

Sigeru Omatu
José Neves
Juan M. Corchado Rodríguez
Juan F. De Paz Santana
Sara Rodríguez González (Eds.)

Distributed Computing and Artificial Intelligence

10th International Conference

 Springer

Editors

Sigeru Omatu
Graduate School of Engineering
Osaka Prefecture University
Osaka
Japan

José Neves
Department of Computing Science
University of Minho
Braga Codex
Portugal

Juan M. Corchado Rodríguez
Department of Computing Science and
Control
Faculty of Science
University of Salamanca
Salamanca
Spain

Juan F. De Paz Santana
Department of Computing Science and
Control
Faculty of Science
University of Salamanca
Salamanca
Spain

Sara Rodríguez González
Department of Computing Science and
Control
Faculty of Science
University of Salamanca
Salamanca
Spain

ISSN 2194-5357
ISBN 978-3-319-00550-8
DOI 10.1007/978-3-319-00551-5
Springer Cham Heidelberg New York Dordrecht London

ISSN 2194-5365 (electronic)
ISBN 978-3-319-00551-5 (eBook)

Library of Congress Control Number: 2013937503

© Springer International Publishing Switzerland 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The International Symposium on Intelligent Systems and Applications (ISA'13) is a forum for solving complex problems in distributed environments, such as environment monitoring, manufacturing, computing, to mention only a few. It is a high added value with scientific and research. These technical research and technical efforts and exchange of ideas between the industry sector is essential to meet the ever-increasing demands of the experience, current work in computing, artificial intelligence solutions to real problems.

This conference is a stimulating community can work towards the future of Intelligence areas. Nowadays, the premier conferences in computing, artificial intelligence. This year's technical program contributions in well-established papers were submitted from all over the world, including the United States, Finland, France, Germany, Italy, Japan, Mexico, Morocco, Poland, Portugal, Russia, and South Africa. This is a truly "wide area network" conference. The program committee has selected 75 papers. As a result, the Journal of Artificial Intelligence, Robotics (IJIR) and the International Journal of Intelligent Systems (IJIS) and the International Journal of Intelligent Systems (IJIS) are the most highly regarded world

Automatic Prediction of Poisonous Mushrooms by Connectionist Systems	341
<i>María Navarro Cáceres, María Angélica González Arrieta</i>	
Adaptive Learning in Games: Defining Profiles of Competitor Players	351
<i>Tiago Pinto, Zita Vale</i>	
Bioinformatics, Biomedical Systems, e-health, Ambient Assisting Living	
Associative Learning for Enhancing Autonomous Bots in Videogame Design	361
<i>Sergio Moreno, Manuel G. Bedía, Francisco J. Serón, Luis Fernando Castillo, Gustavo Isaza</i>	
An Integral System Based on Open Organization of Agents for Improving the Labour Inclusion of Disabled People	369
<i>Alejandro Sánchez, Carolina Zato, Gabriel Villarrubia-González, Javier Bajo, Juan Francisco De Paz</i>	
+Cloud: An Agent-Based Cloud Computing Platform	377
<i>Roberto González, Daniel Hernández, Fernando De la Prieta, Ana Belén Gil</i>	
Representation of Propositional Data for Collaborative Filtering	385
<i>Andrzej Szwabe, Pawel Misiorek, Michal Ciesielczyk</i>	
Face Identification by Real-Time Connectionist System	393
<i>Pedro Galdámez, Angélica González</i>	
QoS Synchronization of Web Services: A Multi Agent-Based Model	401
<i>Jaber Kouki, Walid Chainbi, Khaled Ghedira</i>	
Data mining, Information Extraction, Semantic, Knowledge Representation	
Comparing Basic Design Options for Management Accounting Systems with an Agent-Based Simulation	409
<i>Friederike Wall</i>	
Potential Norms Detection in Social Agent Societies	419
<i>Moamin A. Mahmoud, Aida Mustapha, Mohd Sharifuddin Ahmad, Azhana Ahmad, Mohd Zaliman M. Yusoff, Nurzeatul Hamimah Abdul Hamid</i>	
Collision Avoidance of Mobile Robots Using Multi-Agent Systems	429
<i>Angel Soriano, Enrique J. Bernabeu, Angel Valera, Marina Vallés</i>	

