Intelligent Algorithms Based on Data Processing for Modular Robotic Vehicles Control

MICHAIL PAPOUTSIDAKIS AND DIMITRIOS PIROMALIS
Department of Automation
Technological Institute of Piraeus
P.Ralli & Thivon 250, 12244, Athens
GREECE
mipapou@teipir.gr

FILIPPO NERI
Dept. of Electrical Engineering and Information Technology
University of Napoli "Federico II"
Via Claudio 21, 80125 Naples
ITALY
filipponeri@unina.it

MICHEL CAMILLERI
Dept. of Computer Information Systems
University of Malta
MSD 2080
MALTA

Abstract: - One of the numerous implementations that the Intelligent Systems Lab of the Technological Institute of Piraeus, Greece, has recently accomplished is described in this paper. Within the area of hand-on autonomous robotics, the project combines computer science in modern wireless network protocol communication, with microcontroller-based intelligent motion-control tasks. The design of the presented pair of versatile robots is intended for a “predator and prey” application domain. Both robot designs are based on low cost and easy to find equipment, though modern and up-to-date technology. Two different multitasking and flexible algorithms are designed in order to investigate their performance and compare the outcomes. In particular, an intelligent algorithm based on data capturing and processing within a tolerable elapsed time is presented, with two different performance scenarios explained and compared in detail.

Key-Words: - *hand-on robotics, versatile platforms, wireless communication, intelligent algorithms.*

1 Introduction

Mechatronics and Robotics are considered a young and under development discipline although they both have attracted early on the interest of engineering research. Building and designing autonomous robots requires a diverse set of technical and non-technical skills that encompass mechanical engineering, electrical engineering, data network communications and much of software design. Nowadays the availability of low cost hardware for building robots aided by the steady decline in cost for computational hardware has expanded the field of autonomous robotics applications. Computational power and hardware have finally caught up to the level necessary to achieve the concepts of robot designers. However, with the same hardware technology available to all, the distinguishing factor in most cases becomes the software design. The software for autonomous robots must intelligently control the hardware so that it functions in unstructured, dynamic and uncertain environments while maintaining autonomous characteristics like adaptability. Moreover, the world-wide popularity of robotics competitions, in all their different versions, provides opportunities for constant improvement in teaching computer science and engineering principles at all student levels.

In the field of mechatronics and robotics applications, many researchers in the recent past like, Amerongen and Jongkind, Brussel, Chamilothoris, Mataric , Koenig and Feil-Seifer, Yu and Weinberg, Piepmeier, Bishop and Knowles, have pointed out the fact that development and improvement of such systems arise naturally from the need of delivering knowledge to potential engineers. In Addition to this, universities and technological institutes all over the world emphasize the need of robotics application in everyday life by delivering not only stand alone courses in this field but also host Departments based on Mechatronics and Robotics design and construction.

All robotics and mechatronics platforms are designed and implemented by the staff of the Intelligent Systems Lab of Department of Automation and after that are distributed to students for further testing, experimentation and improvements. Given the fact of the versatility of such systems, it can be easily understood that they are a huge research challenge for further consideration. Instead of consuming huge budget amounts, this work is based on minimum cost (see next chapters) and easy to find equipment in next door hardware stores. I will be proven that for the task of this project clever ideas and intelligent algorithms are substitutes with no worse performance outcomes than other highly expensive and hard to assembly constructed robots. All authors of this paper are unaware of robot applications supported by this specific low budget that might have taken place anywhere else in a university or research centre for instance. Such application, which is a sequel follower of many more in the close past, will be described in this paper.

It refers to the “predator and prey” application, based on a pair of two small mobile and autonomous robots that act like the predator and prey. Keeping in low levels the hardware cost, the two individual robots will perform in a restricted arena; avoid random obstacles and stay within the arena boundaries until the predator ‘touches’ the prey, which means that the distance between them will finally be less than 10mm. Schematics and figures will be provided in the following chapters of this paper in an attempt to fully illustrate the construction details and the motion control of this pair of mobile robots.

3 Mobile Robot Platform Architecture

Both robots bodies were designed and built from off-the-shelf components. The majority of the parts, which were used to produce both platform chassis, were purchased at a local hardware store. The quote of the characteristics and operation details of all equipment and components, which were used in this project, are beyond the purposes of this paper. Instead, all equipment will be briefly mentioned and the necessity of them will be thoroughly explained. Finally, it must be stated that the dimensions of both robots are such in order to be easily carried with one human hand and support the mobility of them, like seen in figure 4.

2.1 The "prey" platform

The most important part of the application is the prey robot construction since it requires more equipment and software design than the predator robot discussed in following chapter. The building architecture that was followed required a compact size, multiple level construction in order to fit all equipment and do not exceed in length the wheel diameter more than the double of it. This way, the obstacle avoidance ability is significantly improved and the agility of the robot highly increased. A side view of the prey robot is given below.

