19

Multi – Agent System Concepts Theory and Application Phases

Adel Al-Jumaily, Moha'med Al-Jaafreh
University of Technology Sydney
Australia

1. Introduction

This chapter presents a recent research studies in multi-agent system, which has been using in many areas to improve performance and quality as well as reduce cost and save time by using available agents to solve many problems. Pursuit problem will be discussed to demonstrate the cooperation of multi agent systems as multi agent robots. Different pursuit algorithms are implemented for acquiring a dynamic target in unknown environment.

The multi-agent system concepts appeared recently and it is extremely distributed in all research areas; to solve problems by many agents cooperation. These agents, which have been using for multi-agent system, are defined as an entity; software routine, robot, sensor, process or person, which performs actions, works and makes decision (Arenas & Sanabria, 2003). In human society's concepts, the cooperation means "an intricate and subtle activity, which has defied many attempts to formalize it" (D'Inverno et al., 1997). Artificial and real social activity in social systems is the paradigm examples of Cooperation. In multi-agent concepts side, there are many definitions for cooperation; the most popular definitions are

Definition 1: "The multi-agents working together for doing something that creates a progressive result such increasing performance or saving time" (Gustasfon & Matson, 2003).

Definition 2: "One agent adopts the goal of another agent. Its hypothesis is that the two agents have been designed in advance and, there is no conflict goal between them, furthermore, one agent only adopts another agent's aim passively."

Definition 3: "One autonomous agent adopts another autonomous agent's goal. Its hypothesis is that cooperation only occurs between the agents, which have the ability of rejecting or accepting the cooperation" (Changhong et al., 2002).

The multi-agents system is divided to theory and application phases (Changhong et al., 2002). Cooperation taxonomy, cooperation structure, cooperation forming procedure and others are related to theory phase. For application phases, the mobile agent cooperation, information gathering, sensor information and communication and others have been studied. The following sections will show the latest researches of multi-agent system from both theory and application phases.

2. Theory of Multi-Agent

The theory of any science forms the core and facilitates understand ability and documentary of that science (e.g. multi agent system). To explain multi-agents cooperation theory the

Source: Mobile Robots, Moving Intelligence, ISBN: 3-86611-284-X, Edited by Jonas Buchli, pp. 576, ARS/plV, Germany, December 2006

Cooperation structure, Cooperative problem solving, Evolution of cooperation, Negotiation, Coalition and Cooperation taxonomy, will be discussed in detail in next sections.

2.1 Cooperation Structure

Cooperation in multi agent system is divided to complete structure and incomplete structure depending on the goal dividing. These complete and incomplete cooperation structures include the following cooperation structures:

a) Cooperation of one agent to one agent coalition (CATC).
b) Cooperation of one agent coalition to one agent (CCTA). (e.g. goal needs a group of agents to complete together, or group of agents can complete more effectively at less cost than one agent.)
c) Cooperation of one agent coalition to another agent coalition (CCTC).
d) Cooperation of two agent coalitions on each other's goal exchanging (CGE). (e.g. two agents (or agent groups) adopt each other's goal).

These structures are illustrated in figure (1).

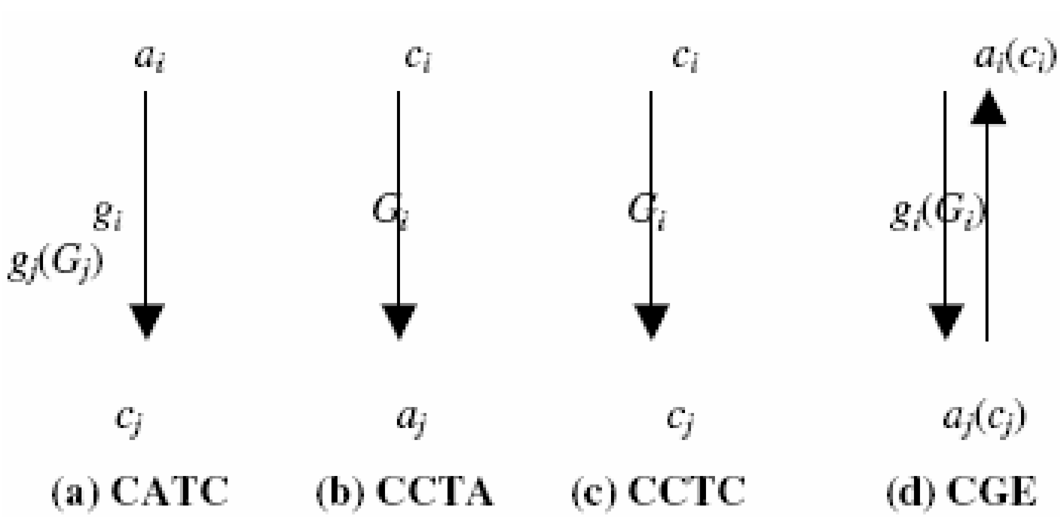


Fig. 1. Types of multi-agents cooperation structure.

Where a_i donates agent *i*, g_i *denotes* the goal of a_i, c_i donates agent coalition *I*, G_i donates the goal of c_i.

The cooperation between agents is implemented through different communication techniques. The Speech Act is the basis communication technique of multi-agent interaction and it can be isolated from the content that speech embodied (Changhong et al., 2002). Speed act technique includes many communication primitives such as request cooperation, reply to the request, receive affirm, reject and other primitives.

In other hand, the cooperation structure is classified to implicit cooperation, explicit cooperation, and dynamic cooperation according to three dimensions of taxonomy (Gustasfon & Matson, 2003):

1) The amount of information that an agent has about other agents.
2) The type of communications between agents.
3) The amount of knowledge that agent has about the goals.

There are different states of taxonomy dimensions; which determine the cooperation types and facilitate the distribution of tasks between agents. These states are illustrated in table (1).

Information	Communication	Goal knowledge
Local state information	Implicit communication	Agent knows about its goal
Neighbourhood state information	Explicit communication "broadcast message"	Agent knows about its goal and the neighbours' goals
Global state information	Send and receive messages between robots	Agent knows about all robots' goals

Table 1. States of taxonomy dimensions.

From states of taxonomy dimensions table:

1) The implicit cooperation type consists of agents know about their goals or about neighbours' goals, use implicit communication and have information of local or neighbourhood agents' states.
2) The explicit cooperation type consists of agents know about all agents' goals, send messages to specific agent and have information of neighbourhood or global information.
3) The dynamic cooperation type consists of agents know about all agents' goals, send and receive messages between agents and have information of global state.

2.2 Cooperative problem solving

Performing cooperative problem solving requests a great deal of knowledge between agents. Agents decompose problem to sub-problems; which facilitates allocating tasks between them. There are many conditions to achieve cooperation between agents.
P. M. Jones & Jacobs listed these conditions (Jones & Jacobs, 2000):

1) There must be two or more reasoning agents.
2) Agents must be sharing the same environment, so that in some sense, their actions or the effects of their actions can be mutually perceived.
3) Each agent must be contributing by some productive elements, and they must be working together
4) Availability of local goals.

Jones's theory assumed some tenets of human-computer cooperative solving. These tenets are human in charge, mutual intelligibility, openness and honesty, multiple perspectives and management of trouble. In addition, some degrees of mutual understanding or shared meaning is required for human–computer cooperative problem solving; as example the computer "partner" is based on a normative model of task requirements. (Jones & Mitchell, 1991)
Cooperative Problem Solving concepts are metaphors, functional purpose, general functions, general and specific material, social mechanisms for cooperative problem solving, and cooperative mechanisms. Table (2) shows the concepts and their elaboration and examples.
Models of cooperative problem require a domain of knowledge to facilitate communications of goals, information and the coordination of activity. Bainbridge argues for the notion of contextual models; which highlights active seeking and structuring of information for building up a "structure of inference" about the current and future potential states of the system; such as prediction, planning, anticipation, flexibility, adaptability, organization of behaviour and management of multiple concurrent activities (Bainbridge, 1993).

Cooperative problem solving concept	Elaboration and Examples
Metaphors	Information processing system, culture and machine
Functional purpose	Augmentative, integrative and debative
General functions	Sharing goals or information and coordination activity (articulation work, redirection of attention)
General and specific material and social mechanisms for cooperative problem solving	Space: remote or proximate Time: synchronous or asynchronous Directness: Direct or machine-mediated Autonomy: distributed or collective e-mails, systems, artefacts, computer software and social organisation
Cooperative mechanisms	Request-based, inference-based and structurally-based

Table 2. Cooperative problem solving concept.

