Publication

- Shahwar Yaseen, Yasar Ayaz, Khawaja Fahad Iqbal, Naveed Muhammad, Syed Omer Gilani, Mohsin Jamil, and Syed Iritza Ali. "Centre of mass avoidance planner using radius of gyration for Reciprocal Velocity Obstacles." In Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on, pp. 118-124. IEEE, 2014.

ABSTRACT

Swarm Robotics is the study of swarms of small simple locally behaving mobile robots and their emergent behaviors as a swarm of intelligent species. The inspiration of these studies comes from nature swarms of ants, fish and birds colonies. Many areas of swarms including sensing, localization, mapping, path planning, interference and behavior modeling are being studied currently.

Path planning being one of the old areas of researches has already been studied in great details for static and moving obstacles. The introduction to intelligent obstacles that react to the robots motion are new addition to the research areas. Swarms of robots are numbers of robot trying to accomplish collective or individual tasks. Hence the area of path planning has taken a new dimension. "Reciprocal Velocity Obstacles" is one of the approaches addressing the same problem. It though lacks a mechanism for decision symmetry breaking and biasing. Biases RVO is one attempt to do so, yet the attempt is more of an inspirational concept as it is not much generic in nature and must be modeled for every different scenario.

Getting inspiration from swarm congestion control techniques and ant foraging models, two approaches we tried for implementation for asymmetric biasing and fast solutions to swarm problems. The first one is a much global behavior technique in which the robots try to avoid the center of mass of the whole swarm weighted by inverse of the radius of gyration of the point masses of the robots. It worked well for simulations possible congestion at one point but lacked better results in situations of multiple congestion zones. The second one is a much local approach. In this approach the robots try to follow their most friendly neighbors, weighted by the distance from their respective goal positions. In this way, it was observed that the robots followed a much humble path when they were away from their goals and became ruder as the goals came nearer. The technique proved results in both local and global congestion zones. The efficiency benchmarking for different scenarios varied from 30% in mobile robot pick and place situations to 50% in a symmetric circle antipodal goals problem.

ACKNOWLEDGMENT

I would like to thank Almighty Allah; whose guidance lead this work to be completed in time and whose blessings and benevolent help kept me sheltered all the time.

I would like to thank my advisor Dr. Yasar Ayaz, for his many contributions to both my work as well as my personal and professional development throughout my time at SMME NUST. And for all of the technical insight he has provided and on the improvements in co-curricular skills on numerous occasions throughout the duration and before the start of this thesis work.

I would like to thank my beloved parents and family, who always guided me in the right way and whose endless effort and support made it possible for me, to be, what I am today.

In the end I would like to thank the entire honorable faculty of RIME Department, whose professional approach and vision groomed me.

Shahwar Yaseen

May, 2015

Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
BFS	Breadth First Search
BRVO	Biased Reciprocal Velocity Obstacle
CC	Collision Cone
CM	Center of Mass
CMA	Center of Mass Avoidance algorithm
CMU	Carnegie Mellon University
CPA	Cluster Planning Algorithm
DFS	Depth First Search
DH	Denavit–Hartenberg
DW	Dynamic Window
FOV	Field Of View
GA	Genetic Algorithms
HRVO	Hybrid Reciprocal Velocity Obstacles
IDS	Iterative Deepening Search
KIST	Korea Institute of Science and Technology
MC	Moving Cells
MIT	Massachusetts Institute of Technology
ML	Machine Learning
OA	Obstacle Avoidance
OGM	Occupancy Grid Map

ORCA	Optimal Reciprocal Collision Avoidance
PEAS	Performance Measure Environment Actuators Sensors
R_C	Radius of Congestion Circle
R_G	Radius of Gyration
RVO	Reciprocal Velocity Obstacles
SCARA	Selective Compliance Assembly Robot Arm
SI	Swarm Intelligence
SLAM	Simultaneous Localization And Mapping
SRI	Stanford Research Institute
VO	Velocity Obstacles

Chapter 1:

Introduction

1.1 Background of Robotics

The history of the concept of robotics dates back centuries. People from the earliest of times have tried to produce machines, mechanism that would produce appraisable actions on their own. People as historic as Da Vinci and even older were fascinated by the idea of mechanical human beings and other machines that could automatically work without much human intervention needed. But the mentionable robotics dates back only few decades. In 1942, a paint-sprayer was designed that could be programmed to perform certain tasks by W. Pollard and H. Roselund for the DeVilbiss Company [1]. In 1946, a general purpose playback device was patented by G. Devol. It was designed for controlling machines using magnetic recordings. In 1951 R. Goertz designed the first tele-operated articulated arm. Generally, this is regarded as a major milestone in force feedback (haptic) technology [2]. In 1954 G. Devol designed the first truly programmable robot called UNIMATE "Universal Automation." [3]. Later, in 1956, G. Devol and J. Engelberger formed the world's first robot company “Unimation” for “universal automation”. Engelberger has been called the 'father of robotics’. One of the first industrial robots became operational in North America in the early 1960’s in a candy producing factory in Ontario. In 1964 MIT, Stanford University and the University of Edinburgh establish Artificial intelligence research laboratories. In 1965, the Robotics Institute is established at Carnegie Mellon. In 1968 Mcgee and Frank created the first computer controlled walking machine at the University of South Carolina. In 1968, SRI built a mobile robot that was equipped with a vision system. In 1969 Ichiro Kato designed WAP-1 that became history’s first biped robot. Air bags connected to the frame were used to stimulate artificial muscles. Later, WAP-3 was designed that could walk on flat surfaces and climb up and down stairs or slopes. In 1977, Dr. Devjanin created a six-legged walking machine with his colleagues at the Russian academy of Science. In 1988, the first service robot was deployed at Danbury Hospital Connecticut. In 1996, P2 was created at HONDA. This was the first step towards creating ASIMO. In 1997 PathFinder robot landed on Mars. The wheeled rover was sent to send images and other data about Mars to Earth. In 1998, LEGO released MINDSTORMS, a development product for robotics. MINDSTORMS is designed for inventing robots using a modular design and LEGO plastic bricks. In 2001, iRobot Packbots searched through the rubble of the world Trade Center. The subsequent versions of the Packbot robots were used in Afghanistan and Iraq. In 2002, iRobot released the first generation of Roomba robotic vacuum cleaners. In 2005, The Korean Institute of Science and Technology (KIST), created HUBO, and claims it is the smartest mobile robot in the world. This robot is linked to a computer via a high-speed wireless connection; the computer does all of the thinking for the robot. 2008 brings the popular Roomba robotic vacuum cleaner over 2.5 million units sale, proving that there is a strong demand for this type of domestic robotic technology.

1.2 Types of robots

There are many classification criteria that are used to classify robots into different categories. There is not one single generic criterion that would classify all robots strictly into distinct useful classes. Yet we mention here some of the commonly known classification criteria and the categories of robots according to these criteria.

- Arm configuration
- Shape of workspace
- Operating method
- Type of controller
- Type of power
- Size
- Type and number of joints
- Type of technology
- Tasks being performed
- Generation of design
- Type of motion

Different robot classes under these criteria are discussed as under

1.2.1 Classification by Arm Configuration

- Rectangular coordinates robots
- Cylindrical coordinates robots
- SCARA robots
- Polar coordinates robots
- Jointed arm Robots

1.2.2 Classification by Controller

- Limited Sequence Robots

  These use mechanical stops to limit their movement
- Point to Point Robots

  These robots have sensory feedback and they hold memory of coordinates of each axis
- Continuous Path Robots

  They have greater memories than point to point robots and can record many coordinates

1.2.3 Classification by power supply

- Electric Powered
- Pneumatic Powered
- Hydraulic Powered

1.2.4 Classification by level of technology

- Low-tech level

  They are used for material handling, pick and place, loading unloading operations
- Medium Tech level

  They are used sorting operations and other medium tech operations
- High-tech

  They are used in intelligent operations and sophisticated tasks

1.2.5 Classification by design

- First generation

  These robots were fixed sequence robots with no sensors on board
- Second generation

  These robots can adapt to situations by use of sensors and a closed loop control
- Third generation

  These robots are intelligent and can react to situations instantaneously

1.2.6 Classification by joint configuration

- Cartesian only
- Cylindrical
- Polar
- SCARA
- Revolute

1.2.7 Classification according to mobility

- Fixed robots

  Suitable for industrial applications
- Flexible robots

  Suitable for delicate light payload applications
- Mobile robots

  Suitable for applications involving land navigation
- Flying robots

  Suitable for aerial applications
- Underwater robots

  Suitable for underwater applications

1.3 Introduction to robot kinematics

Robot kinematics discusses the geometry of motion of robots, it addresses all possible orientations of robots. We try to create a workspace description of a robot and then see if we can solve given problems with provided link configurations. DH notation is commonly used in solving kinematic problems.

1.4 Introduction to robot dynamics

The robots are modeled for real world applications. Without the knowledge of dynamics, we are unable to predict whether a certain task can be accomplished at a certain speed or not. Robot dynamics studies address the problem of perform ability of tasks and provides feasible workspaces and path plans. Jacobians are usually used to calculate robot dynamics.

1.5 Introduction to Artificial Intelligence

In order to understand Artificial Intelligence, we must try to understand Intelligence itself. According to Webster's dictionary, it is the capacity to learn and solve problems. The word artificial intelligence on the other hand can be defined my four categories [33]

Thinking Humanly	**Thinking Rationally**
Acting Humanly	**Acting Rationally**

Table 1: Four Approaches to AI

1.5.1 Thinking Humanly

This approach involves actually trying to understand our brain and try to replicate that on machines and computers.

1.5.2 Acting Humanly

This approach involves the machines trying to act like humans without having the knowledge of the logical explanation behind an act

1.5.3 Thinking Rationally

This approach is based upon logical serial contact formation and solving logical problems with rational thinking approach. With given constraints and functions, a robot should be able to think rationally for the solution based on logical reasoning.

1.5.4 Acting rationally

This approach expects the Intelligent agents to perform rationally even in cases where the agents don't have any idea as to why were these decisions made. The only thing that counts in this approach is that to what extent a robot was able to make rational action decisions.

In order to describe Artificial Intelligence, we can say that it is the combination of Philosophy, Mathematics, Neuroscience and Psychology inside an artificial brain called a computer or a machine these days.