Collision Avoidance of Mobile Robots Using Multi-Agent Systems

Angel Soriano, Enrique J. Bernabeu, Angel Valera, and Marina Vallés

Instituto U. de Automática e Informática Industrial,
Universitat Politècnica de Valencia, Camino de Vera s/n
46022 Valencia, Spain
{ansovi, ebernabe, giuprog, mvalles}@ai2.upv.es

Abstract. This paper presents a new methodical approach to the problem of collision avoidance of mobile robots taking advantages of multi-agents systems to deliver solutions that benefit the whole system. The proposed method has the next phases: collision detection, obstacle identification, negotiation and collision avoidance. In addition of simulations with virtual robots, in order to validate the proposed algorithm, an implementation with real mobile robots has been developed. The robots are based on Lego NXT, and they are equipped with a ring of proximity sensors for the collisions detections. The platform for the implementation and management of the multi-agent system is JADE.

1 Introduction

The area of artificial intelligence (AI) has expanded considerably in recent years. It not only dominates the area of games versus computers, but nowadays it applies in many sectors like databases management or web pages. As it is well known, the main topic of AI is the concept of intelligent agent defined as an autonomous entity which observes through sensors and acts upon an environment using actuators [20]. This definition is very close to services that a robot can provide, so the concept of agent often is related with robots, [4], [27], [16].

On the other hand, detecting and avoiding a collision is a previous step for overcoming the motion planning problem. In fact, collision detection has been inherently connected with the motion-planning algorithms from the very beginning. Current planning algorithms require the collision detection of mobile and nondeterministic obstacles.

Continuous collision detection (CCD) techniques are the most effective when dealing with multiple mobile agents or robots. CCD algorithms basically make a return if a collision between the motion of two given objects is presented or not; and if a collision is going to occur then, the instant in time of the first contact is returned, [2], [3], [7], [19], and [26].

In this paper, collision detection strategies of autonomous mobile robots based on [3] are combined with strategies based on artificial intelligence to offer a new method of collision avoiding management.

The method is divided into three basic concepts which are merged in this paper: obstacle detection by a mobile robot, the concept of abstraction robotic agent as a software agent within MAS, and distributed artificial intelligence as a method of communication and negotiation between these software agents.

Nowadays, there are many sensors on the market that allow robots to know if there is an obstacle that stands between them and its trajectory, and where is that obstacle. This process is usually local, i.e. it is performed inside a robot. In the case of two mobile robots at the same scenario, each one represents an obstacle to the other, but neither is aware of it because it is handled as a local process. The concept of robotic agent in a multi-agent robotic system is proposed as a next level or upper layer to fix it and to manage a more intelligent solution.

Multi-agent robot systems (MARS) represent a complex distributed system, consisting of a large number of agents-robots cooperating for solving a common task. In this case, each agent of MARS represents a real physical mobile robot that informs its software agent of all it perceives. The ability of communication, cooperation and coordination between the agents, allows conversations, negotiations and agreements, which are the basis of the algorithm is presented.

2 Avoiding Collision Method

The aim of this section is reviewing a CCD methodology in [3] for obtaining the instant in time when two robots or agents in motion will be located at their maximum-approach positions while they are following straight-line trajectories.

The mentioned maximum approach is also calculated. Therefore, if the involved robots do not collide while they are following their respective motions, then their minimum separation is returned. Otherwise, their maximum penetration is computed as a minimum translational distance [5]. A remarkable aspect is that both the instant in time and the corresponding minimum separation or maximum approach are computed without stepping any involved trajectory.

Some collision avoiding configurations for the involved robots or agents are directly generated from the computed instant in time and maximum penetration. These collision-free configurations are determined in accordance with a given coordination between the robots or agents.

