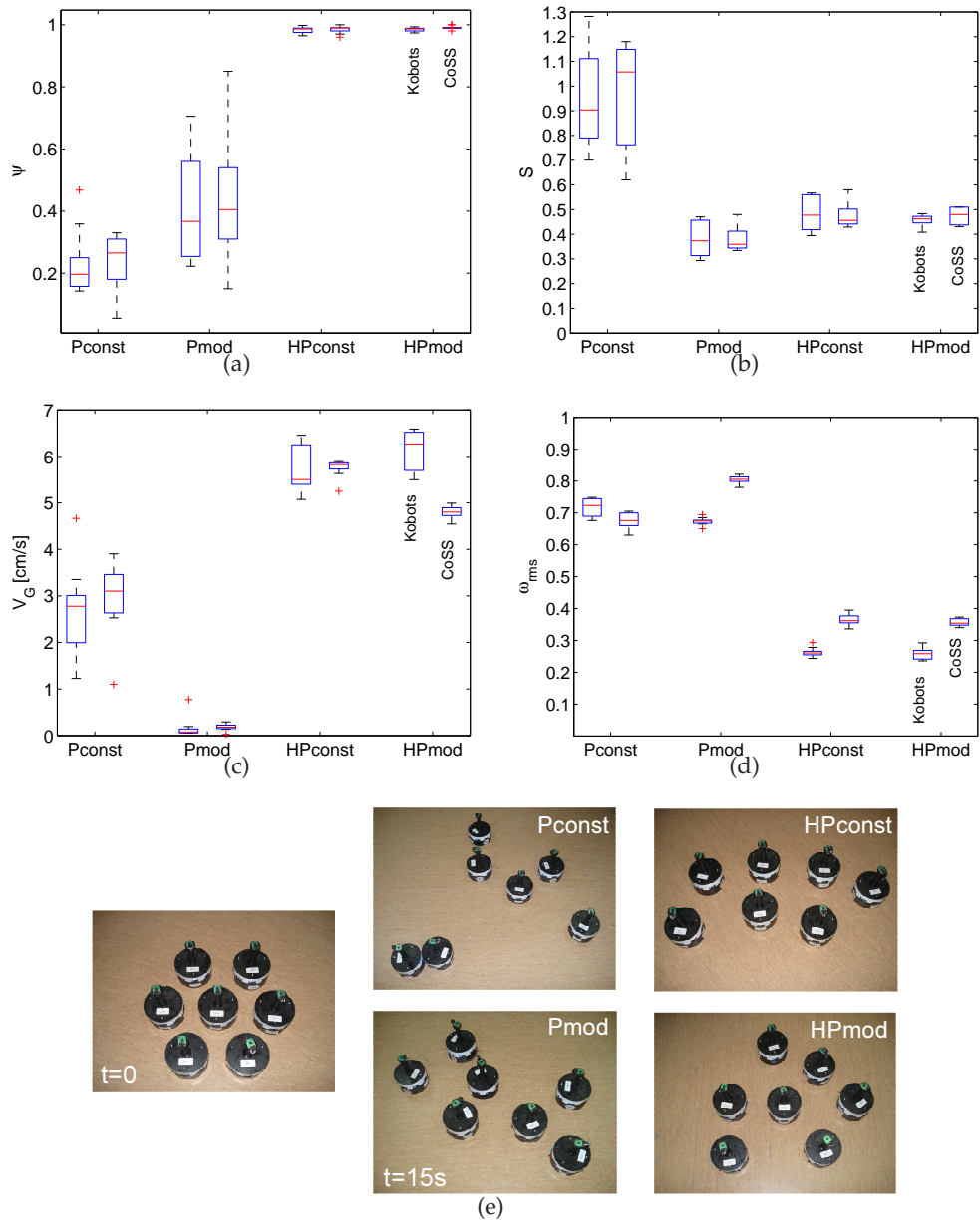


Figure 7.1: *Advance*-phase experiments. (a) Plot of order. (b) Plot of entropy. (c) Plot of normalized average forward velocity. (d) Plot of normalized average angular velocity. (e) Snapshots of the initial and final configurations of Kobots. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by a plus sign.

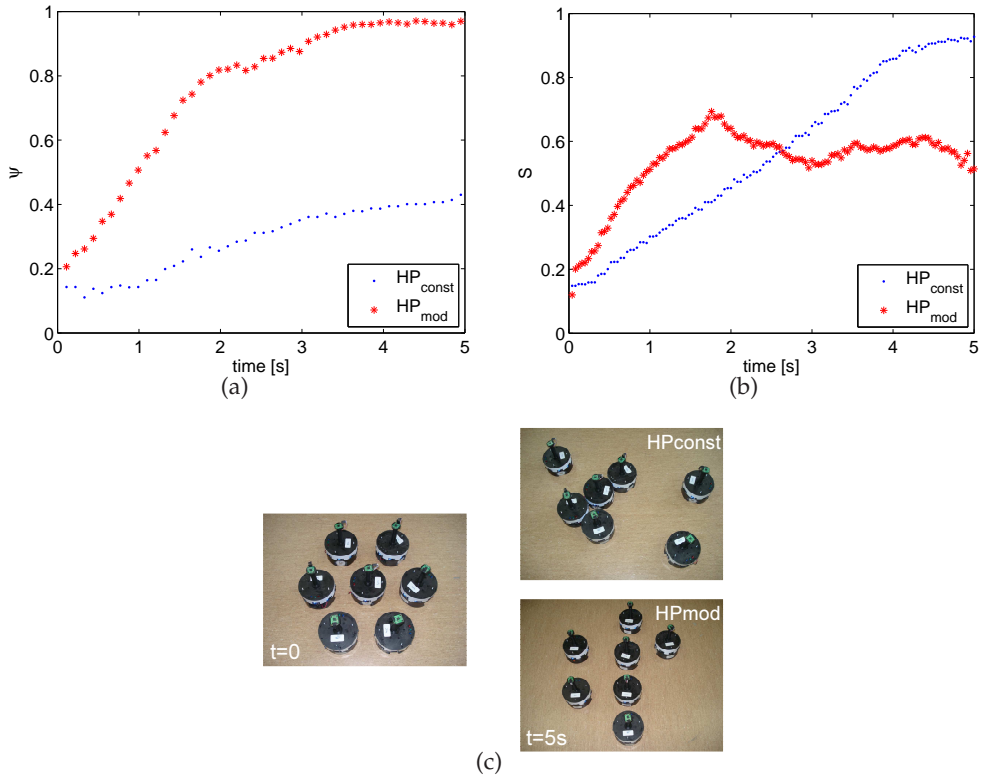


Figure 7.2: The results of the *alignment*-phase experiment. (a) Time evolution of order. (b) Time evolution of entropy. (c) Snapshots of the initial and final configurations of Kobots.

and preventing the flock from attaining a common heading. An independently moving and a stationary group of robots can be observed in Figure 7.2(c). Out of ten experiments, HP_{const} failed in all of them, resulting in a success rate of 0%.

The *alignment*-phase experiments clearly revealed that the HP_{mod} variant is superior to the HP_{const} variant due to the fact that the forward speed of the robots is modulated in HP_{mod} . Modulation has crucial importance in avoiding robots or obstacles, which will become clearer in the *avoidance*-phase experiment.

7.1.3 The Avoidance-Phase

In the *avoidance* phase, robots having a common orientation and moving cohesively face an infinite wall in their path. They bounce off that wall, retaining their alignment and cohesiveness. At the same time, the robots avoid collisions with each other. These experiments are also expected to demonstrate the importance of modulation in velocity as in the *alignment*-phase experiment.

In the avoidance-phase experiments, robots were initially placed against an infinite wall and had a common orientation. They were left to move for 15 s, during which they faced

a wall, and tried to bounce off and move away from it. Order and entropy metrics were utilized during the analysis of the HP_{const} and HP_{mod} variants.

Figures 7.3(a) and 7.3(b) present the time evolution of the order and entropy values of a sample run for the two behaviors, respectively. The snapshots of these runs are shown in Figure 7.3(c).

In the case of the HP_{mod} variant, the robots preserved their alignment before and after they encountered with the wall, keeping the order high (~ 1), as depicted in Figure 7.3(a). The entropy graph shown in Figure 7.3(b) has a hump. Entropy first increases when the robots encounter the wall, and then returns to its initial value, which indicates that the group remains cohesive after bouncing off the wall. The HP_{mod} variant performed successfully in all of the ten runs, attaining a 100% success rate. One of these runs is shown in Figure 7.3(c).

The performance of the HP_{const} variant is similar to its performance in the *alignment-phase* experiment. Due to lack of modulation of forward velocity, the robots get stuck at the wall or collide with each other as shown in Figure 7.3(c), preventing the group from preserving its order as seen in Figure 7.3(a) and staying cohesive, which results in a continuous increase of entropy as shown in Figure 7.3(b). Out of ten runs, the HP_{const} variant had a success rate of 0%.

The results of the experiments in this phase revealed that modulation of forward velocity is a must when avoidance is concerned and this ability is only present in the HP_{mod} variant.

The experiments reveal that the HP_{mod} variant outperforms the other behaviors and satisfies the requirements of *desirable* flocking behavior. Its success depends on two factors. First is the heading alignment, which adjusts the orientation of the robots to a common value. Second is modulation of forward velocity, which prevents collisions among the robots.

7.2 Optimization of the Controller Parameters

Having analyzed the variants of the flocking behavior and decided on the HP_{mod} variant, in this section we optimize the controller parameters for an aligned, cohesive and rapid flocking motion.

In the experiments, the weight of the proximal-control vector (β), maximum forward speed (u_{max}) and proportional gain (K_p) parameters were set to various values and their

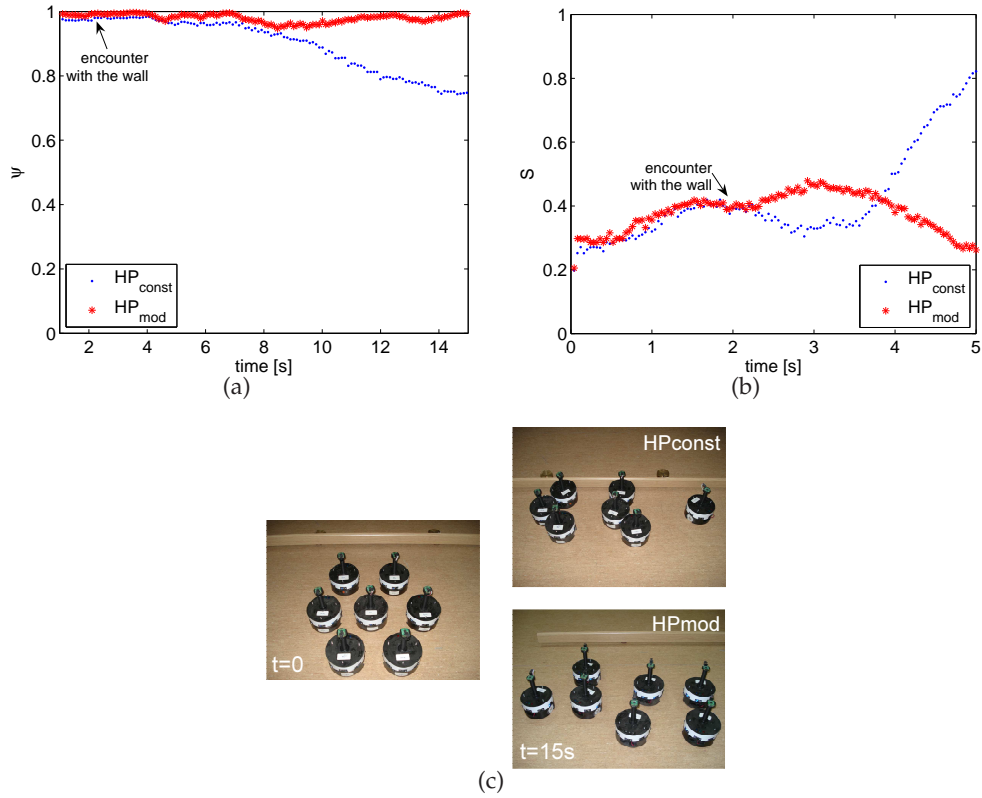


Figure 7.3: The *avoidance*-phase experiment. (a) Plot of order. (b) Plot of entropy. (c) Snapshots of the initial and final configurations of Kobots.

effects were analyzed using order, entropy and average forward velocity metrics. The experiments were conducted with CoSS using 100 robots and repeated 20 times. The same experiments were also conducted using seven Kobots and repeated 3 times. In the experiments, robots had random orientations having a regular hexagonal formation with a center distance of 25 cm. In this way, connectivity of the group was ensured, meaning that each robot was in the IR sensing range of at least one other robot. The experiments were conducted with the default parameter set, as shown in Table 6.1, unless otherwise stated.