1.5.5 Agents

Agents can be defined as anything that can perceive its environment using sensors and act upon it using its actuators. This broad definition puts all living organisms under the hood of agents. But in robotics, we discuss agents as artificially produced species that can sense the environment and act upon it following mathematical principles.

Agents are usually categorized using the following four criteria

- Performance Measure

  What exactly is the above mentioned agent supposed to do?

- Environment

  What environment is it put in?
- Actuators

  What tasks can it perform using its actuators?
- Sensors

  What information can it get from the given above stated environment using its sensors?

In general, agents can also be classified as to the level of complexity they possess. Some agents are simple, they react as reflexes to the changes in the environment that they perceived through on board sensors. Others are model based, that react to the environment if it changes according to their model of understanding it. Some others try to optimize their own goals. Yet others try to learn from the general trends of a normal environment and try to figure out whenever something goes wrong and take necessary decisions.

1.5.6 Environments

We as humans live in the world and we perceive it in a continuous fashion. We can detect minute changes to our environments and react to them to either rectify the problems or to enjoy the perks. Robots on the other hand must be designed for multiple types of environments discreetly. In order to describe the environments, we need to develop some criteria to present the model to the robot. Some criteria are mentioned below

Observability (full or partial)

It describes as to what extent the environment can be observed by the agent

Number of Agents (Single or Multi Agent)

It describes as to whether the agent is alone in the near vicinity or are other agents also around

Deterministic (Deterministic or Stochastic)

It describes as to whether the environment is Deterministic i.e. can be predicted in the near future or it is stochastic or unpredictable

Episodic (Episodic or Sequential)

It describes whether the environment moves in discrete episodes or does it move sequentially with respect to previous states as in a game of chess

Static or Dynamic

Whether the states of objects / concepts around the agent remain stationary with time without agent's intervention or do they change due to other factors as well with time.

Discrete or Continuous

Whether the environment consists of finite number of discrete states or does it exist as infinite and continuous domain of events

1.5.7 Introduction to state space searching

For humans, the world is full of infinite possibilities and we have learned to make decisions as to what tasks to perform and what not to do. The robots on the other hand must develop a state space and then search for a feasible solution to maximize their performance criteria. The state space can be discrete or it can be continuous thus practically expanding the possible orientations and combinations to infinity. The robots must learn to find out a feasible solution within a rational time span so that the action that must be taken within time may not get delayed. Many strategies have been developed to search faster within the state space of the agents. The state space searching problem consists of the following attributes

- All the states
- Initial State
- Goal State
- Possible actions
- Costs of actions
- Heuristics to goals

After inputting this information, the nodes are created, the trees and graphs are created and then a suitable search algorithm is deployed to search for a solution.

- Breadth first search BFS
  Searches for node expansion with breadth priority
- Depth first search DFS
  Expands nodes with depth priority
- Iterative deepening search IDS
  Iteratively deepens the DFS
- Best first search
  Searches for the past best solution
- A* search
  Searches for the past as well as future expected heuristic best solution
- Genetic algorithms
  This family of algorithms try to solve problems using laws of genetics

"Some of the grand challenges in science and technology are that we are still understanding the brain such as how does the brain do reasoning, cognition, creativity etc. and we want to be creating intelligent machines. Arguably AI poses the most interesting challenges and questions in computer science today."

1.6 Introduction to Swarm Robotics

Swarm robotics is defined as the study of coordinating large groups of relatively simple robots by using local rules. Swarm robotics is inspired by insects' societies that are able to perform tasks that may be beyond the capabilities of individual robots. Reference [4] explains this kind of robots' coordination as follows:

"The group of robots is not just a group. It has some special characteristics, which are found in swarms of insects, that is, decentralized control, lack of synchronization, simple and (quasi) identical members."

1.6.1 Social Insect Motivation and Inspiration

The social insects' collective behaviors, such as the dance of the honeybee, the nest building of the wasp, the termite mound's construction, or the ants' trail following, have been considered for a long time strange and mysterious aspects of biology. In recent decades, Researchers have showed that individual agents do not need any representation or sophisticated knowledge to produce such complex behaviors [5]. In socially active insects, the individuals are not informed about the global status of the colony. There exists no leader that guides all the other individuals in order to accomplish their goals. The knowledge of the swarm is distributed throughout all the agents, when an individual is not able to accomplish its task without the rest of the swarm. Social insects are able to exchange information, and for instance, communicate the location of a food source, a favorable foraging zone or the presence of danger to their mates. This interaction between the individuals is based on the concept of locality, where there is no knowledge about the overall situation. The implicit communication through changes made in the environment is called Stigmergy [6, 4]. Insects modify their behaviors because of the previous changes made by their mates in the environment. This can be seen in the nest construction of termites, where the changes in the behaviors of the workers are determined by the structure of the nest [7]. Organization emerges from the interactions between the individuals and between individuals and the environment. These interactions are propagated throughout the colony and therefore the colony can solve tasks that could not be solved by a sole individual. These collective behaviors are defined as self-organizing behaviors. Self-organization theories, borrowed from physics and chemistry domains, can be used to explain how social insects exhibit complex collective behavior that emerges from interactions of individuals behaving simply [7]. Self-organization relies on the combination of the following four basic rules: positive feedback, negative feedback, randomness, and multiple interactions [7]. Şahin [8] lists some properties seen in social insects as desirable in multi-robotic systems: robustness, the robot swarm must be able to work even if some of the individuals fail, or there are disturbances in the environment; flexibility, the swarm must be able to create different solutions for different tasks, and be able to change each robot role depending on the needs of the moment; scalability, the robot swarm should be able to work in different group sizes, from few individuals to thousands of them.

1.6.2 Swarm intelligence

As it is an emerging area of research, many researchers have been attracted to the swarm intelligence. The concept was introduced in 1980s. It now has become a frontier for interdisciplinary studies and many disciplines focus on it such as artificial intelligence, sociology, economics, biology, etc. It has been observed that some species survive in nature taking the advantage of the potential power of swarms, rather the wisdom of individuals. The individuals in swarms are not much intelligent, but they can complete complex tasks by cooperating and division of labor. Therefore showing high intelligence as a whole swarm which is highly self-organized and self-adaptive. Swarm intelligence is a soft bionic of the nature swarms, i.e. it simulates the social structures and interactions of the swarm rather than the structure of an individual in traditional artificial intelligence. The individuals can be regarded as agents with simple and single abilities. Some of them have the ability to evolve themselves when dealing with certain problems to make better compatibility [9]. A swarm intelligence system usually consists of a group of simple individuals autonomously controlled by a plain set of rules and local interactions. These individuals are not necessarily unwise, but are relatively simple compared to the global intelligence achieved through the system. Some intelligent behaviors never observed in a single individual will soon emerge when several individuals begin cooperate or compete. The swarm can complete the tasks that a complex individual can do while having high robustness and flexibility and low cost. Swarm intelligence takes the full advantage of the swarm without the need of centralized control and global model, and provides a great solution for large-scale sophisticated problems.

1.6.3 Main Characteristics

In order to understand what swarm robotics is, a definition taken from Sahin [8] is given:

"Swarm robotics is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behavior emerges from the local interactions among agents and between the agents and the environment."

This definition is complemented with a set of criteria in order to have a better understanding and be able to differentiate it from other multi-robot types of systems [8].

i. The robots of the swarm must be autonomous robots, able to sense and actuate in a real environment.
ii. The number of robots in the swarm must be large or at least the control rules allow it.
iii. Robots must be homogeneous. There can exist different types of robots in the swarm, but these groups must not be too many.
iv. The robots must be incapable or inefficient respect to the main task they have to solve, this is, they need to collaborate in order to succeed or to improve the performance.
v. Robots have only local communication and sensing capabilities. It ensures the coordination is distributed, so scalability becomes one of the properties of the system.

1.6.4 Swarm Robotics and Multi-Robotic Systems

There exist several research areas inspired from the nature swarm, which are often confused with swarm robotics, such as multi-agent system and sensor network. These research areas also utilize the cooperative behavior emerged from the multiple agents in the group for specialized tasks. However, there are several differences between these systems, which can distinguish these systems fundamentally, as shown in Table 2.

From Table 2, it can be easily deduced that the main differences among swarm robotics and other systems are population, control, homogeneity and functional extension. Multi-agent and sensor network systems mainly focus on the behaviors of multiple static agents in the known environments while the robots in the multi-robot systems are quite small, usually heterogeneous and are externally controlled.

Table 2: Comparison of Swarm robotics to other multi agents systems

Comparison of swarm robotics and other systems.				
	Swarm robotics	**Multi-robot system**	**Sensor network**	**Multi-agent system**
Population Size	Variation in great range	Small	Fixed	In a small range
Control	Decentralized and autonomous	Centralized or remote	Centralized or remote	Centralized, hierarchical or network
Homogeneity	Homogeneous	Usually heterogeneous	Homogeneous	Homogeneous or heterogeneous
Flexibility	High	Low	Low	Medium
Scalability	High	Low	Medium	Medium
Environment	Unknown	Known or unknown	Known	Known
Motion	Yes	Yes	No	Rare
Typical applications	Post-disaster relief	Transportation	Surveillance	Net resources management
	Military application	Sensing	Medical care	Distributed control
	Dangerous application	Robot football	Environmental protection	

Since homogeneity and scalability are considered at the beginning of the system design, the swarm robotics shows great flexibility and adaptability compared with other systems. The multi-robot systems usually involve the heterogeneous robots, and may achieve better performance on specialized tasks at the cost of flexibility, reusability and scalability. Besides scalability which is introduced in previous section, the characteristics of swarm robotics among other three cooperative systems are listed in Table 1.

- ## Autonomous

The individuals in swarm robotics systems must be autonomous, i.e. capable of interacting and motioning in the environment. With these key functions, the cooperative mechanisms inspired from the nature swarms can be introduced into the swarm robotics. Although the systems, like sensor networks, are far different from the swarm robotics from such point of view, but the research on the area can indeed throw some lights on swarm robotics research.

- ## Decentralization

With a good set of cooperative rules, the individuals can complete the task without centralized controls which promises the scalability and flexibility of the swarm. At the same time, the swarm can benefit more in the environments when communication is interrupted or lagged and improves the reaction speed and precision of the swarm.

