PATH PLANNING STRATEGY OF AUTONOMOUS CLIMBING ROBOT FOR INSPECTION APPLICATIONS IN CONSTRUCTION

C. Balaguer, V.M. Padrón, A. Giménez, J.M. Pastor, M. Abderrahim

Departamento de Ingeniería Eléctrica, Electrónica y Automática (DIEEA)
University Carlos III of Madrid
c/Butarque, 15, 28911 Leganés (Madrid), Spain.
balaguer@ing.uc3m.es

Abstract: The construction sector has a large number of highly dangerous manual operations related to inspection, assembly and maintenance. A significant part of these operations takes place in large scale environments formed by metallic structures. Specially developed self-supported robots have been introduced for this purpose during the last years. This paper deals with the autonomous climbing robot ROMA which uses the "insect" concept to climb in a complex 3D metallic-based structure. Robot needs to inspect in the structure all the beams, columns and its joints. This is why one of the crucial problems is to optimise the robot energy consumption during this inspection. The minimum energy TSP-like path planning algorithm and return-to-start algorithm are also presented in the paper.

Keywords: Robots, autonomous systems, robot path planning, optimum planning.

1. INTRODUCTION

The development of special climbing, walking and mobile robots for non traditional sectors and service application increases every day. From among them, the most usual are: a) application to wall erection, brick assembly, etc. (Gambao, *et al.*, 1997), and b) Inspection and maintenance (Bevan, et al., 1994). Construction industry has become one of the most appropriate for robotization, due to its current low level of automatization. Another important reason is the high complexity of construction sites and environments, such as buildings, bridges, towers, etc. This is why construction industry demand autonomous climbing robots with a high level of mobility in this type of environments.

During the last years a few well known climbing robots have been developed (Kerley, et al., 1992), (Luk, *et al.*, 1995), (Bach, *et al.*, 1995), etc. Nevertheless, these robots are mainly non autonomous or semi-autonomous in two senses: a) in the control system of the robot, which is normally placed in the "ground", and b) in the power supply source which is also placed in the "ground". The "ground" equipment is umbilically linked to the climbing robot using heavy wires which substantially reduce their mobility. Their control systems work at the actuator level only, but not in the locomotion or inspection ones.

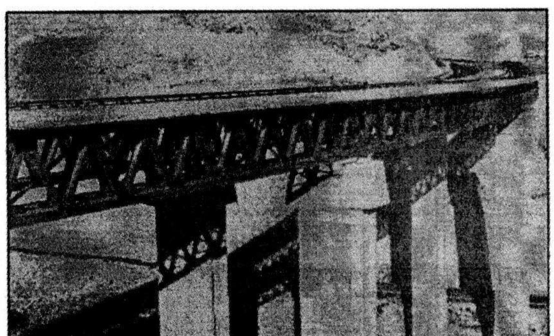


Fig. 1: Metallic-based bridge for ROMA robot inspection

There are many highly dangerous manual operations related to periodical inspection in construction sites. An important part of these operations are performed in large size environments formed by metallic structures with difficult and dangerous access even for skilled workers. The most relevant examples are: inspection of screwed/welded unions of building metallic skeleton or inspection of the painting of the metallic-based bridges with a complex structure (Fig. 1). The possibility of using autonomous robots for these applications will present a very important advantage in safety and quality (Kamei, *et al.*, 1994). The main objective of the ROMA robot is the development of multifunctional autonomous self-supported climbing robot able to travel into complex metallic-based environment. The navigation is performed by the robot CPU in an autonomous way without other help. The robot is able to self-support its locomotion system for 3D movements and it has the possibility of autonomous power supply using the on-board batteries or be umbilically connected to "ground" power supply to increase the robots autonomy. The effectiveness of the ROMA robot in inspection of large structures, like bridges, directly depends on its autonomy.

The robot is equipped with two types of on-board sensors for inspection operations: a) colour cameras and b) laser telemeters. With the camera it is possible to inspect: a) the colour to check for rust, paint and structural defects, b) the geometric features of screws and bolts to check if they are at their required torque. Laser is mainly used for: a) localization of the robot with respect to the metallic structure, and b) helps the camera to obtain the 3D images. These data are used internally by the robot CPU or transmitted to the "ground" computer which perform several processes like initialization, supervision, programming, etc. An example of man-machine interface with sensor data is presented in Fig. 2 (Balaguer, et al., 1998).