7.2.1 Weight of Proximal-Control Behavior

The weight of proximal-control behavior (β) determines the relative strength of the two behaviors when the weight of alignment behavior (α) is assumed to be constant. The α parameter favors heading alignment and increases the average forward velocity of the flock, but does not consider the cohesiveness. In contrast, the β parameter favors proximal control and decreases the average forward velocity of the flock; however, it maintains the cohesiveness. In this section, we try to find an optimum α and β combination which will

preserve the cohesiveness of the flock while keeping the average forward velocity at high values.

In the experiments, we took α as constant and changed β to various values to find its optimum. The plots of CoSS and Kobot experiments are shown next to each for comparison purposes. Figures 7.4(a) and 7.4(b) show the time evolution of order. Time evolution of rate of change of entropy is depicted in Figures 7.5(a) and 7.5(b). Figures 7.6(a) and 7.6(b) show the average forward velocity of the group as a box-and-whisker plot.

For CoSS experiments conducted with 100 robots, it is seen in Figure 7.5(a) that increasing β , increases dS/dt having a value of 0 when $\beta = 0.5$. V_G makes a peak as β increases, then V_G decreases continuously as shown in Figure 7.6(a). In all of the β values, order reaches approximately 1 regardless of the choice of β as depicted in Figure 7.4(a). From these results, we can conclude that for a large group, the optimum value of β is 0.5, since at this value, $dS/dt = 0$ meaning that the group preserves its cohesiveness and V_G has its maximum value.

For the experiments performed with 7 Kobots, it is observed in Figure 7.5(b) that dS/dt is approximately 0 when $\beta = 1$ and $\beta = 2$. V_G has a similar trend as in the 100-robot case, having its maximum when $\beta = 0.5$ as shown in Figure 7.6(b). Order again reaches approximately 1 for all of the tested β values, as depicted in Figure 7.4(b). From these results, we can conclude that the optimum value of β is 1.0 for a small group. This is the value which both ensures cohesiveness of the flock ($dS/dt = 0$) and provides a moderate speed. β being different in 100-robot and 7-robot case is an interesting result that might be a consequence of differences in the dynamics of large and small groups.

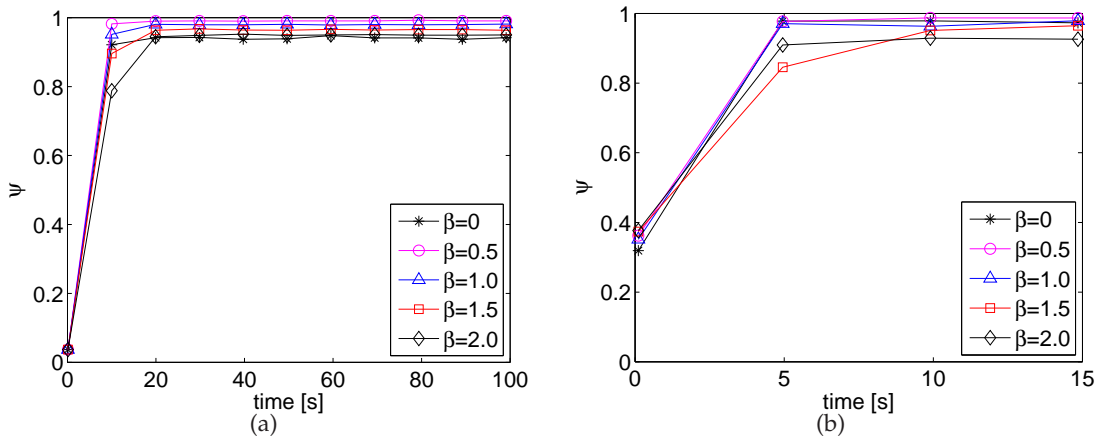


Figure 7.4: Plot of order in β experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars are not shown for clarity.

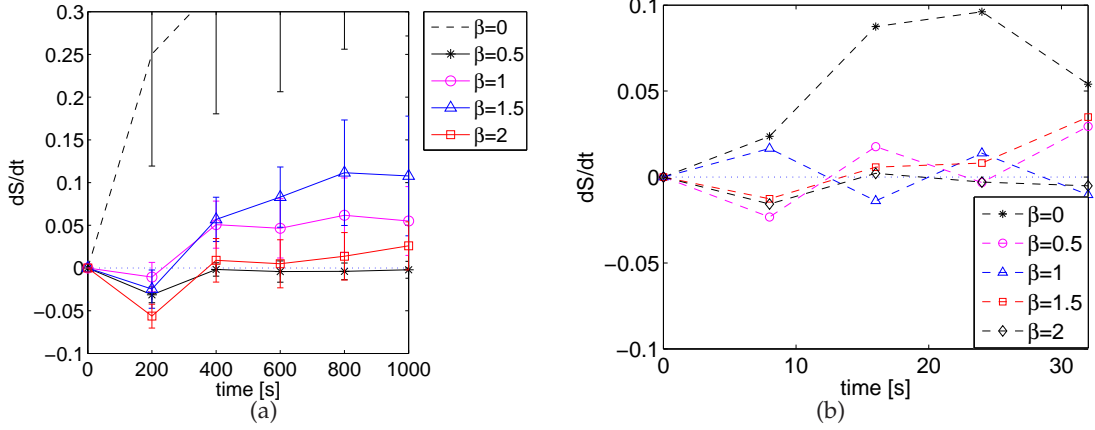


Figure 7.5: Plot of rate of change of entropy in β experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars indicate ± 1 standard deviation from the mean. Error-bars in (b) are not shown for clarity.

7.2.2 Maximum Forward Speed

Maximum forward speed (u_{max}) determines the maximum attainable forward speed (u) of a robot. Experiments were conducted to determine the optimal value of u_{max} that maximizes the average forward velocity of the group while maintaining its cohesiveness.

In the experiments, we varied u_{max} and determined its effect on the flocking behavior. For the Kobot experiments, u_{max} was only set to 4 and 7 cm/s due to mechanical limitations of the DC motors. The results of CoSS and Kobot experiments are presented next to each other for comparison purposes. Figures 7.7(a) and 7.7(b) show time evolution of order. Time evolution of the rate of change of entropy is shown in Figures 7.8(a) and 7.8(b). Average forward velocity is represented as a box-and-whisker plot in Figures 7.9(a) and 7.9(b).

It is observed in Figures 7.7(a) and 7.7(b) that for all of the tested values of u_{max} , the robots attain an order value of approximately 1 at approximately equal times. dS/dt is observed to be 0 when u_{max} is 4 or 7 cm/s both in CoSS and Kobot experiments as shown in Figures 7.8(a) and 7.8(b). It is observed in Figures 7.9(a) and 7.9(b) that increasing u_{max} increases the average forward velocity of the group. As a result, we set the value of u_{max} to 0.7 m/s which ensures the cohesion of the flock, results in a moderate average forward velocity and is the maximum physically realizable speed in Kobots.

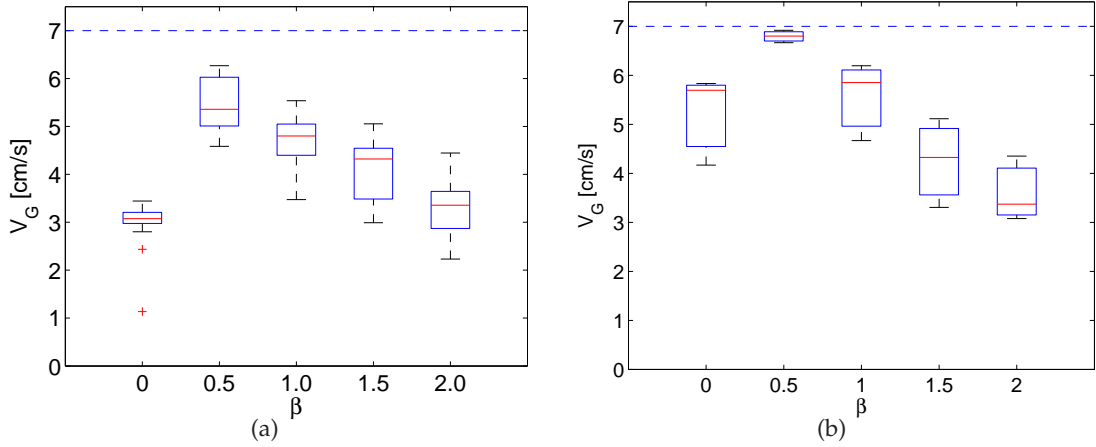


Figure 7.6: Plot of average forward velocity in β experiments (a) CoSS experiments. (b) Kobot experiments. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs.

7.2.3 Proportional Gain

Robots have two velocity components. The first is the forward speed (u), determined by the u_{max} parameter. The second is the angular velocity (ω), determined by a proportional controller having a gain of K_p . The optimum value of the u_{max} parameter was determined in the previous section, and in this section, we will find the optimum value of K_p . When K_p is large, a small deviation from the desired direction causes rapid rotations in robots, eventually decreasing the average forward velocity of the flock. However, when K_p is small, the robots become sluggish in their rotations, resulting in a high average forward

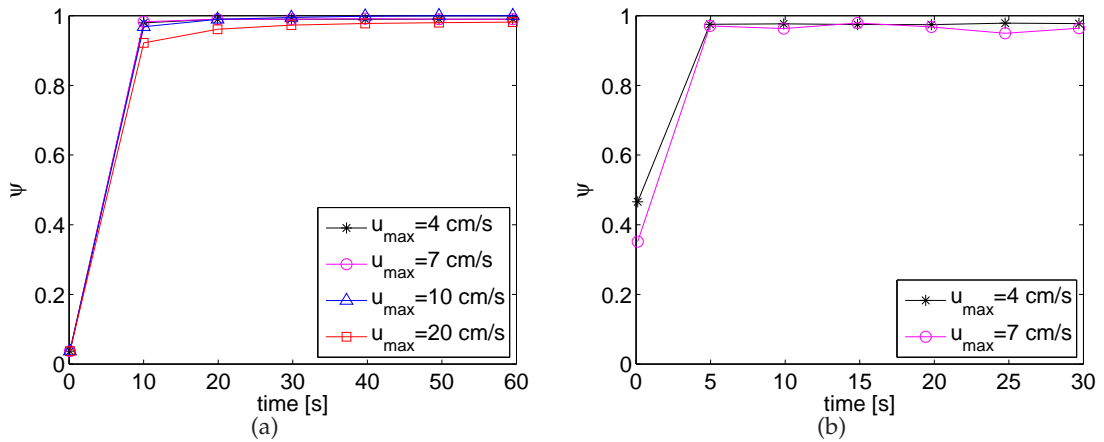


Figure 7.7: Plot of order for u_{max} experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars are not shown for clarity.