2.1 Obtaining the Instant in Time and the Maximum Approach

Consider two robots or agents in motion each one enveloped or modeled by a circle. Let A be a circle in motion whose start position at time t_s is $A(t_s)=(c_A(t_s),r_A)$. Where $c_A(t_s) \in \mathfrak{R}^2$ is the A 's center at t_s and $r_A \in \mathfrak{R}$ is its radius. A is following a straight-line trajectory whose final position at t_g is given by $A(t_g)=(c_A(t_g),r_A)$. Let $v_A \in \mathfrak{R}^2$ be the A 's velocity for the time span $[t_s, t_g]$.

Let B be a second circle in motion whose start and goal positions at the respective instants in time t_s and t_g are $B(t_s)=(c_B(t_s),r_B)$ and $B(t_g)=(c_B(t_g),r_B)$. The B 's velocity for the time span $[t_s, t_g]$ is $v_B \in \mathfrak{R}^2$.

All the infinite interme
is in motion is parameteri

$$A(\lambda)=(c_A(\lambda),r_A):c_A(\lambda)=c_0 + \lambda(c_1 - c_0)$$

Note that the positions A
and $\lambda \in [0,1]$. All the inf
analogously parameterize

Observing equation (1)
between in-motion circle
 $\lambda_c \in [0,1]$ that minimizes $\|$

Let $\lambda_c \in [0,1]$ be the par
by computing the distance

Note that $c_A(\lambda)-c_B(\lambda)$ f
are respectively $c_0=c_A(t_s)-$
is obtained by projecting

$$\lambda_c = \frac{c_0 \cdot (c_1 - c_0)}{\|c_1 - c_0\|^2}$$

Once λ_c is obtained, d_M an

$$d_M = \|c_A(\lambda_c) - c_B(\lambda_c)\|$$

If d_M is negative, then A
collide with the maximum

2.2 Determining A

In case of collision, th
penetration are, as mentio
 $t_M \in [t_s, t_g]$. One of these po
circles into contact by usi

Let $A_f(t_M)$ and $B_f(t_M)$ be

$$A_f(t_M) = (c_{Af}(t_M), r_A)$$

$$B_f(t_M) = (c_{Bf}(t_M), r_B)$$

where $c_A(t_M)=c_A(t_s)+\lambda_c(c_A(t_g)-c_A(t_s))$
is a safety threshold. If
contact. Finally, paramete
applied to each mobile ro
are equal and, consequen
motion. A graphical exam

All the infinite intermediate positions of the mobile circle A for $t \in [t_s, t_g]$ while A is in motion is parameterized by λ with $\lambda \in [0, 1]$, as follows:

$$A(\lambda) = \{ (c_A(\lambda), r_A) : c_A(\lambda) = c_A(t_s) + \lambda \cdot (c_A(t_g) - c_A(t_s)) ; t = t_s + \lambda(t_g - t_s) ; \forall \lambda \in [0, 1] \} \quad (1)$$

Note that the positions $A(\lambda)$ and $A(t)$, with $t = t_s + \lambda(t_g - t_s)$, are equal for all $t \in [t_s, t_g]$ and $\lambda \in [0, 1]$. All the infinite intermediate positions of the mobile circle B are analogously parameterized for $\lambda \in [0, 1]$ as indicated in (1).

Observing equation (1) is easy to conclude that the maximum approach d_M between in-motion circles A and B will be obtained by finding the parameter $\lambda_c \in [0, 1]$ that minimizes $\|c_A(\lambda) - c_B(\lambda)\| - (r_A + r_B)$

Let $\lambda_c \in [0, 1]$ be the parameter that minimizes (2). λ_c is obtained by minimizing by computing the distance from the origin point O to the straight-line $c_A(\lambda) - c_B(\lambda)$.

