5-2011

Control and Coordination in a Networked Robotic Platform

Krishna Chaitanya Kalavacharla
kkalavac@utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Krishna Chaitanya Kalavacharla entitled "Control and Coordination in a Networked Robotic Platform." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

Dr.Seddik M Djoaudi, Major Professor

We have read this thesis and recommend its acceptance:

Dr.Dongjun Lee, Dr. Harry Lee Martin

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Control and Coordination in a Networked Robotic Platform

A Thesis Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Krishna Chaitanya Kalavacharla
May 2011

Dedication

To my mom, dad, priya, chinnu, avani and innumerable valuable friends without whose support and unconditional love, this would have been simply impossible. I love you all.

Control and Co-ordination in a Networked Robotic Platform.

Abstract:

Control and Coordination of the robots has been widely researched area among the swarm robotics. Usually these swarms are involved in accomplishing tasks assigned to them either one after another or concurrently. Most of the times, the tasks assigned may not need the entire population of the swarm but a subset of them. In this project, emphasis has been given to determination of such subsets of robots termed as "flock" whose size actually depends on the complexity of the task. Once the flock is determined from the swarm, leader and follower robots are determined which accomplish the task in a controlled and cooperative fashion. Although the entire control system,which is determined for collision free and coordinated environment, is stable, the results show that both wireless (bluetooth) and internet (UDP) communication system can introduce some lag which can lead robot trajectories to an unexpected set. The reason for this is each robot and a corresponding computer is considered as a complete robot and communication between the robot and the computer and between the computers was inevitable. These problems could easily be solved by integrating a computer on the robot or just add a wifi transmitter/receiver on the robot. On going down the lane, by introducing smarter robots with different kinds of sensors this project could be extended on a large scale for varied heterogenous and homogenous applications.

Table of Contents

List of Figures

Chapter 1

Introduction

Swarm robotics is a new approach to the coordination of large numbers of relatively simple robots. The approach takes its inspiration from the system-level functioning of social insects, which demonstrate three desired characteristics for multi-robot systems: robustness, flexibility and scalability. Robustness can be defined as the degree to which a system can still function in the presence of partial failures or other abnormal conditions. Social insects are highly robust. Their self-organized systems can still work even after losing lots of system components or changing the environment parameters considerably. Flexibility can be defined as the capability to adapt to new, different, or changing requirements of the environment. Flexibility and robustness have partly conflicting definitions. The difference between two occurs in problem level. When the problem changes, the system has to be flexible (not robust) enough to switch to a suitable behavior to solve the new problem. The biological systems have this level of flexibility and can easily switch their behaviors when problems change. For instance, ants are so flexible that they can solve foraging, prey retrieval and chain formation problems with the same base self-organized mechanism. Scalability can be defined as the ability to expand a self-organized mechanism to support larger or smaller numbers of individuals without impacting performance considerably. Although there is a range in which the swarm performs in acceptable performance levels, this range is preferred to be as large as possible.[95]

Swarm robotics is an approach to the coordination of large number of robots. This is inspired from nature and especially from insects and their colonies. These insects demonstrate three highly desirable characteristics for swarm robotic sys-

tems. They are robustness,flexibility and scalability. Robustness n. able to withstand or overcome adverse conditions

1.1 Classification And Literature Review[95]

There are several axes on which the swarm robotics could be classified. With extensive literature survey I concluded on several axes on which swarm robotics could be classified. They could be broadly classified into four sections.

1. Modeling
2. Communication
3. Research
4. Behavioral Design

Each category is explained briefly along with supported research and literature.

1.1.1 Modeling

Modeling is a method used in many research fields to better understand the internals of the system that is investigated. But as we will discuss in the following paragraphs, modeling has some more advantages for swarm-robotics compared to other fields.

The existence of possible risks for the robots and the limited power of the robots require a human observer to follow the experiments and do some house keeping works periodically. The time spent on these experiments and possible risk of losing the robots even if a human observer exists become a bottleneck when several experiments are needed to validate the results of the studies. To

eliminate these problems, it is safer and easier to model the experiments and simulate them on computers.

Another importance of modeling for swarm robotic studies appears when the scalability of the experiments are tried to be tested. Most of the time, scalability requires testing the control algorithms on more than hundreds of robots. But the cost of an individual robot prohibits testing of the experiments on more than a few tens of robots within the current state of the robot technology. Since scalability is an important aim of swarm-robot systems, it seems that the models will be needed until much more cheaper robots are manufactured.

Despite having such advantages of modeling, there is one more point that needs to be considered by swarm robotic researchers. Although models may be valuable for understanding the internals of the system being worked on, there will always be a difference between simulation results and real world results. Although this difference is tried to be minimized by simulator developers, complex dynamics of interactions between the robots and unpredictable noise in the sensors and the actuators of the robots makes simulations impossible to be fully realistic.

Modeling could again be categorized into four types: sensor-based, microscopic, macroscopic and cellular automata modeling. Although adding cellular automata modeling as another type of modeling method is open to discussion and we might consider it as a special type of microscopic modeling method, we chose to add it as another type of modeling method because of the following reasons.

First, it is used as a modeling tool for several self-organized systems in biology [9], which shows that it is an established modeling method for biologists as well as computer scientists. Second, cellular automata is a simple and mature field, which has lots of analytical, tools [34] and is strongly connected to

dynamical systems theory [34]. These properties of cellular automata make it a powerful modeling tool for swarm robotic studies.

1.1.1(a) Sensor-Based Modeling

Sensor-based modeling is a modeling method, which uses the models of sensors and the actuators of the robots and objects in the environment as the main components of the modeled system. After modeling these main components, the interactions of the robots with the environment and the interactions between the robots are modeled. This modeling method is the mostly used and the oldest method for modeling robotic experiments.

The key in this approach is to make interactions discussed above as realistic and as simple as possible since the complexity of these interactions becomes very important when the scalability of the experiments are tried to be tested. They also need to be realistic to be useful for swarm robotic systems. These two aims are contradictory and present a realism-simplicity dilemma in sensor-based modeling.

The examples of this approach can be found in [4] and [58]. The authors physically modeled the environment using an open-source physics engine and run the experiments in parallel over multiple computers connected via a network to overcome the increased complexity of the simulations. Some other examples using this approach are [66] and [65].

1.1.1(b) Microscopic Modeling

Microscopic models robotic experiments by modeling each robot and their interactions mathematically. In this method, behaviors of robots are defined as states and the transition between these states are bound to internal events inside robot and external events in the environment.

The main difference between microscopic models in this section and macroscopic models in the following section is the granularity of the models developed. While microscopic approach models the experiments by modeling each robot, the macroscopic approach models the whole behavior of the system directly.

As a special case of microscopic and macroscopic modeling, probabilistic microscopic and probabilistic macroscopic models are used in swarm robotics. By assigning probabilities to transitions between robot actions (for microscopic models) or transitions between system states (for macroscopic models), the system behavior and the noise in the environment are easily integrated into these probabilistic models.

In probabilistic microscopic models [46], [45], [36], a time unit is defined based on a primitive event 1 to be able to advance the model at each model step. After specifying this time unit, the probability of each state transition is computed with systematic experiments performed with real robots. In other words, the probabilities of all events are computed per time unit of the model. After finding these state transition probabilities, the mathematical model is run for each robot by generating random numbers between 0 and 1 for each possible event transition of the selected robot and comparing these numbers with state transition probabilities. If some of these numbers are lower than the predefined transition probabilities of the associated events, those events are assumed to occur and the state of that robot is changed.

Jeanson et al., [36] studied aggregation strategies in cockroaches. They tried to prove that cockroaches perform the global aggregation from local interactions. To do this they measured the important system parameters from the experiments with cockroach larvae like probability of stopping in an aggregate or probability of starting to move. A numerical model of behaviors of cockroaches is created from these measurements and tried to be validated by

numerical simulations. Although their numerical model reveals a quantitative disagreement with real experiments, they claimed that it also offers strong evidence that aggregation can be explained in terms of local interactions between individuals.

1.1.1(c) Macroscopic Modeling

Another kind of mathematical modeling method of robotic experiments is macroscopic modeling. In macroscopic modeling, the system behavior is defined with difference equations and each of the system states (variables of difference equations) represents the average number of robots in a particular state at a certain time step.

While the system need to be iterated for each robot in microscopic models 2, macroscopic models are solved only once to obtain the steady state of the model. Although this feature allows great speed- ups for macroscopic models when compared to microscopic models, microscopic models allow catching the fluctuations in the experiments. In other words, while macroscopic models allows obtaining a rough global behavior of the robotic system quickly, microscopic models allow to obtain a more realistic global behavior slowly.

Similar to microscopic models, probabilistic version of macroscopic models [46], [42] are used in swarm robotic studies to handle noise in a simple way. Martinoli et al., [46] applied macroscopic modeling to stick pulling problem. The authors presented the model incrementally starting from a basic model, which only contains Search and Obstacle-Avoidance states up to the most complex model, which contains all states in the robot controller. For each stage, a difference equation (DE) is developed and the steady state of the DE system is analyzed to obtain average number of robots in each state at the end of the experiments. Comparisons of microscopic, macroscopic and sensor-based

models are also presented and the limitations of macroscopic modeling for stick pulling problem are described.

Another distinguished feature of this study is the definition and tracking of system-wide guarantees for self-organizing emergent systems. The authors developed an equation-free macroscopic model and system- wide guarantees for an automated guided vehicle warehouse transportation system. They validated the results of the model by comparing the results of the accelerated equation-free macroscopic model with the non-accelerated one. Although they found that some accuracy are lost which is normal for all macroscopic models, the model managed to find the steady state successfully.

Trianni et al., [63] tried to find macroscopic models of aggregation and chain formation problems. But the results of macroscopic model did not fit to the results obtained from sensor-based simulations. They thought that the possible problems are the lack of spatial information in the mathematical model, carrying out the simulation in discrete time and the lack of interaction dynamics in the model. At the end of their experiments, they decided to make their sensor-based simulations more realistic using physical sensor based modeling instead of improving their macroscopic model in their future studies.

1.1.1(d) Cellular Automata Modeling

Cellular automata (CA) is among the simplest mathematical models of complex systems [34]. The CA models contain discrete lattice of cells in one or more dimensions where each cell in the lattice has finite number of possible states.

Each cell interacts only with the cells that are in its local neighborhood and the system dynamics are characterized by the local rules executed locally on the cells in discrete time steps. Several CA models are developed for the natural phenomena [17], [12] around us. In addition to using these models as inspiration

sources of swarm robotic studies, CA can be used as a modeling tool for CA based experiments. The studies of Shen et al., [56], [57] is an example of this type of studies. The details of these studies are summarized in the latter parts of this section.

1.1.2 Communication

I have used the same classification categories Cao et al., [10] used in their survey of cooperative robotics for classifying the swarm robotics studies based on the communication mechanisms used by the swarms.

The first category (interaction via sensing) is the simplest and the most limited type of communication between the robots. This type of communication requires the robots to distinguish between other robots and the environment objects. The details are discussed in the corresponding section below.

In the second category (interaction via environment), the robots used the environment as a communication medium. There are well known examples of this communication type in biology like communication via pheromones in ants [9].

The ants communicate with each other through chemicals called pheromones. For example, when an ant finds food, it will leave a trail along the ground on its way back to home, which in a short time other ants will follow. When they return home they will reinforce the trail, bringing other ants, until the food is exhausted. The slow dissipation property of the pheromone trials will allow the ants to find new food sources when the older ones are exhausted.

Although the communication scheme is simple in this approach, the physical implementation of it is not so easy because of the difficulty of creating special environments allowing communication between agents.

Most of the studies using this approach uses only simulation of this commu-

nication scheme with the help of a short range wireless communication mechanism (e.g. RF or Infrared) [54], [55], [56], [57]. Because of this, I decided not to create a separate section for interaction via environment method and described the simulation attempts of this communication method in interaction via communication section.

The third category (interaction via communication) involves explicit communication with other robots by broadcast messages. Although Cao et al., [10] included the communication via directed messages (using robot identification numbers) in this category, we did not prefer this since swarm robotics prefers to use the communication in a limited way.

Following two sections describe the studies using interaction via sensing and interaction via communication methods subsequently.

1.1.2(a) Interaction via Sensing

The discrimination of interaction via sensing from interaction via communication can be difficult time to time. The guideline to do this discrimination is to look at the aim of the information sender side. If the sender in the interaction aims to give information to other robots intentionally then that study is categorized as interaction via communication instead of interaction via sensing. So if two robots interacting to pull a stick and sensing each others action in a limited way, this work is considered as interaction via sensing. And if a robot broadcasts information packages or switches on/off a light around them to show their state, these studies are considered to be the type of interaction via communication.

Interaction via sensing requires the discrimination of other robots from the environment objects, also called as the kin recognition. Kin recognition is an important feature of animals in nature. With the help of kin recognition, animals can behave different to their kins, work together to accomplish some tasks, and

protect themselves from their enemies better.

I considered kin recognition as a kind of minimalist communication mechanism since just by discriminating the kin and observing their behaviors (without explicit communication), the robots can manage to solve several problems (e.g. flocking, chain formation and cooperative stick pulling) in swarm-robotics. It is also required to solve many problems (e.g. aggregation and flocking) efficiently.

Most of the swarm robotics studies (e.g. [59], [55], [54], [58], [43], [27], [64]) use kin recognition as a communication medium since most of the problems requires (e.g. flocking, chain formation and cooperative stick pulling) discrimination of robots in the environment to obtain acceptable performance. As an example, Soysal and Sahin [58] need the robots to discriminate other robots from obstacles since it is possible for the robots to aggregate near the walls instead of each other in a rectangular arena.

Trianni et al., [66], [65] tried to solve hole-avoidance problem using genetic algorithms to evolve the weights of a simple perceptron based controller. The robots are connected to each other with joints and they have to perform coordinated motion in an environment, which has holes too large to be traversed. The aim of the study is to learn the correct dynamic to move away from the holes as a group when the robot(s) on an edge of the formation sense the hole with its (their) ground sensor(s). The robots can sense their neighbors relative movements with the help of their traction sensors.The communication with the help of traction sensors can be considered as an example of interaction via sensing since there is no intention to send information to other robots in this communication scheme.

1.1.2(b) Interaction via Communication

A more advanced version of communication requires direct communication of robots by broadcasting or one-to-one communication. As mentioned before, one-to-one communication using identification of robots is not preferred in swarm robotics studies since this may reduce the scalability and flexibility of the system.

Nouyan and Dorigo [52] implemented a chain formation behavior. Initially the robots search for other chain members or the nest. Once a robot finds a chain member or the nest, it becomes a chain member depending on two predefined timeouts. The robots distinguish chain members and the nest based on the color of the LED ring around their body. A chain member can have three different colors: blue, green and red. It activates the color blue, if it connects to the nest or to a red chain member. It activates the color green, if it connects to a blue chain member and color red otherwise. This coloring mechanism allows robots to find the direction of the chain. Since having a long chain instead of a chain with several branches is preferred, the robot follows the color to reach to the end of the chain to connect. Nouyan [53] also extended this work with more detailed configurations in his thesis.