- **Local sensing and communications**

Due to the restriction of hardware and cost, the robots in the swarm usually have a limited range of sensing and communicating and thus the whole swarm is distributed in the environment. Actually, the use of global communications will lead to a significant decline in scalability and flexibility, as the communication cost is explode exponentially as the population grows. Nevertheless, certain controlling global communications are acceptable, for instance, updating the controlling strategies or sending the terminal signals, so long as it's not used in the interaction between individuals.

- **Homogenous**

In a swarm robotics system, the robots should be divided into the roles as few as possible and the number of robots acting as each role should be as large as possible. The role here indicates the physical structure of the robot or other states that cannot be changed into one another dynamically during the task. A state in a finite state machine does not count in our definition. This definition indicates a swarm, no matter how large it is, is not considered as swarm robotics if the roles of robots are divided meticulously. For instance, the robots football usually is not considered as swarm robotics, since each individual in the team is assigned a special role during the game.

- **Flexibility**

A swarm with high flexibility can deal with different tasks with the same hardware and minor changes in the software, as the nature swarms can finish various tasks in the same swarm. The individuals in the swarm show different abilities and cooperation strategy when they deal with different tasks. The swarm robotics should provide such flexibility, especially in similar tasks, such as foraging, flocking or searching. The swarm can switch to different strategies according to the environment. The robots can adapt to the environment through machine learning from the past moves and can change to a better strategy.

1.6.5 Common areas of research in swarms

- **Flocking**

Flocking is observed in many social animals including humans. Insects, birds, fish try to move around and navigate in flocks. The numbers tend to be enormous yet it can be seen that the even the birds at the most inner core of the flock do not experience interference from other birds around. In this area, researchers are trying to simulate the same using mathematical models

- **Directed flocking**

With the above mentioned flocking strategies, there emerges another problem in swarms. It becomes difficult for the robots to keep attached to the flock, yet lead the flock in the right direction. Many researchers have tried to address the problem and proposed different methods of solving it.

- **Position and navigation**

In flocking and migration, the positioning of goal, nearby robots and various obstacles in the fields is also an important task. In the application taking place in the large outdoor environments, the global positioning is expensive and requires more hardware, which is unaffordable for swarm robotics. Thus, the local positioning in flocking should be specially focused

- **Obstacle avoidance**

Since obstacle avoidance is key to mobile robotics in particular and to robotics in general, swarm robotics also focuses on obstacle avoidance. Many researchers have tried to bring older obstacle avoidance methods to swarms, yet many others have tried to devise newer methods for swarm obstacle avoidance owing to the fact that they now face different environment conditions as compared to simple mobile robot navigation and path planning problems.

Chapter 2:

Existing methods of Obstacle Avoidance and their analyses

2.1 Obstacle Avoidance

The ability of mobile robots to navigate safely and avoid obstacles static and mobile is necessary for many applications in the real world. Path planning in a dynamic and changing environment is still one of the most difficult and important problems in the field of mobile robotics. The literature that addresses this problem is rapidly growing. Some of the work relating mobile robot obstacle avoidance is discussed in this chapter. Firstly we shall discuss techniques that were originally designed for static obstacle avoidance for mobile robots but later imported to solve for dynamic environments. Then we will discuss the work of people that designed techniques that were specifically modeled, considering the dynamic and changing nature of the environment. Then we will discuss the techniques that were designed for the obstacle avoidance considering the presence of other intelligent agents within the environment. We also intend to discuss the problems related to the above mentioned techniques.

2.1.1 Types of obstacles

According to the dynamic nature of the obstacles, we categorize the obstacles into the following categories

- **Static obstacles**

These obstacles are objects in the environment that do not move. Stationary obstacles limit the workspace of the robot in 3D or 2D space and does not have any effect on changing the path planning approach due to their inherent stationary nature.

- **Dynamic obstacles**

These obstacles are dynamic in nature. Any obstacle that can possible change its position in the world is generally brought under this category

  - **Dumb dynamic obstacles**

These obstacles are dynamic in nature but they keep following their path no matter what else happens in the environment. The following two classes can be brought under dumb moving obstacles

    - **Mathematically modeled obstacles**

These obstacles can be modeled mathematically and predicted in future time. These may include obstacles with constant rectilinear or angular velocity or acceleration. For example an asteroid moving with constant velocity in free space, or an asteroid falling towards the surface of the earth with changing acceleration following the laws of gravitation and friction.

- **Stochastically moving obstacles**

These include obstacles with no predictable mathematical model. Although for all motion that happens within the universe, there exists a mathematical model. But we, in this category bring all the stochastic objects that we do not know the model of. For example a broken (software or hardware) robot acting weird.

Intelligent obstacles

These obstacles are actually other species in the environment that would react to a change in their environment as well. All living organisms are included in this category

- **Artificially intelligent obstacles**

These include other agents in the environment. They can further be divided into two categories as whether they are cooperative or competitive but as we study swarms of robots, we talk about cooperative fellow agents.

- **Really intelligent obstacles**

These include humans, animals, insects and birds. All naturally occurring intelligent species. Of course they all have different level of intelligence and different reaction speeds but one must consider that these species are robust and react to a change in much different way than artificially intelligent ones.

2.2 Modified classic approaches for dynamic environments

We now present the approaches that were initially created to avoid static or stationary obstacles. These approaches are the modified versions of the classic approaches to incorporate dynamic nature of the environment.

2.2.1 Roadmap based dynamic path planning

Roadmaps are possible paths for robot motion. In roadmap based path planning, a search tree is expanded with discrete space locations as its states and the goal node is found by searching the whole tree. The robot's motion constraints determine the roadmap's possible actions. In [10], van den Berg et al. presented a roadmap based solution to dynamic environments. The state space also involved time as a variable as the obstacles changed their locations with respect to time as well.

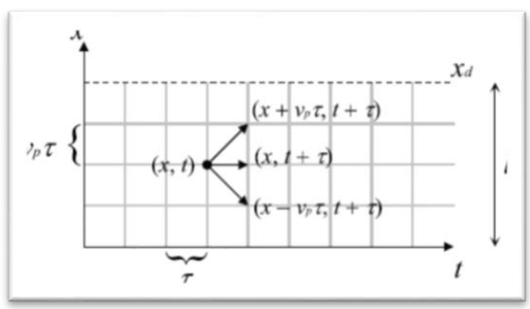


Figure 1: Showing state space with 1D space

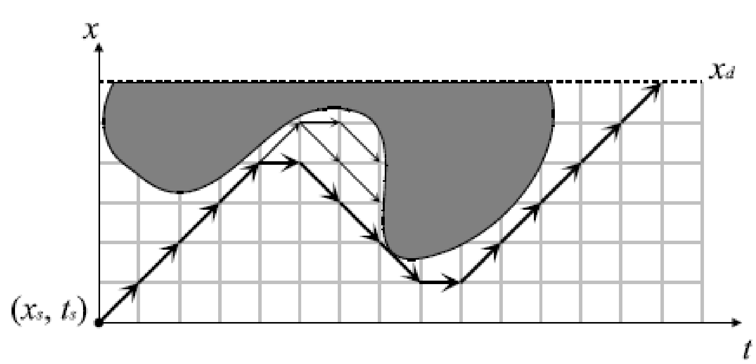


Figure 2: Obstacle avoidance in space time graph

The tree is to be expanded considering the fact that there are obstacles that move around in space and all their possible future positions must be avoided as described in Figure 2.

First of all, this technique lacked the sophistication of models that cannot be predicted for in the future. Also, it created problems like heavy interference of obstacles and subsequent long paths before reaching the goal. One of the results have been shown below for two obstacles moving. First obstacle ‘A’ in an H shape trajectory and the second ‘B’ in a rectangular shaped trajectory. The robot is to move from‘s’ the start state to ‘g’ the goal state. The results show the heavy interference in Figure 3.

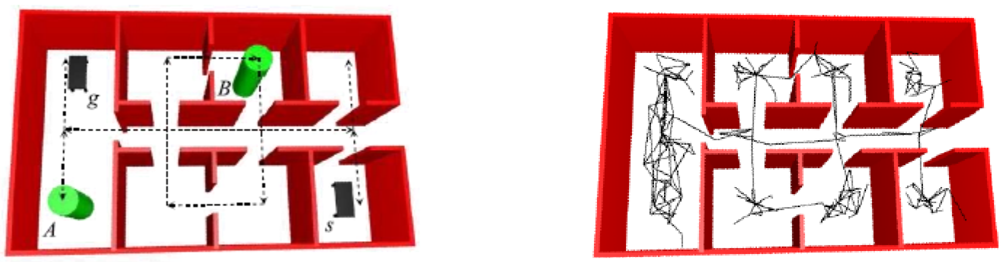


Figure 3: The roadmap technique took a very long route to goal

2.2.2 Dynamic potential field function

The dynamic potential field [11] is a dynamic incarnation of the classic potential field technique. In this technique the potential field of the static objects has been changed to incorporate the velocities of the obstacles as well. The obstacles hold a higher field lift in the direction of their motion hence making the possible collision detectable. But it also carries with it the problem of local minimum. This problem itself is also inherited from the technique of potential fields.

Figure 4 shows the local minimum problem in static potentials with free path to goal.

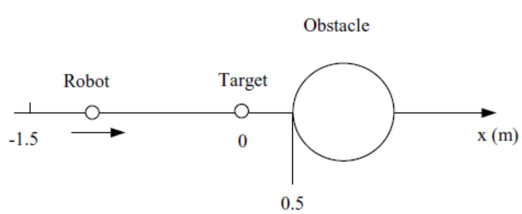


Figure 4: Local minimum problem in static obstacles

Figure 5 shows the local minima problem in dynamic potential fields. The technique originally was not designed to cater for the velocity of the obstacles and had no mechanism of predicting the local minima so it inherited the same problem in the modification as well

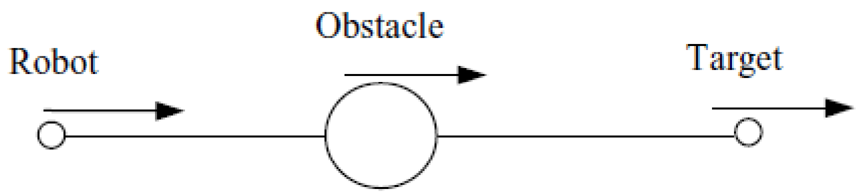


Figure 5: Showing dynamic local minima

2.2.3 Dynamic window approach

DW approach [12] was originally built for static obstacles. It considered the dynamics of the robot itself and calculates all possible paths traversable with the given kino-dynamic constraints. The robot senses a collision if an obstacle comes too close to its dynamically obstructed area. The new approach also includes the trajectories of moving cells that represent obstacle boundaries. The colliding paths are avoided and the motion is optimized to the shortest path to goal. This technique lacks any reactive nature of the obstacles, and cannot plan its path on real time as it being computationally heavy. Figure 6 shows the dynamic window trajectories and subsequent colliding moving cell trajectories.

