SDU International Journal of Technologic Sciences
Vol. 3, No 1/Special Issue, February 2011
pp. 23-36

Computer Technology

# A NOVEL AND SIMPLE-TO-IMPLEMENT FRIEND OR FOE IDENTIFICATION SYSTEM IN MULTI-ROBOT BATTLEFIELD

Atilla ELÇİ[*], Behnam RAHNAMA[**], Ramin BAKHSHI [***], Hamid Mir-Mohammad SADEGHI [****]

**Abstract**

This paper draws a beginning-to-end framework for Friend or Foe (FOF) identification for multi-robot battlefield. Mini-sumo robots are well-known example case while absolute positioning of each teammate is not practical. Our simple-to-implement FOF identification does not require two-way communication as it only relies on decryption of payload in one direction. It is shown that the replay attack is not feasible time wise as the communication is encrypted and timestamp is inserted in the messages. The hardware implementation of cooperative robots is equipped with rotary robot able to detect direction and distance to detected object in addition to gyroscope chipset. Studying dynamics of robots allows finding solutions to attack enemies which are more powerful than friends from sides so they will not be able to resist. Besides, there are certain situations that robots must escape instead of fighting. Experimental part of this research attempts to illustrate results of real competitions of cooperative mini-sumo battlefield as an example of localization and mapping while collaborative problem solving in uncharted environments.

**Keywords:** Friend or Foe, Cooperative Robotics, Positioning, Battlefield, Mini-sumo

## 1. Introduction

Multi-robot systems rely on safety of communication, accuracy of mapping and localization techniques. In battlefield environments, such as minesweeping, and earthquake survival robots, multiple agents act autonomously while they collaborate towards achieving their common goals. Well-known mini-sumo robotics competition class is one of such situations where a team of wrestler robots tries to push enemy out of the battlefield. They require recognizing friends and distinguishing them from enemy so that they will be able to solve the task cooperatively and not to push a teammate out.

Distributed leader election and the scenario of considering one of the team members as master and the rest as slaves complicate the task while there will be a bottleneck for communication. Within this respect, distributed agents act autonomously and work together to reach the common goal without expecting or waiting others to help them. However, they help each other unconsciously. They might attack the same enemy simultaneously.

[*] Eastern Mediterranean University, Department of Computer Engineering and Internet Technologies Research Center, Gazimagusa, Via Mersin 10, TRNC, Turkey, E-mail: atilla.elci@emu.edu.tr; currently with Toros University, Mersin, Turkey.

[***] Eastern Mediterranean University, Department of Computer Engineering and Internet Technologies Research Center, Gazimagusa, Via Mersin 10, TRNC, Turkey, E-mail: behnam.rahnama@emu.edu.tr; currently with European University of Lefke, Gemikonağı, TRNC.

[***] Eastern Mediterranean University, Department of Electrical Engineering,Gazimagusa, Via Mersin 10, TRNC, Turkey, E-mail: ramin.bakhshi@cc.emu.edu.tr

[****] Eastern Mediterranean University, Department of Mathematics, Gazimagusa, Via Mersin 10, TRNC, Turkey, E-mail: hamid.sadeghi@cc.emu.edu.tr

In such situations robots need to detect position of other teammates and enemy to act upon them. In literature, there are several methods available for localization and positioning. Basically, in uncharted, restricted, and real-life environments those map building methods based on only relative positioning of teammates are more preferred.

J. Borenstein et al [1] defined seven categories for positioning systems based on the type of sensors used in controlling the robot.

- ***Odometry*** is based on simple equations which hold true when wheel revolutions can be translated accurately into linear displacement relative to the floor. However, in case of wheel slippage and some other more subtle causes, wheel rotations may not translate proportionally into linear motion. The resulting errors can be categorized into one of two groups: systematic errors and non-systematic errors.
- ***Inertial Navigation*** uses gyroscopes and accelerometers to measure rate of rotation and acceleration, respectively. Measurements are integrated once (or twice, for accelerometers) to yield position.
- ***Magnetic Compass*** is widely used. However, the earth's magnetic field is often distorted near power lines or steel structures. Besides, the speed of measurement and accuracy is low. There are several types of magnetic compasses due to variety of physical effects related to the earth's magnetic field. Some of them are Mechanical, Fluxgate, Hall-effect, Magnetoresistive, and Magnetoelastic compasses.
- ***Active Beacons*** navigation systems are the most common navigation aids on ships and airplanes, as well as on commercial mobile robot systems. Two different types of active beacon systems can be distinguished: trilateration that is the determination of a vehicle's position based on distance measurements to known beacon sources; and triangulation, which in this configuration there are three or more active transmitters mounted at known locations.
- ***Global Positioning System*** (GPS) is a revolutionary technology for outdoor navigation. GPS was developed as a Joint Services Program by the Department of Defense. However, GPS is not applicable in most of robotics fields due to two reasons, firstly, unavailability of GPS signals indoor; and secondly, low accuracy in small prototype single chip GPS receivers used in cellular phones and robot boards.
- ***Landmark Navigation*** is based on landmarks that are distinct features so a robot can recognize from its sensory input. Landmarks can be geometric shapes (e.g., rectangles, lines, circles), and they may include additional information (e.g., in the form of bar-codes). In general, landmarks have a fixed and known position, relative to which a robot can localize itself.
- ***Model Matching*** or ***Map-based positioning***, also known as ***map matching*** is a technique in which the robot uses its sensors to create a map of its local environment. This local map is then compared to a global map previously stored in memory. If a match is found, then the robot can compute its actual position and orientation in the environment. Certainly there are lots of situations where achieving global map is unfeasible or prohibited. Therefore, solutions based on independent sensors carried on robots are more likely valued.