Grob et al., [25] studied the self-assembly problem. The aim of the work is to locate, approach and connect with an object that acts as a seed or connect to other robots already connected to the seed. Similar to the Nouyan and Dorigos previous work described above, a robot discriminates the robots connected to the seed with the help of the LED ring around robots body. The initial color of the robots is set to blue. Once a robot connects to the seed or to a robot already connected to the seed, it activates the color red permanently.

It is also worth to mention the studies performed by Payton et al., [54], [55] Shen et al., [56], [57] in this section since they used broadcasting to simulate the interaction via environment type of communication. The details of their

works related to communication are already described in section 5.1. Although the works of Payton et al., [54], [55] and Shen et al., [56], [57] can be seen as simulation attempts of interaction via environment method, we decided to describe these studies in this section.

Payton et al., [54], [55] simulated the communication mechanism used by ants. The ants communicate with each other through chemicals called pheromones. When an ant finds food, it will leave a trail along the ground on its way back to home, which in a short time other ants will follow. When they return home they will reinforce the trail, bringing other ants, until the food is exhausted. The slow dissipation property of the pheromone trials will allow the ants to find new food sources when the older ones are exhausted.

Shen et al., [56], [57] used a similar approach to be able to simulate the diffusion of hormones in the environment. Although they did not test their ideas on real hardware, they claimed that the diffusion of hormones can be implemented using a short range wireless communication (either using RF or Infrared).

In their experiments, the robots broadcast packets containing the hormone type information. To implement the diffusion of the hormones, each receiver robot determines the direction (e.g. via a directional antenna) of the message and the distance of the signal source (e.g. by measuring the strength of the signal). The robot then applies diffusion function defined in the paper to compute the concentration of that particular hormone at the current and nearby cells. After collecting all hormonal signals coming from neighbor cells for some period of time, the robots computes the reaction of collected hormones and broadcast this information to simulate the diffusion of hormones.

1.1.3 Research

To categorize robots basing on research, several literature surveys have been carried out. Dudek et al.,[16] classified the swarm robotics literature in terms of swarm size, communication range, communication topology, communication bandwidth, swarm reconfigurability and swarm unit processing ability. They prepared a taxonomy instead of a survey on swarm robotics and fit some limited number of sample publications inside this taxonomy.

I believe that swarm size criteria is not much applicable to characterization of swarm robot systems since scalability is one of the desired characteristics of swarm robotics and swarm systems should work with large numbers of system components. I also did not choose communication topology and communication bandwidth as subcategories since the communication should be kept limited as much as possible and preferably communication should be done using broadcasting instead of using robot names or addresses or complex hierarchies based on robot addresses. Although future studies will investigate the communication aspect of swarm systems more; having limited diversity in current studies, require us to have a communication axis which does not include bandwidth and topology of communication as a category in this survey.

Iocchi et al., [35] presented a taxonomy of multi-robot systems and address some multi-robot system studies in their taxonomy. They presented their taxonomy hierarchically using levels. First level is cooperation level, which is divided into aware and unaware categories as the lower knowledge level. Aware category is divided into three more categories namely strongly-coordinated, weakly-coordinated and not-coordinated as the coordination level. Strongly-coordinated category is divided into strongly-centralized, weakly-centralized and distributed categories as the organization level. They also wrote a separate section for describing the application domains of multi-robot systems.

1.1.4 Behavioral Design

Adaptation is any change in the structure or the function of an entity (e.g. a component of a complex system) that allows it to survive more effectively in its environment.

Adaptation in biological systems can be classified as structural, behavioral and physiological adaptation. Structural adaptations are special body parts of an organism that help it to survive in its natural habitat, like its skin color, shape, body covering and teeth. Behavioral adaptations are special ways a particular organism behaves to survive in its natural habitat. Physiological adaptation are subsystems present in an organism that allow it to perform certain biochemistry reactions like secreting slime, being able to keep a constant body temperature or producing pheromones.

An important property of adaptation is its time scale. There are two types of adaptation based on time scale: evolution and learning. Especially structural and physiological adaptations do not develop during an individuals life but over many generations with evolution. In addition to evolution, the individuals may fine-tune their behaviors in their lifetime. This kind of adaptation is performed in a relatively shorter time scale and called learning.

In swarm robotics literature, researchers mostly tried to utilize the behavioral adaptation to control large number of robots to accomplish a task collectively. Because of this and importance of adaptation, we decided to categorize existing behavior design approaches into three sections based on the behavioral adaptation capability of the robot controllers: manual, learning and evolution. While we describe the works, which uses non-adaptive robot controllers in non-adaptive section, the works, which show learning capabilities are described in learning section and the ones, which try to mimic natural selection for adapting the robot controllers, are described in evolution section.

1.1.4.1 Non-Adaptive

Most of the studies utilizing non-adaptive behavior design are categorized into four subcategories: subsumption, probabilistic finite state automata, distributed potential field methods and neural networks. While these categorized studies are described in the following subsections respectively, the non-adaptive studies, which do not belong to these categories, are described below.

Payton et al., [54], [55] described a new approach in swarm robotics called pheromone robotics based on the biologically inspired concept of virtual pheromone. They developed robots with personal digital assistant (PDA) attached at the top, which allows doing computationally expensive operations. The virtual pheromones are signaled between robots with a mechanism attached at the top of the robots, which contains eight radially oriented, directional infrared receivers and transmitters. The information is transferred between the robots as 10-bit messages, which have message type, hop-count, and data fields. The intensity and orientation values obtained from received messages are also used in obstacle detection and in determining distance and direction of neighboring robots.

They defined three main concepts in their studies: virtual pheromone, world embedded computation and world embedded display. Virtual pheromones are working with the help of infrared mechanism described above. With the help of virtual pheromones, the robots may solve problems like generating the map of a field or solving the shortest path problem for a field. This feature is called as world embedded computation. An external observer can also be informed about the results obtained in world embedded computation with the help of a video camera mounted on the observers head, which receives and displays coded infrared signals from each robot. This feature is called as world embedded display.

1.1.4.1(a) Probabilistic Finite State Automata Probabilistic finite state automata (PFSA) is a way to represent dynamical systems with finite state spaces. In probabilistic automata, the transitions between the states of the system are triggered with certain probabilities. The general approach is to model the robot behaviors as states and defining the state transitions with some external input and probabilities. This section will summarize the swarm-robotics studies using this approach.

Soysal and Sahin [58] performed systematic experiments using a probabilistic finite state machine based controller for performing aggregation task. There are four behaviors in the controller, which are connected with subsumption architecture: obstacle avoidance, approach, repel and wait. Normally robots start in approach state and switches to the wait state when they sense another robot. The switches between repel and approach states, and wait and repel states are determined by Preturn and Pleave probabilities respectively. The authors changed the size of the arena to compare different strategies obtained by modifying the Preturn and Pleave parameters. They showed that the best performance is obtained when both of the parameters equal to 1. They also stated that this strategy might not be very feasible on all robotic systems since there is a risk of having large number of robots moving in close proximity and the large power consumption due to continuous movement.

A self-organized model of the aggregation behavior of cockroaches in a bounded circular arena is developed by Jeanson et al., [36] and Garnier et al., [22]. The authors used an approach, which is similar to microscopic modeling developed by Martinoli et al., [46], [45] and Jeanson et al., [36]. They first define a self- organized model for the behaviors of the cockroaches and measured the important transition probabilities between behaviors along with the average time spent on each behavior by real cockroaches. They compared the results

obtained from the developed numerical model with the real experiments results. They claimed that their model better approximates real data than most of the previous global level models, which shows that the cockroaches may behave based on local interaction rules.

1.1.4.1(b) Neural Networks Neural networks [29], [28] are powerful learning mechanisms inspired from nervous system of humans. There are two general types of swarm robotics studies performed using neural networks. The first type uses genetic algorithms to evolve the weights of a neural network to obtain a desired behavior with a fitness function appropriate to the problem. This type of studies [4], [64], [66], [65] are discussed under section 4.3.

The second type of studies with neural networks considers the neural networks as a generalization mechanism and do not use its learning capabilities. The remaining part of this section summarizes this type of studies.

Grob et al., [25] investigated self-assembly problem with a group of robots. They defined the problem as controlling the robots in fully autonomous manner in such a way that they locate, approach and connect with an object that acts as a seed or connect to other robots already connected to the seed. The seed and the robots connected to the seed are discriminated based on the color of the ring around them.

The controller of the robots was a simple perceptron, which connects sensory inputs to motor outputs of the robots. The controller was preprogrammed with the controller obtained from another study. The experiments are done on flat and rough terrains with real robots. The results show that robots achieves self-assembly in a scalable way.

1.1.4.2 Learning

Montemanni and Gambardella [50] presented a distributed protocol for minimum power topology (MPT) problem in wireless networks. The aim in MPT problem is to assign transmission powers to the nodes of a mobile network in such a way that all the nodes are connected by bidirectional links and the total power consumption is minimized.

The authors used one of the previous protocols called MLD (Minimum Link Degree) and made it more distributed. The name of the new protocol is LMPT (Local Minimum Power Protocol), which uses some local information about neighbors to obtain better results.

MLD protocol works as follows: There is an ngb (link degree) parameter, which is used as a minimum number of links any node, should have to obtain full connectivity on the network. The nodes increase their transmission power in small amounts until they reach to ngb number of neighbors. Whenever a node hears another node in this increasing transmission power phase, it realizes that its neighbor has less than ngb neighbors and sets its transmission power as the power of its neighbors transmission power if it is greater than its current transmission power. If it is lower than its current transmission power, then current transmission power is not changed. This phase goes on until each node has at least ngb neighbors. They all stop increasing their transmission powers at this point. The ngb parameter is a heuristic parameter obtained from the global information known about the network. It does not need to be perfect information but the more precise it is, the lower the total transmission power at the end.

1.1.4.2(a) Reinforcement Learning Reinforcement learning (RL) [61] systems consist of a discrete set of environment states, a discrete set of agent actions and a set of scalar reinforcement signals. In robotic studies, environment

states are higher-level representations of sensor readings (e.g. existence of an object in front of the robot based on the thresholded values of front sensor readings). Similarly agent actions are higher-level representations of actuator commands.

The reinforcement value is the core concept in RL which differentiates it from other types of learning methods [49] (e.g. supervised or unsupervised learning) The reinforcement value gives a numerical hint to the agent for the relative success of the executed action in achieving the goal of the agent. The aim of the agent in this setting is to learn a policy (which maps states to actions) that maximizes the cumulative reinforcement values obtained in the long-term.

One of the important properties of RL is that the RL algorithms have clean theoretical convergence properties because of their dynamic programming roots [61]. Despite advantages of RL, there are serious problems in applying RL to multi-robot studies. First, theoretical convergence properties of RL require large numbers of learning trials that are difficult to perform with physical robots.

Another problem is the size of the search space. The RL algorithms are proved to converge on toy problems, which has limited search space, compared to the robotic problems. Large search space (both state and action spaces) of robotic problems requires lots more epochs to be able converge to acceptable results.

Noise is another serious problem while applying RL to multi-robot studies. Besides having lots of noise in sensor readings and actuator actions, interaction between the robots make the environment noisier and more unpredictable. Having multiple robots in the environment also breaks the convergence assumption of some of the well-known popular reinforcement learning algorithms (e.g. Q-learning [67]) since noise converts the environment to a dynamic one from a stationary one.

The last problem is probably the most difficult and classical problem in machine learning: the credit assignment problem. Both temporal and spatial credit assignment problems exist in multi-robot problems since the actions of the robots can be rewarded with a delay and the result may depend on the actions of multiple robots.

We divided reinforcement learning studies into two categories: the studies, which use local reinforcement and the ones, which use global reinforcement. In the former one, the reinforcement is only given to the robots, which are close to the location where the reinforcement is generated. In the latter one, all robots are rewarded as if the last action is a result of the collective actions of all robots. In other words, even if some robots do not contribute to the goal, all of the robots are rewarded in global reinforcement scheme.

As we discussed in section 2, the communication should be kept limited as much as possible in swarm robotic systems. Because of this preference, local reinforcement scheme is more realistic for swarm robotics. But investigating the global reinforcement and comparing its results with local reinforcement may offer new insights in swarm robotics. Yang and Gu presented a survey about multi-robot reinforcement learning studies in [68]. They first discussed preliminaries of the subject starting from Markov decision processes up to relation of multi-agent reinforcement learning with the game theory. Later, they summarized theoretic frameworks for multi- agent reinforcement learning, algorithms utilizing these frameworks and the studies performed with these algorithms. After discussing these, they summarized the works done up to that time for scaling reinforcement learning to multi-robot systems. Finally they described the main challenges of multi-robot systems and future research directions in the field which are mainly obtaining team cooperation, abstracting state and action spaces, generalization and approximation of look-up tables used in re-

inforcement learning algorithms and extending the reinforcement learning into continuous state and action spaces.

Local Reinforcement: In local reinforcement scheme, the reinforcement value generated after achieving a subgoal is only shared by the robots, which contribute in achieving that subgoal. One of the studies using local reinforcement was the study of Li et al., [43]. The authors used Balchs social entropy metric [5] to analyze the effect of diversity and specialization on a stick-pulling experiment. Since Balchs social entropy metric can only be used to measure the diversity of the robot groups, Li et al., defined specialization as a new metric of the correlation between diversity and performance.

The learning algorithm basically starts with a random direction and GTP(Gripping Time Parameter). When a predefined amount of time passes for a robot, an average reinforcement is computed for that time period. Then the GTP value is updated for that robot depending on both the current and the previous average reinforcements. If the current average reinforcement value is greater than the previous one then the GTP is modified in the same direction selected in the previous step. If the current average reinforcement value is lower than the previous one (It means the performance becomes worse.) then GTP is modified in the opposite direction.

Li et al., performed systematic experiments using local and global reinforcement signals with different group characteristics (homogeneous, heterogeneous and caste-based robot groups). Although the performance of the learning swarms achieved the same level of performance independent of the initial GTP, the performance of homogeneous swarms with a fixed GTP is decreased when the initial gripping parameters are increased. This shows that a higher level of robustness is achieved with this learning algorithm.

Global Reinforcement: In global reinforcement scheme, the reinforce-

ments obtained by robots in a specified period of time are shared between robots. Mataric [48] solved foraging problem using reinforcement learning in multi-robot domain. The author defined two challenges for applying reinforcement learning to multi-robot domain. The first one is that even if for single robot experiments the domain has very complex state space; when more than one robot is used, the problem becomes more complex because of the inferences between the robots. The second challenge is the structuring and assigning reinforcement learning. The first problem is handled with the help of behaviors and conditions. The complexity if state and actions spaces are reduced considerably with the help of them. The second problem is handled with the help of shaped reinforcement, which consists of heterogeneous reward functions and progress estimators.