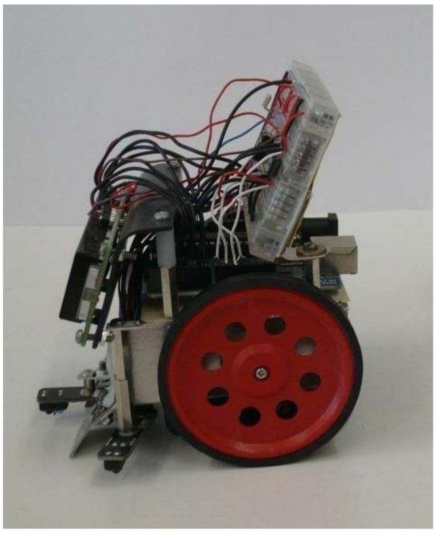

Figure 1. The “prey” robot

The prey platform is remotely controlled based on modern technology wireless data communication, such as Bluetooth, which is used for the task. The user can drive the prey robot on his demand simply by using his mobile phone, tablet PC or PDA in case they support Android software platform. For the implementation of this research paper, a low cost mobile phone was used and a software application was designed especially for this task on Android platform. The next figure illustrates the control buttons on the mobile phone’s touch screen display, which are used to drive the prey robot remotely.

Figure 2. Android mobile phone used as remote control

It can be clearly noticed that all types of move are included in the control panel of the phone, simulating precisely the functions of a joystick. The software design for this manipulation allows fast and relatively long distance guidance of the robot.

The prey robot hosts, on the middle level of its platform, a Bluetooth receiver, a BlueSmirF Gold board, which receives the control commands from the user's mobile phone and transforms them to input signals of the microprocessor. An on board integrated ATMEGA 2560 was used for the prey robot implementation, which is an up to date piece of equipment with a worldwide acceptance in the area of controlling robotics. An overview of the BlueSmirF Gold board attached to the ATMEGA microprocessor is provided in figure 3.

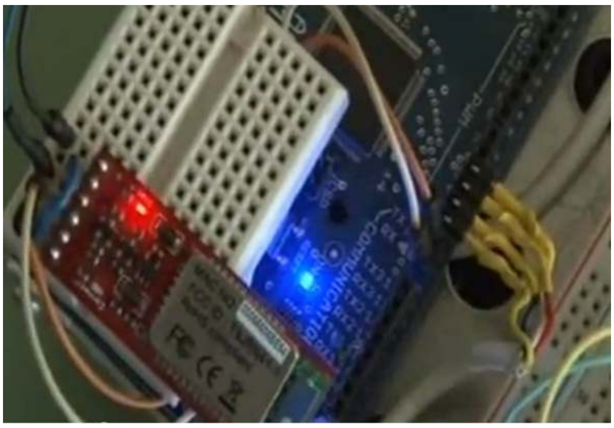

Figure 3. The Bluetooth board (on the down left corner) attached to the microcontroller

In addition to this sensor, infrared (IR) detectors were placed around the platform as well as a supersonic sonar module (HC-SR04) to ensure that the robot will always stay within the experimentation arena boundaries and never overcome them even if the user demands it, as described in chapter III in detail. At the bottom level of the platform one can find the two DC motors that deliver sufficient torque to rotate the two wheels, as well as an adequate power source of the robot that is a 5000mAh lithium rechargeable battery. This so-called 'battery bank' has the capability of recharging via a USB port of a PC: no external charger and more wiring required. At the top level of the platform a very significant piece of equipment is placed on a test board. The infrared (IR) LED, which acts as the location transmitter of the prey robot, operates with a waveform of 40 KHz frequency. This signal represents in real time the prey robot's trajectory as it moves and it is tracked by the predator robot's receiver in order to start the chasing application. The overall cost of the prey robot including all of the equipment (hardware and electronic parts) and also the mobile phone in figure 2, fluctuates in a close range of 300 Euros.

2.2 The "predator" platform

The predator robot consists of a different chassis compared to the prey robot in order to illustrate the individuality of them, the versatility that hand-on robots have and the ability to be transformed upon user's request. It follows the prey robot construction in terms of placing equipment in multiple levels and therefore minimizing its dimensions. It uses an ATMEL 1051/ATMEGA 238P (embedded on a board) microprocessor for C language control code compiling, and two DC motors that drive left and right wheel respectively. It also hosts a LCD display, which provides operation details like running time, distance covered etc, as seen in figure 4, where a view of the predator robot is provided.