Operator function model (OFM) plan and goal graph contextual control are used to model human–machine interaction and assist in the design of user interfaces, intelligent associate systems and intelligent tutoring systems (Mitchell, 1999). OFM is a model of heterarchic-hierarchic network of nodes, which embodies how a human operator manages multiple synchronized activities in a dynamic event-driven world.
In OFM, Expert systems control intelligent associate systems; which are intended to act as intelligent resources for competent, rather than off-line consultants of presumably novice users. There are many examples of intelligent associate systems; such as the expert critic approach, IDEA, pilot's associate system and ISAM-CoSMO-ASPIRE family.

2.3 Evolution of cooperation

The competitive evolution is type of cooperation evolution; which used to facilitate emergent behaviour practically cooperation. This type is employed in pursuit and evasion domain as well as interrelated predator-pray systems. There are different artificial approaches of competitive evolution applied in pursuit domain; single pool approach, plasticity approach and multiple pools approach. A system of robots uses a single genotype illustrates a single pool approach, a system of robots uses learning mechanism characterizes plasticity approach and multiple pools approach is used for robots that share different genotypes (Nitschke, 2003).
Floreano et al. appraised a competitive evolutionary of cooperation in predator-pray scenario by using two mobile robots in evolutionary robotics experiments. Then they compared their procedure with single agent evolution. Finally, a fast comparative evolution of different behavioural strategies was observed with competitive co-evolution (Floreano et al., 1998).
G. Nitschke employed cooperative co-evolution in his research of pursuit evasion game. Briefly, His research is about a team of three robots "pursuers" cooperating to halt one of other three robots "evaders" (Nitschke, 2003). He used the multiple pools competitive evolution approach and yielded great performance compared with single pool and plasticity approaches. The multiple pools present that performance; because each robot of "pursuers and evaders" assigned to genotype from different population, and this encourages behavioural specialization of the team. In the other hand, single pool approach provides simplicity in calculation and behavioural encoding of team fitness as well as plasticity approach allows specialization of team member behaviour.

Nitschke used three strategies of cooperative pursuit by using three pursuers in each strategy:

1) Encirclement strategy: pursuers encircle and in close to an evader to force it to move in same direction then spin in its current position till tired.
2) Entrapment strategy: two pursuers moved in the same direction of evader different sides and third pursuer blocks from front to mobilize evader.
3) Role-switch strategy: two pursuers moved in the same direction of evader different sides and third pursuer moved around the other pursuers to mobilize evader in triangular formation.

2.4 Negotiation in Multi-Agents

The multi-agent cooperation was defined in third definition as “The multi-agents working together for doing something” (Gustasfon & Matson, 2003). The vital member in multi-agents technology is group working; which needs a communication and negotiation between agents. Negotiation means “A key form of interaction that enables groups of agents to arrive at a mutual agreement regarding some belief, goal or plan” (Beer et al., 1998).

The negotiation between agents is implemented by different types, such as argumentation, protocols in the style of the contract net and auctions. The selection of negotiation type depends on the environment of problem, which has to be solved (D’Inverno et al., 1997).

The work group at IWMAS’98 suggested that the cases of negotiation can be distinguished by:

1. Challenges occur in managing agents interactions.
2. The larger processes where agents contribute.
3. Mechanisms will be used to maintain consistency of the challenges.
4. The complexity of negotiation.

Negotiations, according to their parent “going concern”, are characterized:

1. Negotiation techniques differ in the level of “common knowledge” with reference to the context that they presume; such as cooperative negotiations,.
2. Negotiation techniques differ in the side effects and impose on the context. The course of the negotiation in a peace negotiation or labour discussions can significantly effect the subsequent direction of the talks.
3. Negotiation episodes may be connected with another or with other actions in different ways to support the larger context (Parunak, 1999).

Coherent mechanisms dimension examines specific mechanisms; which are used by agents to ensure coherence in pursuit of objectives. Some mechanisms depend on structures in individual participants only “Solipsistic Mechanisms”, while others rely on structures shared among participants “Communal Mechanisms” (Parunak, 1999).

Moreover, Negotiation is classified according to goals’ classes. The next distinctions are specified to clear goals’ classes:

1. Static goals need simple and bounded negotiations.
2. Dynamic goals require renegotiation or meta-level negotiation.
3. An agent negotiating of future action has to address issues of time management which don’t arise in negotiations of present action.
4. Contentious goals need negotiation of mechanisms guarding against harmful threats.

Besides, Communal Mechanisms requires for agents participated in certain and common things, take the form of a common language, this language may exist at different levels of difficulty:

1. First and simplest level is a common ontology, a contract on how to divide up the world, supported by a currency with general agreed-upon valuation and a shared vocabulary.
2. Second level is meta-level which needs a contract to support some types of utterances, such as structured templates for various classes of discourse or rudimentary types of speech acts.
3. Third level is a static protocol which outlines a fixed order of utterances of different categories.
4. Forth and complex level of negotiation is occurred when agents can response to the received stream of speech acts and decide dynamically suitable type of utterances.

The main objective of negotiation is to achieve communication between agents in multi-agent project by protocols. The protocols of the community are used to assess the complexity of a negotiation mechanism (Parunak, 1999).
Several classes of the negotiation protocol are arranged in table (3) in a cline form ascending from less complexity to more complexity.

Class of Protocol	Description	Minimal Language
Reactive	Sense, then Act	Environment
Command	Master agent sends unilateral instructions to servant	Symbolic
Voting	One-shot quantitative statement of interest	Currency
Fixed Protocol	Back-and –front: symbolic and quantitative	Message and Protocol (Contract Net)
Conversation	Arbitrary interchange	Speech Acts (statement, request, commitment and command e.g. KQML)

Table 3. Negotiation protocols' classes.

The conclusions from this discussion of multi-agent negotiation are:

- Negotiation's issues provide many ways to explore in multi-agent researches.
- Negotiation processes are divided up to protocols, communication languages, standards and others (Beer et al., 1998).

2.5 Coalition in Multi-Agent

In "Cooperation structure" section, all types of the cooperation structures CATC, CCTA, CCTC and CGE include agent coalition; so the agent coalition is the most important part in multi-agent cooperation systems. Operations of coalition have to configure legacy or foreign systems (Allsoop et al., 2002).
Agent coalition is special type of agents, which concentrate on the coordination and the communication among agents to collaboratively accomplish tasks. For an example, the

power stations' extension, which costs a lot of money and time, can be solved if many power stations work together as form of agent coalition. In this example, agents in this system are owners of power stations, groups of customers and coordinators. The objective of the multi-agent system is to derive effective with gainful coalitions under the fair play practice subject to the constraints also requirements of power generation and transmission.

The modern industries needed a more efficient approach to facilitate a stable searching for new partners, coalition formation and a fair system used to identify the contribution from each participant (Yen et al., 1998). Some game theory models can be borrowed to improve the theoretical foundation for the multi-agent system.

The recent coalition operations are effected by many factors, such as data overload, starvation of information, labour-intensive information collection and coordination. The agent-based computing presents a new promising approach to effective coalition operations; since this approach embraces the coalition environment's open, heterogeneous diverse dispersed nature (Allsopp et al., 2002).

J. Yen et al. stated "The coalition formation in the multi-agent system is a hill climbing process." The payoff for each agent in the coalition formation should not be worse than the payoff of the previous method to solve same problem. However, such requirement may not be able to find the best solution for all the agents, it may arrested in local minimum (Yen et al., 1998).

Allsopp et al. stated "The coalition agents experiment (CoAX) is an international collaboration carried out under the auspices of DARPA's control of Agent-Based Systems program (CoABS)."

Some research hypotheses are suggested:

- Agents present a useful metaphor for dealing with the difficulty of real-world systems.
- An agent-based on command and control (C2) framework can support agile, robust coalition operations.
- Software agents can enable interoperability between incompatible systems.
- The CoABS Grid can rapidly integrate many different agents and systems, permitting rapid creation of virtual organizations.
- Domain policies can enforce coalition policies and compose agent relationships.
- Intelligent task and process management advance agent collaboration.
- Agent interoperability between distinct coalition controls systems can be improved by semantic web technology.

As an application on Multi-Agent coalition; CoAX team has built a software agent tested and based on the control of Agent-Based Systems program Grid. In this project, a demonstration series were being conducting within increasing difficulty in a stylized yet practical peace-enforcement scenario situated in Binni, a fictitious African state. These demonstration series use agent technologies to compose a coalition command system for intelligence gathering; visualization, campaign, battle, mission planning and execution (Allsopp et al., 2002).

Another application on Multi-Agent coalition; six-bus system of electric distribution has been used to illustrate the planning process of network expansion. The limits of power transmission and power generation are shown on the figure (2). The heuristic approach used to rank the possible locations to add new lines to an existing system. The heuristic approach is a quadratic linear programming problem, which used to identify the degree of solution feasibility (Yen et al., 1998).