Mataric developed a simple reinforcement algorithm called reinforcement summation algorithm, which adds and normalizes the reinforcement values obtained for state action pairs over time. The author compared the results of two different variations of this algorithm with a hand coded optimal solution and pure q-learning algorithm without shaped reinforcement. The first variation of her algorithm was the reinforcement summation algorithm with only heterogeneous reward functions and the second variation was the reinforcement summation algorithm with both heterogeneous reward functions and progress estimators. The results showed that the first variation is better when compared to others and q-learning algorithm is better than the hand coded optimal solution.

1.2 Problem Statement

While most studies have focussed their research in certain areas of Swarm Robotics, our research on solving three major aspects of Swarm Robotics. They

are i) Pattern and Chain formations ii) Self Assembly of Robots and iii) Coordination to achieve tasks.

1.3 Significance of Study —& Experimental Setup

Payton et al., studied the colonization of ants and tried to simulate the pheromone behavior of the ants. They named it the virtual pheromone and tried sending messages using a wireless system. The idea is somewhat similar but limited to a certain group of robots instead of communicating globally among the robots by eliminating the wastage of communication energy.The goal of the project is not only to answer multiple research related problems on mobile robots but also to promote the concept of social robotics.

The following steps define the environment and experimental setup of the system.

1. The robots are from iRobot Corporation called Create. They come with 8 bit micro controllers which could be added to the robots but not used in this particular experimental setup .

2. Class 1 bluetooth dongle attached to Windows XP desktop and Bluetooth communication receivers from Element Serial which can be attached to the robots on a 13 pin slot.

3. A coordinate system setup in the Control System Laboratory of Electrical Engineering and Computer Science department at University of Tennessee, Knoxville.

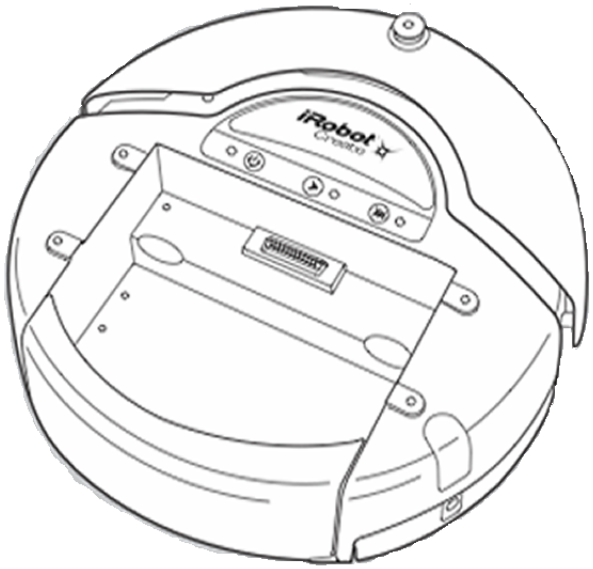

Figure 1: *iRobot Create*

Chapter 2

System

2.1 Introduction

A mobile robot is capable of autonomous motion because it is equipped with motors that are driven by an on board computer. The concept of autonomy is understood as the ability to independently make intelligent decisions as the situation changes. These machines are used in inaccessible environments that are often cluttered with unknown, moving or fixed obstacles, in extreme conditions or for special applications. This is why the study of the mobile robot dynamics is becoming increasingly important.

Wheeled mobile robots are a class of mechanical systems characterized by non-integrable kinematical constraints. The condition of rolling motion without slipping and side-slipping between the wheel and the contact surface demands the presence of non-holonomic constraints, which are the kinematic particularity of this kind of robot.

On the other hand, mobile robots are more complex to control than serial and parallel robots, because of non-holonomic constraints. But, at the level of instantaneous velocities, mobile robots can be treated mathematically as a special type of parallel robot, having different connections to the ground in parallel.

In his paper, Angeles [72] studied some interesting aspects of the mobile robot dynamics using the formalism of Lagrange equations. Other authors (e.g. Colbaugh et al. [73]) gave a characterization of the mechanical non-holonomic systems. Volterra, Appel and Ceaplighin used also the Lagrange equations and

formalism of multipliers in the dynamics of motion with the links. Recently, neural networks appeared as powerful tools for learning dynamics of highly nonlinear systems (Kim et al. [74]).

The analysis of the problems of two-wheeled mobile robot dynamics is being done mainly to solve successfully the control of the motion of such systems. Simple models are very often accepted for a system description, even though they do not take into account the masses of the many mobile elements. Authors describing the dynamics of such systems use classical equations taken from Newtonian mechanics and, most often, they approach the motion of these systems using second-order Lagrange equations [75,76,77,78].

An equivalent parallel robot, consisting of three legs, can model a differentially driven mobile robot with two moving actuators [79, 80]. Pathak et al. [81] analyze the dynamic modeling and the position control of a series of wheeled inverted pendulums (Segway, Quasimoro, JOE) by partial feedback linearization and from a controllability point of view. Using recursive formulation, the kinematics model with a global singularity analysis is carefully discussed in [82]. Chakraborty and Ghosal have presented in their works [83, 84] the kinematics and a set of differential equations for the dynamics modeling and simulation of a wheeled mobile robot.

The Quasimoro prototype of the mobile wheeled- pendulum by Salerno and Angeles [85] is a special quasi-holonomic mechanical system, which comprises two driving wheels and an intermediate central body carrying the payload. Salerno, Ostrovskaya and Angeles studied in the paper [86] the dynamics of a rolling robot, using the second-order Lagrange equations with multipliers.

2.2 Robot Anatomy and its Kinematic Model

Let us consider a mobile robot with three conventional wheels that can roll without slipping on a horizontal surface (Fig. 1). This kind of differentially driven robots needs three non-collinear support points in order not to fall over. In practice, the robot can turn on the spot by giving opposite speeds to both actuated wheels.

The mobile robots are made up of a rigid frame with non-deformable wheels and sometimes they are moving on a fixed horizontal ground. To simplify the graphical image of the kinematical scheme of the robot, in what follows we will represent the intermediate reference systems by only two axes, as is being presented in many robotics papers [72, 80, 93]. The z_k axis is represented for each component element T_k. It is noted that the relative rotation with the angle θ_i must be always about the direction of the z_k axis.

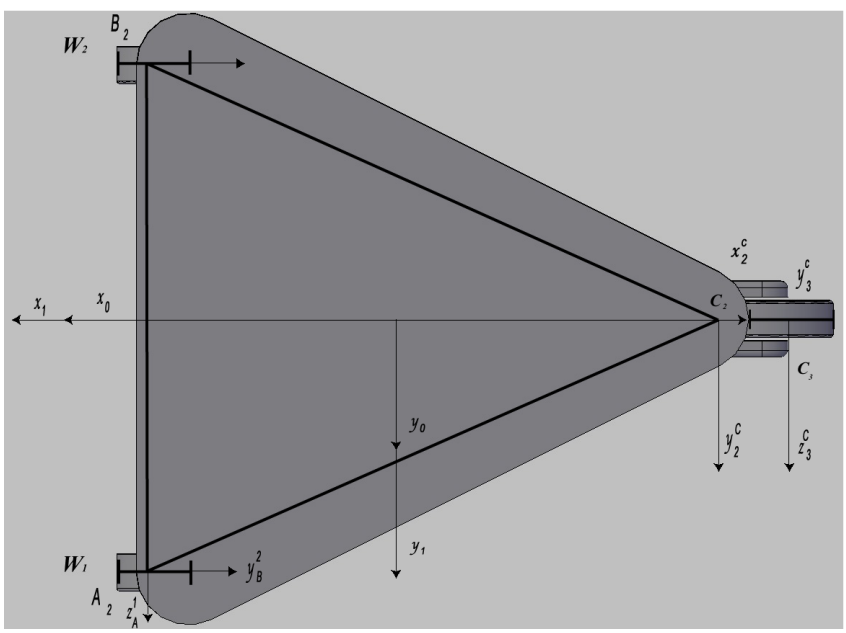

Figure 2: *Kinematic Schematic of the Robot*

The moving platform of the robot,as Staciu derived in his paper[94],is linked to a central reference frame $Gx_1y_1z_1$, is an isosceles triangle with the dimension l for the base and $a + b$ for its height. It has the mass m_1 and the tensor of inertia J_1. Two cylindrical coaxial driving wheels of the same radius r are fixed to the frames $A_2x_2{}^Ay_2{}^Az_2{}^A$,$B_2x_2{}^By_2{}^Bz_2{}^B$ and connect to chassis by means of revolute joints at the points $A_2 = O_1$ and $B_2 = O_2$. They have the masses $m_2{}^A = m_2{}^B$ and the tensors of inertia $J_2{}^A = J_2{}^B$. A crank $C_2C_3 = PO_2$ is jointed to the moving platform at the point $C_2 = P$ of the triangle. Its mass and tensor of inertial with respect to C_2 are respectively m_2^C and J_2^C. This rigid element can orientate permanently the motion of a passive rolling caster wheel of small a radiusr_0, mass m_3^C and tensor of inertia J_3^C (Fig.2). The caster wheel has no kinematical function; its only purpose is to keep the robot in balance.

Let us analyze how the motion of the robot on a curved trajectory in the turning period between two permanent rectilinear motions. The non-holonomic constraints reduce the mobile robots velocity degrees of freedom and hence the robot has only two actuated joints.

In the forward velocity kinematics (FVK), we will consider that the input rotation angles θ_1,θ_2 of the driven wheels can determine completely the instantaneous position and orientation of the robot. Thus, since the platform has a planar motion, its position with respect to a fixed reference frame $Ox_0y_0z_0$ with origin O on the horizontal ground, is given by the coordinates x_{10}, y_{10}, H and by the angle of rotation θ, which form the following matrices:

$$r_{10} = \begin{bmatrix} x_{10} \\ y_{10} \\ H \end{bmatrix}; \qquad a_{10} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

In what follows, we apply the method of successive displacements to the geometric analysis of closed-loop chains and we note that a joint variable is the

displacement required to move a link from the initial location to the actual position. If every link is connected to at least two other links, the chain forms one or more to at least two other links, the chain forms one or more independent closed loops. We call the matrix ${a^{\varphi}}_{k,k1}$,for example, the orthogonal transformation 3x3 matrix of relative rotation with the angle ${\varphi^{a}}_{k,k1}$ of link ${T^{A}}_{k}$ around ${z^{A}}_{k}$axis.

In the study of the kinematics of mobile robots, we are interested in deriving a matrix equation relating the location of an arbitrary T_k body to the joint variables. When the change of coordinates is successively considered, the corresponding matrices are multiplied. We obtain the following orthogonal transformation matrices in the reference frames [87]:

$$
\begin{aligned}
&{a^{A}}_{21} = {a^{\theta_1}}_{z} a_1; \quad {a^{B}}_{21} = {a^{\theta_2}}_{z} a_1; \\
&{a^{C}}_{21} = {a^{\theta_4}}_{z} a_2; \quad {a^{C}}_{32} = {a^{\theta_3}}_{z} a_1;
\end{aligned} \tag{2}
$$

where

$$
a_1 = \begin{bmatrix} \cos\theta_i & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \quad a_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix};
$$

$$
{a^{\theta_1}}_{z} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad where(i = 1, 2, 3, 4) \tag{3}
$$

If the distance $A_2B_2 = l$ between both actuated wheels is known, as well as the characteristic dimensions d,h of the crank PO_3, the following vectors give the invariable positions of the revolute joints A_2, B_2, C_2:

$$
{r^{A}}_{21} = \begin{bmatrix} a \\ -l/2 \\ -h_0 \end{bmatrix}; \qquad {r^{B}}_{21} = \begin{bmatrix} a \\ l/2 \\ -h_0 \end{bmatrix};
$$

$$r^C{}_{21} = \begin{bmatrix} -b \\ 0 \\ 0 \end{bmatrix}; \qquad r^C{}_{32} = \begin{bmatrix} d \\ 0 \\ h \end{bmatrix} \tag{4}$$

So, kinematics of the robot's elements is completely characterized by five relative angular velocities

$$\omega_{10} = \theta u_3; \quad \omega^A{}_{10} = \theta_1 u_3; \quad \omega^B{}_{21} = \theta_2 u_3,$$
$$\omega^C{}_{21} = \theta_4 u_3; \quad \omega^C{}_{32} = \theta_3 u_3; \quad u_3 = [0\,0\,1]^T \tag{5}$$

which are associated with the following skew-symmetric matrices:

$$\omega_{10} = \theta u_3; \quad \omega_{21} = \theta_1 u_3; \quad \omega_{21} = \theta_2 u_3;$$
$$\omega_{21} = \theta_4 u_3; \quad \omega_{32} = \theta_3 u_3; \quad u_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

Since the analyzed system of three rolling wheels is characterized by non-holonomic constraints, the matrix conditions of connectivity (7) will establish five analytical relations between the characteristic velocities of a two-degrees-of-freedom mobile robot:

$$\vec{\nu}_{10} + \tilde{\omega}_{10}\vec{r}^A_{21} = [r\dot{\theta}_1 \quad 0 \quad 0]^T$$
$$\vec{\nu}_{10} + \tilde{\omega}_{10}\vec{r}^B_{21} = [r\dot{\theta}_2 \quad 0 \quad 0]^T$$
$$a^C{}_{21}(\vec{\nu}_{10} + \tilde{\omega}_{10}\vec{r}^C_{21}) + (a^C{}_{21}\tilde{\omega}_{10}\vec{r}^C_{21} + \tilde{\omega}^C_{21})\vec{r}^C_{32} = [-r_0\dot{\theta}_3 \quad 0 \quad 0]^T. \tag{7}$$

These constraint conditions are satised if all wheels do not slip transversally and do not slip longitudinally, so that the distance over which the outer wheel surface rotates equals the distance travelled by the point on the rigid body to which the wheel axle is attached.

Indeed, we assume in FVK problem that the position and orientation of the mechanism at a given instant will be completely determined by the input rotation angles of the two actuated wheels, namely:

$$\theta_1 = \theta^*{}_1[1+\cos(\pi t/3)]\theta_2 = \theta^*{}_2[1+\cos(\pi t/3)] \tag{8}$$

Therefore, the relations (7) can provide rst the Jacobian matrix and then

the expressions of the characteristic velocities of the moving platform:

$\omega_{10} = \dot{\theta} = r(\dot{\theta}_1 - \dot{\theta}_2);$

$\nu_{10}{}^{x} = r\dot{\theta}_1 - 0.5l\dot{\theta};$

$\nu^{y}{}_{10} = -a\dot{\theta};$

$[\dot{x}_{10} \quad \dot{y}_{10} \quad 0]^T = a^T{}_{10}[\nu^{x}{}_{10} \quad \nu^{y}{}_{10} \quad 0]^T;$ (9).