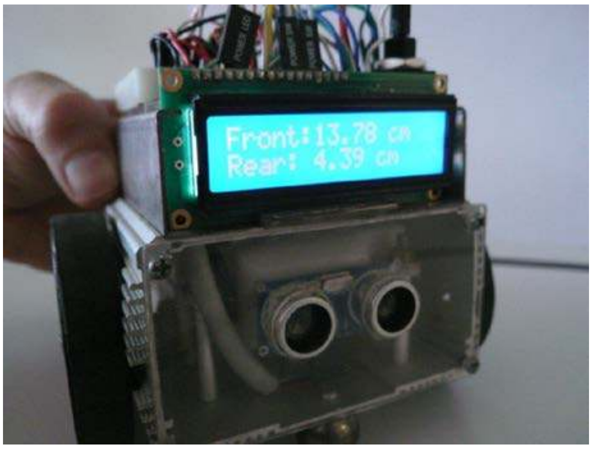

Figure 4. The “predator” robot

Figure 5 is showing the side view of the robot, where one can notice a part of the wiring and connections of the robot and the compact construction of it that ensures the steady operation of it.

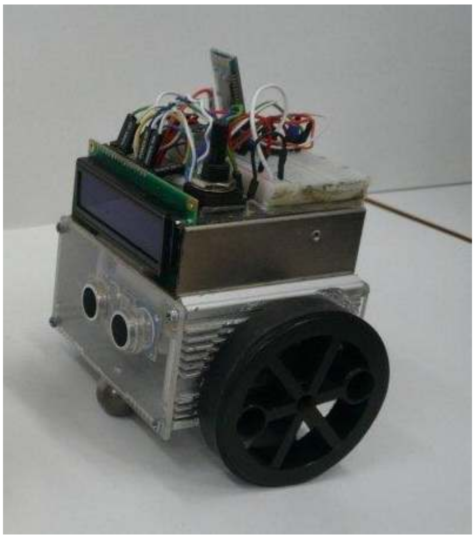

Figure 5. The “predator” robot side view

The ability of the robot to stay within the arena limitations is supported by two infrared (IR) detectors placed at the bottom of the platform that operate as position sensors. Moreover, a supersonic module (HC-SR04) attached at the front of the robot is responsible to detect obstacles and communicate as an input with the microcontroller. It can calculate distance by counting the time its transmitter radiation is reflected and received back in its receiver. The control algorithm processes the signals and afterwards chooses the shortest way available in the arena to capture the prey. The significance of this sensor is high given the fact that the sooner the predator touches the prey, the better the application performance becomes. A light sensitive transistor is responsible for receiving the detection signal that the prey robot transmits. This sensor provides the microcontroller all necessary data in order for the robot to move and adapt its orientation to the prey tracking problem. In case random obstacles get in the way and the transmission signal is obscured, the predator robot keeps moving in the arena until it discovers the prey robot’s signal again. The power supply for all the above is a 5000mAh lithium rechargeable battery that provides enough energy for long term operations of the robot. It is placed between the DC motors (lowest level of the platform) and the ATMEL 1051 microcontroller board (middle level of the platform) and is exactly the same with the prey robot’s battery. The goal of keeping the overall cost of the predator robot below 150 Euros including hardware, sensors, actuators and microcontroller was achieved.

3 Robot Kinematics and Performing Scenario

3.1 2-D Kinematics of robot

Both robots have the same moving philosophy. They employ a two-wheel differential drive structure to achieve mobility in two dimensional (2-D) space, as seen in Figure 6. This architecture provides the platform the ability to move forward and backward by directly controlling the speed of the right and left DC motors, each responsible for right and left wheel respectively, as indicated by researchers, Borenstein, Everett, Feng and Wehe, Dhaouadi and Sleiman, Joo Er and Deng, Yim, Duff and Roufas. Each DC motor of the robots is supplied by a PWM voltage signal amplified by an H-Bridge integrated circuitry.

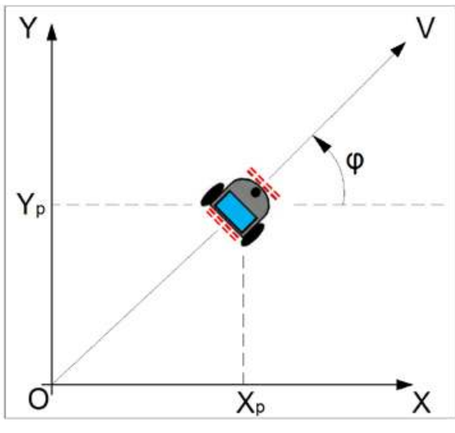

Figure 6. The robotic platform dynamics

In order to clarify the above figure, some more mathematics should be illustrated that describe the robot posture, like:

$$\begin{bmatrix} \dot{X}_p \\ \dot{Y}_p \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} V \\ \omega \end{bmatrix} = F(\varphi)u \quad (1)$$

where,

- V is the linear velocity of the robot
- ω is the rotational velocity
- φ is the orientation of the robot
- X_p and Y_p are the coordinates of the center of the mass of the robot
- the vector u is the control command depending on the right and left wheels speeds