Self localization of autonomous agent has been considered a fundamental problem in mobile robotics for long time. Specially, controlling the large number of mobile robots as members of a team, which are able to move and recognize their position, is an important concern where cooperative solving of a given task is required [2]. C. Belta et al [3] proposed an abstraction method for controlling such groups of robots. For decreasing the stream of communication between the members, based on the definition of a map from the battlefield configuration at firs, they subdivided the area to lower the dimension for each robot. Secondly, they designed and derived controllers for individual robots. However, these controllers utilize feedback that depends on the current state of the robot. As mentioned above, number of communication messages between distributed robots within a decentralized architecture is highly important to be reduced [4].

Another interesting research was done by M. Peasgood et al [5] wherein, it addresses the challenging problem of finding collision-free trajectories for many robots moving toward individual goals within a common environment. Many methods have been proposed for planning the motion of one or more robots whether they are guaranteed to find a solution if one exists. Their multiphase approach to planning problem uses a graph and spanning tree representation to create and maintain obstacle-free paths through the environment for each robot to reach its goal.

Various control strategies for mobile robot formations have been reported in the literature, including behavior based methods, virtual structure techniques, and leader–follower schemes [6]. Among them, the leader–follower approaches have been well recognized and become the most popular approaches.

The basic idea of this scheme is that one robot is selected as leader and is responsible for guiding the formation. The other robots, called followers, are required to track the position and orientation of the leader with some prescribed offsets. The advantage of using such a strategy is that specifying a single quantity (the leader's motion) directs the group behavior. In followers, sliding-mode formation controller is applied which is only based on the derivation of relative motion states. It eliminates the need for measurement or estimation of the absolute velocity of the leader and enables formation control using vision systems carried by the followers. However, it creates bottleneck for message passing and decision making while it can be improved by decentralized autonomous control such as in [7] on the other hand, situations wherein the leader dies is not considered.

Coordination, integration of information collected by different robots into a consistent map, and dealing with limited communication during exploration were integrated in a distributed multi-robot exploration and mapping system developed by D. Fox et al[8]. The approach enables teams of robots to efficiently build highly accurate maps of unknown environments, even when the initial locations of the robots are unknown. In order to avoid wrong decisions when combining their data into shared maps, the robots actively verify their known relative locations. Uncertainty about the relative locations or being involved in situations masking robots to point other teammates causes wrong or possibly corrupted results.

Simultaneous localization and map building (SLAM) problem has attracted more and more attention in recent years, due to the enormous potential of multi-robot exploration of unknown environments. A solution to SLAM problem was presented by M. Di Marco et al [9] for a team of cooperating robots. Each robot plays the role of a moving landmark for all other robots. If only one robot at a time is moving, the others can act as a landmark base in regions where it is difficult to extract reliable features. Moreover, at each time instant, the same feature can be perceived by more than one robot. If the robots share mapping information, this can lead to a more accurate, faster-converging global map.

In the following section, the proposed FOF identification system is presented in detail. Consequently, section 3 covers hardware development of cooperative mini-sumo robots. Section 4 covers cryptanalysis of communication and possible attack scenarios and vulnerabilities of the system. It ends with useful strategies that enhance the performance and results of the algorithm. Experimental results over 50 competitions are shown in section 5. Finally achieved results are concluded in section 5.

## 2. Proposed Scheme

In this paper a novel and simple-to-implement FOF identification system is proposed. The system is composed of ultrasonic range finder rotary radar scanning the circumference for obstacles, and an infrared receiver reading encrypted echo messages propagated from omnidirectional infrared transmitter on the detected object through a fixed direction.

Each robot continuously transmits a message encrypted by a shared secret key between teammates consisting of its unique identifier and timestamp. The simplicity is due to excluding transceiver system for exchanging encoded/decoded messages. System counters replay attack by comparing the sequence of decoded timestamps. Encryption is done using a symmetric encryption technique such as RC5. The reason for selecting RC5 is its simplicity and low decryption time. Besides its hardware implementation consists of few XOR and simple basic operators which are available in all microcontrollers.

The decision making algorithm and behavioral aspects of each robot are represented as follows:

1. Scan for detecting surrounding objects using ultrasonic sensor.
2. Insert a record consisting of distance and position for each detected element into a queue.
3. Fetch the queue head record and direct the rotary radar towards its position.
4. Listen to IR receptor within a certain period (i.e. 100 ms)
5. if no message is received
    a. Clear all records
    b. Attack the object
    c. Go to 1
6. Otherwise,
    a. Decode the message using the secret key
    b. If not decodable Go to 5.a
    c. Otherwise, register the identifier and timestamp besides position and distance for detected object
    d. Listen again to IR receptor within a certain period
    e. Decode the message using the secret key
    f. If not decodable Go to 5.a
    g. Otherwise, match the identifier and timestamp against the one kept before
    h. If identifier mismatches or timestamp is the same or smaller than as it was before, Go to 5.a
    i. Else if detected identifier is the same as the identifier of detector, Go to 5.a
    j. Go to 3

It is assumed that the received message is free of noise and corrupted messages are automatically discarded. This can be done by listening for a limited number of times if message is not decodable. However, transmission is modulated on a 38 KHz IR carrier so sunlight and fluorescent light are not highly distorting the IR transmitted stream.