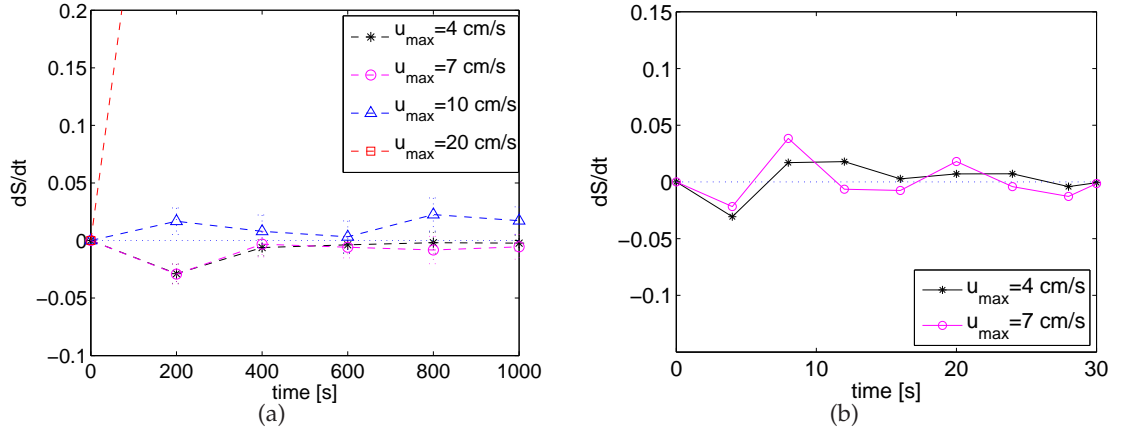


Figure 7.8: Plot of rate of change of entropy for u_{max} experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars indicate ± 1 standard deviation from the mean. Error-bars in (b) are not shown for clarity.

velocity but decreased responsiveness. Experiments were conducted to investigate these counter-effects and find an optimal value of K_p .

In these experiments, K_p was varied and their results for CoSS and Kobot experiments are presented next to each other. Time evolution of order is depicted in Figures 7.10(a) and 7.10(b). Rate of change of entropy is plotted in Figures 7.11(a) and 7.11(b). Average forward velocity is plotted as a box-and-whisker plot and shown in Figures 7.12(a) and 7.12(b).

It is observed in Figures 7.10(a) and 7.10(b) that order reaches a steady-state value of approximately 1 for all of the tested values of K_p . dS/dt is 0 when K_p is 0.5 or 0.9 as seen in Figures 7.11(a) and 7.11(b). For CoSS experiments, V_G shows a decreasing trend as K_p increases shown in Figure 7.12(a) which is not the same in Kobot experiments shown in Figure 7.12(b) such that Kobots cannot attain a high V_G when K_p is 0.1 unlike the CoSS counterpart. This is due the fact that at this value of K_p , flock cannot stay cohesive ($dS/dt > 0$). Robots in a small group move away from each other much easily than robots in a large group, hence; V_G of Kobots is considerably lower than V_G of robots in CoSS. From these results, we can conclude that the optimal value of K_p is 0.5 which makes the flock stay cohesive and attain a moderate average forward velocity.

The experiments conducted in this chapter revealed important facts. First, heading-alignment behavior is crucial in attaining a common orientation among the robots. Noisy IR sensors are not sufficient for this purpose. Second, when robots are in need of avoiding an object, they should have their forward velocities modulated; otherwise, they collide

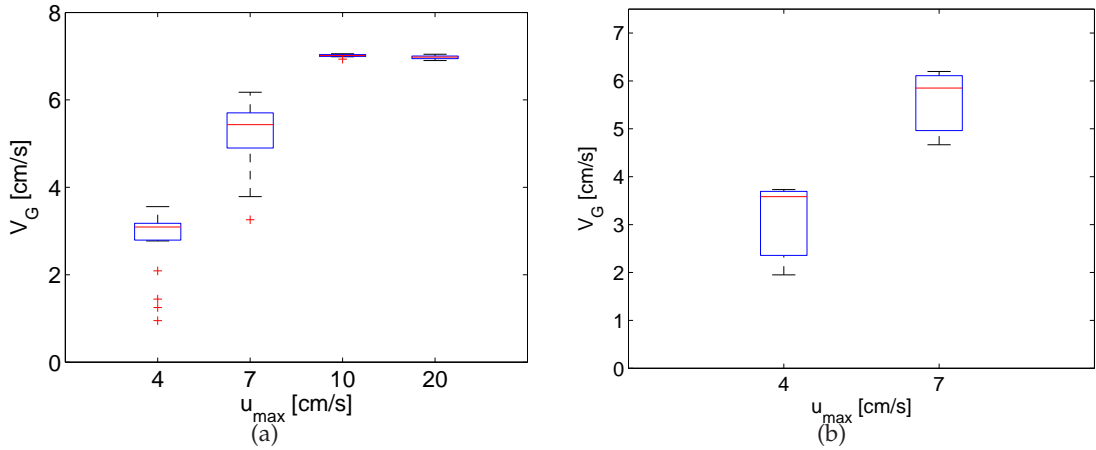


Figure 7.9: Plot of average forward velocity for u_{max} experiments. (a) CoSS experiments. (b) Kobot experiments. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs.

with the other robots or obstacles and usually get stuck. Third, there is an optimum set of β , u_{max} and K_p parameters of flocking that result in *desirable* flocking behavior. In the next chapter, we will consider the effect of VHS characteristics on the flocking behavior.

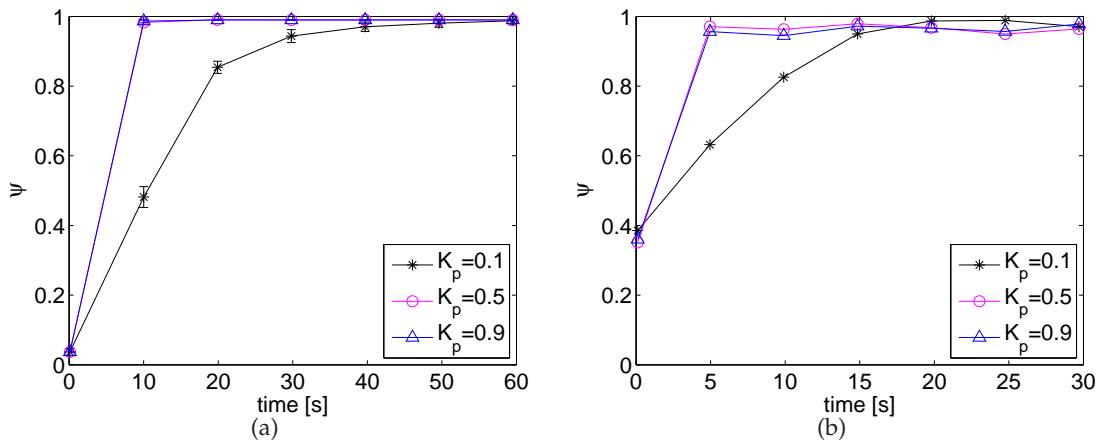


Figure 7.10: Plot of order for K_p experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars are not shown for clarity.

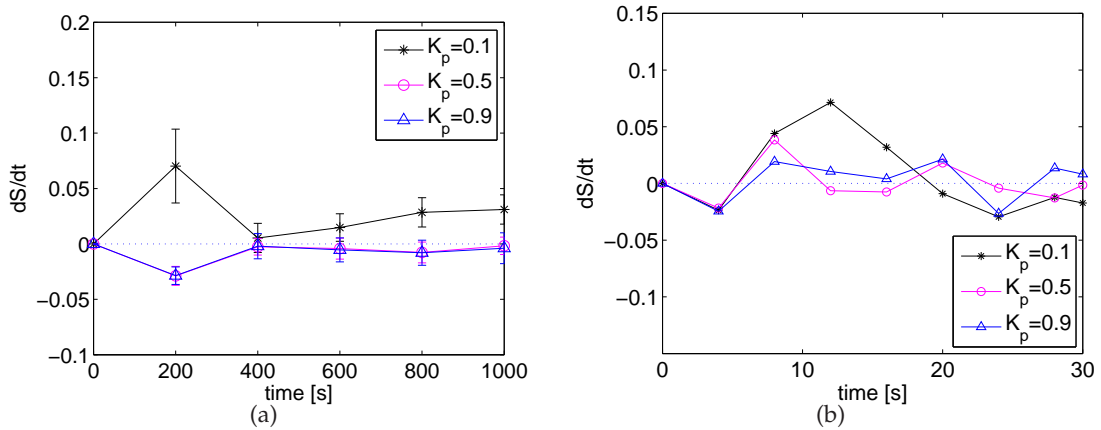


Figure 7.11: Plot of rate of change of entropy for K_p experiments. (a) CoSS experiments. (b) Kobot experiments. Error-bars indicate ± 1 standard deviation from the mean. Error-bars in (b) are not shown for clarity.

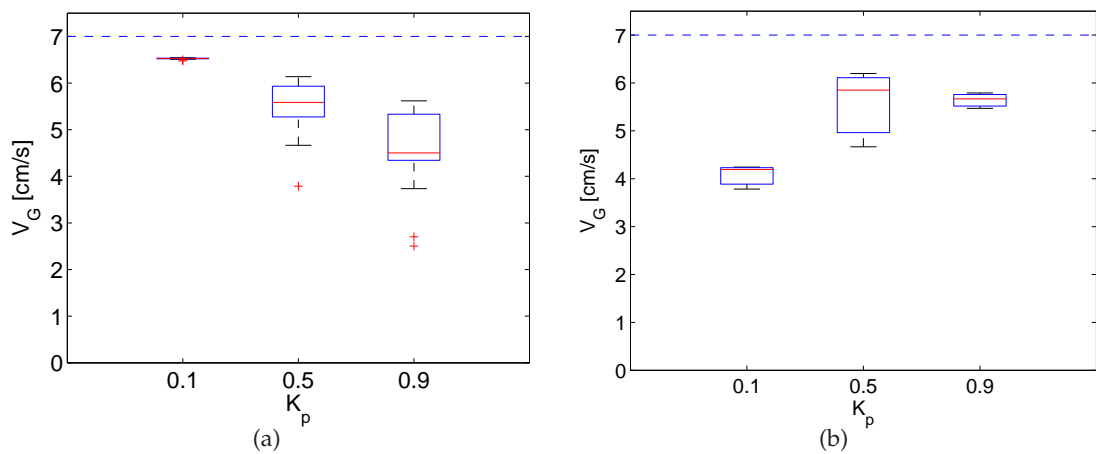


Figure 7.12: Plot of average forward velocity for K_p experiments. (a) CoSS experiments. (b) Kobot experiments. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs.

CHAPTER 8

ANALYSIS OF FLOCKING BEHAVIOR UNDER DIFFERENT VHS CHARACTERISTICS

The performance of flocking depends highly on the sensory characteristics of the VHS. Therefore, in this chapter, we study the performance of flocking under different parameter settings of the virtual heading sensor: (1) the amount and nature of noise of the digital compass, (2) the number of VHS neighbors and (3) the range of wireless communication.

We specifically varied the magnitude (η) and direction (ξ) of the noise vector, the communication range (R) and the number of VHS neighbors (N). The experiments were conducted with 100 robots in CoSS placed with random orientations in a regular hexagonal formation having 25 cm center spacing to ensure initial connectivity of the group. The default parameters as shown in Table 6.1 were utilized in the experiments unless otherwise stated and the experiments were repeated 10 times in each case. The number of VHS neighbors experiments were also performed with seven Kobots and were repeated 3 times.

8.1 Heading Noise

The digital compass utilized in the VHS is susceptible to noise due to the presence of ferrous metals in the environment when operated indoors. This noise is apt to affect the performance of flocking considerably. Therefore, in this section, we try to investigate the effect of heading noise in the virtual heading sensor.