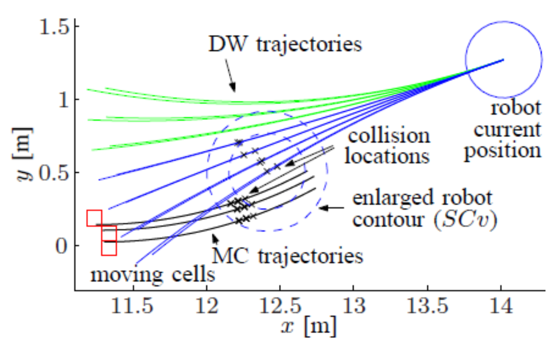


Figure 6: DW and MC trajectories colliding

2.2.4 Path velocity decomposition

This approach deals with the dynamic nature of the environment by

1. Avoiding all the static obstacles to generate a path
2. Adjusting the velocity of the robot to avoid all dynamic obstacles that are supposed to cross the path in a space-time graph

The problem with this kind of approach is that if an obstacles stops somewhere or changes its trajectory on its way, the re-planning of path is not being done online. The reactive nature of this algorithm is limited to the assumption that the environment is free of sudden changes. Another modification to this classic technique is presented in [13]. It uses an Occupancy Grid Map to avoid moving obstacles and tries to predict the motion of obstacles with a probabilistic approach. But this method also is not reactive in nature so it also fails in case of a change in the dynamic environment while execution.

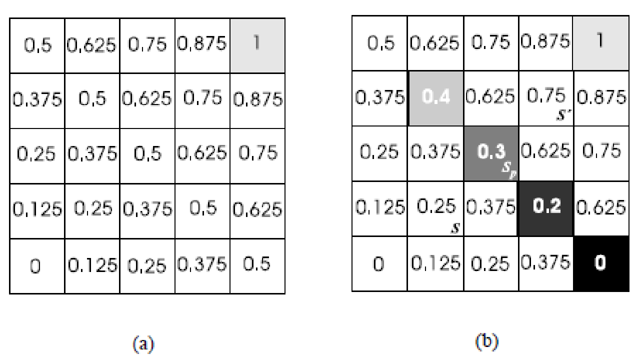


Figure 7: The OGM and a robot trying to move with fast velocity

2.3 Kinematic approaches

These approaches were developed to react to instantaneous collision paths. The following approaches consider the possible collision within a small interval of time ‘t’. The reactive nature requires simpler algorithms that could compute collision free paths for a small time in the future so that the whole path to the goal is the final sequence of the small intervals of the decisions.

2.3.1 The collision cone approach

This approach [14] was introduced to cater for the collision cone that is created by a point robot trying to avoid a circular obstacle. The collision cone that has been developed to avoid the object has been created by two tangents to the circle. The approach is then extended to cater for the collision between a point and a circular obstacle, between a point and an irregular obstacle, between two circular robots, and between two irregularly shaped agents. The approach computes tangents to differentiate between collision course and free available area.

2.3.2 The Velocity Obstacle approach

The velocity obstacle approach [15] considers an agent trying to avoid a linear moving obstacle for a small time interval ‘t’. The time interval is infinitesimal so the observed velocity can be taken as constant over the interval. The collision cones are created by using the above approach and areas for obstacle avoidance are created. This approach also considers the mobile nature of the obstacle and shifts the origin of the collision cone. It then maps the collision cones over the set of admissible accelerations that are provided from the dynamic constraints of the robot. By superimposing the above two, an optimization problem is created. Now based upon the information of the goal location, the optimization problem is solved to maximize speed, avoiding all obstacles and reaching the goal as fast as possible. Figure 8 shows what the optimization problem looks like

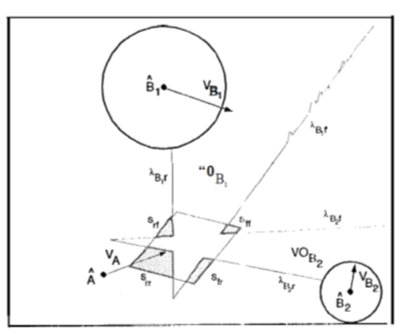


Figure 8: Classification of the reachable avoidance velocities

2.3.3 The Reciprocal Velocity Obstacles

The velocity obstacles approach is reactive in nature and contains all the necessary obstacle avoidance measures considering the kino-dynamic constraints. The VO has an inherent problem. It does not consider the reactive nature of the obstacles. It considers the obstacles to be dumb moving obstacles in the environment. In multi-agent systems, the obstacles are artificially intelligent. Moreover, if the robot is deployed in a human or animal interaction workspace, the environment also has real intelligent species in it.

The problem that was created due to such negligence is that the robots tried to optimize their path to goal on every time-step. So at time ‘t’, the robot A avoids robot B and B avoids A to completely alter their path and go into optimally non colliding directions. But as soon as the next cycle of perception is completed, both the robots see that their original path plan is clear to move and they simultaneously come back to the collision path again. This created robotic dances, the kind of dances that are even common in really intelligent species like humans as a result of poor perception of the path plan of the oncoming. The RVO [16] was designed to reduce robotic dances due to this flaw.

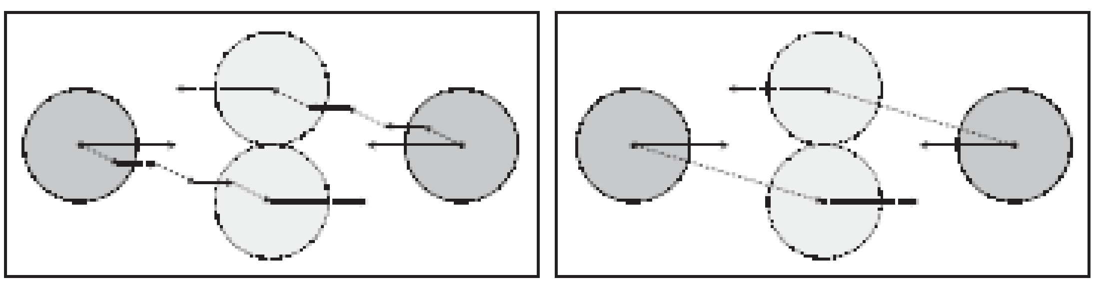


Figure 9: Two robots deploying VO (left) and RVO (right)

RVO plans by calculating the VOs and then each robot tries half the responsibility to avoid the oncoming. In this way, the previous path plan never becomes feasible again and the robots navigate without dancing. Figure 10 shows the VO and the RVO cones with agents A and B.

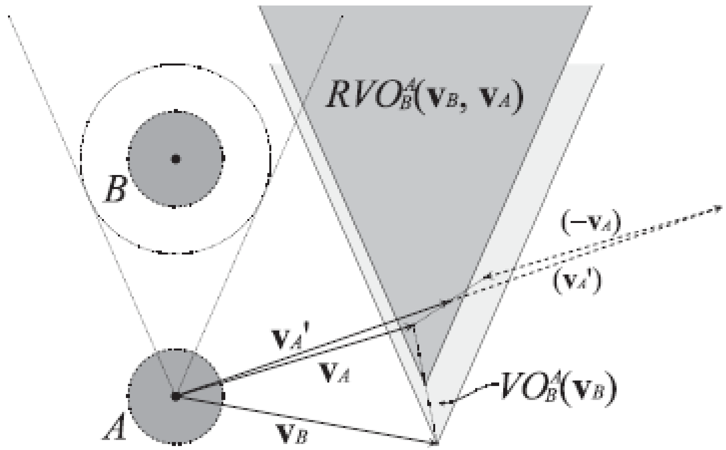


Figure 10: The VO and the RVO collision cones

2.3.4 Optimal Reciprocal Collision Avoidance

The RVO is extended to cater for many robots at a time and create an optimization problem in 2D space in ORCA [17]. It computes the best possible action to be taken as a reaction to number of robots obstructing the traversable immediate environment. Figure 11 shows ORCA in action.

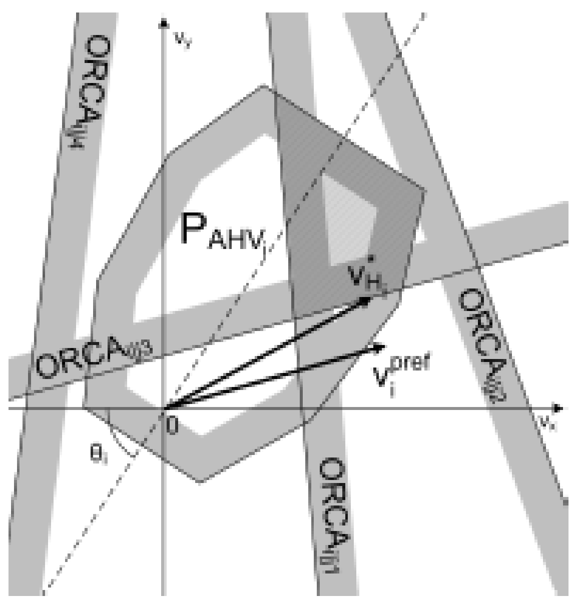


Figure 11: ORCA with a number of immediate neighbors

2.3.5 Biased Reciprocal Velocity Obstacles

Although RVO is a good reactive path planning approach, it lacks a mechanism of breaking the symmetry that is inherent with the collision cones. The perfectly symmetric cones create decision making problems as to how to solve situations where n number of robots try to solve a perfectly circular formation. Biasing of robots is discussed in this approach [18] and criteria for explicitly developing the biasing are discussed

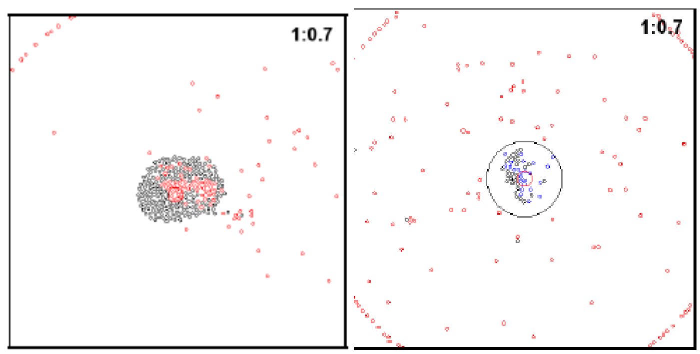


Figure 12: The effects of introducing an explicit biasing scheme

2.3.6 The Hybrid Reciprocal Velocity Obstacle

In this approach [19], the symmetry breaking is made possible by making the collision cone asymmetric and the effects are proven to get better.