Concerning the kinematics of the crank

PO_3 and the passive caster wheel jointed at the point $O_3 = C_3$, from the matrix conditions (7) we will derive a signicant differential equation and a relation containing the angular velocities $\omega^{C}{}_{21} = \dot{\theta}_4 = \dot{\psi}, \omega^{C}{}_{32} = \dot{\theta}_3$ as follows:

$d\dot{\psi} + rth\dot{e}ta_1 \sin\psi - [d + 0.5l\sin\psi + (a+b)\cos\psi]\dot{\theta} = 0$

$r_0\dot{\theta}_3 = [(a+b)\sin\psi - 0.5l\cos\psi]\dot{\theta} + r\dot{\theta}_1\cos\psi.$ (10)

In the forward position kinematics, the estimation of the relative angle of rotation and the absolute pose of the moving platform must be performed by integration of the velocity equations (9), (10).

In order to determine new conditions of connectivity of the accelerations, we could derive the matrix relations (7). Thus, the characteristic accelerations of the moving platform are immediately obtained:

$\varepsilon_{10} = \dot{\omega}_{10} = \ddot{\theta} = r(\ddot{\theta}_1 - \ddot{\theta}_2)$, $\gamma^{x}{}_{10} = r\ddot{\theta}_1 + a\dot{\theta}^2 - l\ddot{\theta}/2$ $\gamma^{y}{}_{10} = r\dot{\theta}\dot{\theta}_1 + a\ddot{\theta} - l\dot{\theta}^2/2$

$[\ddot{x}_{10} \quad \ddot{y}_{10} \quad 0]^T = a^T{}_{10}[\gamma^{x}{}_{10} \quad \gamma^{y}{}_{10} \quad 0]^T$ (11)

Note that the absolute velocities $\vec{\nu}^C_{k0}, \vec{\omega}^C_{k0}$, the accelerations $\vec{\gamma}^C_{k0}, \vec{\varepsilon}^C_{k0}$ and the useful square matrices $\tilde{\omega}^C_{k0}\tilde{\omega}^C_{k0} + \tilde{\nu}^C_{k0}$ of the third leg OC_2C_3 of the robot, for example, can be calculated with some recursive matrix formulae [20-22]:

$\vec{\nu}^C_{20} = a^C{}_{21}[\vec{\nu}_{20} + \tilde{\omega}_{10}\vec{r}^C_{21}],$

$\vec{\nu}^C_{30} = a^C{}_{32}[\vec{\nu}_{20} + \tilde{\omega}_{20}\vec{r}^C_{32}],$

$\vec{\omega}^C_{20} = a^C{}_{21}[\vec{\omega}_{10} + \vec{\omega}^C_{21}],$

$\vec{\omega}^C_{30} = a^C{}_{32}[\vec{\omega}_{20} + \vec{\omega}^C_{32}],$

$\vec{\gamma}^C_{20} = a^C{}_{21}[\vec{\omega}_{10} + (\tilde{\omega}_{10}\tilde{\omega}_{10} + \tilde{\varepsilon}_{10})\vec{\omega}^C_{32}],$

$$
\begin{aligned}
&\vec{\gamma}_{30}^{C} = a^{C}{}_{32}[\vec{\omega}_{20} + (\tilde{\omega}_{20}\tilde{\omega}_{20} + \tilde{\varepsilon}_{20})\vec{\omega}_{32}^{C}],\\
&\vec{\varepsilon}_{20}^{C} = a^{C}{}_{21}\vec{\varepsilon}_{10} + \vec{\varepsilon}_{21} + a^{C}{}_{21}\tilde{\omega}_{10}a_{21}{}^{CT}\vec{\omega}_{21}^{C},\\
&\vec{\varepsilon}_{30}^{C} = a^{C}{}_{32}\vec{\varepsilon}_{20} + \vec{\varepsilon}_{32} + a^{C}{}_{32}\tilde{\omega}_{20}^{C}a_{32}{}^{CT}\vec{\omega}_{32}^{C},\\
&\tilde{\omega}_{20}^{C}\tilde{\omega}_{20}^{C} + \tilde{\varepsilon}_{20}^{C} = a^{C}{}_{21}[\tilde{\omega}_{10}^{C}\tilde{\omega}_{10}^{C}]a^{CT}{}_{21} + \tilde{\omega}_{21}^{C}\tilde{\omega}_{21}^{C} + \tilde{\varepsilon}_{21}^{C} + 2a^{C}{}_{21}\tilde{\omega}_{10}a_{21}{}^{CT}\tilde{\omega}_{21}^{C},\\
&\tilde{\omega}_{30}^{C}\tilde{\omega}_{30}^{C}+\tilde{\varepsilon}_{30}^{C} = a^{C}{}_{32}[\tilde{\omega}_{20}^{C}\tilde{\omega}_{20}^{C}]a^{CT}{}_{32}+\tilde{\omega}_{32}^{C}\tilde{\omega}_{32}^{C}+\tilde{\varepsilon}_{32}^{C}+2a^{C}{}_{32}\tilde{\omega}_{20}a_{32}{}^{CT}\tilde{\omega}_{32}^{C},
\end{aligned}
\tag{12}
$$

2.3 Equations of Motion(Dynamic)

2.3.1 Principle of Virtual Work

Two electric motors that generate the torques $\vec{M}_1 = \vec{M}_1\vec{u}_3$ and $\vec{M}_2 = \vec{M}_2\vec{u}_3$, which have the direction of the common axis $A_2B_2 = O_1O_2$, control the evolution of the driving wheels A_2, B_2 and transmit the motion at the passive caster wheel C_2.

We will study the inverse dynamic problem, in order to establish the variation of the torques M_1, M_2 and the powers P_1, P_2 developed by the two active wheels, during the evolution of the robot between the initial position and the position which corresponds to the stationary motion. Thus, we will use an approach based on the principle of virtual work.

In every analysis, the system is considered initially at rest. It is noteworthy that the simulation runs do not account for either external dissipation, such as rolling friction between the wheels and ground, or internal dissipation, such as friction in the bearings.

The fundamental principle of virtual work states that a mechanical system is under dynamic equilibrium if and only if the virtual work developed by all external, internal and inertia forces vanishes during any general virtual displacement, which is compatible with the kinematical constraints [4,23,25].

A first set of virtual characteristic velocities of the robot bodies results easily

from the constraint conditions (7), namely:

$$\omega^{\nu}{}_{1a} = 1, \omega^{\nu}{}_{2a} = 0, \omega^{\nu}{}_{10a} = r/l, ,$$
$$\nu^{x\nu}{}_{10a} = r/2, \nu^{y\nu}{}_{10a} = -ar/l, \theta^{\nu}{}_{10a} = r/l, ,$$
$$\omega^{\nu}{}_{3a} = r/r_0[0.5\cos\psi + (a+b)/l\sin\psi],$$
$$\omega^{\nu}{}_{4a} = r/d[-0.5\cos\psi + (a+b)/l\cos\psi] + r/l,$$
$$\omega^{\nu}{}_{\varphi} = \omega^{\nu}{}_{4a} - \omega^{\nu}{}_{10a} \tag{13}$$

Assuming that the frictional forces at the joints are negligible, the virtual work produced by the forces of constraint at the joints is zero. Hence, the following compact expression of the torque applied to the right driving wheel A2 is (Staicu[24]) :

$$M_1 = \vec{\nu}^{\nu T}_{10a}\vec{F}_{10} + \vec{\omega}^{\nu T}_{10a}\vec{M}_{10} + \vec{\nu}^{A\nu T}_{20a}\vec{F}^{A}_{20} + \vec{\omega}^{A\nu T}_{20a}\vec{M}^{A}_{20} + \vec{\nu}^{B\nu T}_{20a}\vec{F}^{B}_{20} + \vec{\omega}^{B\nu T}_{20a}\vec{M}^{B}_{20} + \vec{\nu}^{C\nu T}_{20a}\vec{F}^{C}_{20} + \vec{\omega}^{C\nu T}_{20a}\vec{M}^{C}_{20} + \vec{\nu}^{C\nu T}_{30a}\vec{F}^{C}_{30} + \vec{\omega}^{C\nu T}_{30a}\vec{M}^{C}_{30} \tag{14}$$

with its analytical form

$$M_1 = m_1\omega^{\nu}{}_{10a}(0.25l^2\ddot{\theta} - 0.5lr\ddot{\theta}_1 - ar\dot{\theta}\dot{\theta}_1 + a^2\ddot{\theta}) + m_1\omega^{\nu}{}_{1a}(r^2\ddot{\theta}_1 - 0.5lr\ddot{\theta} + ar\dot{\theta}^2) + J^z{}_1\ddot{\theta}\omega^{\nu}{}_{10a} + 0.5m^A{}_2r^2(3\omega^{\nu}{}_{1a}\ddot{\theta}_1 + 0.5\omega^{\nu}{}_{10a}\ddot{\theta}) + 0.5m^B{}_2r^2(3\omega^{\nu}{}_{2a}\ddot{\theta}_2 + 0.5\omega^{\nu}{}_{10a}\ddot{\theta}) + m^C{}_2\omega^{\nu}{}_{3a}r_0[r_0\ddot{\theta}_3 - (d - x^C{}_2)\omega^2{}_{\psi}] + m^C{}_2\omega^{\nu}{}_{\psi a}[r_0(d + x^C{}_2)\dot{\theta}_3\omega_{\psi} + d^3(3h + d)\varepsilon_{\psi}/3(h + d)] + 0.5m^C{}_3r^2{}_0(3\omega^{\nu}{}_{3a}\ddot{\theta}_3 + 0.5\omega^{\nu}{}_{\psi a}\varepsilon_{\psi}), \tag{15}$$

where $\omega_{\psi} = \dot{\psi} - \dot{\theta}, \varepsilon_{\psi} = \ddot{\psi} - \ddot{\theta}$.

The force of inertial and the resultant moment of the forces of inertia have, for example, the following general form:

$$-\vec{F}^{inC}_{k0} = -m^C{}_k[\vec{\gamma^C}{}_{k0} + (\tilde{\omega}^C_{k0}\tilde{\omega}^C_{k0} + \tilde{\varepsilon}^C_{k0})\vec{r}^C_k]$$
$$-\vec{M}^{inC}_{k0} = -[m^C{}_k\tilde{r}^C_k\vec{\gamma^C}{}_{k0} + hatJ^C{}_k\tilde{\varepsilon}^C_{k0} + \tilde{\omega}^C_{k0}hatJ^C{}_k\vec{r}^C_k] \tag{16}$$

where the accelerations $\vec{\gamma^C}{}_{k0}, \tilde{\varepsilon}^C_{k0}$ and the square matrices $\tilde{\omega}^C_{k0}\tilde{\omega}^C_{k0} + \tilde{\varepsilon}^C_{k0}$ can be calculated by relations (12).

For torque M_2 of the couple applied to the left driving wheel B_2, an expression analogous to (15) results in:

$$M_2 = m_1\omega^{\nu}{}_{10b}(0.25l^2\ddot{\theta} - 0.5lr\ddot{\theta}_2 - ar\dot{\theta}\dot{\theta}_2 + a^2\ddot{\theta}) + m_1\omega^{\nu}{}_{2b}(r^2\ddot{\theta}_2 - 0.5lr\ddot{\theta} +$$

$ar\dot{\theta}^2)+J^z{}_1\ddot{\theta}\omega^\nu{}_{10b}+0.5m^B{}_2r^2(3\omega^\nu{}_{2b}\ddot{\theta}_2+0.5\omega^\nu{}_{10b}\ddot{\theta})+0.5m^A{}_2r^2(3\omega^\nu{}_{1b}\ddot{\theta}_1+0.5\omega^\nu{}_{10b}\ddot{\theta})+$
$m^C{}_2\omega^\nu{}_{3b}r_0[r_0\ddot{\theta}_3-(d-x^C{}_2)\omega^2{}_\psi]+m^C{}_2\omega^\nu{}_{\psi b}[r_0(d+x^C{}_2)\dot{\theta}_3\omega_\psi+d^3(3h+d)\varepsilon_\psi/3(h+$
$d)]+0.5m^C{}_3r^2{}_0(3\omega^\nu{}_{3b}\ddot{\theta}_3+0.5\omega^\nu{}_{\psi a}\varepsilon_\psi),$ (17)

where the following virtual velocities must be introduced:

$\omega^\nu{}_{1b}=0,\omega^\nu{}_{2b}=1,\omega^\nu{}_{10b}=-r/l,$

$\nu^{x\nu}{}_{10b}=r/2,\nu^{y\nu}{}_{10b}=ar/l,$

$\omega^\nu{}_{3b}=r(0.5\cos\psi-[a+b]\sin\psi/l)/r_0,$

$\omega^\nu{}_{4b}=-r(0.5\sin\psi-[a+b]\cos\psi/l)/d-r/l,$

$\omega^\nu{}_{4b}=\omega^\nu{}_{4b}-\omega^\nu{}_{10b},$ (18)

The matrix relation (14) constitues the inverse dynamics model of the mobile robot provided with caster wheel.

The various dynamical effects, including the Coriolis, coupling centrifugal forces and the gravitational actions, are considered in equation(14).

2.4 Equations of Motion(Kinematic)

The previous section determines and derives the equations of a dynamically driven non holonomic robot. We have provided this information, to show that this system works on a dynamic robot although we used kinematic robots from iRobot, called as "Create". The equations of motion for such kinematic robots are very straight forward. The robot is modeled by non-linear odinary equations given by

- $\dot{x}=\nu.\cos_\theta$
- $\dot{y}=\nu.\sin_\theta$
- $\dot{\theta}=u$

where ν is the linear velocity, u is the angular velocity, θ is the angular orientation with respect to x-axis.

Chapter 3

Problem, its Solution and its Application

3.1 Problem

In the previous chapter we have determined the kinematics of the robot. Therefore we have reached a point where we could control robot to achieve desired objectives. Now the actual problem is to achieve a Networked Robotic Platform where in a swarm of robots behave cooperatively to achieve a specific task by assigning a leader among themselves and the followers follow the leader in a specific formation.

For our experiments, we used four robots from iRobot, with one robot being the leader and remaining three being the followers. The number of followers is a variable that could be changed when needed but we have chosen to use all the three robots as followers because we wanted to show the formation control robustness of the system.

This entire phenomena is depicted in the following flowchart and each step is explained in detail in the latter sections.

3.2 Flowchart of the system

The entire system can be broadly classified into three sections. This is a vertical process and is illustrated in the following Figure.

Figure 3: *System Process.*

As shown above, this system is classified into three stages.

- Stage 1: Knowledge of Position and Destination. This knowledge is global and is highly necessary for the system to function.
- Stage 2: Protagonist Determination is next step in this hierarchial process which is necessary for the deermination of leader and the followers.
- Stage3: Follower Algorithm is another final step in this system which holds the leader follower and formation control algorithms.