The linear and rotational velocities of both platforms can be obtained from the right and left wheels velocities, according to the next functions:

$$\omega = \frac{d\varphi}{dt} = \frac{V_R - V_L}{b} \quad (2)$$

And

$$V = \frac{V_R - V_L}{2} \quad (3)$$

Where,

- V_R is the velocity of the right wheel
- V_L is the velocity of the left wheel
- b is the distance between the centres of the right and left wheels

All the above, are included in the different control algorithms that each robot is executing in its microcontroller and all sensors and actuators are collaborating harmonically in order to achieve the task of the application. A part of the controller's programming code indicating only the base loops of it for illustration purposes, is shown in Figure 7 that represents a form of a flow chart.

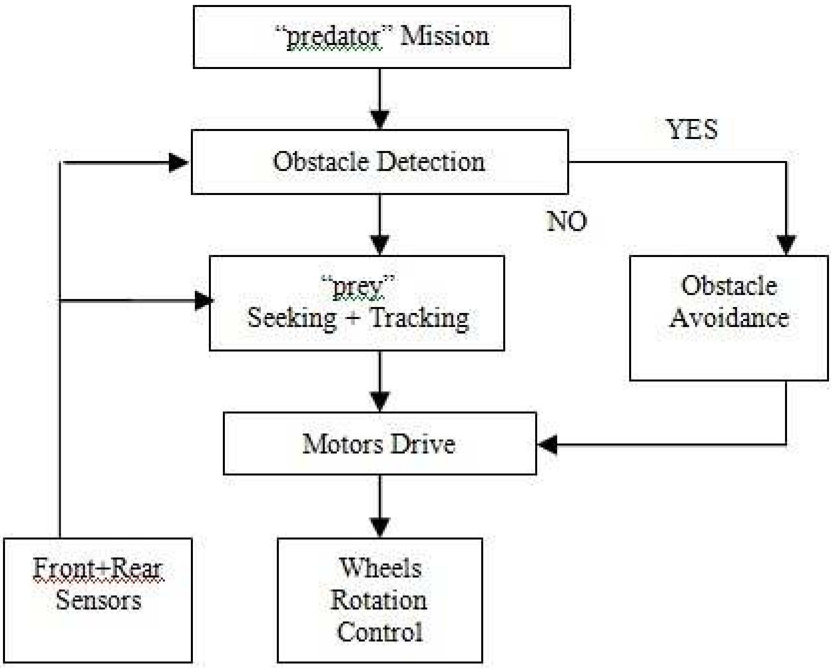

Figure 7. The control algorithm flow chart overview

At this stage, it was considered as a nice idea to include the above kinematic functions and the flow chart conditions in a simulation environment and check out the outcome. The so-called MatLab Simulink was chosen for the task and therefore, the ‘prey’ robot model was built by using the simulation software appropriate tools. The ATMEGA 2560 microprocessor of the robot has command libraries, which are compatible with MatLab so that the communication between the robot and the software was safely established. The input of the simulation model is the commands of the user in order to navigate on the arena surface and avoid the obstacles. A joystick device was chosen for an alternative choice of the mobile phone navigator in an attempt to enchase the system’s versatility. The output of the simulation model is the control signals from the microprocessor to the right wheel motor and the left wheel motor respectively. These control signals are in fact responsible for all possible movements that the robot can achieve like, move forward, move backward, turn right, turn left, etc. An illustration of the simulation model in the MatLab Simulink environment is shown in figure 9.

In such an engineering simulation environment, many feedback signals can be monitored and analysed. It was decided to process the current data in amperes that each motor consumes because this information is critical when, for example, choosing the batteries of the robot. In figure 8, the current consumption in real time for both robots is provided and some comments can be made. Current is expected to increase rapidly at the beginning of the robot motion and stabilise in almost constant values thereafter. This is confirmed by the combined current graph of figure 8 since from 0.8sec to 1sec, from 3.2sec to 3,5sec and from 4,2sec to 4,5sec one can notice ‘spikes’ in current values and smooth waveforms in all other time intervals.