Heading noise is modeled using the vectorial noise model [25], as shown in Equation 5.1. In the vectorial noise model, noise is represented by two components, one being magnitude (η) and the other being angle (ξ). Two experiments were conducted to deter-

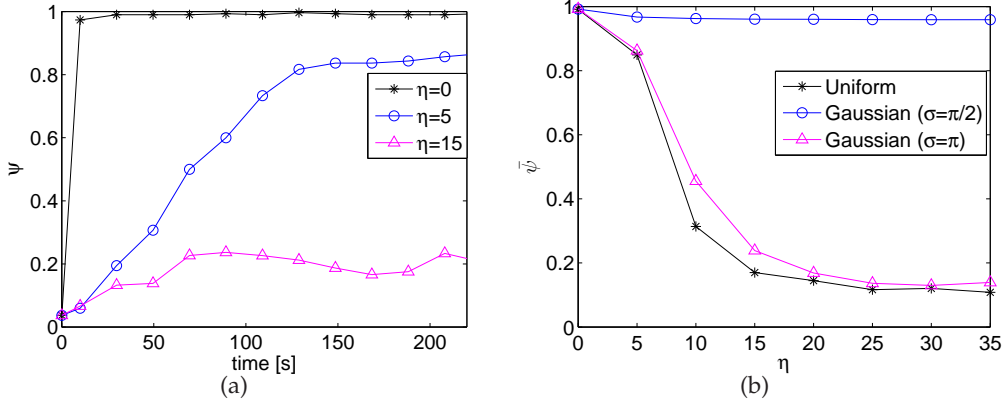


Figure 8.1: Heading noise experiments. (a) Plot of order for different levels of noise (η) when ξ is uniformly distributed in $[-\pi, \pi]$. (b) Plot of the steady-state value of order for Gaussian noise $N(0, \pi/2)$, $N(0, \pi)$ and uniformly distributed noise in $[-\pi, \pi]$.

mine the effect of these two components. In both of the experiments, the number of VHS neighbors is set to 1 to accentuate the effect of heading noise.

8.1.1 Magnitude of Noise

In the first experiment, we investigated the effect of magnitude (η) by assuming that (ξ) was uniformly distributed in $[-\pi, \pi]$ and varied η . Time evolution of order for various η values is plotted in Figure 8.1(a).

Results of the experiment reveal that even for rather high levels of noise ($\eta < 15$), order of the group converges to rather high values, though the speed of convergence is slowed as shown in Figure 8.1(a).

8.1.2 Nature of Noise

In the second experiment, we investigated the effect of the nature of noise by changing the type of distribution of the angle as $\xi \in [-\pi, \pi] = N(0, \infty)$, $\xi \in N(0, \pi/2)$ or $\xi \in N(0, \pi)$, where $N(x, y)$ denotes a Gaussian distribution having a mean of x and a standard deviation of y . η was varied between 0 and 20 for each distribution.

The steady-state values of order are plotted in Figure 8.1(b). The values are calculated by averaging order in the last 5000 time-steps of the experiments. The results indicate that order stays above 0.8 for η values smaller than 5 when uniform or Gaussian distributions ($N(0, \pi)$) are utilized. However, in case of the Gaussian distribution denoted by $N(0, \pi/2)$, order is approximately 1 for all the tested η values.

Two experiments reveal that flocking behavior is robust against sensory noise, both in cases of Gaussian and uniform distributions. In the rest of the experiments, noise is assumed to be Gaussian ($N(0, \pi/2)$), with an η value of 3.0 unless otherwise stated.

8.2 Number of VHS Neighbors

In these experiments, we varied number of VHS neighbors and investigated its effects on flocking using order, entropy and average forward velocity metrics. Two sets of experiments were conducted. In the first set, 100 robots were utilized in CoSS and in the second set, seven Kobots and seven robots in CoSS were utilized.

8.2.1 Large Group

The experiment was conducted using 100 robots in CoSS. Time evolution of order is plotted in Figure 8.2(a), and its steady-state value is shown in Figure 8.2(b). Time rate of change of entropy and average forward velocity are shown in Figures 8.3(a) and 8.3(b). Steady-state order is calculated by averaging the last 5000 time-steps.

An increase in N improves the settling time of order as shown in Figure 8.2(b) and it also increases the average forward velocity of the flock depicted in Figure 8.3(b). The rate of change of entropy is not affected considerably which is approximately 0 for all the tested values of N as seen in Figure 8.3(a). It is observed in Figure 8.2(b) that number of VHS neighbors positively affects system's robustness to noise. An η value of 15 decreases order to 0.2 when $N = 1$, whereas order is approximately 0.8 when $N = 3$.

8.2.2 Small Group

The experiments were conducted using seven Kobots and seven robots in CoSS. Order, entropy and average forward velocity values in Kobot experiments are plotted in Figures 8.4(a), 8.4(b) and 8.4(c), respectively. Order, entropy and average forward velocity values for CoSS experiments are plotted in Figures 8.5(a), 8.5(b) and 8.5(c), respectively.

Similar to the 100-robot case, it is seen in Figures 8.4(a) and 8.5(a) that for all of the tested N values, order approaches to 1 quite rapidly. The latency of the 1-neighbor case in settling time is much less significant in seven robots than in 100 robots, since seven robots assume a common heading more rapidly than hundred robots. The rate of change of entropy is not affected significantly by N as shown in Figures 8.4(b) and 8.5(b). Average forward velocity increases with increasing N , however the increase is less apparent than

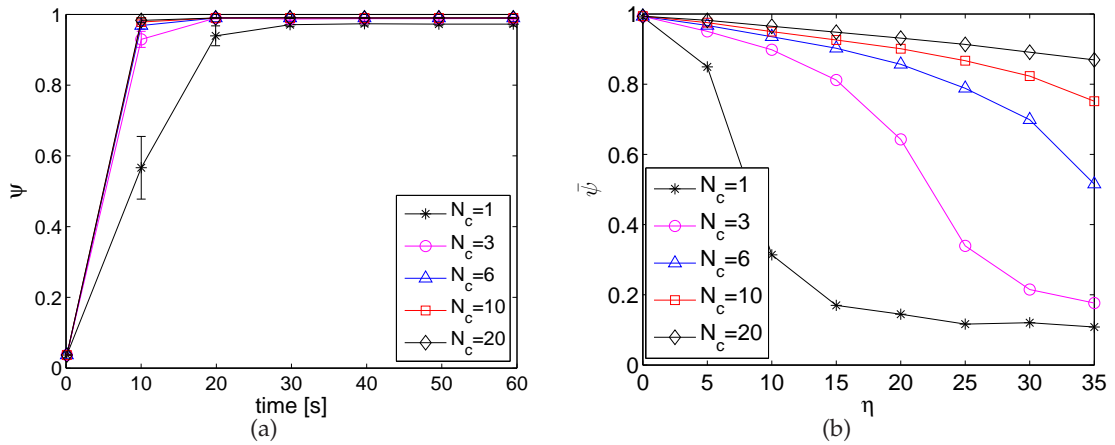


Figure 8.2: Number of VHS neighbors experiment with 100 robots in CoSS. (a) Plot of order. (b) Plot of the steady-state value of order. Uniformly distributed noise is assumed with $\eta = 3$ to accentuate the results.

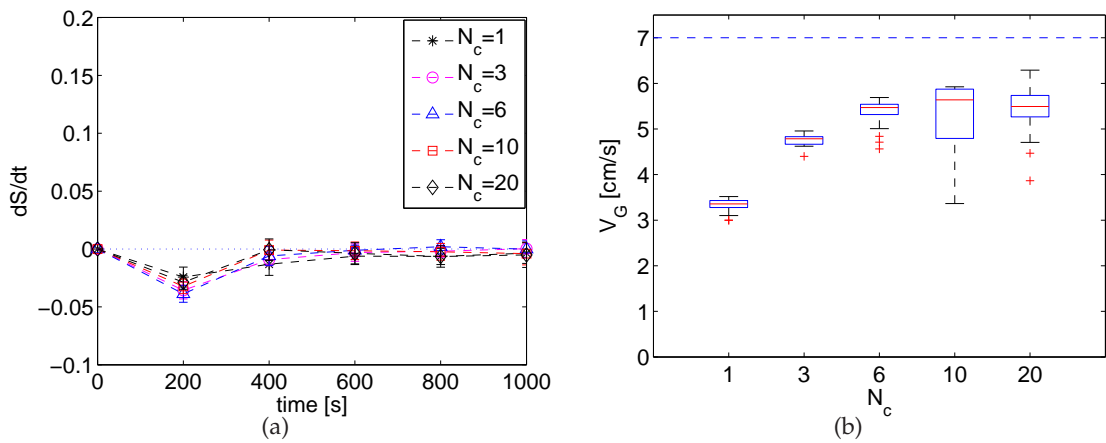


Figure 8.3: Number of VHS neighbors experiment for 100 robots in CoSS. (a) Plot of rate of change of entropy. (b) Plot of average forward velocity. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. The outliers are denoted by '+' signs.

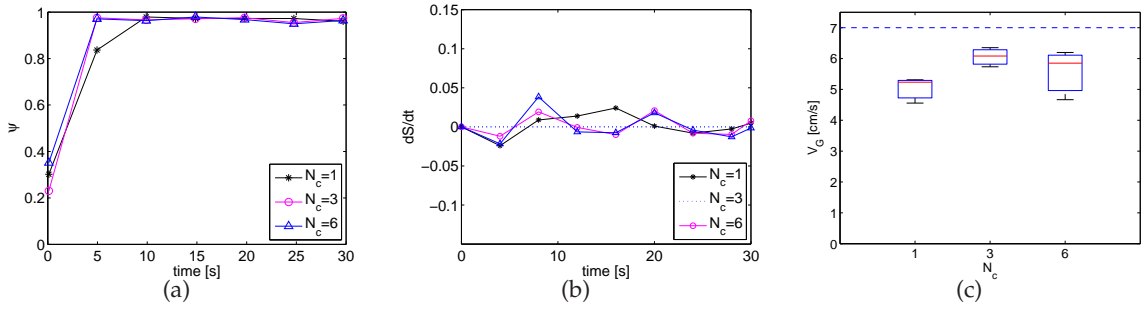


Figure 8.4: Number of VHS neighbors experiments with seven Kobots. (a) Plot of order. (b) Plot of rate of change of entropy. (c) Plot of average forward velocity. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. The outliers are indicated by '+' signs.

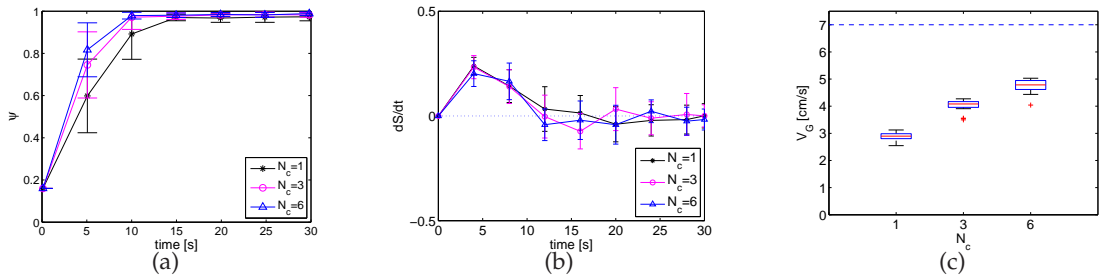


Figure 8.5: Number of VHS neighbors experiment with seven robots in CoSS. (a) Plot of order. (b) Plot of rate of change of entropy. (c) Plot of average forward velocity. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. The outliers are indicated by '+' signs.

the 100-robot case, since the direction of motion in a 7-robot-flock is much more prone to disturbances by momentary changes in the direction of individual robots as seen in Figures 8.4(c) and 8.5(c).

8.3 Range

In this section, we investigate the effect of range on the performance of flocking. Range values (R) were varied such that a local (40 cm), an almost global (100 cm) or a global range (200 cm) was attained for 100 robots. The number of VHS neighbors was taken as the mean value obtained from Prowler experiments (shown in Figure 5.2(b)), which was 5 for 40 cm, 25 for 100 cm, 22 and 20 for 200 cm.