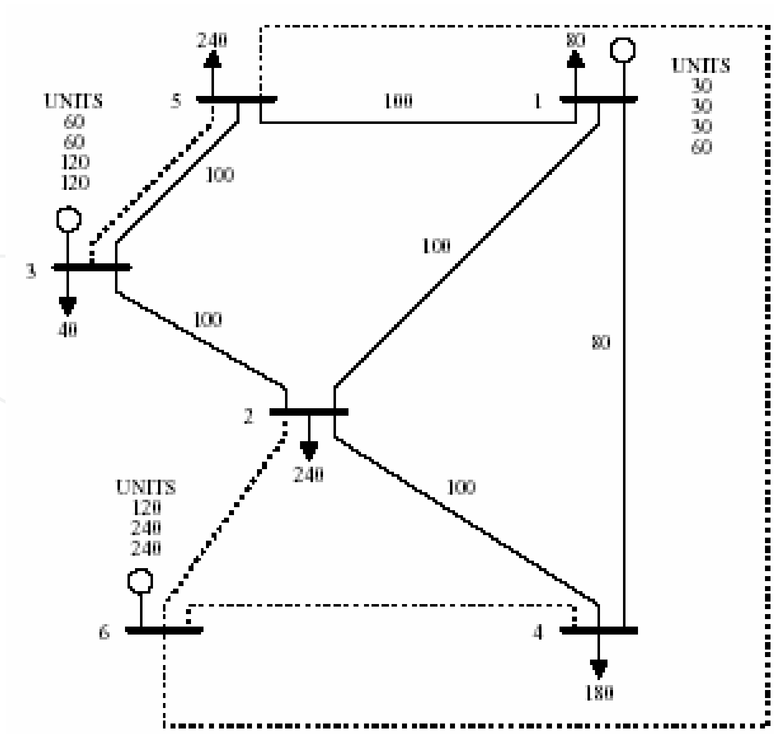


Fig. 2. Six-bus system of electric distribution (Yen et al., 1998).

The conclusions from these applications are:

1. The multi-agent is capable of making decisions for coalition formation and cost allocation, with very limited coordination and synchronization provided by the coordinator, in a fully decentralized environment.
2. Users do not need to rent dedicated lines to support the communications, because this implement is executed on the Internet.
3. Multi-agent systems can easily be applied to solve the problems where formation of coalition is essential and the environment is geographically dispersed, such as, global logistics planning or coalition formation of shipping and transportation firms.

3. Applications of Multi-Agent

There are many applications of Multi-Agent systems in all fields. The popular applications are information gathering, mobile agent cooperation and sensor information and communication. The overview of these applications will be presented in following sections.

3.1 Information Gathering

The most important issue in the applications related to information gathering is making an informed decision from a huge amount of information. One of the most recent researches in this area is a system of information gathering (Li et al., 2002).
This system consists of two agents:

1. Query processing agent.
2. Information filtering agent.

The objectives of information gathering system are:
1. Retrieving relevant information from World Wide Web (WWW).
2. Query the relevant information.
3. Filtering information from vast of information.

The structure of this system is shown in figure (3).

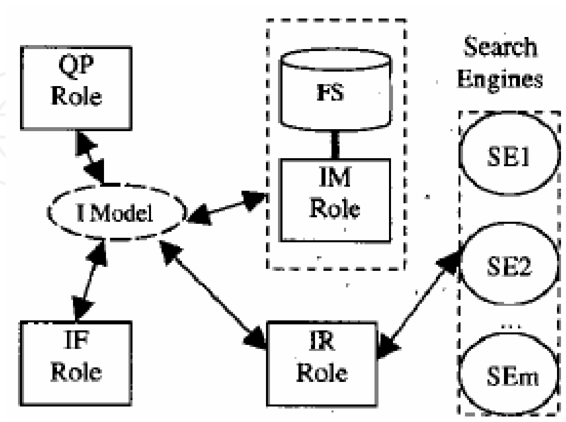


Fig. 3. Information Gathering System (Li et al., 2002).

The information collected from WWW is multi-modal, unorganized and distributed on collections overall world; therefore gathering right information is very difficult. Information of WWW is uncertain. WWW search engines provide users with overload sites or document for certain keywords or expressions. WWW services and documents are extremely changing.
These system achievements are depended on the training documents; because documents contain uncertain and unorganized information.

3.2 Mobile Agent cooperation

Flexibility and efficiency in solutions of distributed systems are the mean objectives of Multi-Mobile Agents (Multi-MA). There are many research studies are presented in this area.
A theoretic model has been suggested to solve some key problems such as fuzzy belief composing and contradiction coordinating. In addition, new ideas of Multi-Mobile Agents on fuzzy knowledge exchange and representation are presented.
Mobile node and mobile agent form this theoretic model. In addition, this model concerns on two important topics the architecture and agent's behavioral interesting aspects. The diagram of this model is shown in figure (4):

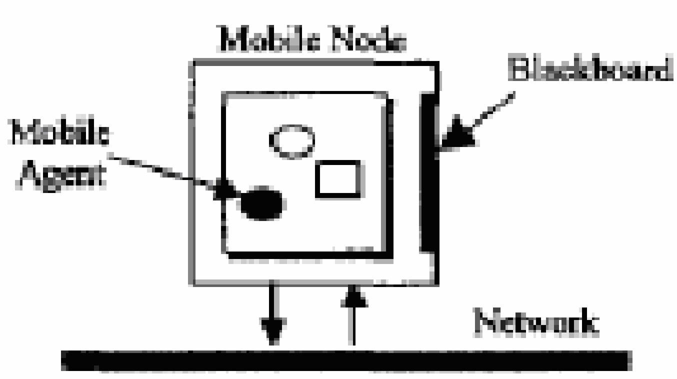


Fig. 4. The reference model of multi-MAs cooperating System (Miniy 2000).

The black board in the model allows agents' interactions; when agent passed the node the information on the blackboard is updated. This information influences on agent's behaviours in the same time. The basic elements of agent are Run-time behavioural code, roaming trial, belief (fuzzy knowledge) and other parameters.

There are two types of agent's migration behaviours:

1. Passive migration; when agent is controlled by other agents.
2. Active migration; when agent can decide by itself and start its movement according to suitable condition (Minyi et al. 2000).

Another application on Multi-Mobile Agents is the Multi-hop ad-hoc wireless Network (MANET). The MANET is defined as "a network architecture where each mobile host is treated as a router and makes a peer-to-peer communication without any base-stations" (Onishi et al., 2001).

The advantages of using mobile agents for control packets are:

1. Instillation new routing system is easy, because the process to instillation is "collect old routing agents and release new agents over the network". This easiness facilitates routing algorithms to adapt themselves to valuable cases.
2. Several services of routing are locally embedded in agents. In addition, in supported agents the heterogeneous networks can be connected together.
3. The used resources will be released after agents left.

In contract, the disadvantages of using mobile agents for control packets are:

1. Malicious hosts can easily attack agents.
2. Deploying a mobile-agent system, which effectively protects a machine against malicious agent by restricting resources, is possible.
3. Agents' codes are usually written in relatively slow interpreted languages and slightly gain weight (Onishi et al., 2001).

3.3 Sensor information and communication

The sensors are used to measure signals, temperature, pressure and other variables. Multi-agent systems have been used in this field to achieve reliable, processed, accurate and accepted measurements through merge different methods (agents) that used to measure one or more variables have common relationships.

Complex maintenance-type assignments for mobile Jet turbofan engine have been executed in coordinated style through cooperative of simple autonomous mobile sensor platforms. These platforms have been developed from a general archetype. The mains platforms of this system are:

1. Original platform collects decision, controls and returns sound, video and other measured variables.
2. Network platform connects all platforms and provides a communication between all platforms.

This system is based on swarm algorithms; which provide architecture scalability, units' flexibility and robustness through redundancy and simplicity. Tethers and necessity for an efficient representation of engine surface are essential keys for this model; since tethers create restriction on agents' movement and need a path planning, and the necessity for an efficient representation of engine is due to limited memory capacity of agents (Litt et al., 2003).