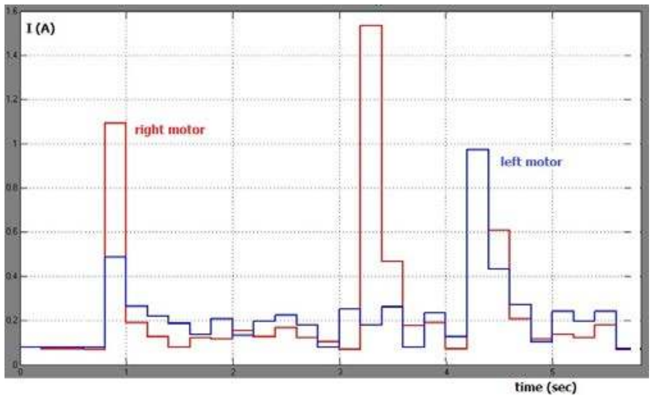

Figure 8. The control algorithm flow chart overview

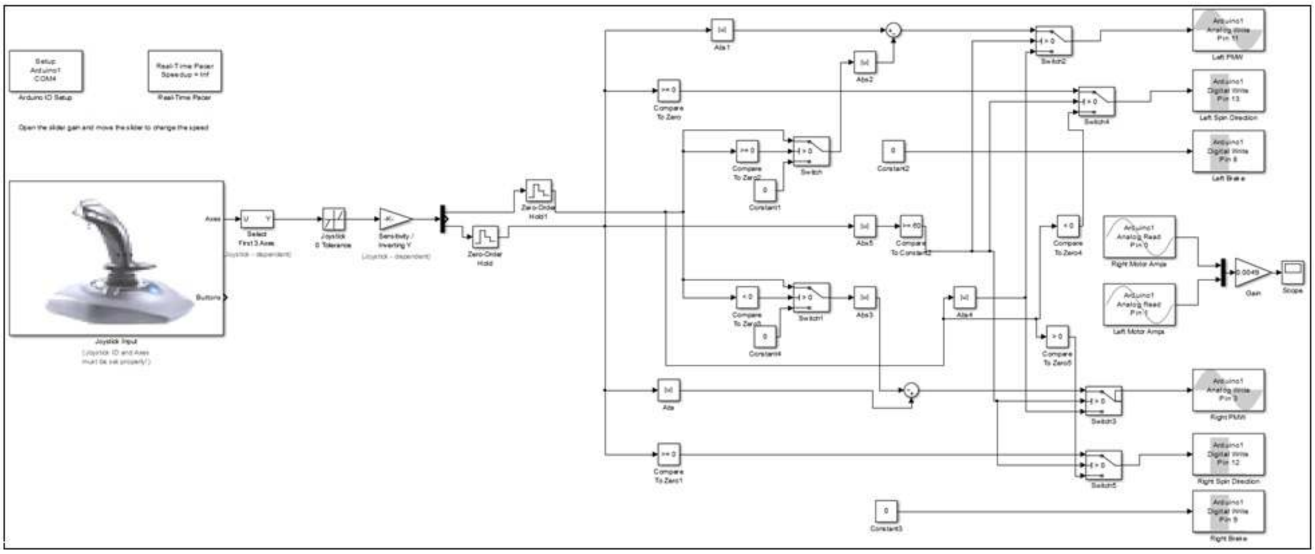

Figure 9. The control algorithm flow chart overview

3.2 The 1st scenario of the "predator-prey" acting

The pair of robots is supposed to be acting like the predator and the prey. In other words, the robot that carries off the prey role moves within the restricted arena, which is especially prepared for this task, while at the same time it avoids obstacles of random number and size. Using its sensors and moving under its control command, the prey robot manages to stay constantly within the arena boundaries as well as transmitting the identification signal explained earlier in this paper. At this point, the predator robot is placed by the user in a random point of the arena and automatically it starts seeking and tracking for the prey. It moves continuously amongst the obstacles and within the arena boundaries until it discovers the transmitting signal of the prey. Then it focuses its moving orientation in tracking the prey, it follows it and finally touches it (get closer than 10mm) as quickly as possible, which is the goal of the application and the end of performing. Having accomplished a significant number of experimentation with this control algorithm of the predator, it can be stated that the necessary time for tracking the prey decreases as the code/algorithm is improved. At the final stage of this research project, the tracking time was recorded between 4 to 7 seconds, running both robots on fully charged batteries. This time variation depends on the number of the random obstacles located on the arena surface as well as the initial distance between the two robots. A representative figure of the arena with random obstacles and the pair of robots placed on it can be seen below, in figure 10.