2. ROBOT STRUCTURE

The mechanical, electromechanical and control design and development of the ROMA robot was performed by the University Carlos III of Madrid. There were several stages during the mechanical design. First of all, the analysis of the different movements of the robot in different scenario was performed in order to select its kinematics structure.

The output of this process was the selection of robot's number DOF and their ranges. Then the electromechanical design was performed using the dynamics analysis and simulation package ADAMS (Balaguer, *et al.,* 1997). This helps to define the length and weight of the robot's parts, and select the electrical motors, batteries. etc. for the previously calculated torques.

The ROMA robot is formed by three joined parts (Fig. 3a): a) the body of the robot, which includes the CPU, the servo controller multiaxis board, one servo motor amplifier (drivers), he batteries, the radio-based Ethernet communication with the "ground" operation centre, and auxiliary electronics, like the multiplexing system; b) the locomotion system of 8 DOF formed by two grippers attached to the robots body controlled by AC servo motors with Harmonic Drive reductors, which permit the 3D movements along complex structures, c) the sensorial platform based on camera and laser telemeter, for inspection operations and for the robot navigation. Fig 3b shows the ROMA robot on the lab test structure.

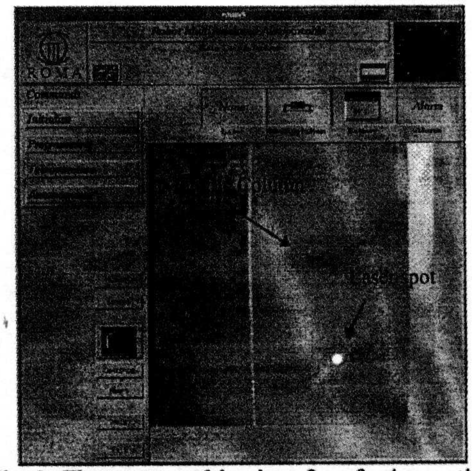
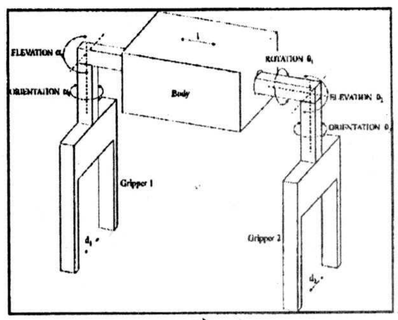


Fig. 2: The man-machine interface for inspection tasks



a)

b)

Fig. 3: ROMA robot structure: a) number of DOF, and b) real robot view.

The robot has eight DOF kinematics: a) four DOF for elevation (α_1 and θ_2) and orientation (α_2 and θ_3) of each gripper, b) one DOF for rotation (θ_1) of the gripper 2, c) one DOF for "extension" (l) of the body, and d) two DOF for grippers (d_1 and d_2). In this way the robot structure has a non redundant kinematics and a minimum possible number of DOF for 3D complex movements. This is one of the main specifications because each DOF (includes its actuator) increases the total robot weight and consequently increases the required torque to move the overall weight of the robot. Tab. 1 summarized the main characteristic of the ROMA robot.

Table 1 The main characteristics of the ROMA robot

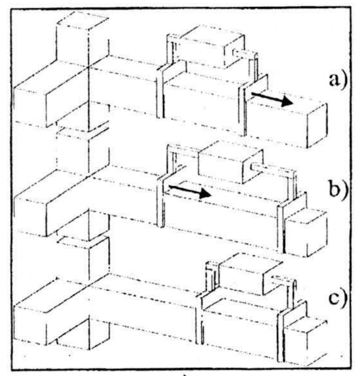
Characteristics	Values
Elevation range	$-10^{\circ} \rightarrow +190^{\circ}$
Orientation range	$-190^{\circ} \rightarrow +190^{\circ}$
Rotation range	$-190^{\circ} \rightarrow +190^{\circ}$
Extension range	500 mm
Grippers extension range	300 mm
Robot weight	75 kg
Maximum linear velocity	$\approx$1 m/min
Autonomy	$\approx$ 3 h.