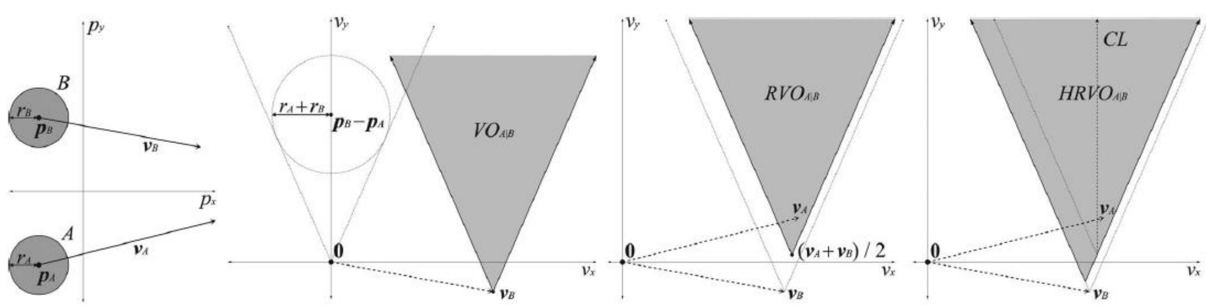


Figure 13: The VO, RVO and HRVO collision cones for robots A and B

The HRVO approach shifts its cone by considering RVO cone on one side of the triangle and VO on the other side. Since both the schemes guarantee obstacle avoidance, the path is collision free in general. But added advantage is of oscillation free plans.

Chapter 3

Generic Biasing to Reciprocal Velocity Obstacles

3.1 Inspirational approaches

We now present some bio inspired techniques that are used in literature for mimicking natural swarm behavior and their effects on general performance of the swarm are studied. We also present the bio inspired biasing criteria that are generic in nature and their workability is discussed

3.1.1 Interference as a major problem

In [20], the authors discuss the problem of interference in an ant foraging task. The obstacle avoidance, or it may be called as reciprocal collision avoidance is one of the major hindrances in efficient foraging task. Figure 14 shows a general picture of possible foraging interference occurrences.

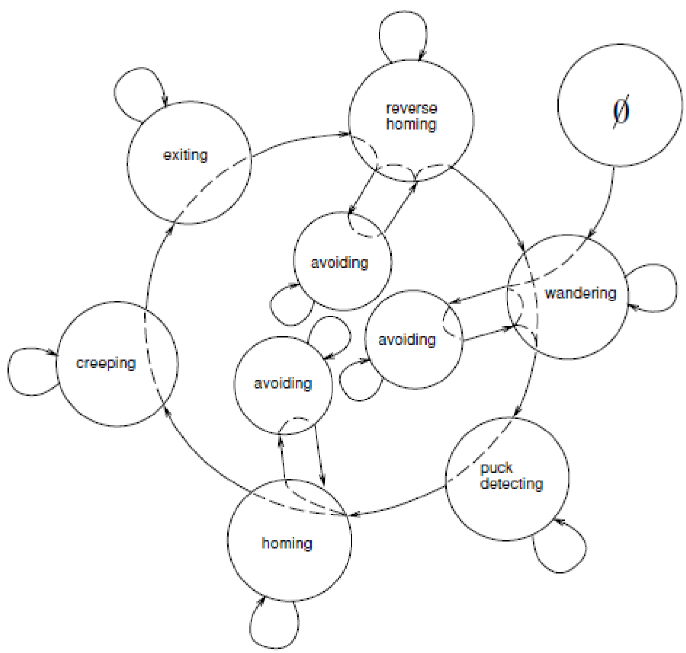
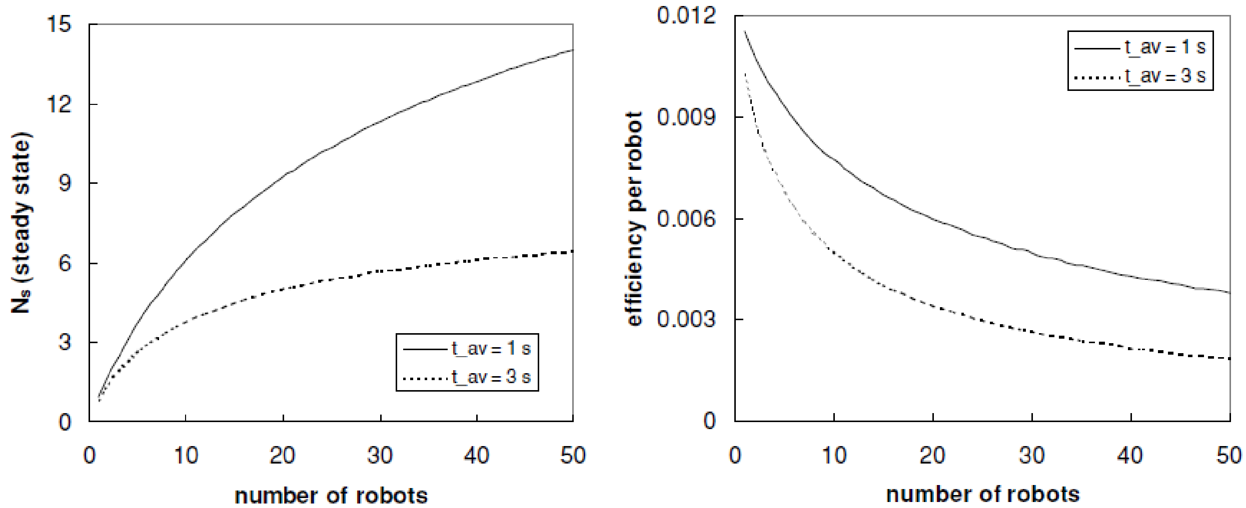


Figure 14: General ant foraging work cycle

The performance of foraging is also mapped by increasing the number of agents to the simulation. As we increase the number of agents in the simulation, the overall efficiency of the swarm gets high as there increases the probability of finding food. But the efficiency per robot gets low due to interference involved. Figure 15 shows the effects of reduced time in avoiding on scaling and efficiency of the swarm and the individual agents.



(a) (b)

Figure 15: Efficiency of the swarm and agents within the swarm

3.1.2 Bucket brigading

To solve the problem of interference, one approach [21] is to pass on the resources to other robots in a mobile robot pick and place operation. The technique has proven its applicability in narrow passages but has also failed to perform in open spaces.

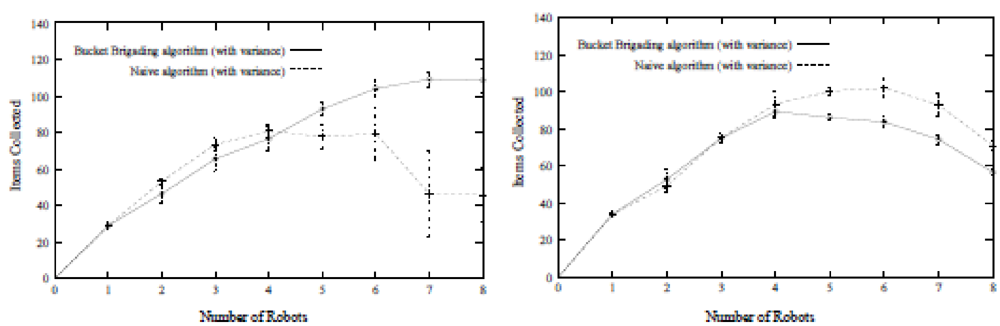


Figure 16: Better results in narrow passages (left) and worse in open fields (right)

3.1.3 Congestion control

Many attempts have also been made to control the congestion of swarm agents on workstations. One of them is [22] that describes an explicit criterion for each simulation and then discusses the effects of different strategies on different scenarios.

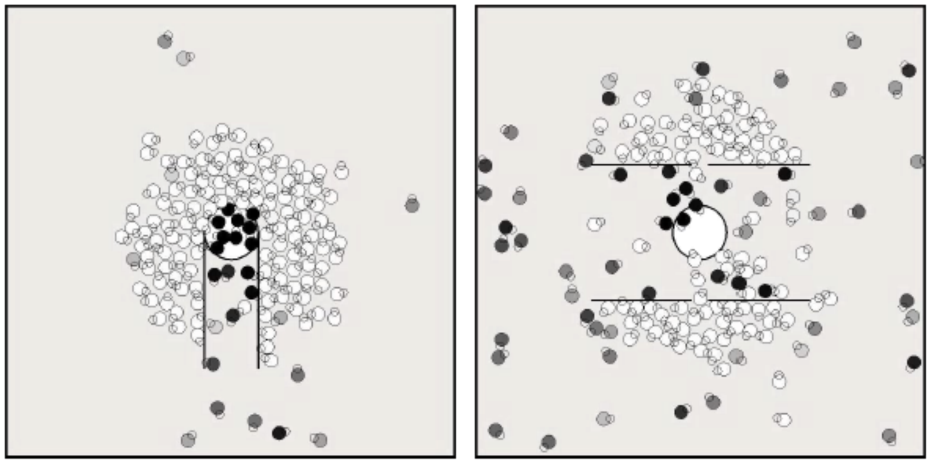


Figure 17: Showing successful congestion control using pipe method and wall method

3.1.4 Waypoint detection with a smaller field of view

This technique [23] reduces interference in robots by using smaller field of view for pheromone detection in robots. This creates multiple local routes with relatively smaller path lengths, yet not a single path with high interference for the agents. Figure 18 shows the paths formed in one simulation.