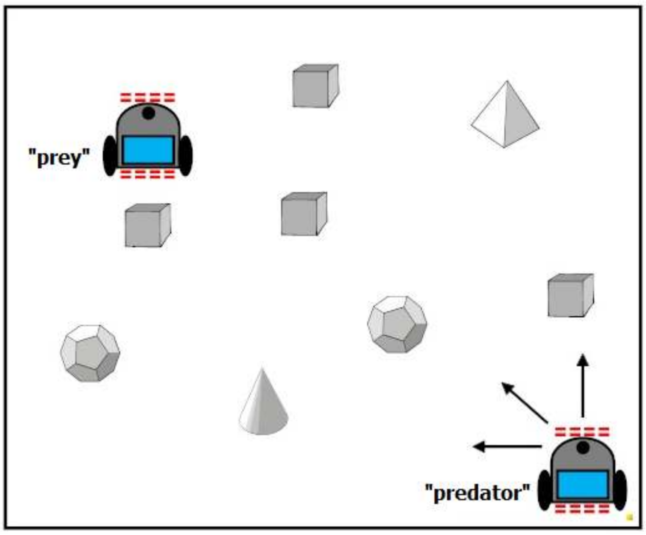

Figure 10. The $4m^2$ arena representation of the application

At this stage it must be highlighted that the prey which moves under the user's commands via Bluetooth communication, cannot overcome the arena boundaries which can be either a black line on a white arena surface or vertically placed small wall. The remote control of the prey robot ensures the high level of tracking difficulty while at the same time the control algorithm running on the prey robot's microcontroller does not allow obstacles to be hit or arena limitations to be overcome even if the user desires that. The predator-prey pair of robots can be viewed in the following figure:

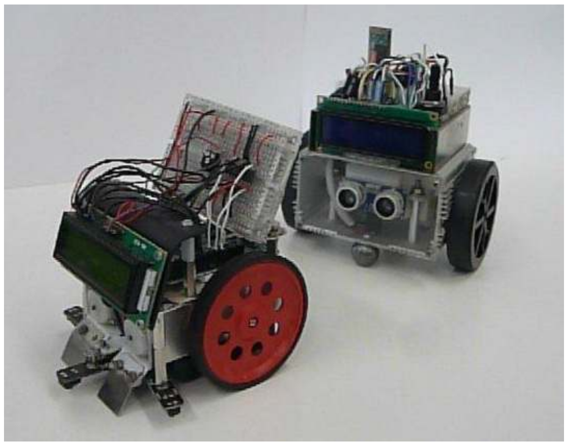

Figure 11. The predator-prey pair of robots

3.3 The 2nd scenario of the "predator-prey" acting

Within the robotics engineering domain, the researcher should not stay focused only in one possible solution of a problem, but should always test various techniques and try to improve the performance and the efficiency of the system under investigation. Therefore, for the application described above, another experiment was designed in order to compare results and furthermore the performance of another algorithm architecture. It was decided to re-program the predator robot and use a kind of "mapping" and movement data collection control structure. The predator robot was left alone without the existence of the prey robot in the same arena among the same obstacles and the updated algorithm design considers the arena area a Cartesian plane. Using its sensors, the predator robot locates all obstacles and registers in its memory coordinates on a (x,y) format starting from (x_1,y_1) to (x_N,y_N), where N represents the number of the random obstacles, as illustrated in figure 12. The biggest N is, the larger the data that need to be memorized become, requiring an extensive use of the microcontroller internal memory like RAM. Supported by the dynamics of robot motion provided by equations (1), (2) and (3) discussed earlier, the microcontroller can estimate the location of the robot continuously.

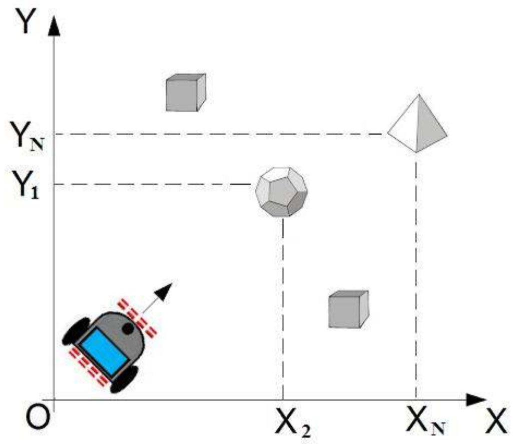

Figure 12. The predator robot "memorising" all obstacle's coordinates

By the time the prey robot appears on the arena, the predator processes all collected data in its microcontroller from the previous "memorizing" phase, recalls them from its RAM and rejects all coordinates that represent the obstacles. The intelligent part of the algorithm is that for the predator robot, at a higher level, all obstacles are "hidden" or do not exist on the arena surface. However, at a lower level, it is able to move amongst them without touching them. All pre-memorized (in a double column format table) 2 dimensions coordinates, referring to the obstacles locations, are running in parallel with the self-moving and tracking the prey task that the microcontroller is undertaking. Therefore, a multitasking process occurs since the predator operates under an auto-selective algorithm scheme for coordinates rejection movement. This large amount of data handling even at lower case level and the fact of continuous real-time processing of them, upgrades the predator's job to a more complex and intelligent one. Although in this scenario of the predator-prey application, a much more complicated control algorithm is executing, the results of adopting this technique are beneficiary enough, while at the same time the construction cost is limited to very low levels, as illustrated earlier. In order to prove the predator robot performance improvement, the time of catching the prey was selected as the key factor of efficiency. It was then recorded that the maneuverings among obstacles in the first scenario is time consuming and it requires more effort for the predator robot to track the prey. During experimentation with the second "intelligent" algorithm, the tracking time was decreased by 1-2 seconds in all cases and although the algorithm itself requires all microcontroller resources for execution, it still navigates the predator faster to track the prey than the first experiment.