Note that $c_A(\lambda) - c_B(\lambda)$ for all $\lambda \in [0, 1]$ is really a segment whose extreme points are respectively $c_0 = c_A(t_s) - c_B(t_s)$ and $c_1 = c_A(t_g) - c_B(t_g)$. Then, the parameter $\lambda_c \in [0, 1]$ is obtained by projecting O onto mentioned segment, O^\perp , as

$$\lambda_c = \frac{c_0 \cdot (c_1 - c_0)}{\|c_1 - c_0\|^2} \quad \text{with } \lambda_c \in [0, 1]. \quad \text{Then } O^\perp = c_0 + \lambda_c \cdot (c_1 - c_0). \quad (2)$$

Once λ_c is obtained, d_M and the associated instant in time t_M are computed as

$$d_M = \|c_A(\lambda_c) - c_B(\lambda_c)\| - (r_A + r_B) \quad t_M = t_s + \lambda_c(t_g - t_s). \quad (3)$$

If d_M is negative, then d_M holds a penetration distance and, then A and B will collide with the maximum penetration d_M at t_M .

2.2 Determining Avoiding Collision Configurations

In case of collision, the positions where A and B present their maximum penetration are, as mentioned, $c_A(\lambda_c)$ and $c_B(\lambda_c)$ respectively with $\lambda_c \in [0, 1]$, $d_M < 0$, $t_M \in [t_s, t_g]$. One of these positions can be minimally translated in order to bring both circles into contact by using the unit vector $\hat{v}_{MTD} = O^\perp / \|O^\perp\|$, with $\|\hat{v}_{MTD}\| = 1$,

Let $A_f(t_M)$ and $B_f(t_M)$ be the mention collision-free configurations,

$$\begin{aligned} A_f(t_M) &= (c_{Af}(t_M), r_A) : c_{Af}(t_M) = c_A(t_M) - \delta \cdot \alpha \cdot d_M \cdot \hat{v}_{MTD} \\ B_f(t_M) &= (c_{Bf}(t_M), r_B) : c_{Bf}(t_M) = c_B(t_M) + \delta \cdot (1 - \alpha) \cdot d_M \cdot \hat{v}_{MTD} . \end{aligned} \quad (4)$$

where $c_A(t_M) = c_A(t_s) + \lambda_c(c_A(t_g) - c_A(t_s))$ and $c_B(t_M) = c_B(t_s) + \lambda_c(c_B(t_g) - c_B(t_s))$. Parameter $\delta \geq 1$ is a safety threshold. If $\delta = 1$, then configurations $c_{Af}(t_M)$ and $c_{Bf}(t_M)$ will be in contact. Finally, parameter $\alpha \in [0, 1]$ configures the degree of motion modification applied to each mobile robot or agent. In this way, if $\alpha = 1$, then $c_B(t_M)$ and $c_{Bf}(t_M)$ are equal and, consequently, mobile robot or agent B do not change its current motion. A graphical example is shown in Fig. 1.

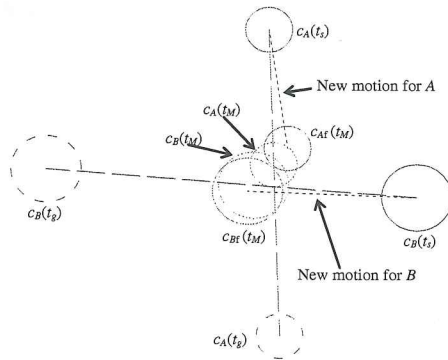


Fig. 1 Avoiding collision configurations with $\alpha=0.7$ and $\delta=1.03$

3 Hybrid Control Collision Avoidance

The implementation of the collision avoidance proposed methodology has six phases (see Fig. 2). A scenario where multiple robots follow a path infinite straight line between two target points is considered. These two points are alternated when they are achieved. All robots have their representation as a software agent in the MAS which encompasses the whole system, so there is no moving object within the scene that is not a software agent.

In phase 1 (**detection**), the local system (each robot) has defined a detection object area. If the local system detects an obstacle that may be a threat of collision (from now threat-object), it calculates the position of threat-object in the global scenario and it sends to the agent who represents the local system in MAS to manage it.

When an agent receives the position of a threat-object (from now threat-position) by the local system, it must identify what kind of threat it is. To know this, in the **obstacle identification** phase, the agent detects the threat (from now detector-agent), consults the other agents to know who is located within that area of threat. If there is not any agent within that area it is identified as a static object threat and directly the collision detection phase is performed. Otherwise, the threat-agent is identified through communication among agents and the next phase starts.