## 3. Hardware Implementation

Our first generation of cooperative mini sumo robot included an electronic compass instead of gyroscope and accelerometer so it was not able to detect skidding errors in any axes. Very common instance is when the robot is pushed by enemy. Fig. 1 presents the first developed board being able to control two DC servomotors, communicate through wireless over 900MHz modulation, and having infrared sensors and bumpers to detect surrounding objects.

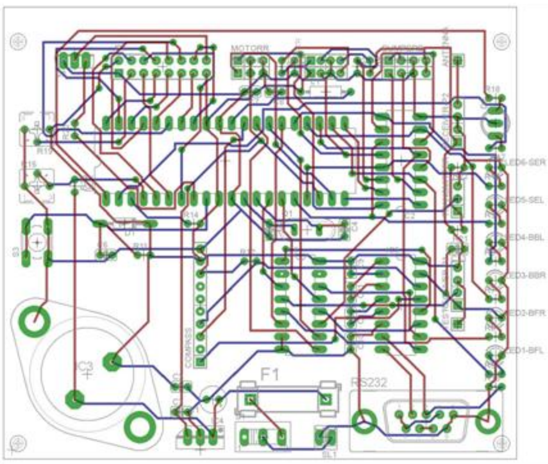



**Fig. 1.** The first generation of cooperative mini sumo platform robots.

In the second design, an extension board suitable for open source Mark III mini sumo robots is presented. The Mark III Robot is the successor to the two previous robot kits designed and sold by the Portland Area Robotics Society. The base robot is serial port programmable. It includes PIC16F877 20MHz microcontroller with boot-loader which has made programming steps easier. In-System Programming (ISP) is provided by boot-loader facility. It is possible to program the robot in Object Oriented PIC (OOPIC) framework. It includes controller for two DC servomotors in addition to three line following and two range finder sensors. Low-battery indicator is an extra feature provided on Mark III. However, there were few requirements to enhance the robot to fit our requirements for cooperative robotics. Wireless Communication, Ultrasonic range finder, infrared modulated transceiver, and gyroscope sensors were added as an extension board as shown in fig. 2. In addition, the robot uses two GWS S03N 2BB DC servomotors each providing 70 dyne cm torque at 6v. However, the battery pack connected to motors is not regulated so it does not provide steady voltage while discharging. It effects center point of Servo calibration which effects servo proper movement. In extension board, a regulator is also included to fix the problem explained above.

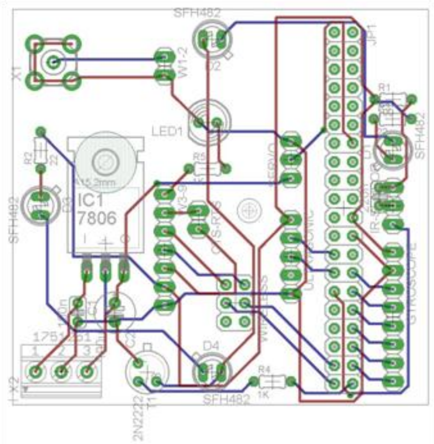



**Fig. 2.** The extension board for Mark III.

## 4. Cipher analysis and attacking Strategies

Following figure represents two of the worst cases for decision making in battlefield. These two crucial situations include 1) An enemy robot masks a friend and copies messages it receives from the masked friend to others so called replay attack. 2) Attacking an enemy by two friendly robots from opposite sides.

### 4.1. Replay Attack

In the first instance, $E_1$ stands between $F_2$ and $F_3$ covering their line of sight, so it is possible for $E_1$ to copy messages propagated from $F_3$ and replay them to $F_2$ and present itself as a friend and then attack against $F_2$. In this situation, $F_2$ assumes that $E_1$ is the friend $F_3$ and it will be targeting the next possible enemy detected by rotary radar however it will be attacked by $E_1$.

Part 6 of the algorithm presented in Section 2 counters replay attack. In order to avoid replay attack, the timestamp included in decrypted message is compared against the one received in advance. Besides, the other friend robot receives the same copy of its own transmitted message including its identifier. Therefore it recognizes the enemy by matching and comparing the identifier of copied message with its own unique identifier. Therefore it recognizes the enemy.

### 4.2. Opposite Side Dual Attack

According to the algorithm represented in previous section, both $F_2$ and $F_3$ start attacking $E_1$ from opposite sides either towards sideways of $E_1$, or one faces front of $E_1$. In both cases, they keep pushing enemy until they see the boundary so they return and start searching for other enemies. However, they either stay in this situation and challenging for a long time or one of friend robots understands that it is pushed out. It is highly possible so any of friends will be detected by other enemies and will be pushed out. Therefore, a convincible strategy is to escape if it is not able to push. Being pushed or challenging without being able to push is simply detectable by checking gyroscope and acceleration sensors. LIS3LV02DL from free samples of ST Microelectronics single chipset gyro-acceleration sensor is used to provide movement and acceleration towards x, y, and z axes.