A multi-agent real-time simulation framework allows high-fidelity virtual sensor models, which incorporate in hardware-in-the-loop (HIL) experiments. Multi-agent real-time framework results are:

1. Full control of the environment.
2. Reproducibility.
3. Easy merge of real/virtual components in model experiment.

Multi-agent real-time framework has been illustrated through a laser range finder sensor and a pre-crash control solution (Papp et al., 2003)

4. Pursuit Problem Demonstration

In this section the pursuit problem will be used to demonstrate the working of multi agent systems as multi agent robots.
Pursuit problem is one of the punch problems in robotics fields as it deals with dynamic targets going in unknown directions. This section takes the notion of pursuit and implements different algorithms for acquiring a dynamic target in unknown environment, the algorithms test without using any localization devices such as the GPS devices or laser reflectors (Beacons). The agents are work in the same environment, negotiate together to decide on the way the target is acquired. The robots will roam in the environment and preserve their formations until the dynamic target is spotted. When the target is spotted the robots will decide who will acquire the target and will use its algorithm to acquire the target.
Robots can be used to carry out many tasks with a wide range of complexity: from simple tasks that can be performed by one agent, to other complicated tasks that need a group of robots collaborating together to reach their goal. The multiagents implemention in the domain of Pursuit problem is concerned with designing and implementing algorithms that allow robots to cooperate together to catch a hider robot. Negotiations and other anti flocking algorithms will take place to make the searching algorithms of different groups more effective.
Since our work considering using behaviors based robotics, robots control can benefit from animal behavior studies. Robotics can draw from these behaviors models to make similar forms of behavior in machines. To Study behavior-based robotics it is essential to start with the biological behavior of animals in terms of ethology, and learn the relationship between animal behavior and multiagents behaviors. The study of ethology is very important for multiagents robotic systems in terms of display behavior (which involves the information signaling by changes in activity). These displays are usually generated by fixed action patterns and may be electrical, visible or audible. The displays include color changing in fish, birdsong, leg waiving in spiders, etc. These displays may benefit some activities such as escaping a predator. [Arkin, 1998]. The "ecological niche" is one of the most important concepts taken from the study of ethology that benefit behavior-based robotics. The ecological niche is the status of an animal in its own community in terms of food and surviving. A successful robotic system is a system that is autonomous and can compete with its environmental surroundings, that's why the ecological niche is very important to robotics. A system that does not find a stable niche will be unsuccessful. Robots must compete with their natural world with static or dynamic objects and the niche allows them to survive with their competitors. When an effective robotic system is to be designed it must be targeted towards some niche. [Arkin, 1998]

4.1 Pursuit in robotics

The contest of pursuit is among the most widespread, challenging, and important optimization problems that face mobile agents, and represents some of the most important

potential applications for robots. In a typical contest of this sort, a predator chases a prey animal around until the prey is captured.
Pursuit contests are usually difficult to handle, because they deal with dynamic targets. Agents that pursue must maintain the sensory information of both the physical environment and the hostile opponent. An effective pursuit may often require prediction and "mind-reading". With this problem, recognizing other robots is very important when designing multiagent systems. A robot has to distinguish other robots from many environmental features and also when working with teams recognizing a hider robot from a seeker robot is very important as well. [Arkin, 1998]
The most commonly used techniques adapted to deal with pursuits are:

1. *Classical calculus of Variations and Optimal Control Technique*: This technique provides a strong tool of analysis and design. This technique has the advantage of giving a real time solution, whenever it exists, since the system and the constraints are represented by a set of differential equations. However this technique has not been widely used in pursuits because it gets complicated with the increase in the number of players.
2. *Dynamic Programming*: A very efficient technique that mainly deals with discrete systems with a value function that needs to be optimized.
3. *Reinforced Machine Learning*: Reinforcement learning is a technique of learning how to map situations to actions.

The learner is not told which actions to take, instead he must decide on the rewarding actions by trying them. Actions may not affect only the immediate reward, but also the next state and, through that, all subsequent rewards. The two characteristics of trial and error search and delayed reward, are the most important features of reinforcement learning. The basic idea is simply to capture the most important aspects of the real problem facing a learning agent interacting with its environment to achieve a goal. This goal is related to the state of the environment. Three aspects are included: sensation, action, and goal.
Many researchers have tackle the pursuit problem, Cassimatis, Trafton, Schultz, Bugajska and Adams from the Naval Research Laboratory in Washington[Staugaard, 2002], take the game hide and seek as an example in their research about cognitive modelling techniques, and address some issues that have to be taken in consideration when developing pursuit algorithms. To find their targets they have to be able to identify other objects and agents, to plan their path and move towards their target trying to avoid the obstacles in their way. In other words any information agents gather perceptually will be valuable when seeking and navigating, and the more efficiently they navigate the better they will be at the game. [Staugaard, 2002]
They give an illustrated example of temporal reasoning about the robot rolls behind an occluding screen and then after that a robot that looks the same comes out. Then someone puts a barrier behind the screen. [Staugaard, 2002]
Many issues from the logical study of intelligence arise here: what can be assumed and why; how can an assumption be falsified and when is there more than one explanation for an event, and how do we choose between them. The authors argue these issues of reasoning have to be taken in consideration whenever we try to build an effective hide-and-seek or a pursuit system. [Staugaard, 2002]
The Authors argue that any system uses hide and seek has to revise provisional beliefs and inferences that followed from it. So seekers must engage in probabilistic interferences in order to keep track of the hiders efficiently. [Staugaard, 2002]

Other work has done with one robot that seeking robot and there are several hiding robots in an arena. The seeking robot will initially wait a certain time until the hiding robots find their places to hide on the arena (the arena contains walls and obstacles that make it difficult for the seeker robot to find the other robots). These hiding robots will have the capability to move as soon as the seeking robot detects them. [Kranz, et al 2006]. The difference in this game is that when the robot detects other robots it begins shooting them with laser not actually chasing the robots. The hiding robots will try to move away to prevent themselves from being hit by the laser. If the hiding robot finds the seeking robot before the seeker finds it, it can attempt to evade the seeker.

The other hide and seek algorithm. A human hides a vehicle anywhere in the hiding area. The seeker will search the hiding area for the opponent and attempt to touch it. A human will try to drive the vehicle away once it is spotted by the seeker. If the robot looses the vehicle during the chase, he should look for it behind nearby objects to see if it is hiding there. The game will be over when the robot touches its opponent [Coulter, 1992]

One of the pursuit algorithms is a tracking algorithm that is based on calculating the curvature that will move a robot from its position to another goal position. First a goal position has to be chosen some distance ahead of the robot on the path. This algorithm is derived from the way humans drive. A human tend to look some distance in front of the car and drive towards that spot. The look ahead distance changes as a human being drive to reflect the twist of the windings of the road. [Staugaard, 2002]

4.2 Formation for pursuit problem

Formation is very important in the field of multi-agents robotics, in order for the robots to roam together to complete a certain tasks they need to form a chain and maintain this formation. This research only concentrates on how to achieve formation coordination without providing the robots with global knowledge of other robots' positions or headings.

Formations can be done in different strategies ranging from simple, behaviors-based, ones to more involved ones relying, on global knowledge of the environment, typically a global coordinate system or maybe knowledge of other robots' positions in the environment and their headings. The first category is characterized by robustness but there is a lack of guarantees that the wanted formation will actually emerge; the second category is characterized by reliability and efficiency but there is a need for the global knowledge of the environment and sometimes complex computation. [Riley et al. 2005]. In [Watson, 1925] three principles of formation control are identified and categorized depend on references.

In [Riley et al. 2005] an algorithm was presented using IDs assigned to the robots, in this algorithm, the robots can change in group size and also switch between formations and avoid obstacles.

Analyzing of the different types of geometric formations that multi-agents can do and they also agree that a geometric formation consists of three main parts; Conductor: which is the robot at the head of a group in a formation, Leader: the leader is the agent who guides the one that follows it, and Follower: All the agents in a formation are followers except the conductor [Fredslund & Mataric´, 2002] and [Balch & Arkin, 1998].

The main objective of this paper is to design, implement and test algorithms that allow many agents to cooperate together in order to catch a hider using agent of sub-team. The sub-teams are multi-agents that communicate and collaborate together in order to reach their goal. Each sub-team of robots has its own pursuit algorithm that is behavior-based and uses some mathematical manipulations to make it efficient. The robots also make use of certain formations to accomplish their task.

4.3 Algorithms-behaviors design

Our systems are dealing with four sub-teams of agents that have their own way to chase the hiding agent. These sub-teams will collaborate together to make their search efficient. Our work with assumptions that the Environment is unknown and none of the robots have any previous information about the environment, No localization devices are used, agents are identical, and the work is only concerned about pursuit algorithms, no algorithms were implemented to the hiding agent.

All the agents have obstacles avoiding behaviors which make each agent capable to detect the obstacles using the sonar. This algorithm will be executed whenever an obstacle is detected with less than 0.4 meters in front of the robots. The sonar beams is covered area by the ranges up to 6 m.

4.3.1 Attacking the target methods

These behaviors and it own algorithms have been design to give the agents capability to deal with target.