When the two involved agents in a possible threat have been identified, the communication between them is used to obtain the information needed to apply the detection algorithm presented in 2.1. In the **time to talk, negotiate and resolve phase**, the inputs of the algorithm are four: the positions of each of the agents involved in that instant ($c_A(t_s)$, $c_B(t_s)$) and the target positions where they will be at time t_g ($c_A(t_g)$, $c_B(t_g)$). This time t_g must be the same for the two robots therefore, to calculate it, the agents communicate to each other to know which one reaches its destination before. Who plans to take more time to reach their destination calculates an intermediate destination from its current trajectory and

Fig. 2 Phases of the proposed

the arrived time of the o
shared the time it takes to

In the **collision detection**
detection algorithm are: c
destination, ($c_A(t_g)$), curr
($c_B(t_g)$). If in the Phase 2
target is the same as the
applied to the algorithm a
object. In case there is no
a collision is detected, the
penetration (t_M). The next
section 2.2. If the object i
entire cost of the collision
negotiations between the
charge is allocated to each

To decide the load pe
avoidance, in the **negotia**
and exchange parameter
maximum speed, etc., wh
offers to change its traject

selection of α , the detecto
The detector-agent, by 1
robot should be achieve :

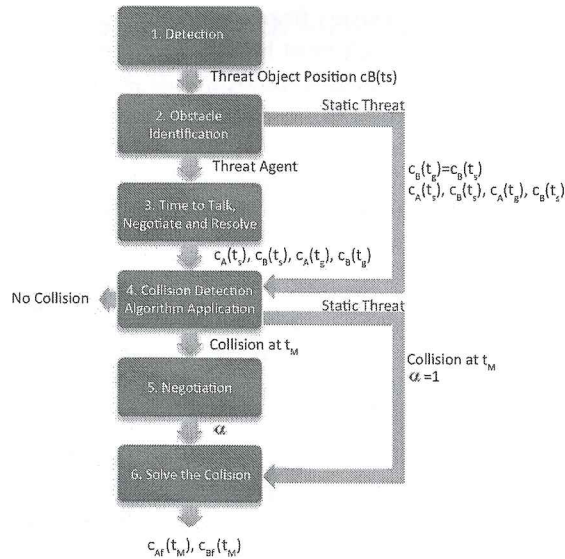


Fig. 2 Phases of the proposed methodology

the arrived time of the other agent to its destination. In this way the two agents shared the time it takes to reach their destination.

In the **collision detection phase**, the input requirements to implement collision detection algorithm are: current position coordinates of detector-agent ($c_A(t_s)$), its destination, ($c_A(t_g)$), current position of threat-object ($c_B(t_s)$) and its destination ($c_B(t_g)$). If in the Phase 2, the threat-object was identified as a static-threat, the target is the same as the initial position ($c_B(t_s) = c_B(t_g)$). Therefore the inputs are applied to the algorithm and it returns the probability of collision with the threat-object. In case there is no collision, threat is discarded and the method ends but if a collision is detected, the method informs to detector-agent the time of maximum penetration (t_M). The next step is to avoid the collision by the method described in section 2.2. If the object is a static-threat, the detector-agent should take over the entire cost of the collision avoidance ($\alpha=1$) and jump to Phase 6. Otherwise the negotiations between the two agents involved are opened to decide how much charge is allocated to each.

To decide the load percentage (α) that each robot will have in the collision avoidance, in the **negotiation phase** the two agents communicate with each other and exchange parameters such as priority, the difficulty of maneuvering, maximum speed, etc., which define the easiness or availability that each agent offers to change its trajectory and avoid collision. Once each agent agreed with the selection of α , the detector-agent runs the last method described below.

The detector-agent, by the method 2.2, computes the two new positions that the robot should be achieve at time t_M to avoid collision in the **solve the collision**

phase. The threat-agent receives, from the detector-agent, the avoidance position ($c_{Bf}(t_M)$) and the time in which must be achieved. Both change their trajectories to go to the new destination partial ($c_{Af}(t_M), c_{Bf}(t_M)$) at the right time. Once it's reached, the collision is resolved, each robot continues its original path and the method ends.