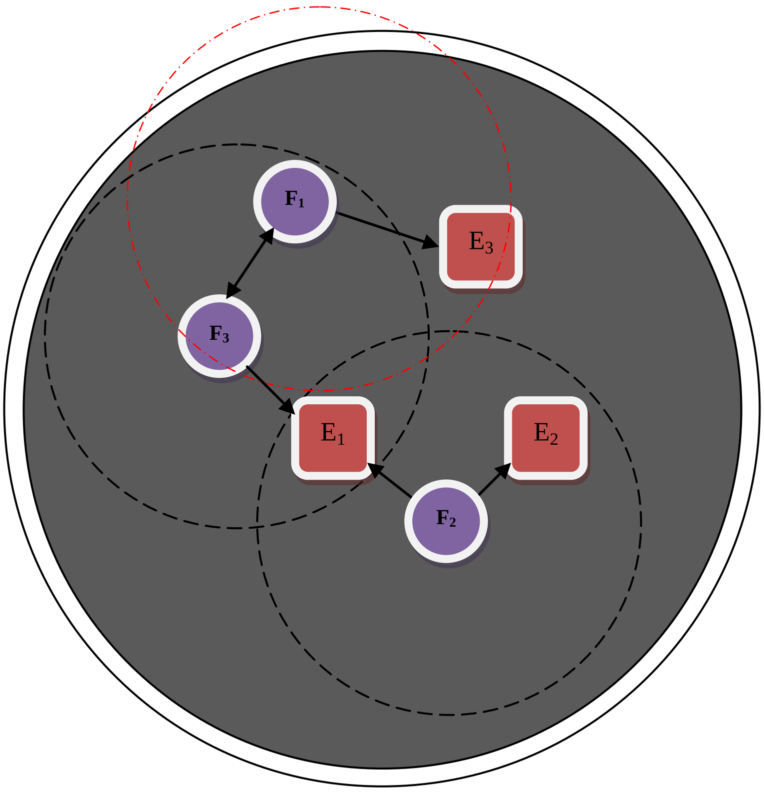



**Fig.3.** An example arrangement of two teams of robots while fighting. Arrows demonstrate detection of objects.

## 4.3. Strategies

Escape strategy simply consists of backing off for a period or rotating around itself with maximum speed and then moving towards a direction so it can start the algorithm from beginning or attack the enemy from a better direction.

Another upgrade in algorithm is to cancel an attack if the enemy is escaped away out of detection radios. The reason is making the system more efficient and spending time on fighting against other enemy robots instead of pursuing an escaping robot which might not be caught in a short while.

It is assumed that the radius of detection range is adjusted to half of radius of the platform. It is due to applying Divide and Conquer (DAC) policy within cooperative robots by assuming to solve each subset of battlefield by one of the robots. In addition it reduces the complexity and collision while communicating with other teammates. Later it is shown that the radius of detection can be dynamically changed based on real-time conditions of match.

A better but more time consuming approach is to detect all enemies in range and then decide which one to attack rather than attacking against first detected enemy. For instance, $E_1$ and $E_2$ are in sight of $F_2$. In this situation $F_2$ should be intelligent enough to choose the best attack.

It is highly possible for robots to be at the boundary so they cannot back off or run away. Therefore the robot has to attack the first detected enemy asking for help from teammates.

Determining the level of power of enemy robots helps deciding to utilize escape strategy more efficiently. The problem refers to the condition where the power level of enemy robots is more than ours. Therefore, in such situations having face to face attack is not desired. Instead, the only way to remedy is to attack from wheel sides of enemy robot. Consequently finding relative movement angle of the enemy robot helps friend robots to decide whether to attack or not. Following are three main concerns.

#### 4.3.1. Determining the level of power of enemy robots

Utilizing gyroscope and matching it with usual speed of the robot in steady state helps measuring movement toward *x,y,z* axes. See fig. 4.

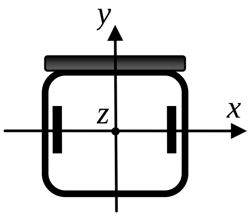



Fig. 4. The direction of axes over the robot while *y* showing the front of the robot.

Let us assume that $A_x$, $A_y$, $A_z$ variables presents gyroscope values respectively. The digital returning value from SPI port indicating gyroscope results follow $A_x, A_y, A_z \in$ (-512,+512). Attacking face to face an enemy robot is when $A_y < -\alpha, \alpha > 0$ or $A_x > \beta, \beta > 0$. $\alpha$ and $\beta$ are threshold values such that $A_y < -\alpha$ shows backward movement and similarly $A_x > \beta$ indicates side movements more than an acceptable threshold for skidding errors. Therefore, $A_y < -\alpha$ indicates that the level of power of the enemy is more than being able to repel against. In this case attacking sideways of enemy is needed. Respectively, the relative angle of enemy should be suitable for attack so that one side of enemy could be caught. The ultrasonic rangefinder on implemented rotary radar determines the distance to detected object. The relative angle is calculated from position of DC servomotor rotating the radar. An example is represented in fig. 5.