4.3.2 Roams the environment

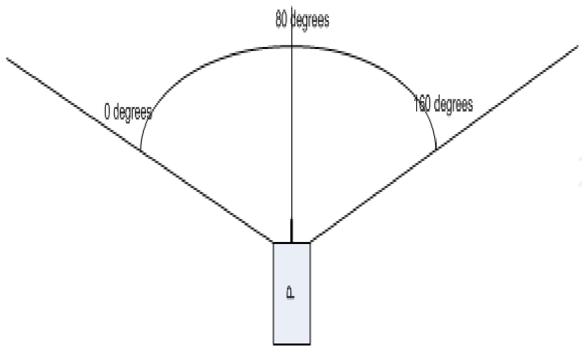
This algorithm is done with one agent that roams the environment and avoiding obstacle. When it sees the hiding agent it starts attacking it. The hiding agent is recognized by its color. This algorithm is derived from the way humans think about attending a desired objective. A human tend to look at the spot where he wants to go and then walk straight towards that spot if there are no obstacles in the way.

Whenever the pursuer in this algorithm sees an agent with the green color it will know that this is the opponent and it will start attacking. First it will speed up from the normal speed to the attacking speed (from 0.4 meters/s to 0.8 meters/s) and it uses the information from the camera to adjust the heading to the target by calculating angle difference between the target and the heading of the robot The information about the angle returned from the camera is between 0 and 160 degrees as illustrated in the figure (5) (the actual heading of the robot is at 80 degrees). At turning speed and the linear speed of 0.8 meters per second the agent will keep on chasing the target until it is caught.

4.3.4 Heading with considering the speed of the hider

In this algorithm the pursuer tries to detect the speed and the heading of the hider relative to its own position and heading without using any localization device. After it detects the speed and the heading of the hider it tries to predict where it will be after a certain time and then it will go straight to that meeting point to catch the hiding agent.

In order to calculate the heading and the speed of the opponent, the pursuer will have to take the distance and the angle between him and the agent targeted. Then the pursuer will have to stop for exactly one second before taking another reading of the distance and the angle between its heading and the target. This can happen in two situations, the first is when the hider heading away from the pursuer and the second is when the hider is heading towards the pursuer. These two situations can also happen in two different ways: when the hider is heading to the left of the pursuer and when the hider is heading to the right of the pursuer. The first situation are illustrated in the figure (6).



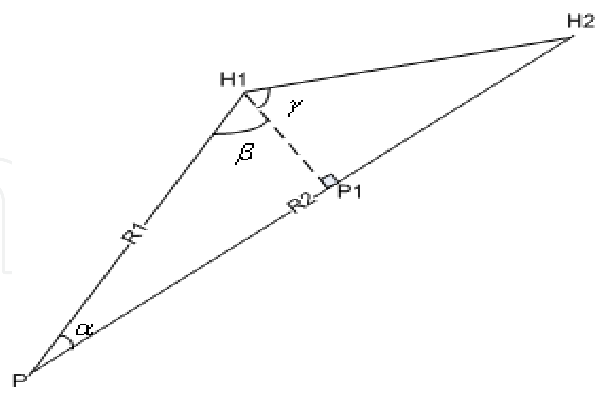


Fig. 5. Camera heading robot-target angle. Fig. 6. Speed detection.

In figure (6) we have:

P: position of the pursuer relative to the other agent.	R_1: the range between the pursuer and the hider before the 1 second interval.
R_2: the range between the pursuer and the hider after the 1 second interval.	H_1: Position of the other agent as soon as it is spotted by the pursuer (H stands for hider).
H_2: position of the other agent after 1 second from when it was first sighted. (Note that the pursuer during the 1 second does not move)	

Let X1 be the angle returned by the camera just before the 1 second stop and X2 be the same angle just after the 1 second stop
Let P1 be the projection of H1 on R2.
The angle alpha in the figure above is the absolute value of $X_1 - X_2$:

$\alpha = \mid X_1 - X_2 \mid$.	$PP_1 = R_1 \times \cos\alpha$.	$H_1 P_1 = R_1 \times \sin\alpha$.
$H_2P_1 = R_2 - PP_1$.	$H_1H_2 = \sqrt{H_1P_1^2 + H_1P_1^2}$	

Having the length of H_1H_2 which is the length of the track of the hider calculated over 1 second we can conclude that the other robot has a speed of H_1H_2 meters/s. We also know that the hiding agent will go 3 x H_1H_2 meters in 3 second, or 5 x H_1H_2 in 5 seconds. Assuming that the hider has a constant speed and it only goes straight until it faces an obstacle the pursuer is now able to predict how far the hider will go in a certain amount of time. Now to meet the hider at the defined point (on figure (7) is H3) after 3 or 5 seconds the pursuer will still have to know the angle that it has to turn from its original position as well as the distance to that meeting point. This is illustrated in figure (7) which is an extension to figure (6) with the addition of H_2H_3 which is the calculated distance that the robot will go in 3 seconds after the 1 second stop interval.
The purpose of this diagram is to calculate PH_3 which is the distance that the pursuer will go to get to the meeting point with the hider. It will also calculate the angle $\angle H_1PH_3$ which will help determining the turning angle of the pursuer however this angle is not the angle that the pursuer has to turn in order to get to his destination, because with absence of a localization device the original heading of the pursuer is unknown.

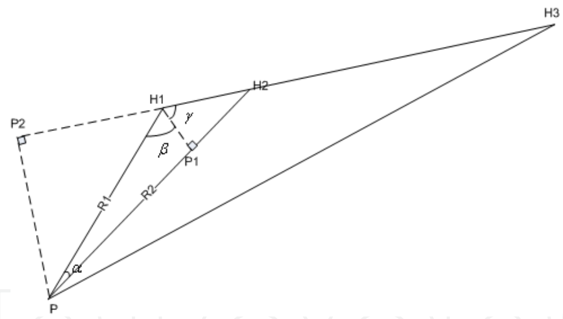
Let P_2 be the projection of P on H_1H_3. From this graph we can conclude

$\gamma = \tan^{-1} H_2P_1/H_1P_1$.	$\beta = \tan^{-1} PP_1/H_1P_1$.
$P_2P = \sin(P_2H_1P) \times R_1$	$P_2H_1 = \cos(P_2H_1P) \times R_1$
$H_1H_3 = 4 \times H_1H_2$ ((1second + 3seconds) x H_1H_2 meters/second).	$P_2H_3 = P_2H_1 + H_1H_3 = \cos(P_2H_1P) \times R_1 + 4 \times H_1H_2$.
Angle $\llcorner P_2H_1P = 180 - (\gamma + \beta)$.	Angle $H_1PP_2 = \sin^{-1} (P_2H_1/PH_1)$
Angle $H_3PP_2 = \tan^{-1} (P_2H_3/P_2P)$	
Angle $H_3PH_1 = H_3PP_2 - H_1PP_2$	$H_3P = P_2H_3 / \sin H_3SP_2$

Having determined H_3P (which is the distance that the robot have to go to get to its objective after 3 seconds we can calculate the linear speed of the pursuer that will make him get there after 3 seconds: speed = $H_3P/3$ m/s.
As mentioned above the camera gives the angle of the line of sight of the hider, this angle is between 0 and 160 degrees with 80 being the heading of the pursuer
With the absence of a localization device the turning angle of the robot has to be calculated relatively to its original heading when it stopped and waited for 1 second. This can be done with the use of the value of the angle H_3PH_1 **c**alculated earlier and the value of X1 and X2 (with X1 being the angle returned by the camera just before the 1 second stop and X2 be the same angle just after the 1 second stop).
To calculate the angle that the robot has to turn.

In the case where the hider is going right (that is X2 > X1):	In the case where the hider is going left (that is X2 < X1):
If X1 < 80 then the hider's initial position before the 1 second stop is (80 – X1) degrees to the left of the pursuer. In this case the turning angle would be 80 –(H_3PH_1 + X1).	If X1 < 80 then the hider's initial position before the 1 second stop is (80 – X1) degrees to the left of the pursuer. In this case the turning angle would be (80 –X1) + H_3PH_1 .
If X1 > 80 then the hider's initial position before the 1 second stop is (X1 - 80) degrees to the right of the pursuer. In this case the turning angle would be – ((X1-80) + H_3PH_1) (note that the – is for the anti-clockwise turning)	If X1 > 80 then the hider's initial position before the 1 second stop is (X1 - 80) degrees to the right of the pursuer. In this case the turning angle would be X1 – 80 - H_3PH_1

To get the turning rate this angle will be divided by 3 seconds.
The second situation is when the hider is heading towards the pursuer: With this situations the graphs and the calculations are different , however the purpose of these graphs is the same as in the previous situation, which is to find the distance that the pursuer have to go to meet the hider on his trajectory, the linear speed and the turning speed. The calculations for the second situation are illustrated in the figure (8)



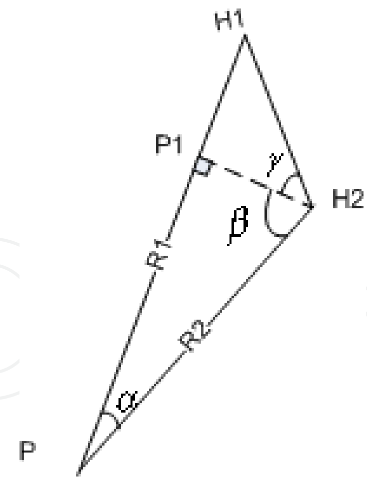


Fig. 7. Linear speed and turning speed calculations. Fig. 8. Speed detection 2.