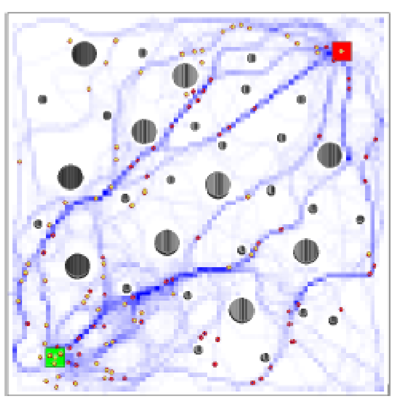


Figure 18: Multiple path creation due to local pheromone detection

3.2 Generic Biasing Criteria

The following is my work in biasing the robots using generic criteria. The ideas are inspired from the approaches mentioned above, yet the need of explicitly defining a criterion for each simulation has been eliminated. RVO has been used as the obstacle avoidance scheme here, but the preferred velocity vector is modified in the following approaches.

3.2.1 Centre of mass avoidance for congestion control

In this work [34], I have devised a generic method of congestion control based on two major attributes of large masses of robots considering all agents as point masses.

- The center of mass of the swarm
- The radius of gyration of the swarm

CMA uses a goal vector modification approach by changing the goal vector of each robot. Firstly, the center of mass of the swarm is calculated as

$$CM = \frac{1}{N}\sum_{i=1}^{i=N}(x_i, y_i)$$

Where:

CM = Coordinates of mass centre of the swarm

N = Number of robots in the simulation

(x_i, y_i) = The coordinates of i^th robot

Then the radius of gyration of the swarm is calculated as

$${R_G}^2 = \frac{\sum_{i=1}^{i=N} m_i r_i^2}{\sum_{i=1}^{i=N} m_i}$$

m_i = mass of i^{th} robot

r_i = distance of robot i^{th} robot from CM

A congestion circle is defined with its center as the center of mass of the swarm and it radius directly proportional to the number of agents in the simulation and inversely proportional to the radius of gyration.

$$R_C \propto \frac{N}{R_G}$$

$$R_C = \frac{C \times N}{R_G}$$

Where

R_C = Radius of circle of congestion

N = Number of robots in swarm

C = Sensitivity constant of the planner

R_G = Radius of Gyration defined as follows

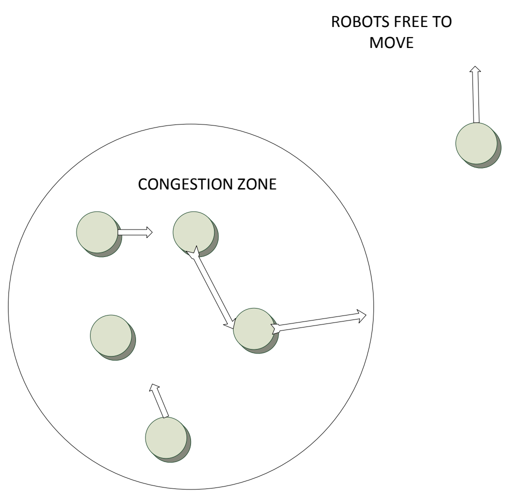


Figure 19: Agents outside the congestion circle can freely follow their path to goal

3.2.2 Cluster Planned Approach

This approach is inspired with the limited FOV approach mentioned above. In this approach, I limited the congestion avoidance to only local congestion avoidance strategy. The robot does not avoid congestion unless it enters a congestion region defined by its limited field of view. As soon as the robot enters a congestion zone, it tries to find friendlier robots that may have velocity vector nearest to its own. It then follows a weighted velocity optimization vector. The vector is weighted by the relative distance from the agent to its goal as compared to its distance from the friendly neighbor.

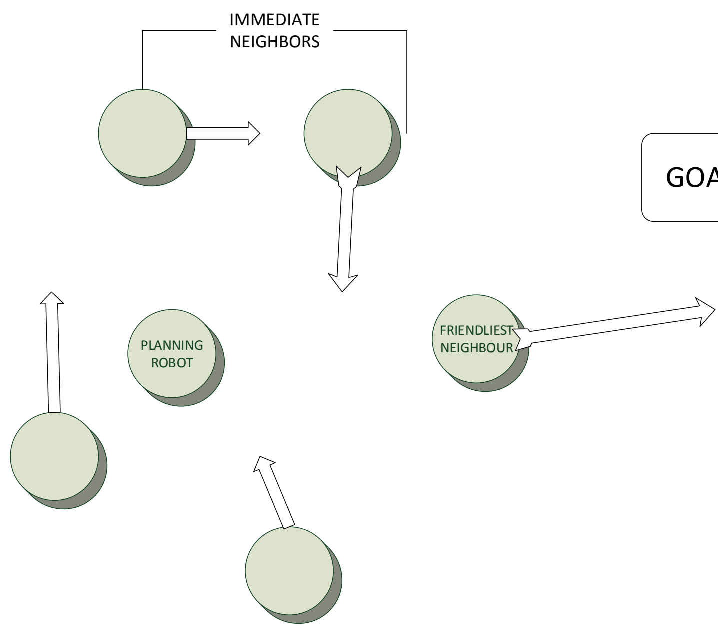


Figure 20: A robot choosing its friendliest neighbor

In case the robot does not find any friendly neighbor, it detours the local congestion zone with a weighted perpendicular velocity to the oncoming, thus creating a natural selection like the evolution of survival of the fittest. This creates bubbles of robots following each other and result in better task accomplishments.

Chapter 4

Simulation results

and discussions

4.1 Tests and benchmark situations

The tests were intended to create maximum possible interference to check for extreme conditions of interference. One such scenario is robots standing in a circle with antipodal goal locations, another is a robot pick and place scenario. These two were selected so as to compare their results with available literature. All tests were performed using RVO2 [32] an open source available C++ library for RVO. The visualizations are made in Processing 2.1.1, an open source java API. And the graphs are made with the help of Office Libre.

4.1.1 Circle with antipodal goals

In this scenario, I have tried to scale the simulation from as low as 2 robots to 150 robots to check for possible trends of the techniques.

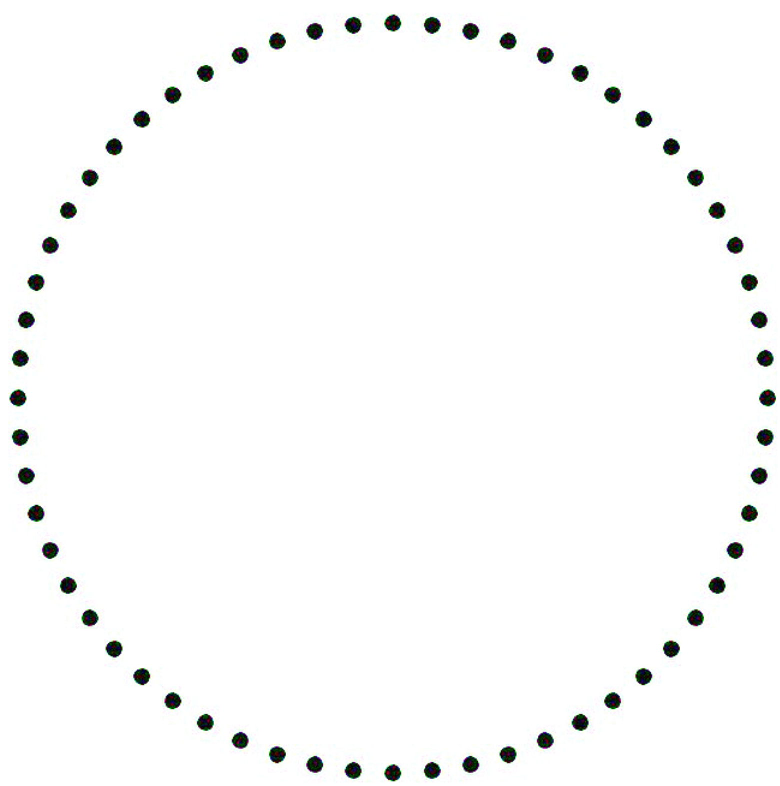


Figure 21: Initial configuration of agents standing in a circle

- **RVO vs CPA**

The agents are colored according to their velocities. Pink shows maximum velocity and blue represents minimum velocity. Black is to represent robots on a halt.

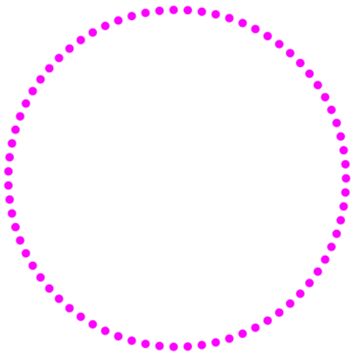
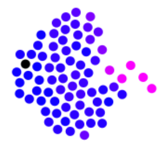
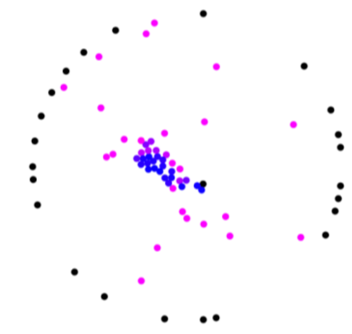
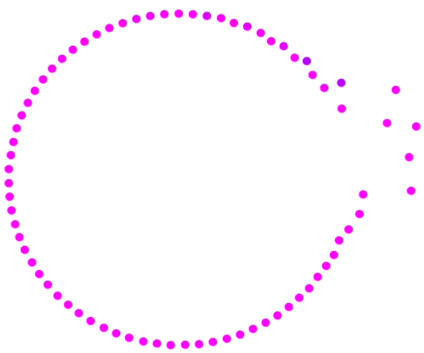
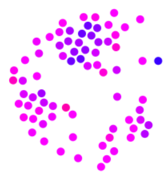
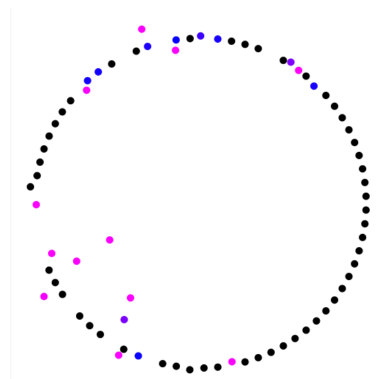
Time Step	100	500	1000
Without Planning			
With Planning			

Table 3: Results for RVO as compared with CPA

The figure clearly shows RVO getting stuck at the middle whereas CPA is showing higher solution speed. Also, the velocity of the swarms remains high during the congestion region due to neighbor following strategy.