3. CLIMBING STRATEGY

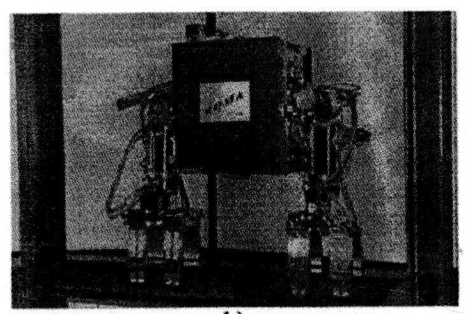
The robot has been designed taking in account not only its mobility among the complex 3D environment but to perform these movements with minimum energy consumption which permits to maximize the robots autonomy. The robots autonomy is one of the most important specification. For this purpose the robot movements were analysed taking in consideration the minimum power consumption. These are three different types of movements: a) 1D movement along the beam or column (Fig. 4), b) 2D movements in the horizontal or vertical planes of the beam faces (Fig. 5 and 6), and c) 3D movements which change the robot position from one plane of the structure to another one (Fig. 7).

To ensure the robot movement in the structure different sequences of elemental movements are possible. These are 1 DOF movements called "movements primitives" (MP). But if the minimum energy consumption criterion is taken into account it is necessary to select among all the possible MP the one with minimum energy consumption. For example, there are three different possibilities to perform the 1D forward movement on the top of a metallic beam (Fig. 8):

1. ***Dragging,*** like a "caterpillar" with the following sequence: a) release one of the grippers, b) extension of the robot body, c) lock the gripper on the beam, d) release the other gripper, e) shrink of the robot body, and f) lock the second gripper to the beam.
2. ***Horizontal rotation,*** like an "centrifuge" with the following sequence: a) release one of the grippers, b) small upwards rotation of the robot (elevation) about the horizontal axis, c) 180° rotation of the robot around the vertical axis, d) small downwards rotation of the robot about the horizontal axis, and e) lock the gripper on the beam.
3. ***Vertical rotation,*** like an "acrobat" with the following sequence: a) release one of the grippers, b) big upwards rotation of the robot above the horizontal axis, c) 180° down rotation of the second robots gripper around the horizontal axis, d) small downwards rotation of the robot above the horizontal axis, and e) lock the gripper on the beam.



a)



b)

Fig. 4: **1D movement along the beam: a) general scheme, b) robot view**

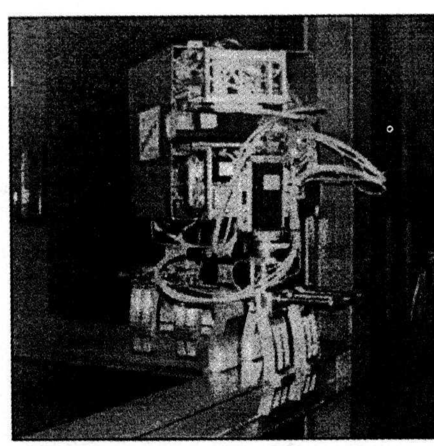
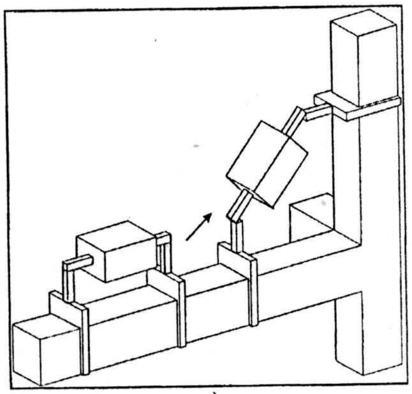
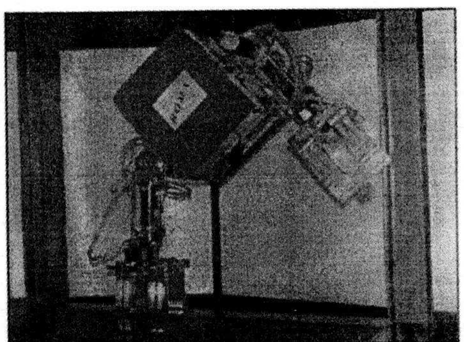


Fig. 5: 2D horizontal robot movement view

As shown in Fig. 8, all three movements are performed by a combination of 7 MP (A, B, C, D, E F and G). This is why we need to compute the energy consumption of each of them and then select the minimum energy consumption strategy for the forward movement. Tab. 2 shows the energy consumption and the robot speed along the beam for the above mentioned alternatives. As result of this study, including the gravity force, the best compromise between energy consumption and robot speed is dragging, being the accepted MP A and B



a)



b)