Same concept as before, that the hiding agent will go 3 x H_1H_2 meters in 3 second, or 5 x H_1H_2 in 5 seconds.

4.4 Cooperate agents and formation

In this algorithm a group of two robots will cooperate together in order to catch the hider. As soon as the robots see each other they will roam the environment together forming a linear formation: a leader and a follower. Then when the hider is sighted the formation will break and the robots will attack the hider one from the left and one from the right. The main benefits of this algorithm is that the attacking robots do not have to worry about the direction of the hider since they attack from both sides.

The formation coordination between the two robots is achieved without providing the robots with global knowledge of other robots' positions or headings. The leader will not communicate to the follower and the follower is the only one responsible of maintaining the formation.

There are two phases in the formation, first starting the formation and then maintaining the formation. The following is done by recognizing the leader’s color and then keeping the right angle with the leader. The follower keeps in line with the leader he by adjusting its heading to match the leader’s heading. At the same time the follower is responsible of keeping a certain distance to the leader and changing its speed accordingly as in figure (9). The formation algorithm is done in a way that makes it very easy for the group to avoid obstacles together. Once the target is sighted after the formation is done the leader will follow the target straight until it is 5 meters from it, then it will speed up to 0.800 meters/second and it will check if the follower is aligned with it (within 10 degrees difference) then it will attack from the right hand side (this is done with the use of the information about the hider’s line of sight’s angle returned by the camera). If the follower is not aligned with the leader (this can happen if the leader has recently faced an obstacle and the follower is still recovering from that) then the leader will wait until the follower fixes its heading before attacking. Then the leader will attack from the right by keeping an angle of 40 degrees to the leader, it will also signal the follower to attack as well. At this time the follower will attack from the left by keeping an angle of 40 degrees from his target to the left. When any of the attacker is at a distance of less than 1.5 meters to the hider it will fix its angle and attack straight until the hider is caught. This is illustrated in figure (10).

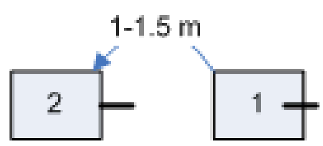


Fig. 9. Following algorithm.

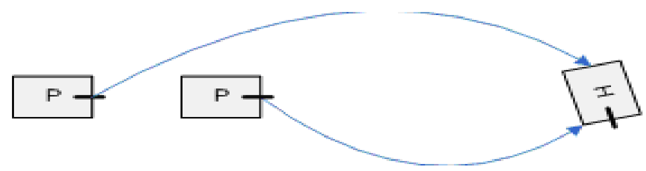


Fig. 10. Formation of reaching the target.

The other type of formation are three robots forming a linear formation. As soon as the target is sighted, the team will chase the hider and then change the formation to a circle formation and surround the target inside the circle.
In this algorithm we have three robots doing the linear formation as shown in figure (11). In this formation the conductor, number 1, which is at the same time a leader to agent number 2. Agent number 2 is a follower to number and a leader to number 3. The conductor is responsible of leading the group and the other two will follow in the same way described in the previous algorithm.
The conductor will decide when to switch from the linear formation to the circle formation. When this is done the robots will have to follow the conductor respectively by maintaining the angle, the speed and the distances calculated depending on figure (12)

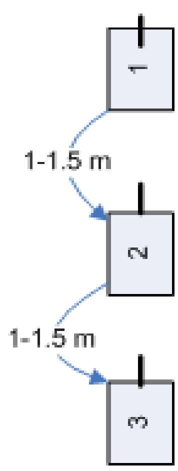


Fig. 11. Linear formation.

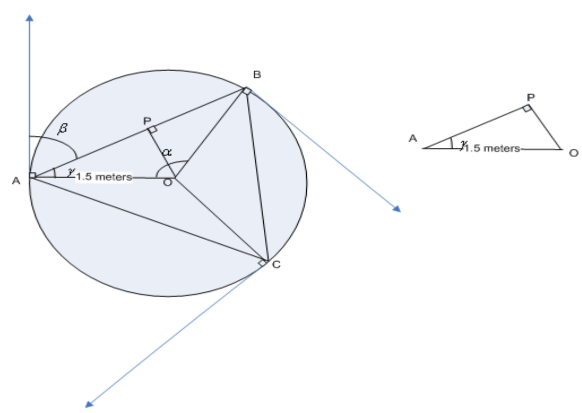


Fig. 12. Circle formation.

4.5 Anti-Flocking

The anti-flocking algorithm used in this work is based on the color detection. Every subteam in this design has its distinctive color, so basically every time the agent in the first and the second attacking methods sees a color other than its color and the target's color a random number between -20 and 20 will be generated and the agent will make this number as its turning speed until the other team or robot is out of sight. This concept is also the same for the leaders in the formations in attacking. In this way the robots will try to stay out of sight of each other and thus searching different areas in the environment. The Anti flocking behavior will only be effective whenever the target is out of sight of the team that performs the anti-flocking behavior. However this antiflocking algorithm has other benefits as well: It prevents the teams from coming in the way of each other; it also prevents blocking the view and preventing other teams from seeing the target as well as helps clearing the way of a team attacking the target.

4.6 Negotiations

The negotiations will take place to decide who will chase the target as only one team can chase the target at one time.
If a team (e.g. Team1) sees the target for the first time it will signal the other teams that the target is being chased and it will also keep on sending its distance to target 10 times a second. Then if another team (e.g. Team2) sees the target while being chased it checks its own distance to target. If that distance is less than the distance of team1 to the target it will signal the other teams that the chase is taken over by team2 this way team1 will know that the target is closer to another team and it will stop chasing leaving the chase to the other team. So there are two messages to be sent between the subteams, the first is the distance to target when target is sighted, and the second is the signal to tell the other teams that the chase is done by this team.

4.7 Implementation and simulation results

This paper presents the acquiring of a dynamic target by multi-agents collaborating. The environment is considered to be unknown; no localization system is used. The heading and the speed of the target are considered unknown. Other problems are the environment is quite small and have obstacles, the robots will have to search different parts of the environment and avoid flocking together, and the communication between the robots should be minimized. The simulation of the design is done using a simulation Stage and player program and using Pioneer Robots. The final implementation including 8 agents.
Most of the simulation results show the screen capture of two windows Stage/ player and G2 taken at the same time when the robots are running inside the environment. This way the full track of the robots is shown and the robot's location at the time when the capture is taken can be seen as well.

4.7.1 Exploration and obstacle avoidance

The first feature tested in the system is the basic roaming of the robots inside the environment, and the obstacle avoidance whenever an obstacle comes up. To test both static and dynamic obstacle avoidance, two robots were used and located inside facing each other figure (13) shows the robots just after meet each other:

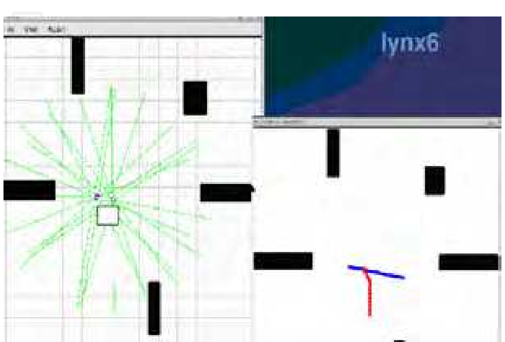


Fig. 13. Obstacle avoidance Phase 1.