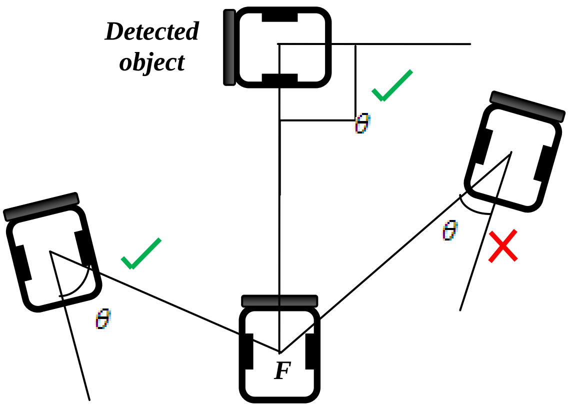



**Fig. 5.** $\theta$ is acceptable if $\theta > \theta_0$.

#### 4.3.2. Determining the speed of enemy

Estimating the velocity of an enemy robot is done through two ways. Firstly, while the enemy attacks directly towards friend. Therefore, $v_e = \frac{l}{s}$ , $v_e$ is velocity of enemy robot, and *l* is the distance traversed is *s* seconds. Secondly, we can estimate speed of enemy robot using radar. At first detection of enemy, assuming its distance is $l_1$ and detecting it again in a short while as *s* seconds in distance $l_2$ with $\theta$ degrees angular rotation of rotary radar, speed can be calculated as follows using law of Cosines as shown in fig. 6. $l_s = \sqrt{(l_1)^2 + (l_2)^2 - 2l_1 l_2 \cos\theta}$ and $v_e = \frac{l_s}{s}$.

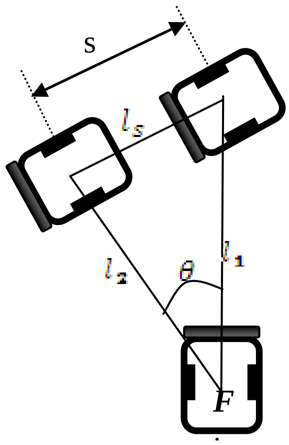



**Fig. 6.** Second way of calculation of the average speed of enemy.

### 4.3.3. Determining the relative angle of enemy robots

The relative angle is considered in both static and dynamic situations. Static situation (see fig. 7) is while a friend robot does not move. Reversely, dynamic situation concerns when the friend robot is moving. Following figure depicts such situations.

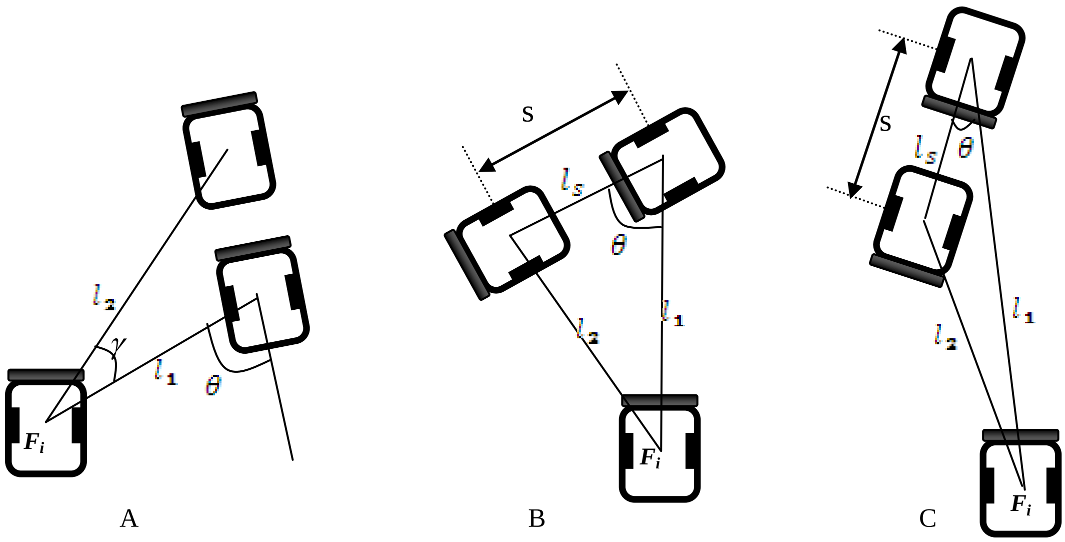



Fig. 7. A) While enemy is going away from the friend robot. B) The enemy gets closer with a desirable angle. C) While enemy gets closer with an angle more than threshold.

In fig. 7.A. $l_2 > l_1$ then in attacking strategy it is decided to follow the enemy if it is in an acceptable range considering that the enemy would not be able to change the role of front and back side of the robot. Otherwise, leaving the target is a better decision as enemy probably has time to attack the friend robot.

In fig. 7.B. $l_2 < l_1$. , $\alpha$ is an acceptable threshold for speed of decreasing distance of enemy towards friend. Satisfying above inequality allows attacking the enemy. $l_s = v_e \cdot s$ , $l_s$ is the distance traversed by enemy robot in *s* seconds.

In the situation shown by fig. 7.C. the friend robot is not allowed to attack. Therefore escape strategy is executed thus the friend robot runs away. In other words, $\frac{l_1 - l_2}{s} > \alpha$ , which shows that the enemy is in good state to attack friend.

$\bar{l} = l_1 - l_2$ , $\bar{v} = \frac{\bar{l}}{s}$ , $\bar{v}$ stands for velocity of enemy moving towards friend robot. Final results of the static situation are as follows:

1. If $\bar{v} \cong 0$ then the movement of enemy is octagonal to our robot.
2. If $0 < \bar{v} < \alpha$ then enemy is getting close with an acceptable relative angle for the friend to attack.
3. If $\bar{v} \cong v_e$ then the enemy is able to attack straight.