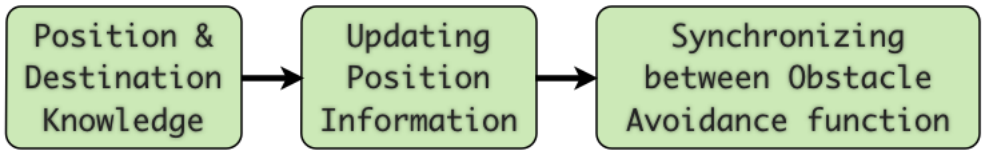

Figure 4: *Position and Destination Knowledge.*

3.2.1 Position and Destination Knowledge.

To begin with, this system needs global knowledge of position information of all the robots in the environment. This knowledge provides us the advantage of minimizing the energy consumption in the system. As shown above this stage has two substages. i) Updating the information ii) Syncronizing between Obstacle Avoidance functions.

3.2.1.1 Updating the Information:

At every interval, (0.2 secs in our case) , the position information is read using the sensors present on the robot. With the availability of a GPS, this information would be much more be qualitative. This updating is required for obstacle avoidance functions as our robots lack in other forms of perceptions.

3.2.1.2 Syncronizing between Obstacle Avoidance functions:

At every update the position information is synchronized with the obstacle avoidance functions. This step is the necessary work around to avoid obstacles when other forms of perception are absent. There are two forms of obstacles, static and dynamic. The position information of the static obstacle are also known to the system. Since we are dealing with four robots in our experiments, each robot will have three dynamic obstacles. For simplicity, no other dynamic

obstacles are present in the neighbourhood.

The next stage in the process is "Protagonist Determination". This process is determining the leader robot among the swarm of robots. As the number of robots in the swarm increase determination of "Protagonist Robot" becomes complicated. The selection process is elaborated in the following flowchart.

3.2.2 Protagonist Determination

3.2.2.1 Determining position and destination information

The robots are controlled by a central control system (which is a Windows powered computer in this case). At the end of each iteration of the cycle, the robots' positions and next destination are updated in this control system.

$$R = [R_1 R_2 R_2 .....R_n]^T \tag{19}$$

where R_i is the position information of the $'i'_{th}$ robot.

Thus 'R' matrix contains the position information of the entire population of the swarm.

3.2.2.2 Calculating nearest robot

Once the positions of the robots and destination are determined, distance between each robot and destination are determined using the formula

$$d_n = \sqrt{(x_n - x_d)^2 + (y_n - y_d)^2} \tag{20.1}$$

$$D = [d_1 \quad d_2 \quad d_3 .....d_n]^T \tag{20.2}$$

where d_i is the distance between the i_{th} robot and destination. (x_i, y_i) are the coordinates of the i_{th} robot and (x_d, y_d) are the coordinates of the current destination.

After detemining the matrix 'D', the least distance to the destination $D_{min} = min(D)$ is found out. Then all the robots that are $'D'_{min}$ meters apart from

the destination are populated in the 'L' matrix. This is the step that marks the beginning of the protagonist determination. Please look at the following flowchart.

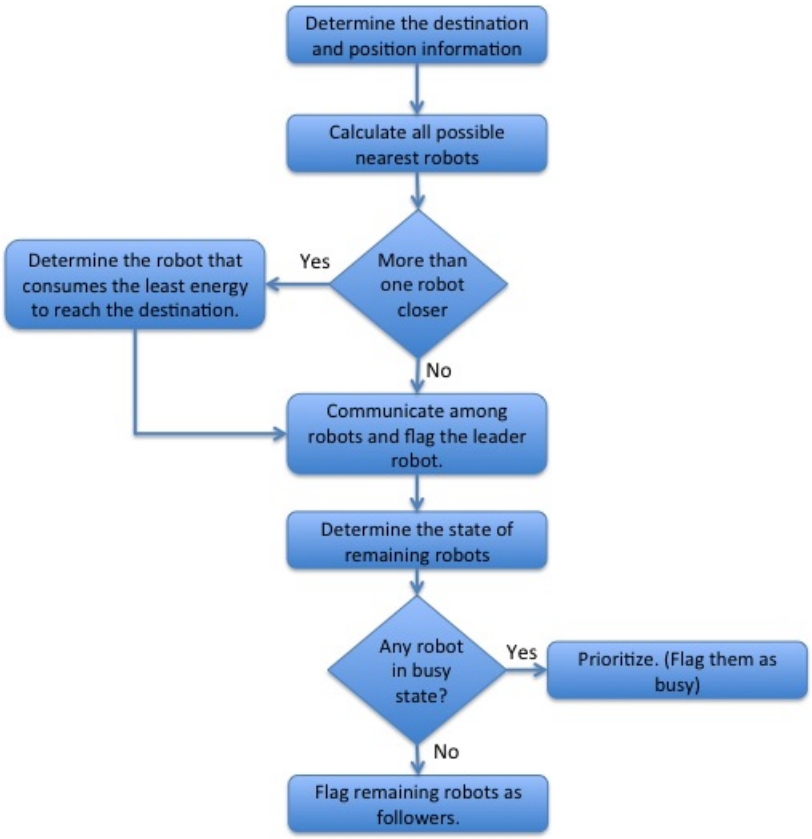

Figure 5: *Protagonist Determination.*

As depicted in the above flowchart, the robot locations and destination are updated at the end of each iteration. The algorithm then proceeds to determine all the closest possible robots to the destination and populates them in L matrix ('L' matrix is a n x 1 matrix, which contains the IDs of the closest possible robots). As the size of the swarm becomes larger, the probability of this matrix L becoming larger is closer to 1. Deploying the least energy consumption

phenomena, a protogonist robot is analytically determined from this L matrix.

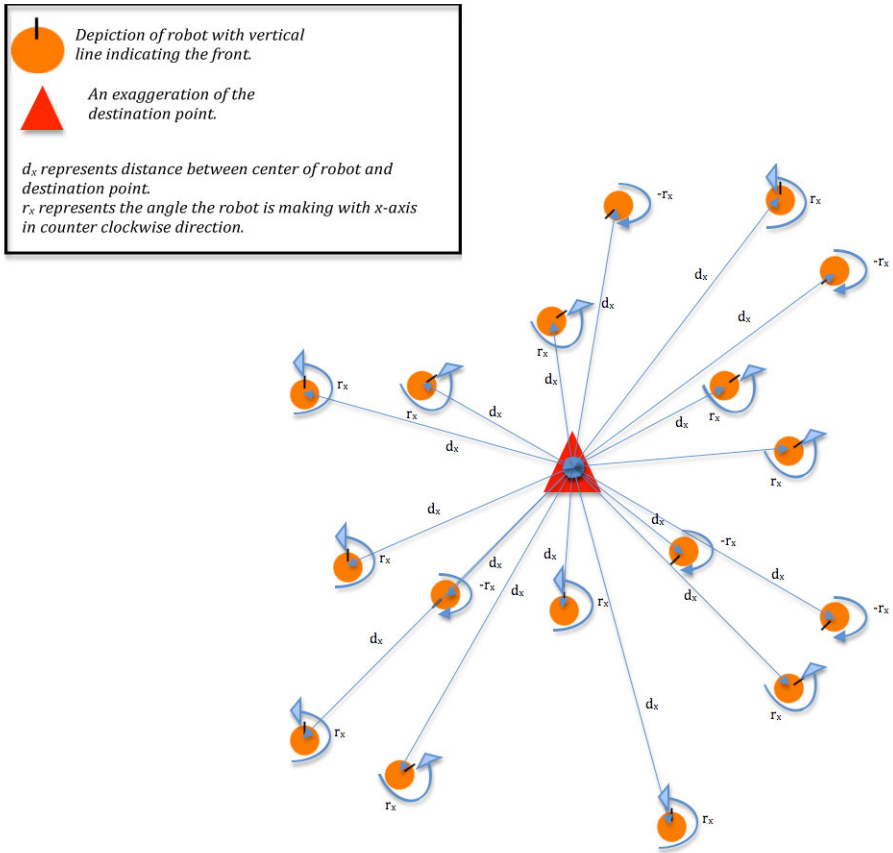

Figure 6: *An illustration of distance calculation between robots and the destination.*

$$L = [R_a \quad R_b \quad R_c....]^T$$

where R_a, R_b.. denote the "Identification (I.D) tags" of the robots. These are the tags that are assigned to each robot at the beginning of the experiment.

This implies that 'L' matrix is populated with the potential protagonist robots.

At this stage all the robots which are not a part of 'L' matrix potentially become followers and are populated in the prioritization matrix.

$$P = R - L$$

Prioritzation is discussed a few subsections below. At this stage, a protagonist a.k.a leader robot must be selected from L matrix. To determine this, energy consumed by each robot is calculated with respect to oreintation of the robot.

3.2.2.3 Communication

Once the angle is calculated and the protagonist is determined, information is passed to all the robots from the Control Console. Each robot in the system thus gets the following information.

- ID of Protagonist robot.(For Protagonist Robot, it gets the information that it is flagged as the leader robot)
- Locations of all the robots.(including location of themselves).

3.2.2.4 State Determination

Our next step is to determine the states of all the robots (excluding the protagonist). At this stage the Control Console receives the state of all the robots. '1' for busy and '0' for idle. This state determination is the crucial step for determining the follower robots which will be discussed in the "Prioritization" section.

3.2.2.5 Prioritization

This is the stage where the follower robots are determined. Before getting into proritization techniques, there is another latent phenomena called as the "Task

Management". Depending upon the robots and the tasks they could perform, a choice is given to the user to prioritize the tasks according to the need. An illustration of setting up the tasks is shown in the picture below.

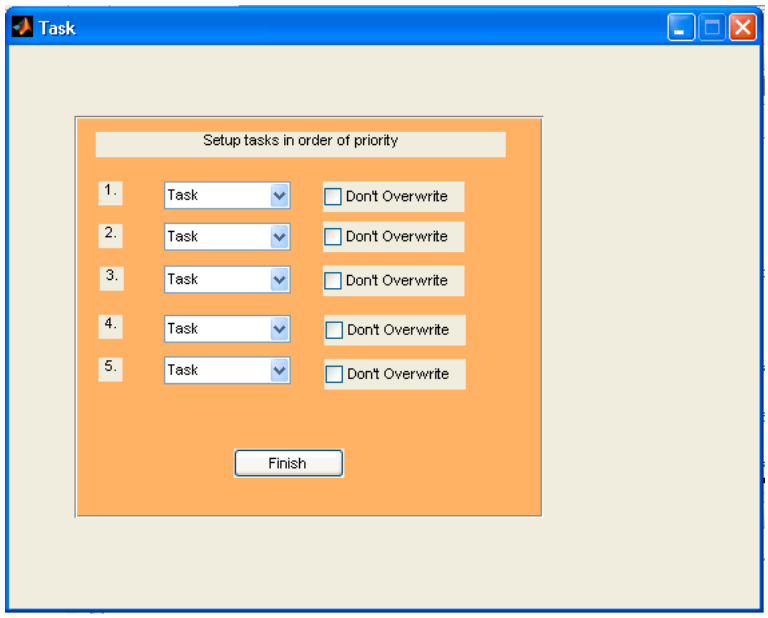

Figure 7: *An illustration of a GUI that enables to setup the prioritization prior to the experiment.*

Notice from the picture that, if the robot is busy in a task that is lower in priority to the currently required one, then it halts the current job and becomes a follower. In addition to this, if " Don't Overwrite" box is checked , then the robot does not become a follower at all until the current task is completed.

3.3 Follower Algorithm

To this point, the leader and the follower robots have been selected. Now let us look into the details of Leader - Follower Algorithm. This section broadly encompasses three major stages.

- Obstacle Avoidance

- Formation Control
- Synchronized Tracking

3.3.1 Obstacle Avoidance

As shown above, Obstacle Avoidance function is further divided into two stages. a) Static Obstacle Avoidance and b) Coordinated Obstacle Avoidance. To avoid a collision first we define a area around a robot that detects the presence of another object (obstacle or a robot). Look at the Figure below.

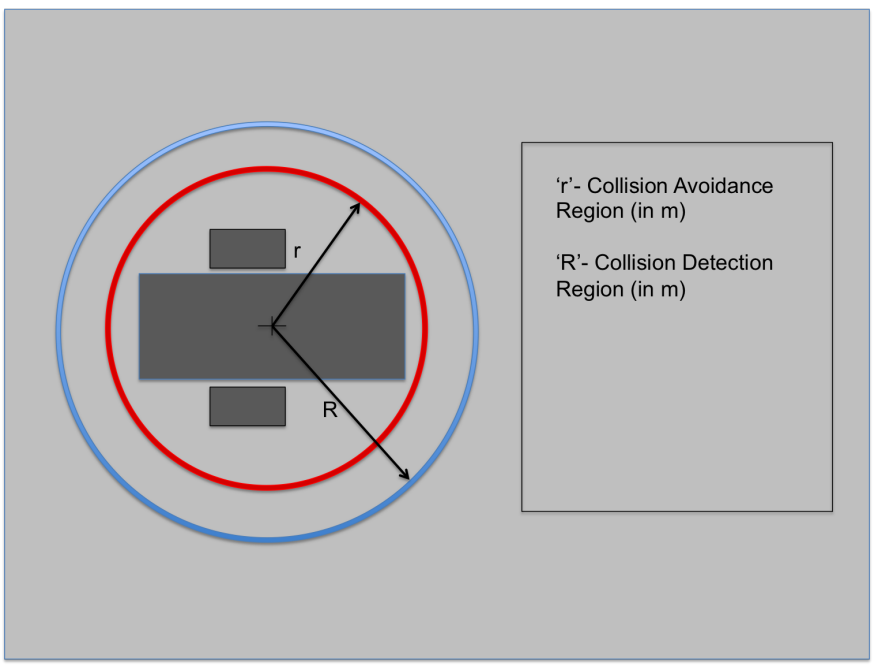

Figure 8: *An illustration of "Collision Avoidance and Detection Regions" around a robot*

Let's say that the location of the robot is (x, y) where $x\epsilon\mathbb{R}$ and $y\epsilon\mathbb{R}$ are the Cartesian coordinates. The destination is defined by (x_D, y_D) and location of obstacle is (x_O, y_O). Now position errors are defined by

$$e_x = x - x_D; e_y = y - y_D$$

. The distance function to avoid collision with an obstacle is given by

$$d_o = \sqrt{((x - x_o)^2) + ((y - y_o)^2)}$$

and distance function to reach destination is given by

$$d_D = \sqrt{((x - x_D)^2) + ((y - y_D)^2)}$$

. Now obstacle avoidance function is defined as

$$V_a = (min(0, \frac{(d_o^2 - R^2)}{(d_o^2 - r^2)}))^2$$

where $r > 0$ and $R > 0$ and R is the obstacle detection radius and r is the obstacle avoidance radius.