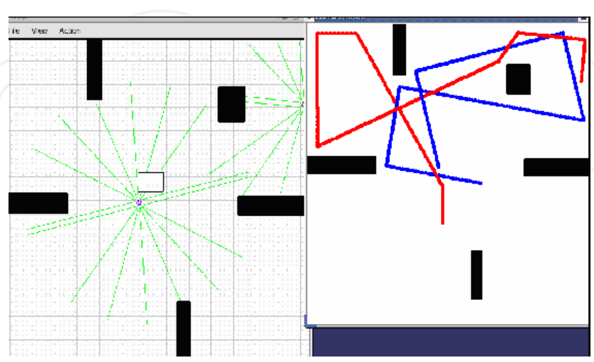


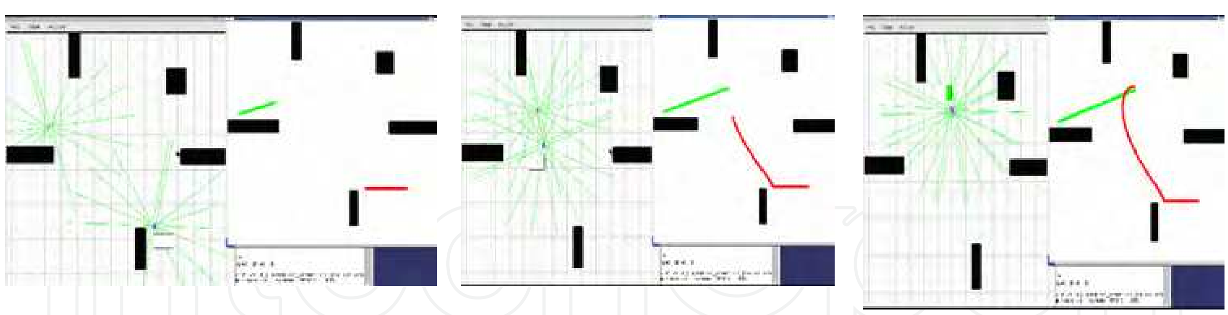
Fig. 14. Obstacle avoidance Phase 2.

It can be seen how the robot with the red path avoided the other robot when it came within it sonar range of avoidance (0.4 meters). The robots run further in figure (14) : and how they avoid different static obstacles.

The degree of the robot to turn left or right depending on the range calculations explained in section, then as soon as the obstacle is avoided the robot goes back to the normal exploration of the environment. The problem of the robot trapped in corner problems or the obstacles interact to block the robot had been eliminated, as shown in the in figure (14) by introducing a second turn of 30 degrees/second as soon as the obstacle is detected without checking the ranges within that second. So the robot turns for about 1 second then it checks on the obstacle to see if it s still there. As it can be seen from these simulation captures, the behaviors of obstacle avoidance works well in this environment and the robots are avoiding the dynamic and static obstacles successfully.

4.7.2 Attacking Behaviours

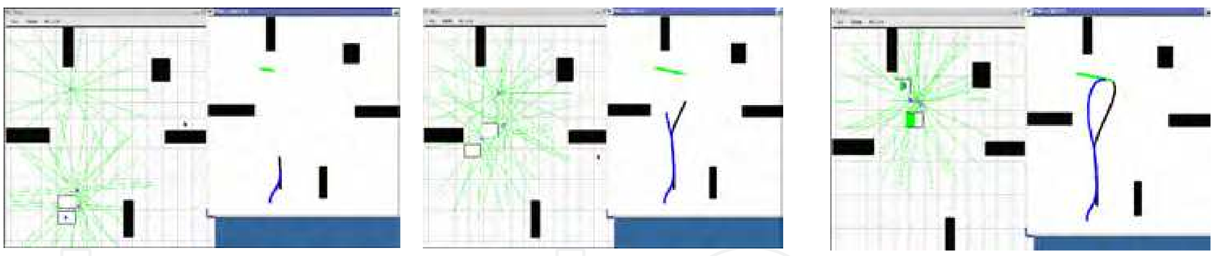
The test of the attacking behaviors, quit few simulation test have done, one is when the target is sighted the agent, it will raise its speed and follow the target straight as shown in figure (15) , the target starting from the left side and attacker starting from the right side of the screen and heading towards an obstacle. As it tries to avoid the obstacle to the right it sees the target then it raises the speed and it starts heading towards the target. In the a of the figure (15) the red line and the green line being about the same length, however in phase two it can be seen that the red line is longer than the green line even though they started at the same time. This is due to the increase in speed of the attacker after it spots the target. The curve in the red line of figure (15) illustrates the path of the attacker when adjusting its heading towards the target. In other attacking scenario, the features of formation starting, formation maintaining, and attacking were all tested. In this algorithm the team of two robots start the linear formation as soon as they spot each other, it starts the formation by adjust its speed accordingly until the formation is achieved and then roam the environment maintaining the formation when avoiding obstacles



a b c

Fig. 15. Attacking behavior for one agent(robot).

As soon as the group spots the target it is advance towards until the leader is 5 meters away and signal the follower to break the formation, both robots attack the target one from the left and the other from the right as illustrate in figure (15).
The robots are placed initially in the environment where the one in green is the target and it starts from the top. The two attackers with the blue and black colors start from the bottom of the screen initially 1 meter apart so the formation can be seen before the attack starts.



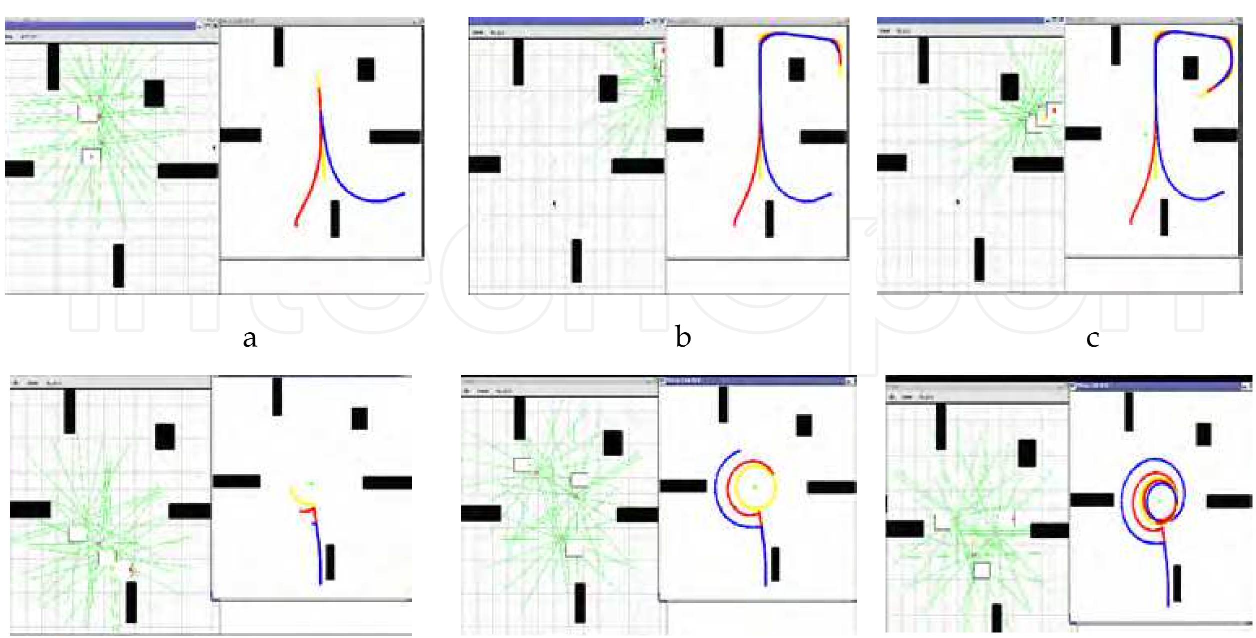
a b c

Fig. 16. Attacking behavior for one agent(robot) with formation.

In a of the figure (16), the robot is black starts the formation by following the robot in blue and then the formation is kept. At the same time the robot in blue spots the target and starts heading towards that target, while the robot in black maintains the formation. when the leader gets to 5 meters from the green target that is heading towards the bottom, It signals the other robot to break the formation and at the same time it starts attacking from the right, while the follower starts attacking from the left as shown in b of the figure (16), and the formation is broken. The final phase of the attack is illustrated in c after the formation is broken the robots increase their speed and attack from both sides until they are 1.5 meters away from the target, the robots attack straight and this way the target is acquired from both sides.

The other test have done for a team of three robots start the linear formation as soon as they spot each other, and then roam the environment maintaining the formation when avoiding obstacles.

As soon as the group spots the target it is supposed to advance towards until the leader is 1.5 meters away and then the leader should signal the other 2 followers to break the initial formation and start another formation that surrounds the target, then the three robots will circle the target and trap him inside the circle by maintaining the exact distances between each other and changing their speed accordingly as figure (17) illustrated.



a b c

d e f

Fig. 17. Demonstration of pursuit problem.

The three robots with the current speeds in the circle formation can lock up a dynamic target of up to a speed of 0.300 meters/second. With that speed whenever the target tries to get out of the circle, it is blocked by one of the attackers and it has to avoid them and go back inside the circle. At worst case if the target's speed is slightly more than 0.300 the target is caught by one of the attackers doing the circle formation, but it can never escape.

5. References

Allsopp, D. N.; Beautement, P.; Kirton, M.; Bradshaw, J.M.; Suri, N.; Durfee, E. H.; Knoblock, C. A.; Tate, A. & Thompson, C. W. (2002). Coalition Agents Experiment: Multiagent Cooperation in International Coalitions. *IEEE Intelligent Systems,* Vol., 17, (May/Jun 2002) page numbers (26 – 35)

Arenas, A. E. & Sanabria, G. B.(2003), Modelling Intelligent Agents for Organisational Memories, In: *Lecture Notes in Computer Science*, Vol., 2773, page numbers (430 – 437), Springer, 0302-9743, Berlin, Germany.