Order and rate of change of entropy values are plotted in Figures 8.6(a) and 8.6(b) for various R . The size of the largest cluster is depicted in Figure 8.3. It is observed in

Figure 8.6(a) that order converges to 1 for R values of 1.0 and 2.0 m. dS/dt is also 0 for these values indicating a cohesive group as shown in Figure 8.6(b) which is also evident in Figure 8.3.

As a conclusion, we can say that, size of the largest cluster approaches to the size of the group when the range of communication tends to be global. We do not consider this as a limitation since the range of VHS is 20 m in indoor environments which is well enough to cover thousands of robots in its range.

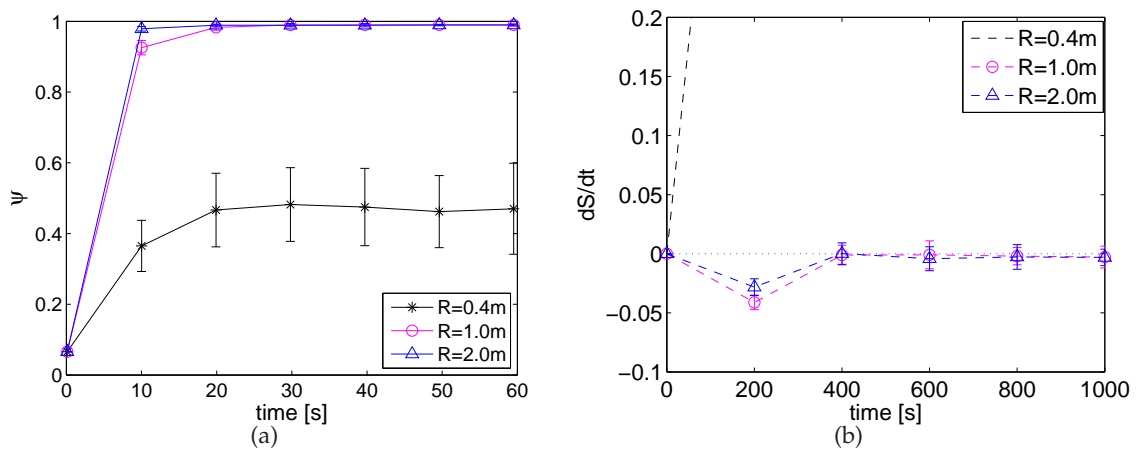


Figure 8.6: Range experiments. (a) Plot of order. (b) Plot of rate of change of entropy.

VHS characteristics have notable effects on the performance of flocking. First, flocking is quite robust against sensing noise. Even for very high noise values, robots can still attain a common heading and advance. Second, a large number of VHS neighbors increases the robustness of flocking against sensing noise. Third, locality in the communication range causes segmentation in the group, resulting in several independently-moving aligned and cohesive clusters. The range should be global in order to attain a single cluster.

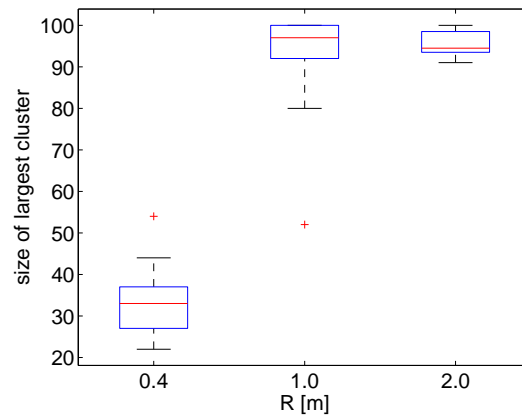


Figure 8.7: Plot of the size of the largest cluster. The ends of the boxes and the horizontal line in between correspond to the first and third quartiles and the median values, respectively. Outliers are indicated by '+' signs.

CHAPTER 9

MODELING PHASE TRANSITION IN FLOCKING

"Make everything as simple as possible, but not simpler."

— Albert Einstein

It was observed in the experiments related to heading noise in Section 8.1 and number of VHS neighbors in Section 8.2 that the unaligned-to-aligned transition of the robots resembles phase transition observed in most physical systems. Hence, in this chapter, we model the phase transition of flocking.

Modeling can be described as an abstraction of a system that is used to predict its response under certain conditions¹³. Specifically, in our case, the model is supposed to predict phase transition characteristics of flocking as a function of heading noise and VHS neighbors, yet simple enough to be treated analytically.

In this study, we extend an existing particle-based model, which is the vectorial network model, and utilize it in modeling the phase transition of flocking. The vectorial network model (VNM) [6] is a simple particle-based model proposed to analyze the predictions of the SDP model, utilizing stationary particles with random neighbors [2]. The VNM predicts similar phase transition characteristics to the SDP model. Under certain conditions, VNM can be treated analytically to show the existence of phase transition which is not present in any of the similar models.

¹³http://en.wikipedia.org/wiki/Mathematical_model. Last visited: April 2008.

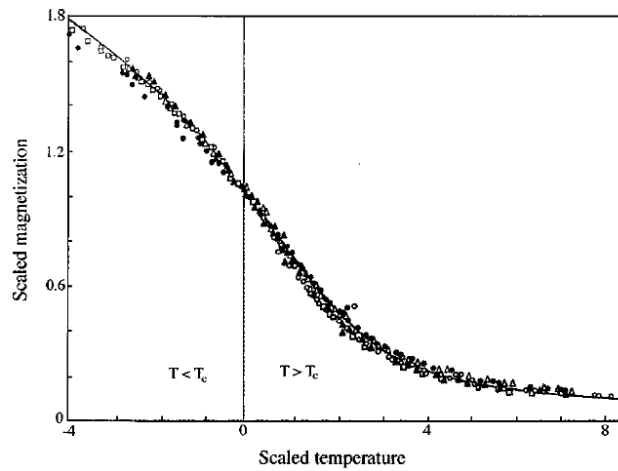


Figure 9.1: Plot of magnetization as a function of scaled temperature for five different magnetic materials. Reproduced from [5].

9.1 Phase Transition in Physical Systems

Phase transition¹⁴ is a widely-observed phenomenon in physical systems in which systems transform from one phase to another under the influence of a certain physical parameter, such as pressure or temperature. Phase transition can be categorized as: being first- or second-order.

First-order phase transition is the most frequent one encountered in our daily lives. Boiling of water, melting of ice and sublimation of naphthalene are some typical examples. In this type of transition, there is an abrupt or discontinuous change in the phase. Second-order phase transition is not witnessed frequently. Ferromagnetic and super-fluid transitions are two examples of this type of phase transition. In a second-order phase transition, phase change is continuous, as shown in Figure 9.1, an abrupt jump is not observed.

At this stage, it is interesting to note the similarities between Figures 9.1 and 9.2 despite the great differences in the systems they represent. Figure 9.1 depicts magnetization as a function of reduced temperature for five different magnetic materials, and Figure 9.2 shows order as a function of η for various number of VHS neighbors attained by 100 robots in CoSS. This similarity is a clear indication of phase transition in unaligned-to-aligned transition of robots under the heading noise. Furthermore, the transition is also argued to be of second-order due to its continuous nature.

A similar observation was also reported by Baldassarre [4] in a recent study. In this study, robots having random orientations were connected to each other physically. They

¹⁴http://en.wikipedia.org/wiki/Phase_transition. Last visited: April 2008.

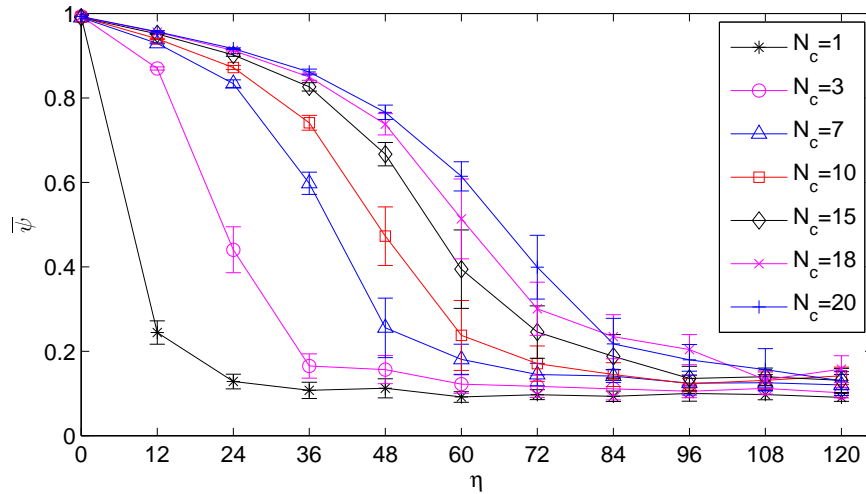


Figure 9.2: Plot of order. Experiments are conducted in CoSS using 100 robots for various number of VHS neighbors. Each experiment is repeated 10 times. Steady-state value of order is calculated by averaging its value for the last 5000 time-steps. Error-bars indicate ± 1 standard deviation from the mean.

were left to rotate and attain a common direction of motion and move in this direction. Robots started pushing and pulling their neighbors while measuring the net force exerted on their bodies, and finally, attained a common orientation and moved as a group. Results of the experiments revealed that unaligned-to-aligned transition of the robots under the influence of sensing noise showed second-order phase transition characteristics.

Order is utilized to study the second-order phase transition phenomenon. It represents the state of a system, such as magnetization, against an independent variable, such as temperature, in a system undergoing ferromagnetic phase transition, as shown in Figure 9.1. At temperatures greater than the critical temperature, indicated by T_c in the figure, the system has a low-order value, indicating a paramagnetic (un-magnetized) system. However, when the temperature is lowered gradually, order increases continuously, showing second-order phase transition characteristics. The system reaches an ordered state at temperatures less than the critical temperature, indicated by a vertical line in the figure. The temperature at which the transition occurs is called the Curie temperature (T_c)¹⁵. The details of this phenomena is not fully understood and remains as an open problem in statistical physics [5].

¹⁵http://en.wikipedia.org/wiki/Critical_phenomena. Last visited: April 2008.

9.2 Vectorial Network Model

The vectorial network model models the phase transition of the orientation of a group of M stationary particles placed evenly in a 2-D square lattice. The particles are only allowed to change their orientations, i.e., modify their heading vectors; $\mathcal{H}(t) = \{\vec{h}_1(t), \vec{h}_2(t), \dots, \vec{h}_M(t)\}$ where $\vec{h}_j(t) = e^{i\theta_j(t)}$ is the heading vector of the j^{th} particle at time t , having an angle of θ_j and a magnitude of unity.

The heading vector of each particle is updated to the average of the heading vectors of its neighbors with the addition of a noise term. The vectorial noise model [25] is utilized in the model. The update equation with the vectorial network model is:

$$\vec{h}_j(t+1) = \frac{1}{N} \sum_{k=1}^N e^{i\theta_k(t)} + \eta e^{i\xi(t)} \quad (9.1)$$

where N is the number of neighbors of each particle. η is the noise coefficient, and ξ is a delta-correlated random variable¹⁶ uniformly selected from $[-\pi, \pi]$.

The main difference of the VNM from other models is the way neighbors are determined as illustrated in Figure 9.3. In the VNM, neighbors are picked up from either the nearest particles or any random particle in the group. The degree of locality is determined by a parameter denoted by p . When $p = 0$, neighbors are all picked up randomly from the group and when $p = 1$, neighbors are selected from the nearest particles only.

VNM predicts a transition from an unaligned state to an aligned state of particles when the noise parameter (η) is decreased below the critical noise value ($\eta_c \simeq 4.5$), as shown in Figure 9.4. The phase transition is only observed at a non-zero noise value when there exists at least one random neighbor in the neighboring set of a particle, as observed when $p = 0.99$ in Figure 9.4. This is due to the fact that long-range interactions are essential to converge to an ordered state in one- or two-dimensional systems in the presence of noise, as stated by Mermin-Wagner in [20]. In the VNM, long-range interactions can only be made possible by introducing a random distant particle ($p < 1$), since the particles are stationary. However, in the SDP model, long-range interactions are facilitated by the movement and diffusion of particles that introduce long-range interactions; hence, a phase transition at a non-zero noise value is still possible for locally interacting particles [23].