The method only involves two agents in the agreement. When a third object intervenes, for now the scheme of Fig. 3 is performed.

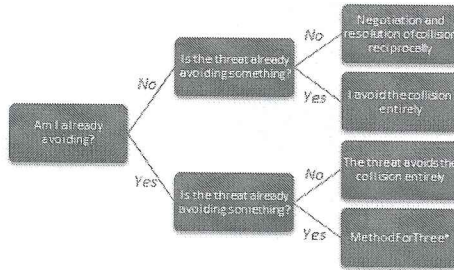


Fig. 3 Performance binary tree

The critical point occurs when an agent (a1) which is already avoiding a collision with other agent (a2) detect that exists a new collision with a third agent (a3), who is already avoiding other agent too. MethodForThree* is a provisional method that progressively reduces the speed of the agents a1 and a2, and may even stop them completely until a3 is no longer a threat to a1. Collision detection by agreements involving more than two agents is also being developed.

In order to test the effectiveness of this method, different scenarios with mobile robots have been simulated. Figures 4 shows the execution obtained in two instants ($t=3s, t=10s$). There are six robots (circles of different colors) that they must arrive to the opposite location (marked by the same color star). The figure also shows the detection area (a trapezoid in front of each robot), three static objects (black squares) and the path described by each robot for the first seconds of the simulation.

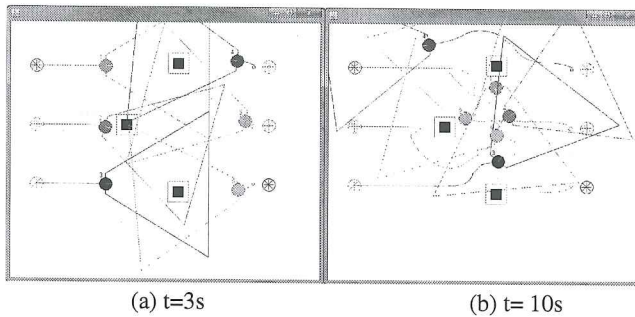


Fig. 4 Collision avoidance simulation with six robots

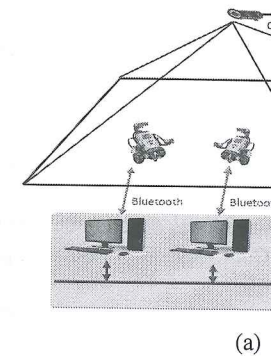


Fig. 5 Control architecture

4 Practical Imp

A practical implementation of the algorithm was tested using the robustness of the platform Mindstorms NXT [12] and the JADE [11].

Two LEGO Mindstorms NXT robots have been defined with two destinations. The strategy is based on a pure negotiation. The robots have been equipped with sensors to detect obstacles.

Robots are connected to two computers. Those computers are part of a multi-agent system. The connection diagram shows how the robots detect its position from a camera. A camera is also used to monitor the environment.

These robots have multiple modules. One of them has one module that manages the IR sensors and sends information to the software.

While IR sensors detect obstacles, they also inform the software about the path. If the solution leads to a collision, the module receives the new path.

The management of the robots receives the position of the threat agent and they execute the path. If a collision is detected, they send the new destinations.

4 Practical Implementation with Mobile Robots

A practical implementation with mobile robots has been developed in order to test the robustness of the presented algorithm. The mobile robots used are LEGO Mindstorms NXT [12] and the platform for the management of MAS chosen was JADE [11].

Two LEGO differential wheeled mobile robots have been built [9]. Each robot has defined two destinations points. In order to achieve the trajectory, a control strategy based on a pure pursuit algorithm [28] was implemented in the robots. The robots have been equipped with a ring of proximity sensors to detect possible obstacles.

Robots are connected to their software agents (computers) via Bluetooth and those computers are part of a network that forms the overall MAS through JADE. The connection diagram is presented in Fig. 5a. Each robot carries a triangle to detect its position from an overhead camera located at the top of the scenario. This camera is also used to monitoring and minimizing odometry problems.