This function is infinite at the boundary of avoidance region and is zero outside the detection region. Thus by taking the partial derivatives for the comfort of choosing the values for different regions, we have [96]

$$if d_o >= R$$

$$\frac{\partial V_a}{\partial x} = 0; \frac{\partial V_a}{\partial y} = 0$$

$$if r < d_o < R$$

$$\frac{\partial V_a}{\partial x} = 4\frac{(R^2 - r^2)(d_o^2 - R^2)(x - x_o)}{(d_o^2 - r^2)^3}; \frac{\partial V_a}{\partial y} = 4\frac{(R^2 - r^2)(d_o^2 - R^2)(y - y_o)}{(d_o^2 - r^2)^3}$$

$$if d_o < r$$

$$\frac{\partial V_a}{\partial x} = 0; \frac{\partial V_a}{\partial y} = 0$$

Thus the desired orientation $\theta_d = Atan2(-e_y - \frac{\partial V_a}{\partial y}, -e_x - \frac{\partial V_a}{\partial x})$ and therefore the error in orientation is $e_\theta = \theta - \theta_d$.

The Figure above shows two regions around a robot schematic. One region circumfenced in blue is the "Obstacle Detection Region" and is "R" metres away from the location of the robot (geometric centre, x, y). When any object comes within this region of the robot, the robot detects the obstacle, halts for a moment i.e.,($\dot{x}, \dot{y} = 0$) immediately calculates its implications on continuing

in its current state. This means, it gives higher priority to avoid collision than to reach its destination.

And the region circumfenced in red is the "Obstacle Avoidance Region" and is "r" metres apart from robot. When any object comes within this region, the robot, halts instantly (if in motion) and moves away from the obstacle. This means that robot instantaneously changes its current state to avoid collision. Both of these phenomena are detailed in further sections.

3.3.1.1 Static Obstacle Avoidance

Consider a scenario where a robot at (x, y) has to reach a destination (x_d, y_d) and avoid a static obstacle located at (x_o, y_0) in its path. To avoid collision with this static obstacle and stay in its path as much as possible the following controller must be applied.[96]

$u = K_u.e_\theta; v = K_v.cos(e_\theta).d_D$ for gains $K_u, K_v > 0$.

Following the work of Silvia Mastellone et.al IJRR 2008, the proof is as follows.

$$E_x = e_x + \frac{\partial V_a}{\partial x}$$
$$E_y = e_y + \frac{\partial V_a}{\partial y}$$

The error dynamics are

$$\dot{e_x} = v(cos(e_\theta + \theta_d)) - \dot{x_d},$$
$$\dot{e_y} = v(sin(e_\theta + \theta_d)) - \dot{y_d},$$
$$\dot{e_\theta} = u - \dot{\theta_d}$$

Now manipulating the above equations by applying the controller, we have

$$\dot{e_x} = K_v[e_x.cos^2(e_\theta) - e_y.cos(e_\theta).sin(e_\theta)] - \dot{x_d},$$
$$\dot{e_y} = K_v[e_y.cos^2(e_\theta) + e_x.cos(e_\theta).sin(e_\theta)] - \dot{y_d},$$
$$\dot{e_\theta} = K_u.e_\theta - \dot{\theta_d}$$

Considering a Lyapunov like function

$$V = V_t + V_a$$
$$= \frac{1}{2}(e_x^2 + e_y^2 + e_\theta^2) + (min(0, \frac{(d_o^2 - R^2)}{(d_o^2 - r^2)}))^2.$$ Derivating 'V' we get

$$\frac{dV}{dt} = e_x.\dot{e_x} + e_y.\dot{e_y} + e_{theta}.\dot{e_\theta} + \frac{\partial V_a}{\partial y}.\dot{y} + \frac{\partial V_a}{\partial x}.\dot{x}$$
$$\leq -K_v.cos^2(e_\theta)(e_x^2 + e_y^2) - e_x\dot{x_d} - e_y\dot{y_d} - e_\theta(K_u e_\theta)$$

When obstacle is outside the the detection region we have $\frac{\partial V_a}{\partial x} = \frac{\partial V_a}{\partial y} = 0$ and thus the inequality comes into action.

$$\frac{dV}{dt} \leq -K_v.cos^2(e_\theta)(e_x^2 + e_y^2) - e_x\dot{x_d} - e_y\dot{y_d} - e_\theta(K_u e_\theta)$$
$$= \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}^T M \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} - \begin{bmatrix} e_x \\ e_y \\ \left\| e_\theta \right\| \end{bmatrix}^T \begin{bmatrix} \dot{x_d} \\ \dot{y_d} \\ -\epsilon_\theta \end{bmatrix}, \leq$$
$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}^T M \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \left\| \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} \right\| \left\| \begin{bmatrix} \dot{x_d} \\ \dot{y_d} \\ -\epsilon_\theta \end{bmatrix} \right\| \text{ where}$$
$$M = \begin{bmatrix} K.cos^2(e_\theta) & 0 & 0 \\ 0 & K.cos^2(e_\theta) & 0 \\ 0 & 0 & K_\theta \end{bmatrix}$$

and assuming that

$\left\| e_\theta \right\| \neq \frac{\pi}{2}$ we have $cos^2(e_\theta) > 0$. Therefore $\frac{dV}{dt}$ whenever $\left\| e \right\| > \frac{\left\| d \right\|}{\lambda_{min}(M)}$ where $e = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}$ and $d = \begin{bmatrix} \dot{x_d} \\ \dot{y_d} \\ \dot{\epsilon_\theta} \end{bmatrix}$. Therefore the stability of the error dynamics and hence tracking with bounding error, are guaranteed outside the detection region. Moreover, by increasing the gains K_u, K_v we can decrease the tracking error.

When the robot is inside the detection region($r \leq d_a \leq R$), the inequality

becomes $\frac{dV}{dt} \leq -K_v.cos^2(e_\theta).d_D^2 - \left\| e_\theta \right\| (K_u \left\| e_\theta \right\| - \epsilon_\theta)$, which is negative definite for $\left\| e_\theta \right\| > \frac{\epsilon_\theta}{K_u}$

Hence, as shown by Stipanovic et al.(2007), since $\frac{dV}{dt}$ is negative definite, then V is non-increasing inside the detection region. Since

$\lim_{\left\| z - z_a \right\| \ to \ r^+} V_a = \infty$ where $z = \begin{bmatrix} x & y \end{bmatrix}^T, z_d = \begin{bmatrix} x_d & y_d \end{bmatrix}^T$, then collision avoidance is guaranteed.

3.3.1.2 Coordinated Obstacle Avoidance

So far we have seen static collision avoidance where the robot was efficiently avoiding static obstacles where the location of the obstacle is known. As mentioned earlier, no perceptive mechanisms are available on the robot and hence except for robots and static obstacles, no other dynamic obstacles are present in the environment and hence the name of the section is "Coordinated Obstacle Avoidance" and not just dynamic obstacle avoidance. In such a situation, instantaneous location of the other robots must be known rather than location and to determine such an instantaneous locations, all the trajectories must be known to the robot to predict the motion of the robot in its trajectory and avoid collision.

Therefore the collision algorithm is exactly the same but instead of location of obstacle, we consider the trajectory of the obstacle. This calculation becomes heavier and complicated with the increase in number of robots in the environment. To avoid this cumbersome calculations, a database structure is used which is updated with trajectories, destination and location information after every cycle. (A cycle here is defined as one complete process shown in Figure 6) Thus calculations are just referenced from database that updates with information and avoids the system lag.

All being said about the similarity, the following controller is applied for

Coordinated Obstacle Avoidance.

$$u = K_{u_i}.e_{\theta_i}; v = K_{v_i}.cos(e_{\theta_i}).d_{D_i} \text{ for gains } K_u, K_v > 0.$$

With dynamic obstacles the needed information is the instantaneous location of the obstacles, i.e., the trajectories of the remaining robots.

3.3.2 Formation Control and Synchronized Tracking

Before barging into the control theory of formations, a little introduction to "Graph theory "[97] is given. **Graph theory:** In mathematics and computer science, graph theory is the study of graphs, mathematical structures used to model pairwise relations between objects from a certain collection. A "graph" in this context refers to a collection of vertices or 'nodes' and a collection of edges that connect pairs of vertices. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another. To form a particular formation, we have considered a " Cycle Graph" in the beginning and later it can emerge into any graph depending on the requirement of the user. A cycle graph is a 2-regular graph and the complete graph on n vertices is *(n-1)-regular.*(as defined by Mesbahi et.al 2010). Considering the agreement protocol which concurs with the controller applied in the system, the speed of the robots are written as (for a three robots presence in our situation),

$$v_1(t) = \frac{1}{2}((s_3(t) - s_1(t)) + (s_2(t) - s_1(t))); v_2t = s_1(t) - s_2(t); v_3t = s_2(t) - s_3(t)$$

which assumes the form

$$v(t) = \begin{bmatrix} -1 & \frac{1}{2} & \frac{1}{2} \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} .s(t)$$

where $s(t) = \begin{bmatrix} s_1(t) & s_2(t) & s_3(t) \end{bmatrix}^T$ Therefore a triangle formed by three follower robots are drawn towards their centroid until they come into each other's obstacle avoidance regions and a straight line formed by three follower robots are drawn towards their center. The illustration is shown below.

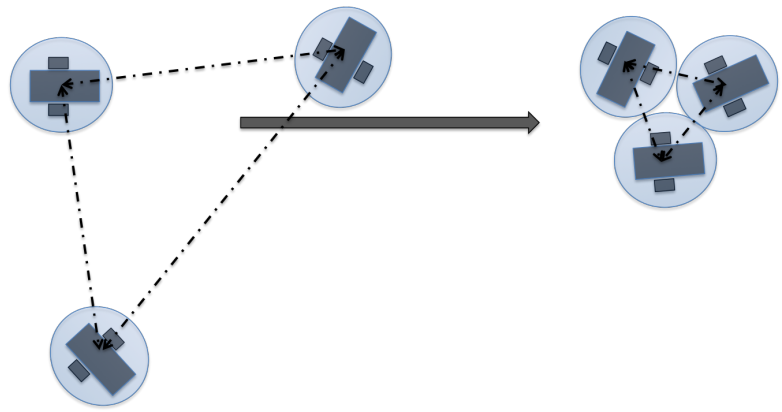

Figure 9: *Robots initially in a triangular shape drawn towards their centroid.*

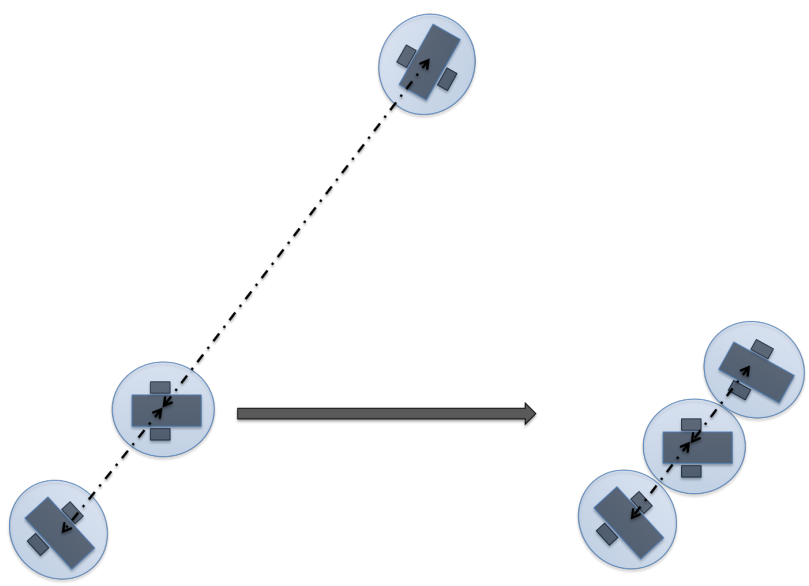

Figure 10: *Robots aligned in a straight line but still drawn towards their centroid.*

Once such a formation is accomplished, the next step is to follow the leader robot. To accomplish this phenomena, the task is divided into two stages, shape of the formation and synchronized tracking.

3.3.2.1 Formation Control.

To determine the formation control, the current state of the robot locations and alignment must be known. As seen above, the follower robots use graph theory to gather towards their centroid, but on a close observation, the robots do not maintain the same alignment when they reach the centroid destination. This is because our controller simultaneously controls the linear and angular velocities, and thus to perfectly follow the leader robot, the alignment should be taken care of. For this the alignment of the leader is determined and followers are adjusted to match their leaders' alignment. While the followers are busy in reaching their centroid, the leader aligns itself towards the final destination (or a trajectory) such that the path it travels is simple. This step is not necessary but recommended to avoid proper formation control among followers.

For example, if the leader makes ψ_l degrees with x-axis (cc) and each follower makes $\psi_{f1}, \psi_{f2}, ...\psi fn$ degrees with x-axis(cc) then each follower robot must turn $\theta_{req} = \psi_l - \psi_{fi}(i = 1, 2, 3...n)$. The next step is to calculate the relative angles(ρ_i) and distances(d_i) between the leader and each follower. This idea is shown in the Figure below.

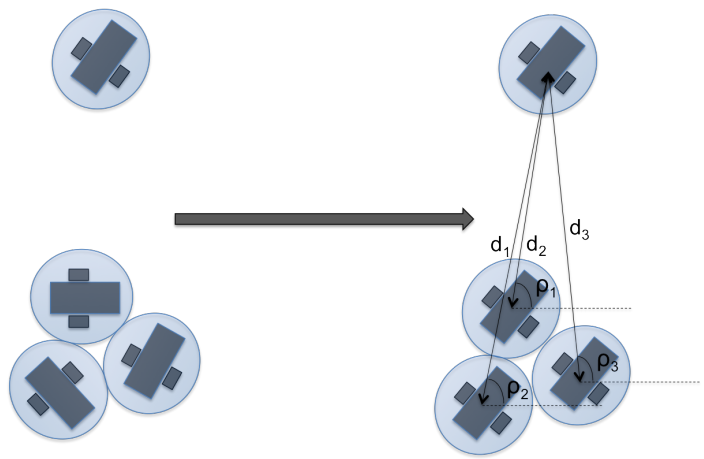

Figure 11: *Follower bots aligned w.r.t leader and relative measurements are calculated.*

As these values are calculated, it is also obvious that arrangement of these robots form a polygon. (Here at all cases only simple non-intersecting polygons are considered) and the follower robots to retain the formation follow the trajectory of the centroid of thus formed polygon. The centroid of the simple polygon is calculated as,

$$c_x = \frac{1}{6A}\sum_{i=0}^{p}(x_i + x_{i+1})(x_i y_{i+1} - y_i x_{i+1}); c_y = \frac{1}{6A}\sum_{i=0}^{p}(y_i + y_{i+1})(x_i y_{i+1} - y_i x_{i+1})$$

where p = Number of sides -1 and A is the area of polygon which is defined by

$$A = \frac{1}{2}\sum_{i=0}^{p}(x_i y_{i+1} - y_i x_{i+1})$$

3.3.2.2 Synchronized Tracking

Once these relative angles and distances are measured, it is the job of the leader to lead the group. So the leader defines the velocity and the directional angle. Thus the controller is applied to this leader which is somewhat similar to the one mentioned earlier. $v = V_L; u = K_L * (\psi_l - \alpha)$ where α is the directional angle. Common sense states that when α is changed then the shape of the formation could be changed.