¹⁶A random variable is called delta-correlated when its autocorrelation function is a Dirac-delta function, meaning that the current value of the random variable does not depend on its past values. It is also called white-noise, and it does not depend on the way the random variable is distributed. http://en.wikipedia.org/wiki/White_noise. Last visited: April 2008.

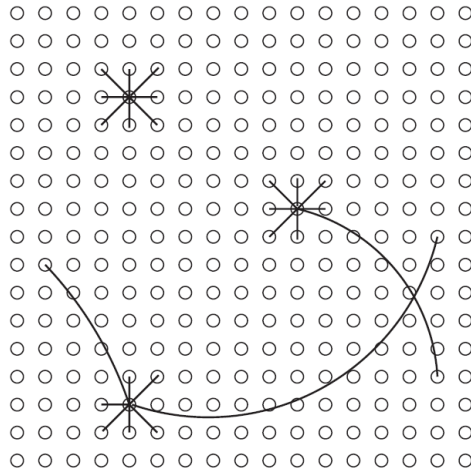


Figure 9.3: Neighbor selection in the vectorial network model. Three possible neighbor configurations for $p = 7/8$ when the number of neighbors (N) is set to be 8. Reproduced from [6].

Being a minimalist model, the VNM can successfully model the phase transition phenomena of simple systems and can be treated analytically as well. However, due to its simplicity, it cannot be used to directly model phase transition of flocking. The main reason for this failure is the lack of a proximal-interaction term in the VNM to account for the close-range interactions among the particles present in the flocking behavior. Close-range interactions make the robots sluggish so that they have a hard time turning to a desired direction. The particles in the VNM can turn to a desired direction freely, without any constraints.

9.3 Stiff Vectorial Network Model

We extended the VNM model to model phase transition of flocking. We add a term, called the persistence term, to the VNM and call the resulting model S-VNM. The persistence term consists of a coefficient (κ) multiplied by the current heading of the particle to account for the current heading when the desired heading is being calculated.

We utilized the vectorial noise model [25], in which noise is considered as a vector having a random direction and a variable magnitude that is added to each heading vector of the neighbors of a particle. Neighbors are selected randomly from the group ($p = 0$) due to two reasons: (a) The virtual heading sensor picks up neighbors randomly from the group regardless of the topology of the neighbor as shown in the receive topology experiment, Figure 5.3(b). (b) An analytical solution of the model is only possible in case of random

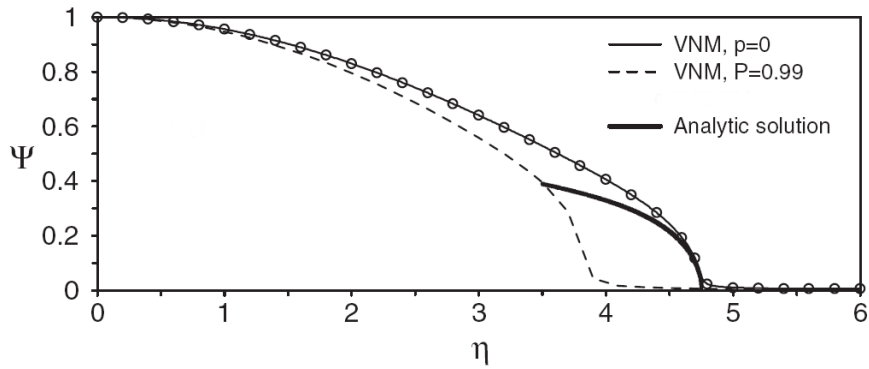


Figure 9.4: Phase transition diagram predicted by the VNM with $p = 0$ (solid line) and $p = 0.99$ (dashed line). The simulations are carried out using 2×10^4 particles with 15 neighbors. η denotes the magnitude of noise, and γ is the order parameter of the group. Adapted from [6].

neighbors. S-VNM consists of M stationary particles placed in a 2-D lattice. The particles can modify their headings vectors $\mathcal{H}(t) = \{\vec{h}_1(t), \vec{h}_2(t), \dots, \vec{h}_N(t)\}$, where $\vec{h}_j(t) = e^{i\theta_j(t)}$ is the heading vector of the j^{th} particle at time t having an angle of θ_j and a magnitude of unity.

$$\vec{h}_j(t+1) = \kappa e^{i\theta_j(t)} + \lambda \sum_{k=1}^{N_c} e^{i\theta_k(t)} + \eta \sum_{k=1}^{N_c} e^{i\xi_k(t)} \quad (9.2)$$

where N_c is the number of neighbors selected randomly from the group. N_c is chosen to be the same for all particles. λ is the interaction coefficient and η is the noise coefficient. ξ_k is a delta-correlated random variable uniformly selected from $[-\pi, \pi]$.

The steady-state characteristics of the model are investigated using the time-averaged version of the order ($\bar{\psi}$) as shown in Equation 6.6. Another important point related to the steady-state characteristics of the system is the critical noise coefficient (η_c). The critical noise coefficient is the value of the noise coefficient (η) when order (ψ) is equal to 0.2. At this value at which the system changes state. For η values greater than η_c , the system is unaligned and in a disordered state. For η values smaller than η_c , the system is aligned, and in an ordered state.

The effect of the persistence coefficient (κ) is shown by simulating the S-VNM in Matlab with 100 particles, each having one neighbor as shown in Figure 9.5(a). Each simulation is repeated three times for 10000 simulation time-steps, and $\bar{\psi}$ is calculated by averaging order in the last 5000 time-steps.

It is observed that an increase in κ flattens the order curve both in the high- and the low-

noise regions, whereas an increase in λ flattens the curve mostly in the low-noise region as illustrated in Figure 9.5(b).

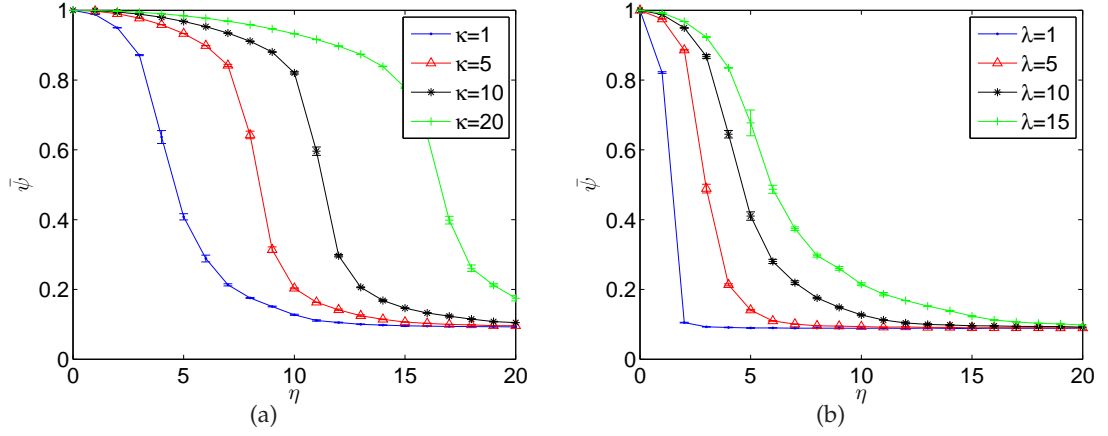


Figure 9.5: The effect of κ and λ coefficients on $\bar{\psi}$. Plot of $\bar{\psi}$ as a function of η . (a) $\kappa \in \{1, 5, 10, 20\}$ and $\lambda = 10$. (b) $\lambda \in \{1, 5, 10, 15\}$ and $\kappa = 1$. Error-bars indicate \pm standard deviation from the mean.

9.4 Analytical Treatment of S-VNM

One of the main appeals of the S-VNM lies in the possibility of analytical treatment of the model. We will propose a solution near the transition point to estimate η_c in terms of κ , λ and N [61].

We will first find the probability distribution functions (PDF) of each term in Equation 9.2, and then solve this equation to find η_c in the stationary (steady-state) condition.

Since only the direction of h is of consideration, we rescale the equation by dividing by λ , and using $\tilde{\kappa} = \kappa/\lambda$ and $\tilde{\eta} = \eta/\lambda$ to simplify notation, and we will drop "tildes" until the end of the calculation.

We first consider the noise term which corresponds to the sum of N steps (each of length 1) in a random direction on the plane. This is equivalent to a standard random walk. As such, we know that the PDF of the position on the $x y$ plane after N steps will be:

$$P_\eta(x, y) = \frac{1}{N\eta^2\pi} e^{-\frac{x^2+y^2}{N\eta^2}} \quad (9.3)$$

We then take the 2-D Fourier transform of Equation 9.3 and obtain:

$$\hat{P}_\eta(\lambda_x, \lambda_y) = e^{-\frac{1}{4}N\eta^2(\lambda_x^2 + \lambda_y^2)} \quad (9.4)$$

The PDF of the interaction term is calculated in [6]. In the calculation, it is assumed that all neighbors of a particle are picked randomly from the group, and hence neighbor inputs are statistically independent. They also have identical PDFs since they are all uncorrelated. In this case, we can utilize the Central Limit Theorem¹⁷ for $N > 5$ and obtain:

$$\hat{P}_{N\theta}(\lambda_x, \lambda_y) = e^{iN(\Delta_{1,0}\lambda_x + \Delta_{0,1}\lambda_y) - \frac{N}{2}(\sigma_c^2\lambda_x^2 + \sigma_s^2\lambda_y^2 + 2\sigma_{cs}^2\lambda_x\lambda_y)} \quad (9.5)$$

where $\sigma_c^2(t) = \Delta_{2,0}(t) - \Delta_{1,0}^2(t)$, $\sigma_s^2(t) = \Delta_{0,2}(t) - \Delta_{0,1}^2(t)$, and $\sigma_{cs}^2(t) = \Delta_{1,1}(t) - \Delta_{1,0}\Delta_{0,1}$.

$\Delta_{m,n}(t)$ is the instantaneous sine-cosine moment of the distribution which is calculated in [6] as:

$$\Delta_{m,n}(t) = \int_{-\pi}^{\pi} P_\theta(\alpha; t) \cos^m(\alpha) \sin^n(\alpha) d\alpha$$

We need to find the PDF of the heading of a single particle ($P_\theta(\alpha; t)$) to evaluate $\Delta_{0,1}$, $\Delta_{1,0}$, $\Delta_{1,1}$, $\Delta_{2,0}$ and $\Delta_{0,2}$, and hence, find the PDF of the interaction term ($\hat{P}_{N\theta}(\lambda_x, \lambda_y)$) in Equation 9.5. Since we are interested in the stationary solution close to the critical noise value (η_c), we only need to approximate $P_\theta(\alpha; t)$ near this point.

We know that when the noise (η) is larger than the critical noise value (η_c) the particles are in disordered state and each of them has the same probability of being in any direction, which results in a uniform $P_\theta(\alpha; t)$ distribution having a magnitude of $\frac{1}{2\pi}$. Decreasing the noise (η) causes small perturbations to grow and transforms the system continuously to the ordered state. Therefore, we assume a $P_\theta(\alpha; t)$ having a perturbation term consisting of a small constant (δ) multiplied by a continuous function such as cosine function. Without loss of generality, the stationary distribution of $P_\theta(\alpha; t)$ near η_c is taken as:

$$P_\theta(\alpha) = \frac{1}{2\pi} + \delta \cos(\alpha) \quad (9.6)$$

$P_\theta(\alpha)$ is the distribution of the heading of an agent, which is the same for all agents due to the fact that they are all uncorrelated random variables. In addition to this, all $P_\theta(\alpha)$ evolve using Equation 9.2 to a stationary solution; this is also the distribution of the persistence term in Equation 9.2 and of the left hand side (LHS) of this equation. With the assumed form for P_θ , Equation 9.5 becomes:

$$\hat{P}_{N\theta}(\lambda_x, \lambda_y) = e^{i\pi N\delta\lambda_x - \frac{1}{2}N\left[\left(\frac{1}{2} - \pi^2\delta^2\right)\lambda_x^2 + \frac{\lambda_y^2}{2}\right]} \quad (9.7)$$

¹⁷http://en.wikipedia.org/wiki/Central_limit_theorem. Last visited: April 2008.