These robots have multiple threads running different functional modules. Each of them has one module to control the robot trajectory, a second one for detection that manages the IR sensors and a third one for communication that receives and sends information to the software agent (see Fig. 5b).

While IR sensors does not detect anything, the robot follows its fixed trajectory, but when something is detected by IR, the communication module informs to the software agent and expects a solution to the possible collision from MAS. If the solution leads to a new destination for the robot, the communication module receives the new destination and sends it to the control module for change the path.

The management of the agents in JADE is simple. When a software agent receives the position of a detected threat, the agent asks everyone if anyone is located in the threat area. Thus, if other robot is the threat, it's identified as the threat agent and they exchange their destinations and speeds to verify if there will be a collision or not. If finally there is it, they negotiate the way to avoid it and send the new destinations and speeds to their respective robots.

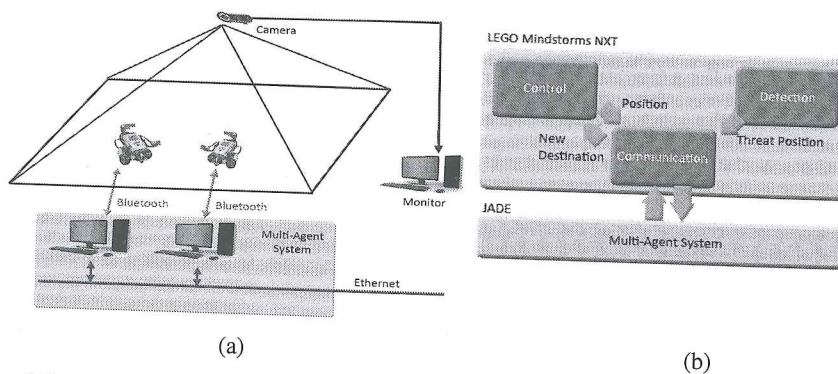


Fig. 5 Control architecture and modules connection scheme

In <http://idecona.ai2.upv.es>, a video demonstration of practical experiment with Lego robots (*Robots Móviles* folder, at videos multimedia gallery) and two compiled versions of the platform that allow the simulation with robots (*Desarrollos de Software* folder, at Results option, Project menu) can be obtained.

5 Conclusions

A collision avoidance method that takes advantages and benefits of MAS has been presented in this work. This method is located one level above the traditional methods of obstacle avoidance where the management is performed locally and the possible communications between the local systems are solved functionally. The application of techniques provided by the area of artificial intelligence to the robotic area opens a wide range of possibilities that offers more natural results and gives human characteristics of communication like negotiation between robots. This work has succeeded in unifying concepts of agent theory with concepts from the area of mobile robotics, providing more intelligence to robots and offering solutions that otherwise cannot be provided. The methodology has been tested both in simulations and in real executions with mobile robots.

The kinematic configuration of the used agents is holonomic, then considering only linear trajectories might be acceptable. However, as a future work, the collision detection using another kind of movements, like natural Splines and Bezier curves are being considered.

Acknowledgements. This work has been partially funded by the Ministerio de Ciencia e Innovación (Spain) under research projects DPI2011-28507-C02-01 and DPI2010-20814-C02-02.