Next, the dynamic situation is considered. As shown in fig. 8. $l_{s_e} = v_e \cdot s$ , $l_{s_e}$ is the distance traversed by enemy in *s* seconds. Similarly, $l_{s_f} = v_f \cdot s$ , $l_{s_f}$ is the distance traversed by friend in *s* seconds. Then results of the dynamic situation are as follows:

1. If $\bar{v} < v_f$ then enemy is going away (see fig. 8.A.).
2. If $\bar{v} \cong v_f$ then the movement of enemy is octagonal to our robot.
3. If $v_f < \bar{v} < \alpha$ then enemy is getting closer (see fig. 8.B.)
4. If $\alpha < \bar{v} \leq v_f + v_e$ then the enemy is able to attack straight.

Conditions 2 and 3 are desirable for friend to attack. However, a better strategy in condition 4 is to escape away. Condition 1 depends on the ratio of speed of friend and speed of enemy. This ratio can be used in decision making strategy whether to attack or leave the enemy.

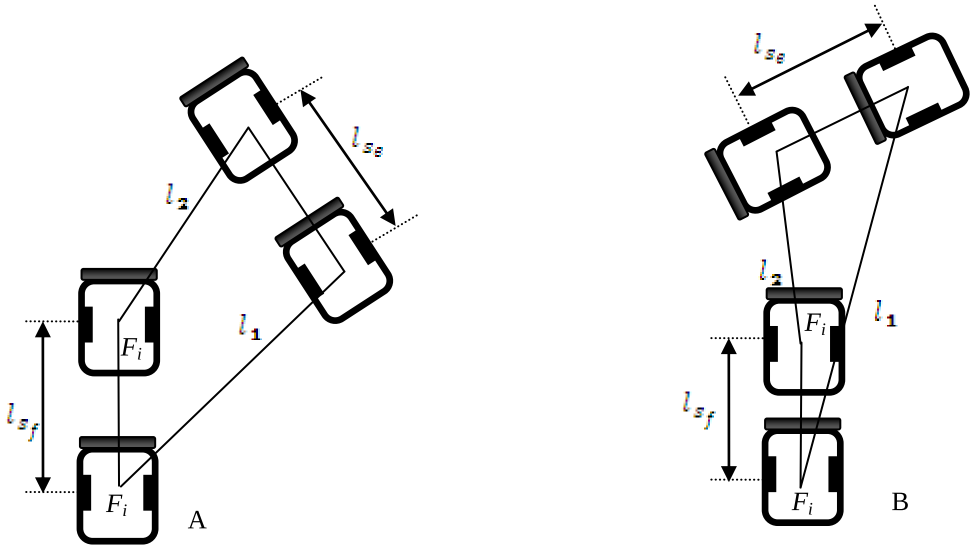
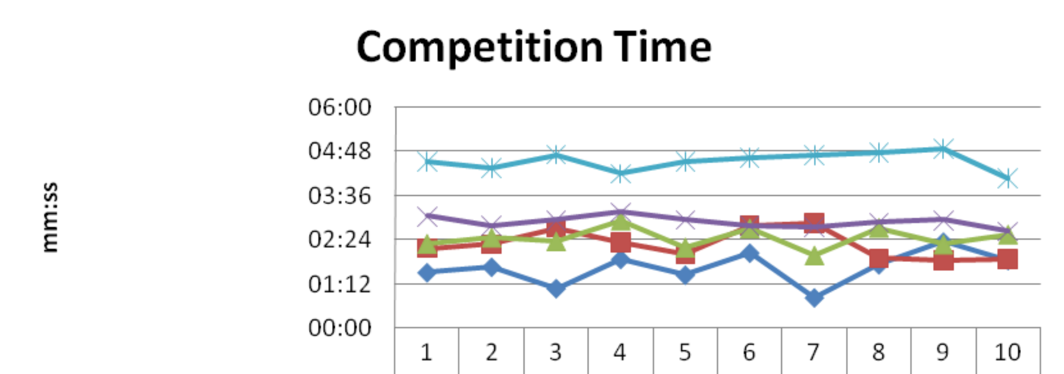



**Fig. 8.** A) $\bar{v} < v_f$. B) $v_f < \bar{v} < \alpha$ .

If the enemy comes towards friend straightly and there would be no possibility to escape, friend should start attack while announcing request for help over wireless medium. Notice, it is already known that level of power of enemy is higher than level of power of friend. Therefore more likely, friend will lose the battle. Now teammates can decide to help the challenging friend if the distance is acceptable or if friend is in the range of their radar, or leave the friend to die.