Then we multiply $\hat{P}_\eta(\lambda_x, \lambda_y)$ and $\hat{P}_{N\theta}(\lambda_x, \lambda_y)$, and take the inverse Fourier transform resulting in the combined PDF of the interaction and noise terms.

$$P_{N\theta\eta}(x, y) = \frac{1}{\pi N \sqrt{(1 + \eta^2)(1 + \eta^2 - 2\pi^2\delta^2)}} e^{-\frac{1}{N} \left[\frac{(x - N\pi\delta)^2}{1 + \eta^2 - 2\pi^2\delta^2} + \frac{y^2}{1 + \eta^2} \right]} \quad (9.8)$$

We then find the probability distribution of the RHS by taking the convolution of $P_{N\theta\eta}$ with PDF of the persistence term, which is equal to $P_\theta(\alpha)$. We finally obtain:

$$P_{RHS}(x, y) = \int_{-\pi}^{\pi} P_{N\theta\eta}(x - \kappa \cos \theta, y - \kappa \sin \theta) P_\theta(\theta) d\theta \quad (9.9)$$

Since we are considering a small δ , we expand Equation 9.9 to first-order in λ by considering small δ perturbation. We then express the resulting equation in polar coordinates (R, Φ) and integrate over R . The resulting equation, which is the PDF of the RHS is:

$$P_{RHS}(\Phi) = \frac{1}{2\pi} + \lambda \Gamma \cos(\Phi) \quad (9.10)$$

where Γ is:

$$\Gamma = \frac{\sqrt{\pi} e^{\frac{-\kappa^2}{2N(1+\eta^2)}}}{2\sqrt{N(1+\eta^2)}} \left[(N + \kappa) I_0 \left(\frac{\kappa^2}{2N(1+\eta^2)} \right) + \kappa I_1 \left(\frac{\kappa^2}{2N(1+\eta^2)} \right) \right] \quad (9.11)$$

where I_0 and I_1 are Modified Bessel Functions¹⁸ of the first kind which are defined as the solution to the differential equation: $z^2 y'' + zy' - (z^2 + n^2)y = 0$.

In the stationary solution, the LHS of Equation 9.2 is equal to $P_\theta(\phi)$. Hence the LHS and the RHS have the same form, resulting in an equality of $P_{RHS}(\Phi) = P_\theta(\Phi)$ which can be expressed as:

$$\frac{1}{2\pi} + \delta \cos(\Phi) = \frac{1}{2\pi} + \lambda \Gamma \cos(\Phi) \quad (9.12)$$

We solve this equation for Γ , then use Equation 9.11 to determine the critical noise value (η_c). We know that just above the critical noise value, the perturbations should grow and put the system into an ordered state. When $\Gamma < 1$, the perturbations will die out, resulting in a disordered state; conversely, when $\Gamma > 1$, the perturbations will grow and put the system in an ordered state. We therefore set Γ to 1, which corresponds to the critical point.

We can find an explicit approximate form for η_c by carrying out certain approximations in the regime that we are considering. We set κ and λ to 1.5 and 22, respectively, to capture

¹⁸<http://mathworld.wolfram.com/ModifiedBesselFunctionoftheFirstKind.html>. Last visited: April 2008.

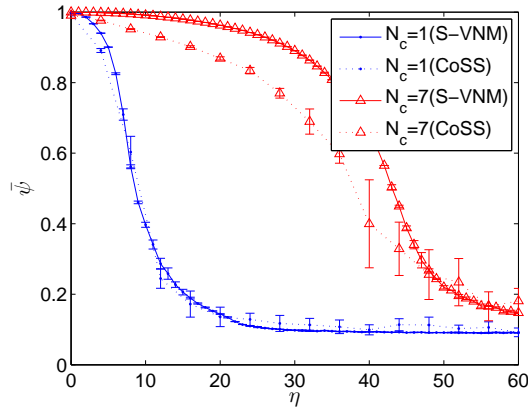


Figure 9.6: The phase transition diagram obtained using simulation of S-VNM. In simulations, κ is set to 1.5 and λ is set to 22.

the dynamics of the flocking behavior. For this set of coefficients, $\frac{\kappa^2}{2N(1+\eta^2)} \ll 1$, which makes $I_0 \sim 1$ and $I_1 \sim 0$. κ is neglected since it is very small when compared to the other coefficients. After substituting $\eta = \tilde{\eta}\lambda$, we obtain a simple equation for η_c as:

$$\eta_c = \lambda \sqrt{\frac{N\pi}{4}} \quad (9.13)$$

9.5 Results of S-VNM

The S-VNM can be utilized in two ways to predict the phase-transition of flocking. On one side, the S-VNM can be easily solved numerically to obtain the full phase-transition diagram of flocking that result from a given set of parameters. On the other side, the S-VNM can be solved analytically to predict η_c .

Phase transition diagram is obtained by simulating S-VNM in Matlab. The simulation is performed with 100 particles for 10000 steps utilizing 1 and 7 neighbors. $\bar{\eta}$ for the last 5000 steps is plotted in Figure 9.5 together with the results of CoSS experiments. It is observed that, predictions of S-VNM is in close agreement with CoSS when $N = 1$. A slight deviation is observed in $N = 7$ case for small η values.

In predicting the critical noise value of flocking, we utilized Equation 9.13 for different N . The results are plotted in Figure 9.7 together with the results of CoSS simulations. We can see that the two results are in close agreement both for small and large N values. However, we should note that Equation 9.13 is actually valid for $N > 5$ due to the Central Limit Theorem utilized in the analytical treatment.

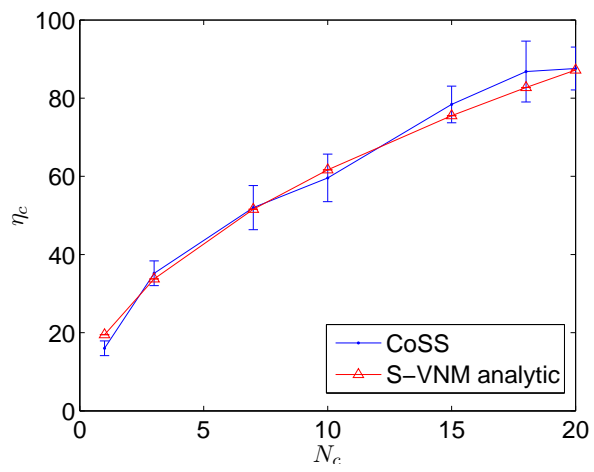


Figure 9.7: Critical noise values for various number of neighbor values.

CHAPTER 10

CONCLUSION

In this thesis, we study self-organized flocking in a swarm of mobile robots without any fixed or elected leader. We proposed a behavior based on the heading alignment and proximal control behaviors. The behavior was first implemented using nine Kobots. We then utilized a physics-based simulator to investigate the performance of flocking under different controller parameters and virtual heading sensor characteristics. We finally modeled the flocking behavior to study its stability conditions under different VHS characteristics.

This thesis has major contributions to various fields of research. First, the Kobot robotic system, having a distinctive design, has two novel sensing systems: the short-range sensing system and the virtual heading sensor. The short-range sensing system is designed to perform proximity measurements at a range of 20 cm. It has the ability to distinguish kin-robots from obstacles with minimal interference from other robots and the environment. The virtual heading sensor, which broadcasts digital compass readings through a wireless-communication module and obtains the relative headings of a group of robots. This method, based only on the assumption that the sensed North direction remains the same as the neighboring robots, is scalable, and holds great promise for use in swarms of mobile robots as well as UAVs.

Second, the flocking behavior that is proposed in the thesis creates self-organized flocking in a group of robots. Different from previous flocking studies with mobile robots, this behavior does not require "simulated sensors", a goal heading that is sensed by the whole group or an elected or designated leader. In this sense, we claim that, to the best of our knowledge, this study is the first truly self-organized flocking attained in a group of mobile robots. It is also the first study that has proposed quantifiable metrics to analyze the flocking behavior systematically under different parameters.

Third, systematic analysis of flocking behavior revealed important facts. The flocking

behavior is quite robust to large amounts of noise, even when each robot hears from only a single neighbor, although the robustness against noise and average forward velocity of the group increases significantly by increasing the number of VHS neighbors. We have also observed that communication range determines the size of the flock, which is consistent with models of flocking in natural systems. Segmentation of the group is observed when the communication range is kept at local values, which is due to the quasi-static movement of robots after adopting a common heading. Due to this fact, the heading information cannot spread throughout the group unless the communication range of the virtual heading sensor is large enough.

Fourth, to the best of our knowledge, this is the first study to model and analytically predict conditions of stability in flocking behavior implemented using a group of robots.

REFERENCES

- [1] C. Reynolds, "Boids (flocks, herds and schools: a distributed behavioral model)." <http://www.red3d.com/cwr/boids/>, Last visited: February 2008.
- [2] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, 1995.
- [3] S. Andelkovic, I. Velickovic, M. Rasic, and G. S. Dordevic, "Digital compass as heading sensor for hexapod robot," *Journal of Automatic Control*, vol. 13, no. 2, pp. 11–16, 2003.
- [4] G. Baldassarre, "Self-organization as phase transition in decentralized groups of robots: A study based on Boltzmann entropy," in *Advances in Applied Self-Organizing Systems* (P. Mikhail, ed.), pp. 127–146, Berlin: Springer Verlag, 2008.
- [5] H. E. Stanley, "Scaling, universality and renormalization: Three pillars of modern critical phenomena," *Reviews of Modern Physics*, vol. 71, no. 2, pp. 358–366, 1999.
- [6] M. Aldana and C. Huepe, "Phase transitions in self-driven many-particle systems and related non-equilibrium models: A network approach," *Journal of Statistical Physics*, vol. 112, no. 1-2, pp. 135–153, 2003.
- [7] M. Dorigo and E. Sahin, "Special issue: Swarm robotics," *Autonomous Robots*, vol. 17, pp. 111–113, 2004.
- [8] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics Workshop: State-of-the-art Survey* (E. Sahin and W. Spears, eds.), no. 3342 in *Lecture Notes in Computer Science*, (Berlin Heidelberg), pp. 10–20, Springer-Verlag, 2005.

- [9] S. Camazine, N. Franks, J. Sneyd, E. Bonabeau, J.-L. Deneubourg, and G. Theraulaz, *Self-Organization in Biological Systems*. Princeton, NJ, USA: Princeton University Press, 2001. ISBN: 0691012113.
- [10] L. Bayindir and E. Sahin, "A review of studies in swarm robotics," *Turk J. Elec. Engin.*, vol. 15, no. 2, pp. 115–147, 2007.
- [11] B. Szabo, G. J. Szollosi, B. Gonci, Z. Juranyi, D. Selmeczi, and T. Vicsek, "Phase transition in the collective migration of tissue cells: experiment and model," *Physical Review E*, vol. 74 (061908), 2006.
- [12] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. O. G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the National Academy of Sciences, USA*, vol. 105, pp. 1232–1237, January 2008.
- [13] S. J. Simpson, G. A. Sword, P. D. Lorch, and I. D. Couzin, "Cannibal crickets on a forced march for protein and salt," *Proceedings of the National Academy of Sciences, USA*, vol. 103, pp. 4152–4156, March 2006.
- [14] B. Partridge, "The structure and function of fish schools," *Scientific American*, pp. 114–123, June 1982.
- [15] I. Couzin, "Collective minds," *Nature*, vol. 445, p. 715, 2007.
- [16] K. S. Kruszelnicki, "Physics of flocks." <http://www.abc.net.au/science/k2/moments/gmis9845.htm>, Last visted: February 2008.
- [17] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *SIG-GRAPH '87: Proc. of the 14th annual conference on computer graphics and interactive techniques* (M. C. Stone, ed.), (New York, NY, USA), pp. 25–34, ACM Press, July 1987. ISBN: 0-89791-227-6.
- [18] J. Buhl, D. J. T. Sumpter, I. Couzin, J. Hale, E. Despland, E. Miller, and S. J. Simpson, "From disorder to order in marching locusts," *Science*, no. 312, pp. 1402–1406, 2006.
- [19] A. Czirok, H. E. Stanley, and T. Vicsek, "Spontaneously ordered motion of self-propelled particles," *Journal of Physics A: Math. Gen.*, vol. 30, pp. 1375–1385, 1997.