References

1. Austin, J.L.: How to Do Things With Words. Oxford University Press, Oxford (1962)
2. Bernabeu, E.J.: Fast generation of multiple collision-free and linear trajectories in dynamic environments. *IEEE Trans. Robotics* 25(4), 967–975 (2009)
3. Bernabeu, E.J., Tornero, J., Tomizuka, M.: Collision prediction and avoidance amidst moving objects for trajectory planning applications. In: *Proceedings of the IEEE Int. Conf. Robot. Automat.*, pp. 3801–3806 (2001)
4. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Ito, T., Abadi, M. (eds.) *TACS 1997*. LNCS, vol. 1281, pp. 415–438. Springer, Heidelberg (1997)
5. Cameron, S., Culley, R.K.: Determining the minimum translational distance between two convex polyhedra. In: *Proceeding of the IEEE Int. Conf. Robot. Automat.*, pp. 591–596 (1986)
6. Choi, Y.-K., Wang, W., Liu, Y., Kim, M.-S.: Continuous collision detection for two moving elliptic disks. *IEEE Trans. Robotics* 22(2), 213–224 (2006)
7. Campion, G., Bastin, B.: Kinematic and dynamic control of mobile robots. *Robotics and Automation* (1997)
8. Java Agent Development Framework (JADE) (2000)
9. LEGO Mindstorms h
10. Michalewicz, Z.: *Genetic Algorithms + Data Structure = Evolutionary Algorithms*. 2nd edn. Springer, Heidelberg (2005)
11. Redon, S., Kheddar, R.: Collision avoidance of rigid bodies. *Comput. Graph. Appl.* (2000)
12. Russell, S.J., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs (1995)
13. Searle, J.: *Speech Act Theory and Pragmatics*. Cambridge University Press, Cambridge (1969)
14. Van den Bergen, G.: Collision detection and distance calculation. In: *Game Developers Conference*. <http://www.dtecon.com> (1997)
15. van Leeuwen, J. (ed.): *Artificial Intelligence in Games*. Springer, Heidelberg (1995)
16. Wallace, R., Stentz, A.: *Real-time Motion Planning in Robotics*. Results in Robot Road (2000)

7. Campion, G., Bastin, G., Dandrea-Novel, B.: Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation* 12(1), 47–62 (1996)
8. Java Agent Development Framework, <http://jade.tilab.com>
9. LEGO Mindstorms home page, <http://mindstorms.lego.com>
10. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd edn. Springer, Heidelberg (1996)
11. Redon, S., Kheddar, A., Coquillart, S.: Fast continuous collision detection between rigid bodies. *Computer Graphic Forum* 21(3), 279–288 (2002)
12. Russell, S.J., Norvig, P.: *Artificial Intelligence: A modern approach* (2003)
13. Searle, J.: *Speech acts: An essay in the philosophy of language*. Cambridge University, Cambridge (1969)
14. Van den Bergen, G.: Continuous collision detection of general convex objects under translation. In: *Game Developers Conf.* Morgan Kaufmann (2005), <http://www.dtecta.com/interesting>
15. van Leeuwen, J. (ed.): *Computer Science Today*. LNCS, vol. 1000. Springer, Heidelberg (1995)
16. Wallace, R., Stentz, A., Thorpe, C., Moravec, H., Whittaker, W., Kanade, T.: *First Results in Robot Road-Following* (1985)

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Sigeru Omatu · José Neves · Juan M. Corchado Rodríguez · Juan F. De Paz Santana
Sara Rodríguez González (Eds.)

Distributed Computing and Artificial Intelligence

The International Symposium on Distributed Computing and Artificial Intelligence 2013 (DCAI 2013) is a forum in which applications of innovative techniques for solving complex problems are presented. Artificial intelligence is changing our society. Its application in distributed environments, such as the internet, electronic commerce, environment monitoring, mobile communications, wireless devices, distributed computing, to mention only a few, is continuously increasing, becoming an element of high added value with social and economic potential, in industry, quality of life, and research. This conference is a stimulating and productive forum where the scientific community can work towards future cooperation in Distributed Computing and Artificial Intelligence areas. These technologies are changing constantly as a result of the large research and technical effort being undertaken in both universities and businesses. The exchange of ideas between scientists and technicians from both the academic and industry sector is essential to facilitate the development of systems that can meet the ever increasing demands of today's society. This edition of DCAI brings together past experience, current work, and promising future trends associated with distributed computing, artificial intelligence and their application in order to provide efficient solutions to real problems. This symposium is organized by the Bioinformatics, Intelligent System and Educational Technology Research Group (<http://bisite.usal.es/>) of the University of Salamanca. The present edition was held in Salamanca, Spain, from 22nd to 24th May 2013.

ISSN 2194-5357


ISBN 978-3-319-00550-8



9 783319 005508

Available
online

springerlink.com

 springer.com