## 5. Experimental Results

The developed system is tested on teams of three robots, i.e., friends and enemies. The teams of enemies and friends each consists of three cooperative robots with basic abilities which include IR transceiver for FOF identification. The test is done for ten rounds. Last remaining robot's team wins the game. There were five different situations to test robots. Therefore, fifty different rounds of competition were conducted. These five situations included basic, wireless enabled, radar and wireless enabled, radar and wireless with gyroscope, and finally everything in addition to utilizing escape strategy. Wireless communication helps robots to talk to each other, share their information, and ask for help. Rotary radar is an ultrasonic range finder. Gyroscope shows movement in three axes. Finally escape strategy is a software enhancement as mentioned in the earlier section. Following figure presents five sets of competitions each in ten rounds. The absolute duration for each competition resulting loss of one team is considered separately in terms of mm:ss.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Base | 01:31 | 01:40 | 01:05 | 01:53 | 01:27 | 02:03 | 00:50 | 01:44 | 02:21 | 01:51 |
| Wireless Enabled | 02:10 | 02:17 | 02:44 | 02:20 | 02:01 | 02:47 | 02:52 | 01:54 | 01:50 | 01:53 |
| Radar & Wireless Enabled | 02:18 | 02:28 | 02:22 | 02:56 | 02:12 | 02:43 | 01:59 | 02:44 | 02:17 | 02:33 |
| Wireless & Radar & Gyroscope | 03:02 | 02:47 | 02:56 | 03:09 | 02:57 | 02:47 | 02:45 | 02:53 | 02:56 | 02:38 |
| All with Escape Strategy Enabled | 04:32 | 04:21 | 04:43 | 04:12 | 04:32 | 04:38 | 04:42 | 04:46 | 04:53 | 04:04 |

**Fig. 9**. Competition time for 50 different tests

The average time for each set of ten competition rounds per testing situation is presented in fig. 10. The curve shows increasing average time in each set based on addition of equipments and utilities. By enabling the wireless communication system the average time is increased by 138%. Adding the rotary radar increases the average time by 150% over the basic situation. Adding the gyroscope to the previous step increases the average time of the competition approximately by 221%, and enabling the escape strategy increases the average time by 270% of the base system. The following figure concluded the results shown in fig. 9 for all five situations.

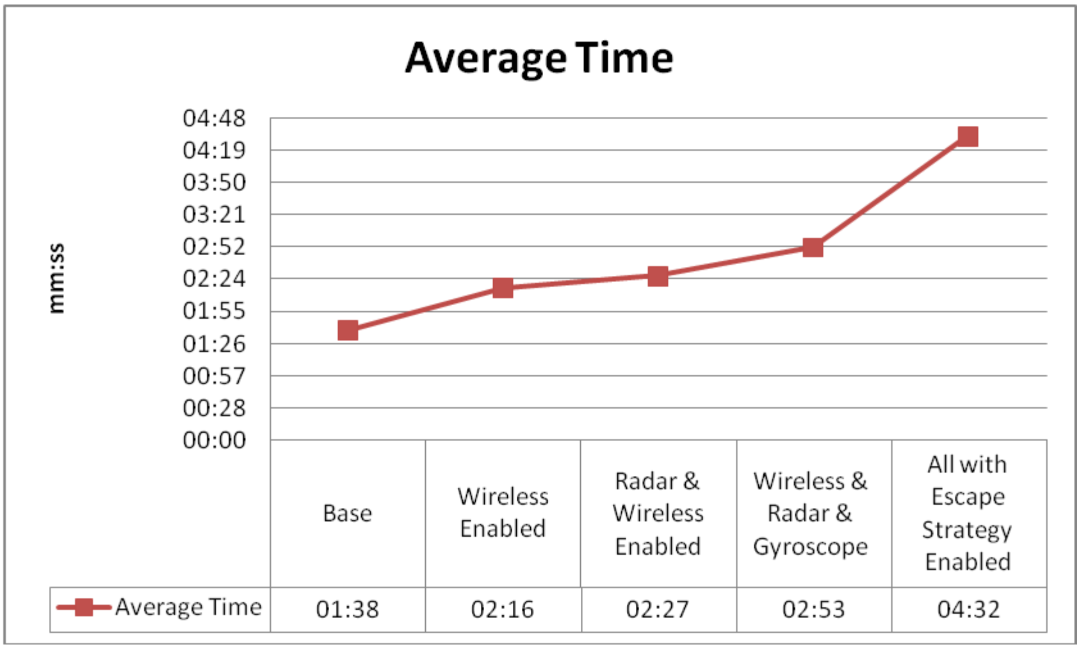



**Fig. 10.** Average time of five competition situations.

The results of competition at each step are displayed in fig. 11. The performance of the final system based on total wins in each set proves the enhancements of the new FOF identifications and intelligent strategies to reach more cooperatively solving shared problems. As shown below, the wining ratio has been increased from 40% to 90% by strategies introduced in this research.