3.4 Experimental Setup

The entire experiment is setup at 3C Systems Laboratory in Electrical Engineering Department at University of Tennessee. The robots used in this experiment are from iRobot corporation and are called "Create". This experiment is conducted by four such Create robots. The position information is physically updated in the beginning of the iteration in the GUI and from the next iteration, the position data is updated into the system. A Windows XP powered computer equipped with MATLAB acts as central control station. Communication is setup using Class 1 Bluetooth technology.

A Figure of GUI is depicted below.

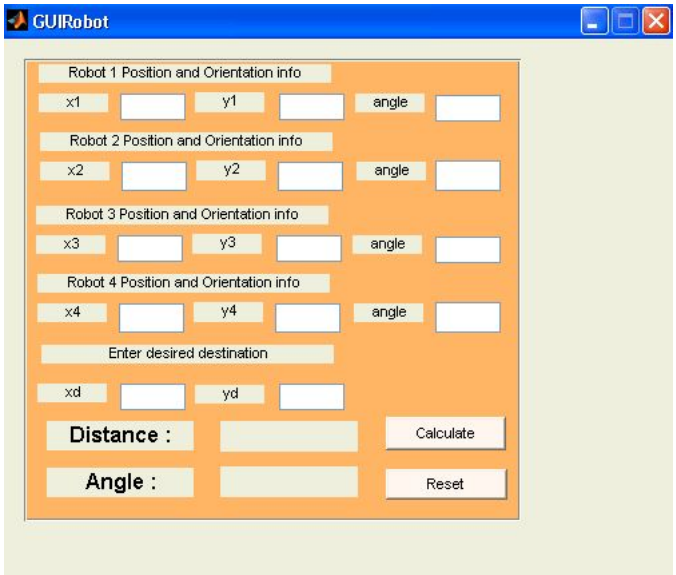

Figure 12: *GUI to enter position and destination information. The distance travelled and the angle turned by the protagonist robot in reaching the destination is shown at the bottom of this GUI.*

As shown above, a GUI is developed where in the user enters the location and orientation information of all the robots. These values could be added at any time of operation. If the are added at the beginning of the experiment,

the robots go through the entire process and the distance travelled and current orientation of robots are populated in the database. On the other hand if the values are entered during an accomplishment of certain task then the system decides whether they have to continue their current pursuit or abandon it and proceed with the new task. This decision is again tied back to the task prioritization. On extrapolating this system to a large population of robots (even better if robots are heterogeneous) this could be utilized in simultaneous task accomplishments which are discussed in detail in the next chapter.

Chapter 4

Conclusion

4.1 Future Work

This system could be developed with many additions. Many research teams across the country are using similar robots for numerous purposes. With the increase in actual number of robots, the problem complexity is increased thus increasing the efficiency of the system. Instead of using Bluetooth technology, if the robots are connected by wireless internet, many problems in communication and task accomplishments could easily be solved.

4.1.1 Simultaneous Implementation

Instead of one robot flock achieving one task at a time, several robots can be made to achieve several tasks simultaneously. Imagine a swarm of robots and more than one task is given then several flock of robots are formed among this swarm and each team would again select a protagonist to achieve the task. Once protagonist robots are decided and they are busy in achieving the task,then task(s) can be given to the remaining robots thus forming a continuous system. Applications of this could be several.

As shown in the figure above, different flocks of robots are formed and eventually a protagonist robot is picked up from each flock which will be engaged in accomplishment of the given task.

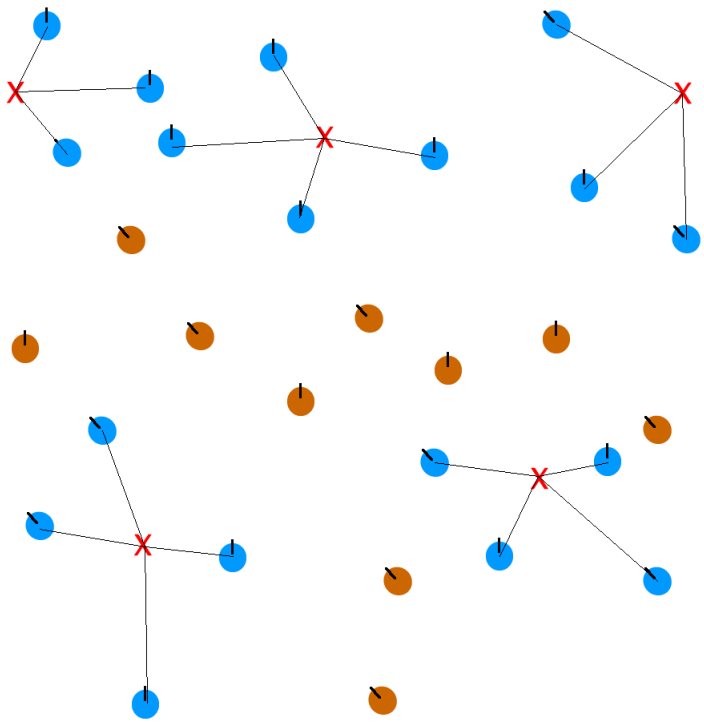

Figure 13: *An illustration of simultaneous task accomplishment.*

4.1.2 Simultaneous Multiple Task Implementation

If we closely observe this system is independent of robots that accomplish the task. So given a control system and provided with different sets of robots at different locations, different tasks could be achieved simultaneously and under a single central control system. A small depiction of this system is shown below. As mentioned on various occasion in previous chapters by introduction of heterogeneous robots the task prioritization becomes an interesting aspect by itself and optimization of that phenomena can lead to a very efficient system.

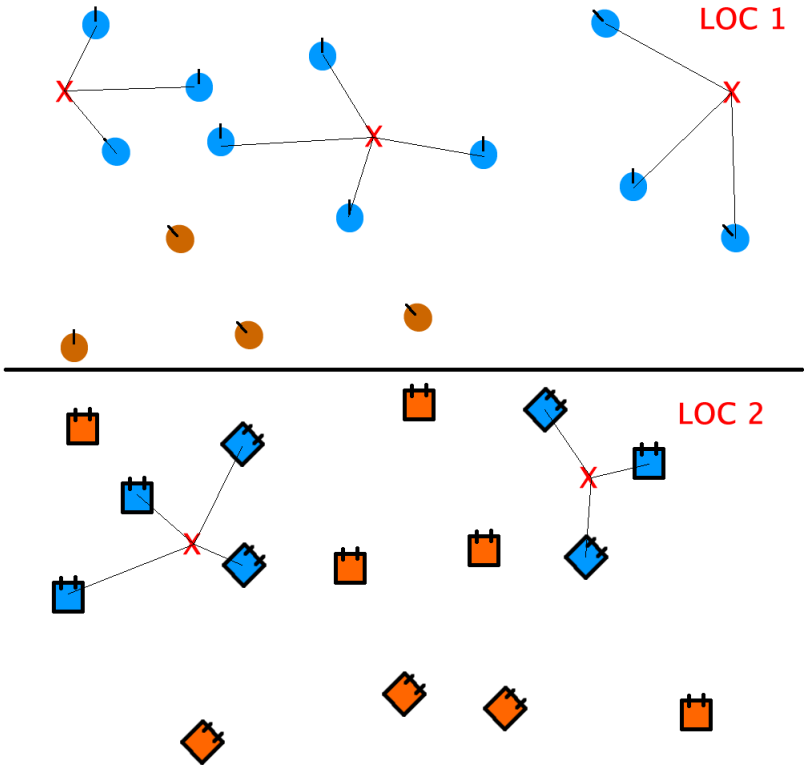

Figure 14: *A sketch depicting simultaneous task accomplishment in different locations simultaneously*

4.1.3 Machine Learning

With provision of a camera and access to the ROM on the robot, machine learning could be implemented on the robot. Static obstacles could be detected. Once the robot encounters with an obstacle, and does not recognise as a robot, then machine learning could be adapted in such a way that obstacle is detected in such a path at the particular point and the system will never take that path again. On multiple iterations of the same environment, this system could be made false proof.

Of course it is provided that the working environment might not be the same all the time but whenever an environment is changed and with a database of history of possible shapes of obstacles, the system could be adapted from one environment to another with little time for training.

4.2 Applications

Several areas like Military, Commercial and Domestic could make use of this system. For example, in military areas, majority of the bomb detection systems are currently tele-operated systems, where an operator has to continuously control the robot. With the use of this system, these robots could helped in locating hazardous places, take a picture of something suspicious and send it back to the control station. With addition of tele-operation to this system, this system could be both autonomous and semi-autonomous at the same time handling more than one task simultaneously.

Consider a swarm of robots that is running this system. Each robot apart from being autonomous is also equipped with a camera and GPS to identify its location. In war grounds these robots could be effectively used for various purposes. With a central control system provided,the user can ask the robots

to reach any point and transmit information back to the central station. With integrated of tele-operation system into the robots, the control of any robot could be taken with touch of a button. The system is illustrated here below.

Figure 15: *An example of an application where four protagonist robots are in accomplishment of tasks*

Its to be kept in mind that these images are illustrations of the actual application. As shown above the robots are three protagonist robots from different flocks attending to different tasks simultaneously.

Figure 16: *An application which shows live images transmitted from robots*

Shown above is the picture of the hand-held computer, the central control system with the user with the images transmitted by the four protagonist bots that were formed out a swarm and attending to four different tasks simultaneously . As the screen shows, one of the four images seem to be suspicious while the other three images are normal. At this stage using the provided controls, the user can switch robot 'R2' from autonomous to tele-operation.

Extensive Research is being conducted in water and air borne robots which

are applied at so many areas including military. With this system becoming self sufficient, this system could be implemented not only on ground vehicles but also in bots that operate in water.

4.3 Conclusion

Thus this system proves to be a useful system with varied applications. The only dependencies of this system is a reliable wireless network, for which researchers have found numerous ways that could cater economic to very expensive needs and customers. A few limitations of this system currently is the Bluetooth wireless system and the Create robots which are unable to transmit data among themselves but only to a Central Control System. Wireless Internet on the other hand could be a very positive alternative to it, although at this point I do not know the pros and cons using such a system on these Create Robots.

As stated in section 4.1.2, since this system offers a robust solution on optimizing the system and independent of robots, this system could be adapted to various industrial applications. This system could also open doors to industrial, commercial and domestic networks in various situations like regulation of appliances or machines in an environment etc.

References

1. R. Arkin, Motor Schema-Based Mobile Robot Navigation, The International Journal of Robotics Research 8 92112, 1989.

2. R. Arkin, Behavior-Based Robotics, The MIT Press, 0262011654, 1998.

3. E. Bah ceci, O. Soysal, E. S ahin, A Review: Pattern Formation and Adaptation in Multi-Robot Systems, Technical Report CMU-RI-TR-03-43. Carnegie Mellon Univ, Pittsburgh, PA, USA, October 2003.

4. E. Bah ceci, E. S ahin, Evolving Aggregation Behaviors for Swarm Robotic Systems: A Systematic Case Study, Proc. of the IEEE Swarm Intelligence Symposium, Pasadena, California, 2005.

5. T. Balch, Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity, Au- tonomous Robots, vol. 8, pp. 209-238, 2000.

6. T. Balch, M. Hybinette, Social Potentials for Scalable Multi-Robot Formations, IEEE International Conference on Robotics and Automation (ICRA-2000), San Francisco, 2000.

7. R. Brooks, A robust layered control system for a mobile robot, IEEE Journal of Robotics and Automation, 2(1):14 23, 1986.

8. R. Brooks, P. Maes, M. Mataric, G. More, Lunar base construction robots, In Proc. IEEE Work- shop on Intelligent Robots and Systems, Tsuchiura, Japan, 1990.

9. S. Camazine, J. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, Self-Organization in Biological Systems, Princeton University Press, 0691012113, 2001.

10. Y. Cao, A. Fukunaga, A. Kahng, Cooperative Mobile Robotics: Antecedents and Directions, Autonomous Robots, vol. 4, pp. 7-23, 1997.

11. A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, In F. Varela and P. Bourgine, editors, Proceedings of the European Conference on Artificial Life, pages 134142, Amsterdam, ECAL, Paris, France, Elsevier, 1991.

12. A. Deutsch, S. Dormann, Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis, A Birkhauser book, 0817642811, 2005.

13. T. De Wolf, G. Samaey, T. Hovoet, Analysis and Synthesis of a Bio-inspired Swarm Robotic System, In E. S ahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 56-70. Springer Verlag, Berlin, Germany, 2006.

14. Heiko Hamann and Heinz Worn, An Analytical and Spatial Model of Foraging in a Swarm of Robots, In E. S ahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 43-55. Springer Verlag, Berlin, Germany, 2006.D. Roose, Decentralised Autonomic Computing: Analysing Self-Organising Emergent Behaviour using Advanced Numerical Methods, Proceedings of the 2nd International Conference on Autonomic Computing (ICAC05), IEEE Computer Society Press, pp 52-63 June 2005.

15. M. Dorigo, E. S ahin, Swarm Robotics - Special Issue, Autonomous Robots, vol. 17, pp. 111-113, 2004.

16. G. Dudek, E. Jenkin, D. Wilkes, A taxonomy for swarm robots, In Proc. 1993 IEEE International Conference on Intelligent Robots and Systems, pp 441447, 1993.

17. B. Ermentrout, L. Edelstein-Keshet, Cellular automata approaches to biological modeling, Journal of Theo- retical Biology, 160:97-133, January 1993.

18. J. Fredslund, M. Mataric, Robots in Formation Using Only Local Sensing and Control, The 7th International Conference on Intelligent Autonomous Systems (IAS-7), Marina del Rey, California, USA, March 25-27, 2002.

19. J. Fredslund, M. Mataric, A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication, IEEE Transactions on Robotics and Automation, 18(5):837-846, 2002.

20. N. Ganguly, B. Sikdar, A. Deutsch, G. Canright, P. Chaudhuri, A survey on cellular automata, Technical report, Centre for High Performance Computing, Dresden University of Technology, December 2003.

21. M. Gardner, Mathematical Games: The Fantastic Combinations of John Conways New Solitarire Game Life, Scientific American, Volume (4)223, 120-123, Oct. 1970.

22. S. Garnier, C. Jost, R. Jeanson, J. Gautrais, M. Asadpour, G. Caprari, G. Theraulaz, Collective decision- making by a group of cockroach-like robots, In Proceedings of the 2nd IEEE Swarm Intelligence Symposium, Pasadena, California, USA, 8-10, BEST PAPER AWARD, June 2005.

23. V. Gazi, B. Fidan, Coordination and Control of Multi-agent Dynamic Systems: Models and Approaches, In E. S ahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm

Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 71-102. Springer Verlag, Berlin, Germany, 2006.