- [20] N. D. Mermin and H. Wagner, "Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic Heisenberg models," *Physical Review Letters*, vol. 17, no. 22, pp. 1133–1136, 1966.
- [21] A. Czirok, A.-L. Barabasi, and T. Vicsek, "Collective motion of self-propelled particles: Kinetic phase transition in one dimension," *Physical Review Letters*, vol. 82, no. 1, pp. 209–212, 1999.
- [22] A. Czirok and T. Vicsek, "Collective behavior of interacting self-propelled particles," *Physica A*, vol. 281, pp. 17–29, 2000.
- [23] M. Nagry, I. Daruka, and T. Vicsek, "New aspects of the continuous phase transition in the Scalar Noise Model (SNM) of collective motion," *Physica A*, vol. 373, pp. 445–454, 2007.
- [24] C. Huepe and M. Aldana, "Intermittency and clustering in a system of self-driven particles," *Physical Review Letters*, vol. 92, no. 16, 2004.
- [25] G. Gregoire and H. C. Y. Tu, "Moving and staying together without a leader," *Physica D*, no. 181, pp. 157–170, 2003.
- [26] C. Huepe and M. Aldana, "New tools for characterizing swarming systems: A comparison of minimal models," *Physica A*, vol. 387, pp. 2809–2822, 2008.
- [27] G. Gregoire and H. Chate, "Onset of collective and cohesive motion," *Physical Review Letters*, vol. 92, no. 2, pp. 1–4, 2004.
- [28] M. Aldana, V. Dossetti, C. Huepe, V. M. Kenkre, and H. Larralde, "Phase transitions in systems of self-propelled agents and related network models," *Physical Review Letters*, vol. 98, no. 9, 2007.
- [29] N. Moshtagh, A. Jadbabaie, and K. Daniilidis, "Vision-based control laws for distributed flocking of nonholonomic agents," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Orlando, Florida, USA), pp. 2769–2774, May 2006.
- [30] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents part I: Fixed topology," in *Proc. of the 42nd IEEE Conference on Decision and Control*, vol. 2, (Hawaii, USA), pp. 2010–2015, December 2003.

- [31] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents part II: Dynamic topology," in *Proc. of the 42nd IEEE Conference on Decision and Control*, vol. 2, (Hawaii, USA), pp. 2016–2021, December 2003.
- [32] A. Regmi, R. Sandoval, R. Byrne, H. Tanner, and C. Abdallah, "Experimental implementation of flocking algorithms in wheeled mobile robots," in *Proc. of the American Control Conference*, vol. 7, pp. 4917–4922, June 2005.
- [33] A. Jadbabaie, J. Lin, and A. S. Morse, "Emergent behavior of a system of self-driven particles." -.
- [34] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [35] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [36] A. Cezayirli and F. Kerestecioglu, "Otonom gezgin robotlarin baglantili grup halinde gezinimi," in *Proc. of the TOK'07 Conference*, (Istanbul, Turkiye), pp. 236–241, September 2007.
- [37] M. Lindhe, P. Ogren, and K. Johansson, "Flocking with obstacle avoidance: A new distributed coordination algorithm based on Voronoi partitions," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Barcelona, Spain), pp. 1785–1790, April 2005.
- [38] Y. Hanada, L. Geunho, and N. Chong, "Adaptive flocking of a swarm of robots based on local interactions," in *Proc. of the IEEE Swarm Intelligence Symposium*, (Honolulu, Hawaii), pp. 340–347, April 2007.
- [39] V. Gervasi and G. Prencipe, "Coordination without communication: the case of the flocking problem," *Discrete Applied Mathematics*, vol. 14, no. 3, 2004.
- [40] I. D. Couzin and N. Franks, "Self-organized lane formation and optimized traffic flow in army ants," *Proceedings-Royal Society of London B*, vol. 270, no. 1511, pp. 139–146, 2002.
- [41] M. J. Mataric, *Interaction and Intelligent Behavior*. PhD thesis, MIT, 1994.

- [42] I. Kelly and D. Keating, "Flocking by the fusion of sonar and active infrared sensors on physical autonomous robots," in *Proc. of the Third Int. Conf. on Mechatronics and Machine Vision in Practice*, vol. 1, (Guimaraes, Portugal), pp. 14–17, September 1996.
- [43] A. Hayes and P. Dormiani-Tabatabaei, "Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Washington, DC), pp. 3900–3905, May 2002.
- [44] H. Hu, I. Kelly, D. Keating, and D. Vinagre, "Coordination of multiple mobile robots via communication," in *Proc. of the SPIE'98, Mobile Robots XIII Conference*, (Boston, USA), pp. 94–103, November 1998.
- [45] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Autonomous Robots*, vol. 17, pp. 137–162, 2004.
- [46] O. Holland, J. Woods, R. Nardi, and A. Clark, "Beyond swarm intelligence: the ultraswarm," in *Proc. of the IEEE Swarm Intelligence Symposium*, (Pasadena, California), pp. 217–224, June 2005.
- [47] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameraon, "Experiments in automatic flock control," *Robotics and Autonomous Systems*, vol. 31, pp. 109–117, 2000.
- [48] C. M. Cianci, X. Raemy, J. Pugh, and A. Martinoli, "Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics," in *Second International Workshop on Swarm Robotics at SAB 2006* (E. Sahin, W. M. Spears, and A. F. T. Winfield, eds.), vol. 4433, (Berlin, Germany), pp. 103–115, Springer Verlag, 2006.
- [49] G. Caprari and R. Siegwart, "Mobile Micro-Robots ready to use: Alice," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3295–3300, August 2005.
- [50] U. of Stuttgart, "Open-source microrobotic project." <http://www.swarmrobot.org>, Last visited: March 2007.
- [51] F. Mondada, G. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L. Gambardella, and M. Dorigo, "Swarm-bot: a new distributed robotic concept," *Autonomous Robots*, vol. 17, no. 2-3, pp. 193–221, 2004.

- [52] J. McLurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," in *Proc. of the 7th International Symposium on Distributed Autonomous Robotic Systems*, June 2004.
- [53] D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone robotics," *Autonomous Robots*, vol. 11, pp. 319–324, 2001.
- [54] K-Team, "Khepera ii." <http://www.k-team.com>, Last visited: February 2008.
- [55] G. M. University, "Flockbots." <http://cs.gmu.edu/~eclab/projects/robots/flockbots/pmwiki.php>, Last visited: February 2008.
- [56] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. of the 1st USENIX/ACM Symposium on Network Systems Design and Implementation*, (San Francisco, California, USA), 2004.
- [57] P. Levis and D. Culler, "Mate: A tiny virtual machine for sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. 5, 2002.
- [58] "Tinyos." <http://www.tinyos.net>, Last visited: June 2007.
- [59] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, "Simulation-based optimization of communication protocols for large-scale wireless sensor networks," in *Proc. of the IEEE Aerospace Conference*, vol. 3, (Big Sky, MT), pp. 1339–1346, March 2003.
- [60] T. Balch, "Hierarchic social entropy: An information theoretic measure of robot group diversity," *Autonomous Robots*, vol. 8, no. 3, pp. 209–237, 2000.
- [61] A. E. Turgut, C. Huepe, H. Celikkanat, F. Gokce, and E. Sahin, "Modeling phase transition in self-organized flocks," in *Submitted to ANTS'08 Conference*, 2008.
- [62] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Kimketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, "Centibots: Very large scale distributed robotic teams," in *Proc. of the International Symposium on Experimental Robotics*, 2004.
- [63] L. Brey and F. Guinea, "Phase diagram of diluted magnetic semiconductor quantum wells," *Physical Review Letters*, vol. 85, no. 11, pp. 2384–2387, 2000.

VITA

PERSONAL INFORMATION

Surname, Name: Turgut, Ali Emre

Nationality: Turkish (TC)

Date and Place of Birth: 5 January 1974, Ankara

Marital Status: Single

Phone: +90 312 210 5545

email: ali.turgut@gmail.com

EDUCATION

- MS, METU Mechanical Engineering, 1999
- BS, METU Mechanical Engineering, 1996
- High School, TED Ankara College, Ankara, 1992

WORK EXPERIENCE

- 2004-Present, METU Department of Computer Engineering, Kovan Lab, Research Assistant.
- 2003-2004, Atılım University, Department of Mechatronics Engineering, Instructor.
- 1996-2003, METU Department of Mechanical Engineering, Teaching Assistant.

FOREIGN LANGUAGES

English

RECENT PUBLICATIONS

[1] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in swarm robotic systems," accepted to *Swarm Intelligence: Special Issue on Swarm Robotics* (editors: E. Şahin and A. Winfield), Springer.

[2] E. Şahin, S. Girgin, L. Bayındır and A. E. Turgut, "Swarm Robotics", in *Swarm Intelligence - Introduction and Applications* (editors: C. Blum and D. Merkle), Springer. (in press)

[3] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking with a mobile robot swarm," in *Proc. of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, (Padgham, Parkes, Muller and Parson, eds.), (Estoril, Portugal), May 2008.

[4] E. Uğur, A. E. Turgut, E. Şahin. "Dispersion of swarm of robots based on realistic wireless intensity signals," in *Proc. of the 22nd International Symposium on Computer and Information Sciences (ISCIS'07)*, (Ankara, Turkey), pp. 419-424, November 2007.

[5] A. E. Turgut, F. Gökçe, H. Çelikkanat, L. Bayındır, and E. Şahin, "Kobot: A mobile robot designed specifically for swarm robotics research," Tech. Rep. METU-CENG-TR-2007-05, Dept. of Computer Eng., Middle East Tech. Univ., Ankara, Turkey, 2007.

[8] A. E. Turgut, F. Gökçe, H. Çelikkanat, L. Bayındır, E. Şahin (2007), "Kobot: Sürü robot çalışmaları için tasarlanmış, gezgin robot platformu (Kobot: A mobile robotic platform designed for swarm robotics research)," in *Proc. of Otomatik Kontrol Ulusal Toplantısı (Turkish National Conference on Automatic Control) (TOK'07)*, (Istanbul, Turkey), pp. 259-264, September 2007.

[9] A. E. Turgut, M. Çakmak, E. Şahin and B. Koku. "Oğul robot sistemleri için sinyalleme sistemi tasarımı (Design of a signaling system for swarm robotic systems)," in *Proc. of*

Otomatik Kontrol Ulusal Toplantısı (Turkish National Conference on Automatic Control) (TOK'06), (Ankara, Turkey), pp. 413-418, November 2006.

[10] H. Çelikkanat, A. E. Turgut, E. Şahin and B. Koku. "Oğul robot sistemleri için basit bir görüntüleme sistemi tasarımı (Design of an imaging system for swarm robotic systems)," in *Proc. of Otomatik Kontrol Ulusal Toplantısı (Turkish National Conference on Automatic Control) (TOK'06), (Istanbul, Turkey), pp. 407-412, November 2006.*

HOBBIES

Football, Commodore 64, Music, Technology, Cartoons.