Fig. 6: 2D robot vertical movement: a) scheme, and b) robot view

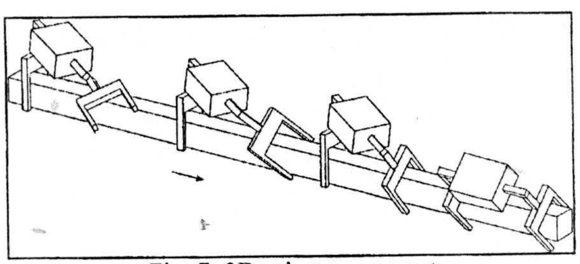


Fig. 7: 3D robot movement

For the 2D movements between different beams the sequence is similar to the above adding to it the elevation and and/or orientation of the grippers. Finally, most of the complex 3D movements are ensured by combining all individual movements including the gripper rotation. As in the above mentioned example, for 2D and 3D movements the minimum energy consumption MP are selected.

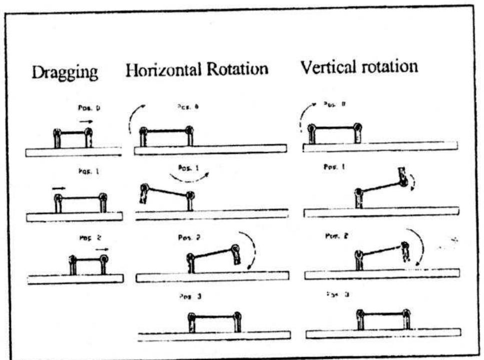


Fig. 8: Forward robot movement alternatives

Table 2 Comparative study of different 1D forward movement

Motion	MP	Energy	Speed(m/min)
Dragging	A,B	392	1
Rot. around	C,D,E	784	0.4
Rot. Above	F,G,E	940	0.25

4. ENVIRONMENT MODELLING

The environments of the ROMA robot are mainly metallic closed structures of buildings or bridges. To plan the robots path in this type of environments it is necessary to built a environment model. This is done using "environment primitives". There are two main primitives: a) beam primitive, which includes beams and columns of the structure, and b) beams-cross, which join the beams and columns to form 2D or 3D structures. An example of the 2D open beam-cross structure is presented in Fig. 9, where the three level coding of the primitives ("a.b.c") has the following structure: "a" is the number of the beam, "b" is the number of the face of the beam, and "c" is the reference point on the beam (starting, centre or end point). The environment is modelled as a graph using the described primitives (Fig. 9).

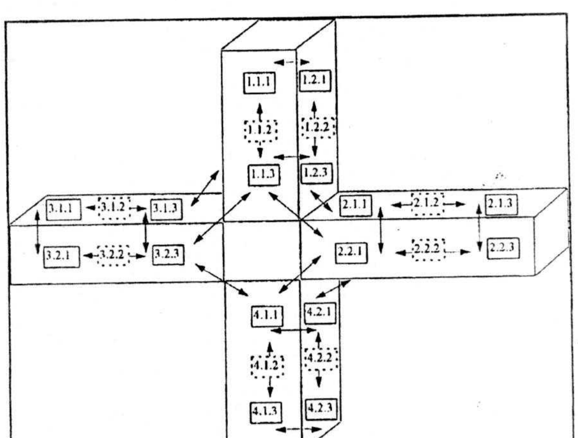


Fig. 9: 2D environment formed by one cross-beam and four beams.

5. PATH PLANNING STRATEGY

The common inspection task is performed through the whole metallic structure, which makes necessary to plan the robot path with the following requirements: a) compute the round trip path taking in to account the power consumption, and b) perform the inspection of all faces of all beams, i.e. visit all the nodes and transition of the environment graph. These two requirements entails solving a TSP-like problem -Travelling Salesman Problem- starting and finishing the inspection task in the same point, transiting, if possible, only once along each beam face, and optimizing the robot consuming energy.

To solve the TSP problem it is necessary that the graph should be Hamiltonian, i.e. the graph in which it can be possible to visit all the nodes without repeating one. To determine if the graph is Hamiltonian, a difficult non-deterministic polynomial (NP) problem should be solved. This is why, usually the graph is transformed into a complete one which, as it is well-known, is always a Hamiltonian (Glover and Punnen, 1997). However, there is a case when the environment graph is easier to transform directly into a Hamiltonian, less redundant than the complete one. This is when each beam-cross primitives can be visited an even number of times, allowing this way to depart and return to same point transiting each beam face exactly once. To complete this transformation is sufficient to: a) introduce a central point beam primitive virtual node (node "a.b.2" in Fig. 9) for each beam face, and b) complete all the transitions in the beam-cross primitives in order to connect each node to all other nodes of the beam-cross.