24. D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Professional, 0201157675, 1989.

25. R. Grob, M. Bonani, F. Mondada, M. Dorigo, Autonomous Self-assembly in a Swarmbot, In K. Murase, K. Sekiyama, N. Kubota, T. Naniwa, and J. Sitte, editors, Proceedings of the Third International Symposium on Autonomous Minirobots for Research and Edutainment, pages 314322. Springer Verlag, Berlin, 2006.

26. H. Gutowitz, Cellular Automata: Theory and Experiment, The MIT Press; 1st Mit Pr edition, 0262570866, 1991.

27. A. Hayes, P. Dormiani-Tabatabaei, Self-Organized Flocking with Agent Failure: Off-Line Optimization and Demonstration with Real Robots, Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation, Washington DC, USA, pp. 3900-3905, May 2002.

28. S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 0132733501, 1998.

29. J. Hertz, A. Krogh, R. Palmer, Introduction to the Theory of Neural Computation, Perseus Books Group, 0201515601, 1991.

30. A. Howard, M. Mataric, G. Sukhatme, An Incremental Self-Deployment Algorithm for Mobile Sensor Net- works, Autonomous Robots, Special Issue on Intelligent Embedded Systems, 2002.

31. A. Howard, M. Mataric, G. Sukhatme, Mobile Sensor Network Deployment using Potential Fields: A Dis- tributed, Scalable Solution to the Area Coverage Problem, DARS 02, Fukuoka, Japan, June 2002.

32. A. Ijspeert, A. Martinoli, A. Billard, L. Gambardella, Collaboration through the exploitation of local inter- actions in autonomous collective robotics: The stick pulling experiment, Autonomous Robots, vol. 11, no. 2, pp. 149171, Kluwer Academic Publishers,Analysis and Synthesis of a Bio-inspired Swarm Robotic System, In E. S ahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 56-70. Springer Verlag, Berlin, Germany, 2006.

33. Heiko Hamann and Heinz Worn, An Analytical and Spatial Model of Foraging in a Swarm of Robots, In E. S ahin, W. Spears and A. Winfield, editors, Proceedings of the Second International Workshop on Swarm Robotics at SAB 2006, volume 4433 of Lecture Notes in Computer Science, pages 43-55. Springer Verlag, Berlin, Germany, 2006. 2001.

34. A. Ilachinski, Cellular Automata: A Discrete Universe, World Scientific Publishing Company, 9810246234, 2001.

35. L. Iocchi, D. Nardi, M. Salerno, Reactivity and Deliberation: A Survey on Multi-Robot Systems, Balancing Reactivity and Social Deliberation in Multi-Agent Systems, From RoboCup to Real-World Applications (selected papers from the ECAI 2000 Workshop and additional contributions), 3-540-42327-3, 934, Springer-Verlag, London, UK, 2001.

36. R. Jeanson, C. Rivault, J. Deneubourg, S. Blancos, R. Fourniers, C. Jost, G. Theraulaz, Self-Organized aggregation in cockroaches, Animal Behaviour, 69, 169-180, 2005.

37. I. Kevrekidis, C. Gear, J. Hyman, P. Kevrekidis, O. Runborg, C. Theodoropoulos, Equation-free, coarse-grained multiscale computation: enabling

microscopic simulators to perform system-level tasks, Communications in Mathematical Sciences, 1:715-762, 2003.

38. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research 5(1):9098, 1986.

39. T. Labella, M. Dorigo, J. Deneubourg, Self-Organised Task Allocation in a Group of Robots, In R. Alami, editor, Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS04). Toulouse, France, June 23-25, 2004.

40. T. Labella, M. Dorigo, J. Deneubourg, Efficiency and Task Allocation in Prey Retrieval, In A.J. Ijspeert, D. Mange, M. Murata, and S. Nishio, editors, Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT2004), Lecture Notes in Computer Science, pages 32-47. Springer Verlag, Heidelberg, Germany, 2004.

41. T. Labella, Prey Retrieval by a Swarm of Robots, Technical Report TR/IRIDIA/2003-16, IRIDIA, Universit Libre de Bruxelles, Brussels, Belgium, DEA thesis, May 2003.

42. K. Lerman, A. Martinoli, A. Galstyan, A Review of Probabilistic Macroscopic Models for Swarm Robotic Systems, Proc. of the Swarm Robotics Workshop at the Eight Int. Conference on the Simulation of Adaptive Behavior SAB-04, E. S ahin and W. Spears, editors, July 2004, Los Angeles, CA. Lecture Notes in Computer Science, 2004.

43. L. Li, A. Martinoli, Y. Abu-Mostafa, Learning and Measuring Specialization in Collaborative Swarm Systems, In: Adaptive Behavior, vol. 12, num. 34 (2004), p. 199212, 2004.

44. A. Martinoli, F. Mondada, Collective and cooperative group behaviours: Biologically inspired experiments in robotics, In O. Khatib and J. K. Salisbury, editors, Proceedings of the Fourth International Symposium on Experimental Robotics ISER-95, pages 310, Stanford, U.S.A., Springer Verlag, June 1995.

45. A. Martinoli, K. Easton, Modeling Swarm Robotic Systems, Proc. of the Eight Int. Syrup. on Experimental Robotics ISER-02, SantAngelo dlschia, Italy, July, 2002. Springer Tracts in Advanced Robotics, pp. 285-294, 2003.

46. A. Martinoli, K. Easton, W. Agassounon, Modeling Swarm Robotic Systems: A Case Study in Collaborative Distributed Manipulation, Special Issue on Experimental Robotics, B. Siciliano, editor, Int. Journal of Robotics Research, Vol. 23, No. 4, pp. 415-436, Invited paper, 2004.

47. M. Mataric, Designing Emergent Behaviors: From Local Interactions to Collective Intelligence, Proceedings, From Animals to Animats 2, Second International Conference on Simulation of Adaptive Behavior (SAB-92), J-A. Meyer, H. Roitblat and S. Wilson, eds., MIT Press, 432-441, 1992.

48. M. Mataric, Reinforcement Learning in the Multi-Robot Domain, Autonomous Robots 4, 73-83, 1997.

49. T. Mitchell, Machine Learning, McGraw-Hill Science/Engineering/Math, 0070428077, 1997.

50. R. Montemanni, L. Gambardella, Swarm approach for a connectivity problem in wireless networks, Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2005), 265-272, Pasadena, U.S.A., June 2005.

51. J. Neumann, The general and logical theory of automata, L.A. Jeffress, ed.,

Cerebral Mechanisms in Behaviour ,The Hixon Symposium, John Wiley Sons, New York, pp. 1-31, 1951.

52. S. Nouyan, M. Dorigo, Chain Formation in a Swarm of Robots, Technical Report TR/IRIDIA/2004-18, IRIDIA - University Libre de Bruxelles, Belgium, March 2004.

53. S. Nouyan, Path Formation and Goal Search in Swarm Robotics, Technical Report TR/IRIDIA/2004-14, IRIDIA - University Libre de Bruxelles, Belgium, DEA Thesis, September 2004.

54. D. Payton, M. Dally, R. Estkowski, M. Howard, C. Lee, Pheromone robotics, Autonomous Robots, 11(3), 2001.

55. D. Payton, R. Estkowski, M. Howard, Pheromone Robotics and the Logic of Virtual Pheromones, SAB 2004: 45-57.

56. W. Shen, C. Chuong, P. Will, Simulating Self-Organization for Multi-Robot Systems, International Conference on Intelligent and Robotic Systems, Switzerland, 2002.

57. W. Shen, P. Will, A. Galstyan and C. Chuong, Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms, Autonomous Robots, vol. 17, pp. 93-105, 2004.

58. O. Soysal, E. S ahin, Probabilistic Aggregation Strategies in Swarm Robotic Systems, Proc. of the IEEE Swarm Intelligence Symposium, Pasadena, California, 2005.

59. W. Spears, D. Spears, J. Hamann, R. Heil, Distributed, Physics-Based Control of Swarms of Vehicles, Autonomous Robots, Volume 17(2-3), August 2004.

60. L. Steels, Cooperation between distributed agents through self-organisation, In Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Elsevier Science Publishers Holland, 175196, 1990.

61. R. Sutton, A. Barto, Reinforcement Learning: An Introduction, The MIT Press, 0262193981, 1998.

62. P. Tangamchit, J. Dolan, P. Kosla, The necessity of average rewards in cooperative multirobot learning, In IEEE International Conference on Robotics and Automation. ICRA 02., pages (2)1296 1301, 2002.

63. V. Trianni, T. Labella, R. Grob, E. S ahin, M. Dorigo, J. Deneubourg, Modeling Pattern Formation in a Swarm of Self-Assembling Robots, Technical Report TR/IRIDIA/2002-12, IRIDIA, Universit Libre de Bruxelles, Bruxelles, Belgium, May 2002.

64. V. Trianni, R. Grob, T. Labella, E. S ahin, M. Dorigo, Evolving Aggregation Behaviors in a Swarm of Robots, In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim and J. Ziegler, editors, Advances in Artificial Life - Proceedings of the 7th European Conference on Artificial Life (ECAL), Lecture Notes in Artificial Intelligence 2801, pages 865-874, Springer Verlag, Heidelberg, Germany, 2003.

65. V. Trianni, M. Dorigo, Emergent Collective Decisions in a Swarm of Robots, In Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005), pages:241-248 June 8-10, 2005

66. V. Trianni, S. Nolfi, M. Dorigo, Cooperative Hole Avoidance in a Swarm-bot, Robotics and Autonomous Systems, Volume 54, number 2, pp. 97-103, 2005.

67. C. Watkins, Learning from Delayed Rewards, Thesis, University of Cambidge, England, 1989.

68. E. Yang, D. Gu, Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey, CSM-404, Technical Reports of the Department of Computer Science, University of Essex, 2004.

69. Agullo,J.,Cardona,S.,Vivancos,J.:Kinematics of vehicle with directional sliding wheels. Mech. Mach. Theory 22(4), 295301 (1987)

70. Abou-Samah, M., Krovi, V.: Cooperative frameworks for multiple mobile robots. In: CCToMM Symposium on Mechanisms, Machines and Mechatronics, Canadian Space Agency (Saint-Hubert) Montreal, Canada (2001)

71. Muir, F.P., Neuman, C.P.: Kinematic modeling of mobile robots. J. Robot. Syst. 4(2), 281304 (1987)

72. Angeles, J.: Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms. Springer, New York (2002)

73. Colbaugh, R., Trabatti, M., Glass, K.: Redundant Non- Holonomic Mechanical System. Characterization and Control. Robotica, vol. 17. Cambridge University Press, Cambridge (1999)

74. Kim, Y.H., Lewis, F.L., Abdallah, C.T.: A dynamic recur- rent neural-network based adaptive observer for a class of non-linear systems. Automatica 33(8), 15391543 (1997)

75. Giergiel, J., Zylski, W.: Description of motion of a mobile robot by Maggies equations. J. Theor. Appl. Mech. 43(3), 511521 (2005)

76. Hendzel, Z.: An adaptive critic network for motion control of a wheeled mobile robot. Nonlinear Dyn. 50(4), 849855 (2007)

77. Velinski,S.A.,Gardner,J.F.:Kinematics of mobile manipulator and implications for design. J. Robot. Syst. 17(6) (2000)

78. Papadopoulos, E., Poulakakis, E.: On motion planning of nonholonomic mobile robots. In: Proceedings of the International Symposium on Robotics, Montreal, Canada, pp. 7782 (2000)

79. Carricato, M., Parenti-Castelli, V.: Singularity-free fully- isotropic translational parallel mechanisms. Int. J. Robot. Res. 21, 2 (2002)

80. Merlet,J.P.:Parallel Robots.Kluwer Academic Press,Dordrecht (2000)

81. Pathak,K.,Franch,J.,Agrawal,S.K.:Velocityandposition control of a wheeled inverted pendulum by partial feedback linearization. IEEE Trans. Robot. 21(3), 505513 (2005)

82. Gracia, L., Tornero, J.: Kinematic modelling and singularity of wheeled mobile robots. Adv. Robot. 21(7), 793816 (2007)

83. Chakraborty, N., Ghosal, A.: Kinematics of wheeled mo- bile robots on uneven terrain. Mech. Mach. Theory 39(12), 12731287 (2004)

84. Chakraborty,N.,Ghosal,A.:Dynamic modelling and simulation of a wheeled mobile robot for traversing uneven terrain without slip. J. Mech. Des. 127(5), 901909 (2005)

85. Salerno, A., Angeles, J.: On the nonlinear controllability of a quasiholonomic mobile robot. In: Proceedings of the IEEE International Conference on Robotics Automation ICRA2003, Taipei, Taiwan, pp. 33793384 (2003)

86. Salerno, A., Ostrovskaya, S., Angeles, J.: The dynamics of a novel rolling robot-analysis and simulation. In: Proceedings of the 11th World Congress in Mechanism and Machine Science, Tianjin, China, pp. 19561960 (2004)

87. Staicu, S., Zhang, D., Rugescu, R.: Dynamic modelling of a 3-DOF parallel manipulator using recursive matrix relations. In: Robotica, vol. 24, pp. 125130. Cambridge Uni- versity Press, Cambridge (2006)

88. Staicu, S., Liu, X.J., Wang, J.: Inverse dynamics of the HALF parallel manipulator with revolute actuators. Non- linear Dyn. 50(12), 112 (2007)

89. Staicu,S.,Zhang,D.:Anoveldynamicmodellingapproach for parallel mechanisms analysis. Robot. Comput. Inte- grated Manuf. 24(1), 167172 (2008)

90. Staicu, S., Carp-Ciocardia, D.C.: Dynamic analysis of Clavels delta parallel robot. In: Proceedings of the IEEE International Conference on Robotics Automation ICRA2003, Taipei, Taiwan, pp. 41164121S (2003)

91. Staicu, S.: Inverse dynamics of a planetary gear train for robotics. Mech. Mach. Theory 43(7), 918927 (2008)

92. Staicu, S.: Relations matricielles de recurrence en dynamique des mecanismes. Rev. Roum. Sci. Tech. Ser. Mec.Appl. 50(13), 1528 (2005)

93. Tsai, L.W.: Robot Analysis: The Mechanics of Serial and Parallel Manipulators. Wiley, New York (1999)

94. Stefan Staicu:Dynamics equations of a mobile robot provided with caster wheel.Nonlinear Dyn (2009) 58: 237248

95. Levent Bayindir, Erol Sahin: A Review of Studies in Swarm Robotics.

96. Silvia Mastellone, Dusan M.Stipanovic, Christopher R.Graunke,Koji A.Intlekofer, Mark W.Spong: Formation Control and Collision Avoidance for Multi-agent Non-holonomic Systems: Theory And Experiments.

97. Graph Theory: Wikipedia(en)

Vita

Krishna C Kalavacharla received his B.S. degree in electrical engineering in 2003 from Anna University, India. He is currently pursuing his M.S. degree at University of Tennessee Knoxville. His research areas include robot formation control, robot colonization(inspired from Ant colonization). Web development and Java are his two most favourite hobbies along with music and computer games. Also currently working on his patents for Banking sector. Aspires to become a entrepreneur in future.