A significant restriction of the predator robot performance is the operation range of the supersonic sensor (HC-SR04), which is used for the obstacle avoidance and mapping. The limitation of the 4m range of the sonar sensor was overcame by replacing it with another similarly low cost sensor, the US020, which operates even at 7m distances. The dimensions of the square arena were increased and more random size obstacles were placed on the same surface. It is important to mention that the microprocessor was not influenced from this larger amount of data management hence the task of the application stays within the capabilities of the specific microprocessor. This new experimentation process was designed in an attempt to prove not only the versatility of the robots but also to investigate how the bigger test arena dimensions and the obstacle number increase, affects the overall performance. It must be highlighted at this point that the greater the arena dimensions become and hosts more obstacles, the greater the tracking time for the predator becomes. Throughout all experimentation, an additional delay up to 2-3 seconds was always present in the scenario fulfillment. The delay time of application completion varies because the predator can start its task in randomly selected places on the arena, closer or not to the also randomly placed prey.

3 Conclusions and Future Work

Experimentation with robotics in general is a dynamic and adaptive process, with multiple feedback loops running at different speeds. This is in line with the broad range of benefits that the robotics can bring, for example, to the control engineering education, since they have became very helpful for developing student capabilities of team work technical communication, interdisciplinary work, problem based thinking and more. The key factor of this research was to prove how perfect outcomes can be recorded, by spending less money. It can be understood that this specific application can also be undertaken if the equipment selection criterion is not based on the cost. The differentiation between them arises naturally from the constant need of reducing costs in nowadays research works.

Resuming, in this paper, two integrated robotic platforms were presented. They are performing as a pair in the 'predator-prey' scenario in a limited arena and they manage to fulfill the application's needs. Their build-architecture, although based in low cost solutions, still ensures the efficiency of the platforms for many applications. The multitasking character of the robots is remarkable, considering the easy assembly process required for them. It is simple to alter the capabilities of this pair of robots by adding or removing sensors and other equipment. Thus, the study of motion control system design and analysis, as well as the mobile robot navigation, is encouraged. The addition of board proximity and range sensors allows the detection and avoidance for obstacles of random shape and size. Both algorithms that were used in this application are discussed in this paper and the outcome is presented supported by comparison comments. After a very large number of experiments, which are still being executed for performance improvement, a clear conclusion was recorded. The processing time for all data handled by both microcontrollers is always significantly less than the predator-prey scenario pursuit – time for the two robots. The fact that algorithms 'run' faster than the time that is required for the mission completion, adds another proof for the algorithm efficiency. In addition, the intelligent format of the second algorithm is proven to improve the predator robot's performance by reducing the required time for tracking the prey robot.

The huge availability of additional sensors nowadays, opens the door for developing motion control algorithms, which allow prospective researchers to learn robots how to navigate in an autonomous way with minimum user intervention. In addition to this, the involvement of machine learning algorithms can be considered, and many more experimentation can take place in order to predict the movements of the robots and further reduce the tracking time. More specifically, new sensors can be attached to the predator robot in order to view more than a single obstacle at the same time. The updated algorithm, which will be based on machine learning methods, will make the robot pre-decide how to navigate between obstacles by checking all possible routes among obstacles and execute the optimum-shortest one. The last task represents a big challenge that follows and the research team of this project will undertake it in the short-term future.

References:

[1] Amerongen J.V. and Jongkind W., "Mechatronics in the Netherlands," IEEE/ASME Transactions on Mechatronics, pp. 106-110, 1991.

[2] Borenstein J., Everett H., Feng L. and Wehe D., "Mobile robot positioning: sensors and techniques", Journal of Robotic Systems, vol. 14, No 4, April 1997

[3] Brussel H.V., "Mechatronics – a powerful concurrent engineering framework," IEEE-ASME Transactions, vol. 1, pp. 127-127, 1996.

[4] Chamilothoris G. and Papoutsidakis M., "Shaping the Mechatronics Courses for the Control Curriculum," The 16th IFAC World Congress, Prague, 2005.

[5] Chamilothoris G., "Development of a Mechatronics Curriculum in Greece", Proceedings of IEE-IME Conference on Mechatronics, pp131-136, Enschede, 2002

[6] Chamilothoris G. and Voliotis I., “A low-cost platform for mechatronics education in Greece”, International Workshop in Research and Education in Mechatronics, Bochum, Germany, 2003.