Figure 22 shows the scaling comparison of RVO and CPA

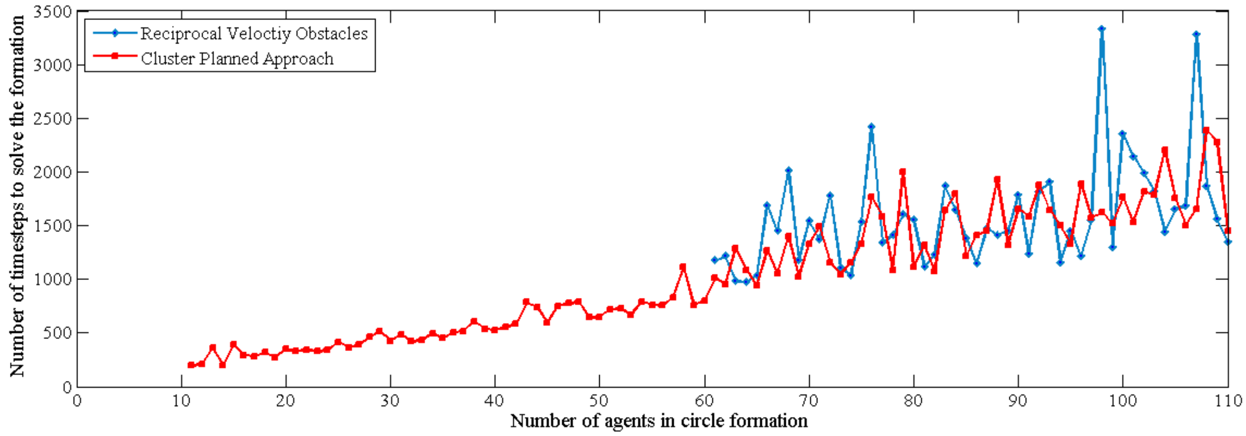


Figure 22: Scaling agents from 10 to 110 CPA vs RVO

As can be seen, RVO fails to solve for less number of agents. The reason being less perturbation in lower number of symmetrically aligned agents. Whereas CPA can solve for lower number of agents as well. Also the variation in solution times lower the predictability of RVO and CPA performs with a higher confidence.

We also present predictable solution profiles with CPA. The following figure shows how CPA solves 70, 100 and 130 robots in an antipodal goal circle formation problem with better predictability and RVO fails to provide a predictable robot reaching time profile

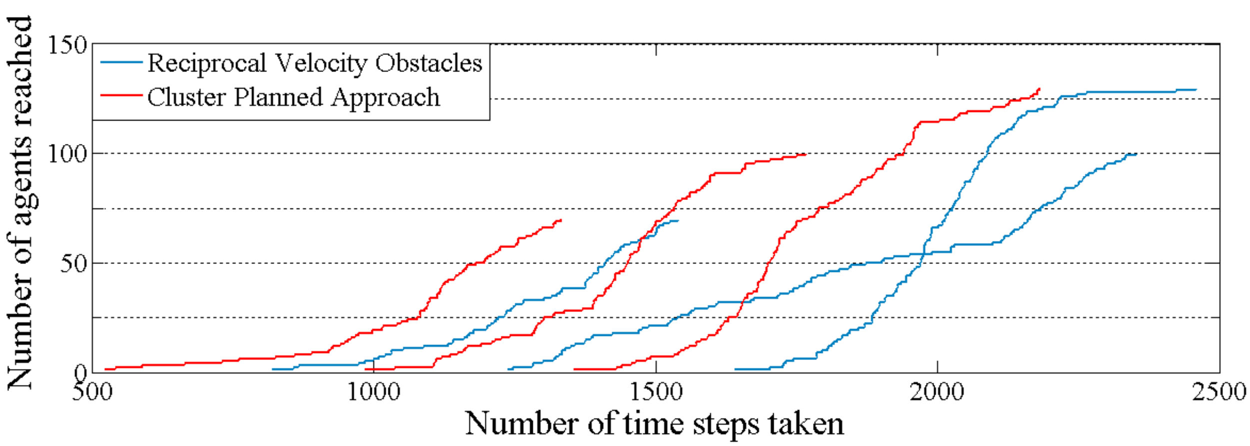


Figure 23: Agents reach their respective goals with predictable times with CPA

- **RVO vs CMA**

Owing to the stagnant nature of RVO, we also added some perturbation to the simulation and monitored its effects against CMA. The spike for 3-5 robots is an emergent anomaly that is yet to be solved

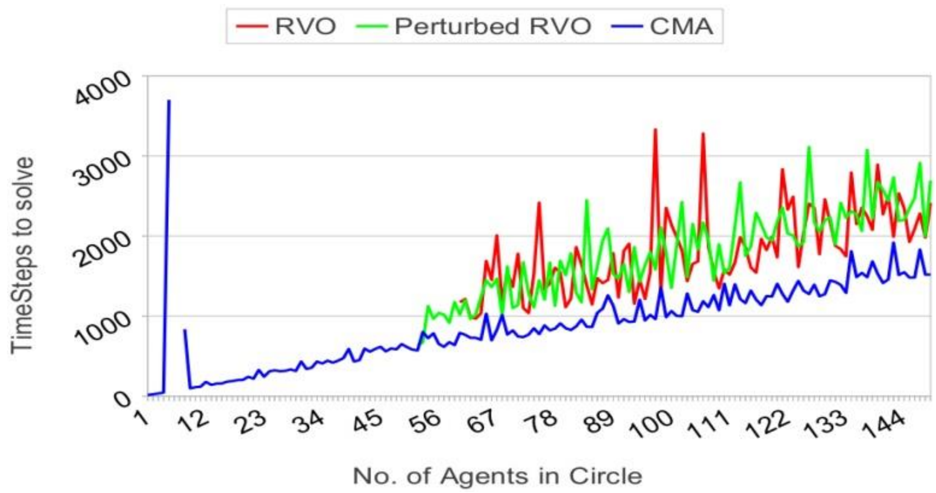


Figure 24: CMA vs RVO and RVO with perturbation

4.1.2 Package transportation problem

We now present the pick and place problem of mobile robots. Possible congestion zones are at the source and sink locations

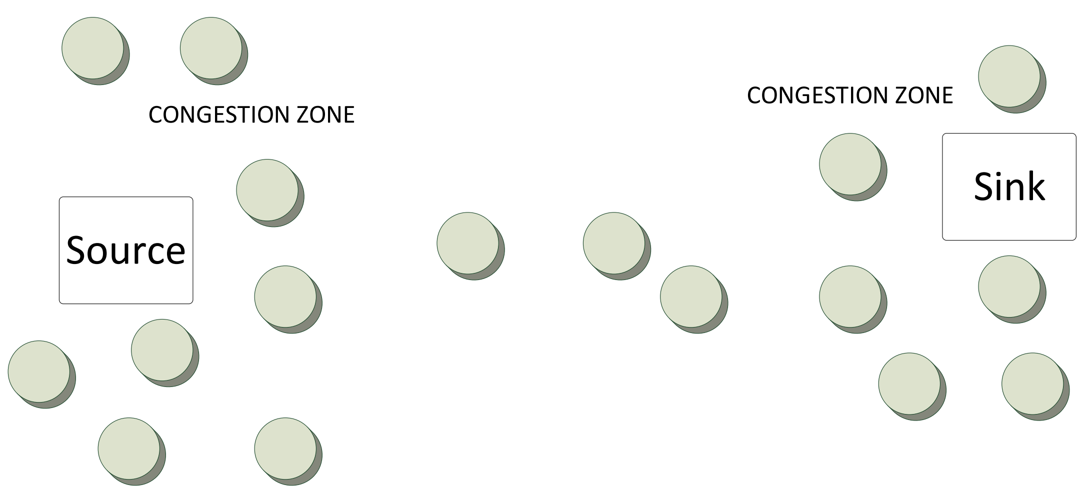


Figure 25: Showing Package transportation problem

- **RVO vs CMA**

The package transportation is being done on a higher gradient than RVO which shows the trends that can be extrapolated to as longer a time span as we wish. Although CMA is not a good approach for situations where there are multiple centers of masses, yet it has proven to work better than RVO itself. Figure 26 shows CMA creating a less functional congestion zone

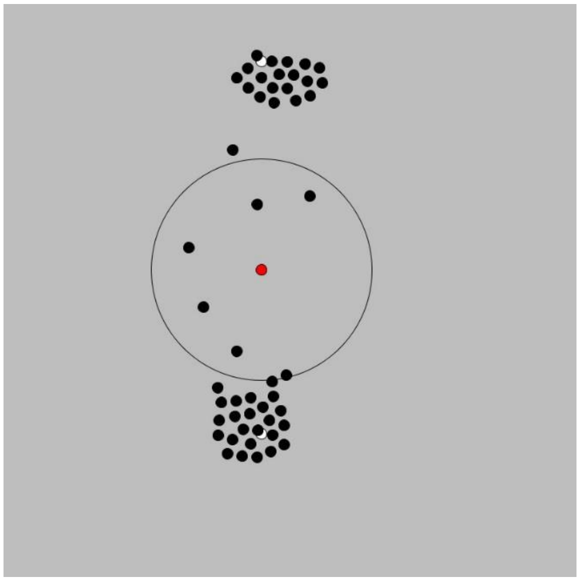


Figure 26: Package transport problem with source, sink center of mass and congestion circle

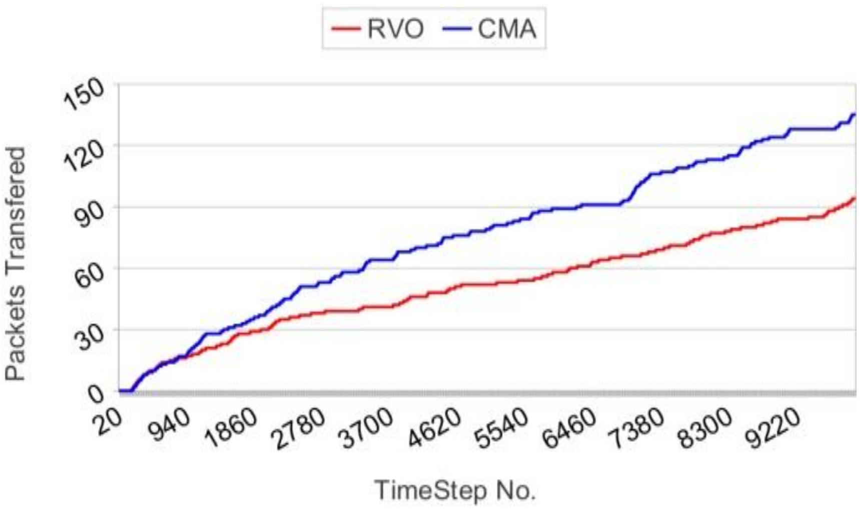


Figure 27: Package Transportation RVO vs CMA

- **RVO vs CPA**

Figure below shows the transportation of packages on a higher rate. The simulation time can be extended to establish the results and is obvious