Arkin, R. C.(1998). *Behavior-based robotics,* MIT Press, 0262011654, London, England Chapters 1,2,3 and 9.

Bainbridge, L. (1993). The change in concepts needed to account for human behavior in complex dynamic tasks, *Proceedings of IEEE Systems Man and Cybernetics Conference*, Vol. 1, pp. 126 - 131, Westminster hotel, October 1993, IEEE, Le Touquet – France.

Balch, T. & Arkin, R. C. (1998). Behavior-based Formation Control for Multi-robot Teams. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, (December 1998) pp. (926-939), 1042-296

Beer, M.; D'Inverno, M.; Luck, M.; Jennings, N.; Preist, C. & Schroeder, M. (1999). Negotiation in Multi-Agent Systems, *The Knowledge Engineering Review*, Vol. 14, Issue 3, (Sep 1999) page numbers (285-289), ISSN: 0269-8889.

Cassimatis, N. L.; J. Trafton, G.; Schultz, A. C.; Bugajska, M. D. & Adams W. (2002), A Task Domain for Combining and Evaluating Robotics and Cognitive Modeling Techniques, Measuring the Performance and Intelligence of Systems: Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop, Washington DC, 2002

Changhong, L.; Minqiang, L.; & Jisong, K.; (2002). Cooperation Structure of Multi-agent and Algorithms, *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems*, pp. 303–307, ISBN:0-7695-1733-1, September 2002, IEEE, Divnomorskoe, Russia.

Coulter, R. C., (1992). *Implementation of the pure pursuit path tracking algorithm,* The Robotics Institute Camegie Mellon University Pittsburgh, Pennsylvania.

D'Inverno, M.; Luck, M. & Wooldridge, M. (1997). Cooperation Structures, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 600-605, August 1997, Nagoya, Japan.

Floreano, D.; Nolfi, S. & Mondada, F. (1998). Competitive co-evolutionary robotics: from theory to practice, *Proceedings of the Fifth International Conference on Simulation of Adaptive Behaviour on From animals to animates 5*, pp. 515 – 524, ISBN:0-262-66144-6, University of Zurich, Zurich, Switzerland, 1998, Mit Press, Cambridge.

Fredslund, J. & Mataric´, M. J. (2002). A General, Local Algorithm for Robot Formations. *IEEE Transactions on Robotics and Automation,* Vol. 18, No. 5 , Oct 2002. pp. (837-846), 1042-296

Gustasfon, D. A.; & Matson, E. (2003). Taxonomy of Cooperative Robotic Systems, *Proceeding of 2003 IEEE International Conference on Systems, Man and Cybernetics*, pp.1141-1146, ISBN: 0-7803-7952-7, Crystal City Hyatt Regency, October 2003, IEEE, Washington DC.

Jones, P. M.; & Mitchell, C. M. (1991). Human-machine cooperative interaction in the supervisory control of a complex dynamic system, *Proceedings of the 1991 IEEE International Conference on Systems, Man and Cybernetics,* Vol. 2, pp.1301-1306, Omni Charlottesville hotel and the University of Virginia, Charlottesville, Virginia, October 1991, IEEE.

Jones, P. M.; & Mitchell, C. M. (1995). Human-computer cooperative problem solving: Theory, design, and evaluation of an intelligent associate system for supervisory control, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No.7, (May 1995) page numbers (1039–1053), ISSN: 1094-6977.

Jones, P.M.; & Jacobs, J. L. (2000). Cooperative Problem Solving in Human–Machine Systems: Theory, Models and Intelligent Associate Systems, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 30, No.4, (November 2000) page numbers (397–407), ISSN: 1094-6977.

Kranz, M.; Rusu, R. B.; Maldonado, A.; Beetz, M. & Schmidt A. (2006). A Player/Stage System for Context-Aware Intelligent Environments. To appear in *Proceedings of the System Support for Ubiquitous Computing Workshop (UbiSys 2006)*, at the 8th Annual Conference on Ubiquitous Computing (Ubicomp 2006), Orange County, California, September 2006.

Li, C. S.; Zhang, C. & Zhang, Z. L. (2002). An Agent-Based Intelligent System for Information Gathering from World Wide Web Environment, *Proceedings of the 2002 International Conference on Machine Learning and Cybernetics,* Volume 4, pp.1852–1857, ISBN: 0-7803-7508-4, Prime Hotel, Beijing, China, November 2002, IEEE.

Litt, J.; Wong, E.; Krasowski M. & Greer L., (2003). Cooperative Multi-Agent Mobile Sensor Platforms for Jet Engine Inspection--- Consept and Implementation, *Proceedings of the* International Conference on Integration of Knowledge Intensive Multi-Agent Systems. KIMAS'03: Modeling, Exploration, and Engineering, October 2003 pp. 716-721, ISBN: 0-7803-7958-6, Royal Sonesta Hotel, Cambridge, MA, USA. Minyi,

Shaowen, W.; Y. & Mingtain, Z. (2000). Cooperative Mobile Agents in Dynamic Network Environment, *Proceedings of the 36th International Conference on Technology of Object-Oriented Languages and Systems*, pp.162-167, China, November 2000, IEEE

Mitchell, C. M. (1999). Model-based design of human interaction with complex systems In: *Handbook of Systems Engineering and Management*, Sage, A. P. & Rouse, W. B., page numbers (745–810), John Wiley & Sons, ISBN: 0-471-15405-9, Canada.

Nitschke, G. (2003). Co-evolution of cooperation in a Pursuit Evasion Game, *Proceedings of the IEEE/RSJ International Conference on intelligent Robots and Systems*, Volume 2, pp. 2037-2042, ISBN 0-7803-7860-1, Las Vegas, Nevada, October 2003, IEEE.

Onishi, R.; Yamaguchi, S.; Morino, H.; Aida, H. & Saito, T. (2001). The Multi-agent System for Dynamic Network Routing, *Proceedings of the 5th International Symposium on Autonomous Decentralized Systems,* pp. 375-382, ISBN 0-7695-1065-5, Dallas, Texas, March 2001, IEEE.

Papp, Z.; Labibes, K.; Thean, A. H. C. & Eikm, G. V. (2003). Multi-Agent Based HIL Simulator with High Fidelity Virtual Sensors, *Proceedings of the IEEE IV2003 Intelligent Vehicles Symposium,* pp. 213 – 218, ISBN: 0-7803-7848-2, Hilton hotel Colummbus, Ohio, USA, June 2003, IEEE.

Parunak, H. V. D. (1999). Characterizing Multi-Agent Negotiation, *Proceeding of the Workshop on Multi-agent Systems, Negotiation Science and Technology: Opportunities and pitfalls,* Centre for Electronic Commerce, ERIM.

Staugaard, A. C. (2002) *Structured Object-Oriented Problem Solving Using C++*, Prentice-Hall, 0130284513, New Jersey.

Watson, J.B. (1924/1925). *Behaviorism*. New York: People's Institute Publishing Company; and Watson, J.B. & McDougall, W. (1928). *The Battle of Behaviorism.* London: Kegan Paul, Trench, Trubner & Co.

Yen, J.; Yan, Y. H.; Wang, B. J. & Sin, P. K. (1998). Multi-agent Coalition Formation in Power Transmission Planning, Proceedings of the Thirty-First Hawaii International Conference on System Sciences, Volume 4, pp. 433 – 443, ISBN: 0-8186-8255-8, The University of Hawaii, January 1998, IEEE.

Mobile Robotics, Moving Intelligence
Edited by Jonas Buchli

ISBN 3-86611-284-X
Hard cover, 586 pages
Publisher Pro Literatur Verlag, Germany / ARS, Austria
Published online 01, December, 2006
Published in print edition December, 2006

This book covers many aspects of the exciting research in mobile robotics. It deals with different aspects of the control problem, especially also under uncertainty and faults. Mechanical design issues are discussed along with new sensor and actuator concepts. Games like soccer are a good example which comprise many of the aforementioned challenges in a single comprehensive and in the same time entertaining framework. Thus, the book comprises contributions dealing with aspects of the Robotcup competition. The reader will get a feel how the problems cover virtually all engineering disciplines ranging from theoretical research to very application specific work. In addition interesting problems for physics and mathematics arises out of such research. We hope this book will be an inspiring source of knowledge and ideas, stimulating further research in this exciting field. The promises and possible benefits of such efforts are manifold, they range from new transportation systems, intelligent cars to flexible assistants in factories and construction sites, over service robot which assist and support us in daily live, all the way to the possibility for efficient help for impaired and advances in prosthetics.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821