[7] Dhaouadi R. and Sleiman M., “Development of a Modular Mobile Robot Platform”, IEEE Industrial Electronics Magazine, vol.5, No 4, December 2011.

[8] Joo Er M. and Deng C., “Obstacle Avoidance of a mobile robot using hybrid learning approach”, IEEE Transactions of Industrial Electronics, vol. 52, No 3, June 2005

[9] Mataric M., Koenig N. and Feil-Seifer D., “Materials for Enabling Hands-On Robotics and STEM Education”, AAAI Spring Symposium on Robots and Robot Venues: Resourses for AI Education, Stanford, CA, March 2007.

[10] Piepmeier A., Bishop B. and Knowles K., “Modern robotics engineering instruction”, IEEE Robotics and Automation Magazine, vol.10, No 2, June 2003.

[11] Yim M., Duff D. and Roufas K., “PolyBot: a modular reconfigurable robot”, International Conference on Robotics and Automation, IEEE, California, USA, 2000.

[12] Yu X. and Weinberg J., “Robotics in education: New platforms and environments”, IEEE Robotics and Education Magazine, 10(3), September 2003.

[13] Yu X. and Weinberg J., “Robotics in education: New platforms and environments”, IEEE Robotics and Education Magazine, 10(3), September 2003.

[14] Obaid Ur-Rehman, Natasa Zivic, ‘Fuzzy Authentication Algorithm with Applications to Error Localization and Correction of Images’, WSEAS Transactions on Systems, Issue 8, Volume 12, August 2013

[15] Christos K. Volos, Nikolaos Bardis, Ioannis M. Kyprianidis, Ioannis N. Stouboulos, ‘Motion Control of a Mobile Robot Based on Double-Scroll Chaotic Circuits’, WSEAS Transactions on Systems, Issue 9, Volume 11, September 2012

[16] Adrian Korodi, Marian Corman, ‘Wheeled Mobile Robot Model and Cooperative Formation Control’, WSEAS Transactions on Systems, Issue 11, Volume 11, November 2012

[17] Hua-Ching Chen, Dong-Hui Guo, ‘Fuzzy Embedded Mobile Robot Systems Design through the Evolutionary PSO Learning Algorithm’, WSEAS Transactions on Systems , Issue 8, Volume 10, August 2011

[18] Haitian Wang, Ge Li, ‘Motion Control and Trajectory Tracking Control for a Mobile Robot Via Disturbance Observer’, WSEAS Transactions on Systems, Issue 1, Volume 9, January 2010

[19] Michail Papoutsidakis, Dimitrios Piromalis and George Chamilothoris, ‘Low Cost Swarm Robotic Platforms Operating with Open-Source Software for Cooperative Applications’, 14th IEEE International Symposium on Computational Intelligence and Informatics, November 2013, Budapest, Hungary

[20] Hájek, P., Neri, F., "An introduction to the special issue on computational techniques for trading systems, time series forecasting, stock market modeling, financial assets modeling", WSEAS Transactions on Business and Economics, 10 (4), pp. 201-292, 2013.

[21] Azzouzi, M., Neri, F., "An introduction to the special issue on advanced control of energy systems", WSEAS Transactions on Power Systems, 8 (3), p. 103, 2013.

[22] Bojkovic, Z., Neri, F., "An introduction to the special issue on advances on interactive multimedia systems", WSEAS Transactions on Systems, 12 (7), pp. 337-338, 2013.

[23] Pekař, L., Neri, F., "An introduction to the special issue on advanced control methods: Theory and application", WSEAS Transactions on Systems, 12 (6), pp. 301-303, 2013.

[24] Guarnaccia, C., Neri, F., An introduction to the special issue on recent methods on physical polluting agents and environment modeling and simulation WSEAS Transactions on Systems, 12 (2), pp. 53-54, 2013.

[25] Neri, F., "An introduction to the special issue on computational techniques for trading systems, time series forecasting, stock market modeling, and financial assets modeling", WSEAS Transactions on Systems, 11 (12), pp. 659-660, 2012.

[26] Muntean, M., Neri, F., "Foreword to the special issue on collaborative systems", WSEAS Transactions on Systems, 11 (11), p. 617, 2012.

[27] Pekař, L., Neri, F., "An introduction to the special issue on time delay systems: Modelling, identification, stability, control and applications", WSEAS Transactions on Systems, 11 (10), pp. 539-540, 2012.

Volos, C., Neri, F., "An introduction to the special issue: Recent advances in defense systems: Applications, methodology, technology" WSEAS Transactions on Systems, 11 (9), pp. 477-478, 2012.