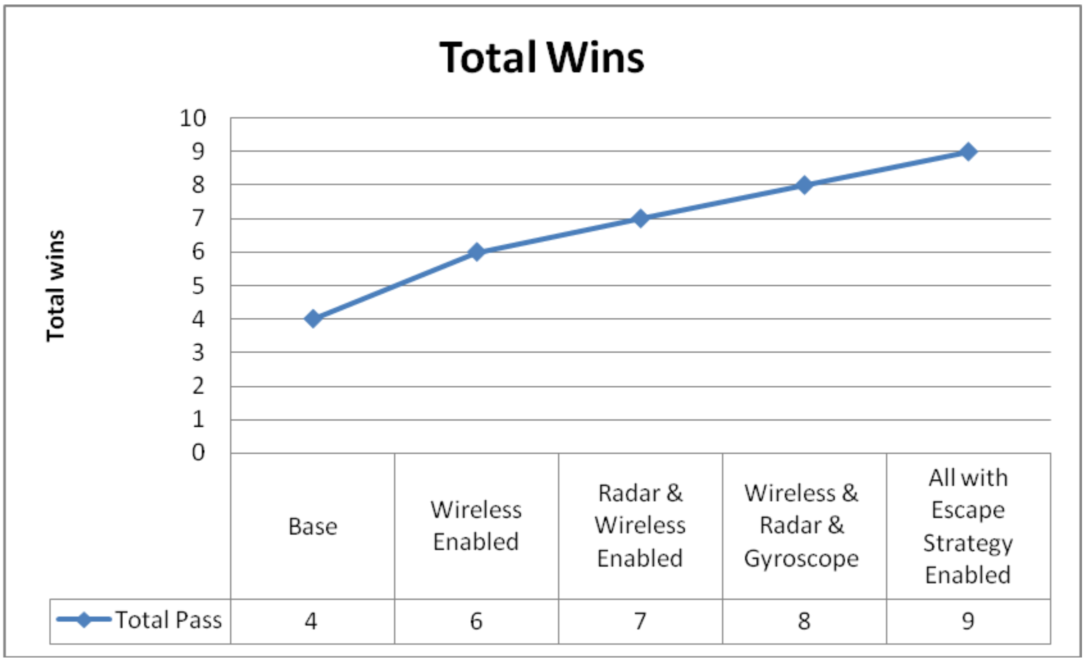



**Fig. 11**. Total number of wins per each competition situation.

## 6. Conclusion

This paper draws a beginning-to-end framework for friend-or-foe identification for cooperative robotics. Mini-sumo robots were the example case of this study while absolute positioning of each teammate was not possible. Our designed friend-or-foe strategy does not require two-way communication as it only relies on decryption of payload in one direction.

The replay attack is not feasible time wise as the communication is encrypted and timestamp is inserted in the messages.

There were two sets of hardware implementations for the test robots. The second version is equipped with rotary robot able to detect direction and distance to detected object in addition to gyroscope chipset.

There were several facts that could be enhanced in decision making algorithm. In situations where robots were losing the game against more powerful enemies, studying dynamics of robots allowed finding solutions to attack such enemies from sides so they will not be able to resist.

There were certain situations that robots must escape instead of fighting. These states were based on the relative position, speed and other properties of enemies which could be calculated by friend robots helping them to act more intelligently in cooperative environments.

Experimental part of this research attempted to illustrate results of real competitions of cooperative mini-sumo battlefield as an example of localization and mapping besides collaborative problem solving in uncharted environments. In order to compare them with the theoretical predictions, measurement variables such as time of competition, mean time and number of wins in each situation was presented. It is found out that the presented theoretical aspects meet the experiences presenting cooperative robots with 90% accomplishing their mission comparing against ordinary robots. However, enhancements applied on the robot increased the time by 270% from base robots mainly due to applying escape strategy through which the robot understands that it is not able to fight alone at the current position.

## References

[1] Mobile Robot Positioning & Sensors and Techniques. Borenstein, J., et al. Special Issue on Mobile Robots, s.l. : IEEE, April 1997, Invited paper for the Journal of Robotic Systems, Vol. 14, pp. 231-249.

[2] Defining a Strategy to Select Either of Closed/Open World Assumptions on Semantic Robots. Elçi, Atilla, Rahnama, Behnam and Kamran, Saman. Turku, Finland : IEEE, 2008. COMPSAC2008. pp. 417-423.

[3] Abstraction and control for Groups of robots. Belta, C. and Kumar, V. 5, s.l. : IEEE, October 2004, IEEE Transactions on Robotics, Vol. 20, pp. 865 - 875.

[4] Applying Semantic Web in Engineering a Modular Architecture for Design and Implementation of a Cooperative Labyrinth Discovery Robot. Elçi, Atilla and Rahnama, Behnam. Gemikonağı, Northern Cyprus : s.n., 30 Nov - 1 Dec 2006. 4th FAE International Symposium on Computer Science and Engineering. pp. 447-451.

[5] A Complete and Scalable Strategy for Coordinating Multiple Robots Within Roadmaps. Peasgood, M., Clark, C.M. and McPhee, J. 2, s.l. : IEEE, April 2008, IEEE Transactions on Robotics, Vol. 24, pp. 283 - 292.

[6] Sliding-Mode Formation Control for Cooperative Autonomous Mobile Robots. Defoort, M., et al. 11, s.l. : IEEE, November 2008, IEEE Transactions on Industrial Electronics, Vol. 55, pp. 3944 - 3953 .

[7] Elçi, Atilla and Rahnama, Behnam. Semantic Robotics: Cooperative Labyrinth Discovery Robots for Intelligent Environments. [book auth.] Andreas Tolk and Lakhmi C. Jain. Complex Systems in Knowledge-based Environments: Theory, Models and Applications. Berlin Heidelberg : Springer-Verlag, 2009.

[8] Distributed Multirobot Exploration and Mapping. Fox, D., et al. 7, s.l. : IEEE, July 26, 2006, Proceedings of the IEEE, Vol. 94, pp. 1325 - 1339.

[9] Simultaneous localization and map building for a team of cooperating robots: a set membership approach. Di Marco, M., et al. 2, s.l. : IEEE, April 2003, IEEE Transactions on Robotics and Automation, Vol. 19, pp. 238-249.