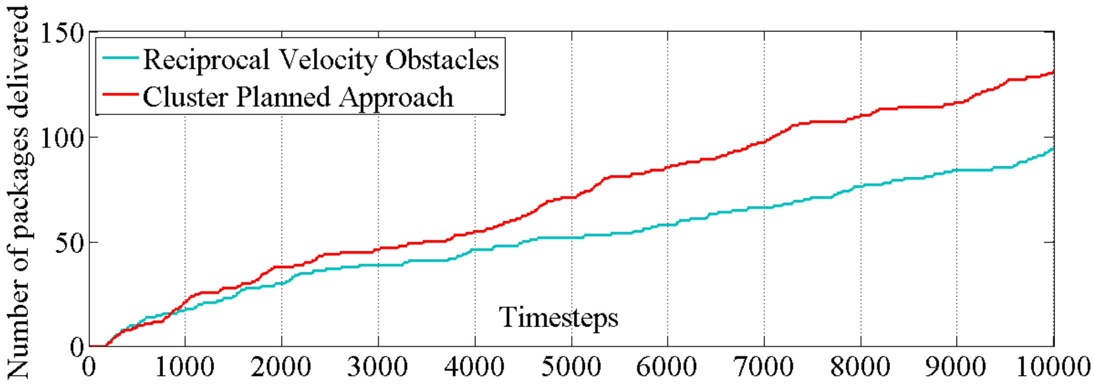


Figure 28: Package Transportation RVO vs CPA

Chapter 5

Conclusions and Future Directions

5.1 Conclusions

Reactive path planning and obstacle avoidance is key to navigating swarms of robots safely to their goals. It may not produce globally optimum solutions or best path plans but it makes the swarm robust to immediate changes in dynamic environments.

RVO stands as one of the fastest, reactive and swarm-applicable obstacle avoidance approach. It inherits the simplicity of VO approach, yet it also considers the presence of intelligent fellow agents without putting much computation burden on the machine.

RVO requires a symmetry breaking mechanism for dense robotic populations. Swarms are supposed to be dense populations so the implementation of RVO in swarms require inspirations from natural swarms.

The applications of swarms require symmetry breaking mechanisms that are generic in nature and implicit schemes are required for fast problem solving.

Inspiration from nature is most likely to produce better results for swarm obstacle avoidance algorithms as the concept of swarms itself is a nature inspired concept.

The two modifications to RVO done have proven results in two different ways. So we discuss each strategy differently and comment upon both independently.

The CMA approach works better in problems where there is a global center of mass and congestion is most likely to take place on one and only one location. It yet lacks the possibility of as simple as two congestion zones. Although, if the parameters of congestion constant be tweaked well enough, it performs relatively good in situations where it is not even supposed to produce any results

The CPA approach is more swarm friendly as it involves the sensing of local neighbors only. The agents are not supposed to predict major congestion zones. We also witness congestion occurrences in apparently large numbers of robots but due to friendly nature of the algorithm itself, it manages to solve the congestion in a quicker way than RVO itself.

5.2 Future recommendations and directions

We discuss CPA and CMA independently for future needs and directions to follow

- **CMA**

  - It needs to be localized and should be modified to sense and avoid local congestion zones.
  - It should be optimized for the local FOV radius.
  - The radius of gyration should be experimented upon for powers other than 1 to search relevant dependence for efficient solution.
  - The center of mass should be experimented upon to care for Gaussian distribution instead of linear distribution of agents

- **CPA**

  - The friendliest neighbor should be calculated in a more sophisticated way
  - The weighted following should be experimented upon to find the optimum weights
  - The oncoming zones should be predicted using general perception patterns

References

[1]. US Patent No. 2,286,571

[2]. US Patent 2679940

[3]. US patent 2 998 237

[4]. Gervasi, V., & Prencipe, G. (2004). Coordination without communication: the case of the flocking problem. Discrete Applied Mathematics, 144(3), 324-344.

[5]. Garnier, S., Gautrais, J., & Theraulaz, G. (2007). The biological principles of swarm intelligence. Swarm Intelligence, 1(1), 3-31.

[6]. Holland, O., & Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. Artificial life, 5(2), 173-202.

[7]. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm intelligence: from natural to artificial systems (No. 1). Oxford university press.

[8]. Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In Swarm robotics (pp. 10-20). Springer Berlin Heidelberg.

[9]. Wang M, Zhu YL, HE XX. A survey of swarm intelligence. Comput Eng 2005;31(22):194e6 [in Chinese].

[10]. Van Den Berg, J. P., & Overmars, M. H. (2005). Roadmap-based motion planning in dynamic environments. Robotics, IEEE Transactions on, 21(5), 885-897.

[11]. Ge, S. S., & Cui, Y. J. (2002). Dynamic motion planning for mobile robots using potential field method. Autonomous Robots, 13(3), 207-222.

[12]. Seder, M., & Petrovic, I. (2007, April). Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In Robotics and Automation, 2007 IEEE International Conference on (pp. 1986-1991). IEEE.

[13]. Foka, A. F., & Trahanias, P. E. (2003, October). Predictive control of robot velocity to avoid obstacles in dynamic environments. In Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on (Vol. 1, pp. 370-375). IEEE.

[14]. Chakravarthy, A., & Ghose, D. (1998). Obstacle avoidance in a dynamic environment: A collision cone approach. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 28(5), 562-574.

[15]. Fiorini, P., & Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. The International Journal of Robotics Research, 17(7), 760-772.

[16]. Van den Berg, J., Lin, M., & Manocha, D. (2008, May). Reciprocal velocity obstacles for real-time multi-agent navigation. In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on (pp. 1928-1935). IEEE.

[17]. Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., & Siegwart, R. (2013). Optimal reciprocal collision avoidance for multiple non-holonomic robots (pp. 203-216). Springer Berlin Heidelberg.

[18]. Sadat, S. A., & Vaughan, R. T. (2012, May). Bravo: Biased reciprocal velocity obstacles break symmetry in dense robot populations. In Computer and Robot Vision (CRV), 2012 Ninth Conference on (pp. 441-447). IEEE.

[19]. Snape, J., van den Berg, J., Guy, S. J., & Manocha, D. (2011). The hybrid reciprocal velocity obstacle. Robotics, IEEE Transactions on, 27(4), 696-706.

[20]. Lerman, K., & Galstyan, A. (2002). Mathematical model of foraging in a group of robots: Effect of interference. Autonomous Robots, 13(2), 127-141.

[21]. Ostergaard, E. H., Sukhatme, G. S., & Matari, M. J. (2001, May). Emergent bucket brigading: a simple mechanisms for improving performance in multi-robot constrained-space foraging tasks. In Proceedings of the fifth international conference on Autonomous agents (pp. 29-30). ACM.

[22]. Scheidler, A., Merkle, D., & Middendorf, M. (2008). Congestion control in ant like moving agent systems. In Biologically-Inspired Collaborative Computing (pp. 33-43). Springer US.

[23]. Sadat, S. A., & Vaughan, R. T. (2010, May). Blinkered lost: Restricting sensor field of view can improve scalability in emergent multi-robot trail following. In Robotics and Automation (ICRA), 2010 IEEE International Conference on (pp. 947-952). IEEE.

[24]. Bentes, C; S. J. dos Campos, Brazil ; Saotome, O., "Dynamic Swarm Formation with Potential Fields and A* Path Planning in 3D Environment" in Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian. pp. 74 – 78.

[25]. Borenstein, Johann, and Yoram Koren, “The vector field histogram-fast obstacle avoidance for mobile robots” in IEEE Transactions on Robotics and Automation (Volume: 7, Issue: 3), 1991, pp. 278 – 288

[26]. Dong Eui Chang, Shawn C. Shadden, Jerrold E. Marsden, and Reza Olfati-Saber, “Collision Avoidance for Multiple Agent Systems” in Proceedings of the 42nd IEEE Conference on Decision and Control Maui, Hawaii USA, December 2003, pp. 539-543.

[27]. Fiorini, Paolo, and Zvi Shiller, “Motion planning in dynamic environments using the relative velocity paradigm” in IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993.

[28]. Fox, D. ; Bonn Univ., Germany ; Burgard, W. ; Thrun, S, “The dynamic window approach to collision avoidance” in Robotics & Automation Magazine, IEEE (Volume:4 , Issue: 1) 1997, pp. 23 – 33.

[29]. Thangarajah, J, Padgham, L and Winikoff, M 2003, 'Detecting and avoiding interference between goals in intelligent agents', in Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 9-15 August 2003

[30]. Jur van den Berg, Stephen J. Guy, Ming Lin, Dinesh Manocha, “Reciprocal n-Body Collision Avoidance” in Robotics Research Springer Tracts in Advanced Robotics Volume 70, 2011. pp 3-19.

[31]. Miner, Don, and Niels Kasch. "SwarmVis: a tool for visualizing swarm systems." UMBC Computer Science 636.

[32]. van den Berg, Jur, Stephen J. Guy, Jamie Snape, Ming C. Lin, and Dinesh Manocha. "RVO2 library: Reciprocal collision avoidance for real-time multi-agent simulation." (2011).

[33]. Stuart J. Russell and Peter Norvig, “Intelligent Agents” in Artificial Intelligence A Modern Approach, 3rd ed, chapter 2.

[34]. Yaseen, S., Ayaz, Y., Iqbal, K. F., Muhammad, N., Gilani, S. O., Jamil, M., & Ali, S. I. (2014, April). Centre of mass avoidance planner using radius of gyration for Reciprocal Velocity Obstacles. In *Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on* (pp. 118-124). IEEE.