To solve the TSP on the obtained graph several algorithms are applied: an exact algorithm and a set of heuristics. The exact algorithm is based on the "Branch & Bound 1-Tree" technique implemented by Hurwitz (Hurwitz, 1992). Due to the fact that TSP problem is NP-complete, the computing time of this algorithm increases in an exponential way. This makes it impractical to apply it for the ROMA robot environments which have a big number of nodes (more than 100).

To solve this problem several traditional heuristics have been checked in order to obtain the sub-optimum solution. The following heuristics are proposed: a) "nearest addition" which consists of adding to an initial tour (close robot path) a nearest node which is not in the tour, b) "farthest insertion" which consists of inserting to an initial tour the farthest node, c) "easytour" which follows the order of definition of the nodes, etc. To improve the results of these heuristics an "improver" is applied (Lawler, *et al.,* 1985). It consists in a local iterative search which exchanges the nodes positions in each heuristic tour in order to optimize the energy criterion. In this case the exchange strategy is based on a variation of a well known Lin and Kernighan algorithm (Lin and Kernighan, 1973). Finally, a exhaustive 3-Opt improver and the Lin and Kernighan (again) are applied to the best obtained result. As can be observed this set of algorithms ("Best Heuristics" after Hurwitz) fails to solve the problem, this is a classical result for TSPs on sparse graphs [GoldBoDoySte1980] and also for the ones with a cost matrix no satisfying the triangle inequality criteria [SahniGonz1976].

To overcome this difficult a new ROMA heuristic is developed. It is based on an algorithm to extract Eulerian circuit, i.e. a robot closed path in which no beam or column face is repeated. This algorithm travel the structure transiting between its cross-beams, and selecting in a cross-beam a new face to reach the next one. This selection is done applying a "Nearest Neighbour" criteria (Rosenkrantz, et al., 1977). As improver a modification of the well-known Or (Or, 1976) algorithm is employed. Fig. 10 and 11 show the effectiveness of the proposed path planning algorithm which can be fast computed for a big number of nodes.

During the movement the robot's battery level can be checked continuously. Using this data the robot predict the minimum battery level necessary to perform the returning movement to the starting point. This prediction is based on the modified algorithm A* using also energy criteria (McHugh, 1990).

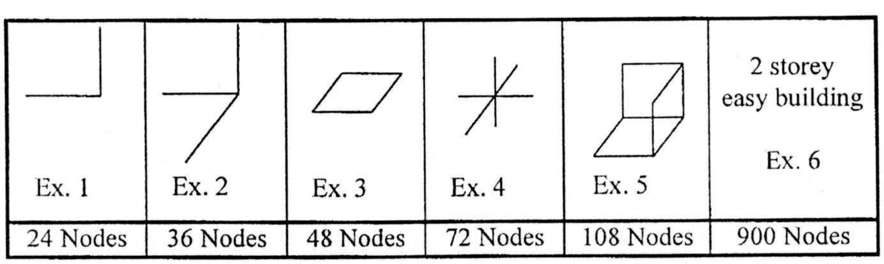


Fig. 10: Environment examples

Algorithm	Optimum		Best Heuristics			ROMA		
Example	Energy (J)	Time (s)	Energy (J)	Time (s)	Over Energy (%)	Energy (J)	Time (s)	Over Energy (%)
1	2885	0.31	2896	0.38	0.38	2885	0.02	0
2	3633	1.63	3633	0.08	0	4002	0.05	10.15
3	4272	3.26	4480	0.25	4.86	4272	0.05	0
4	6850	486.52	7052	0.78	2.94	7052	0.3	2.94
5	11948	1196069	∞	1.76	∞	13055	0.21	9.26
6	84620*	-	∞	2417.57	∞	94916	2.09	12.16

Fig. 11: Results of the path planning algorithms (* Held-Karp estimation).

6. CONCLUSIONS

The developed ROMA robot has several important advantages in the field of climbing robots: it has a very dextrous kinematic structure, it is equipped with servo-controlled axes, it uses a powerful on-board sensorial system, etc. In addition, the main advantage is the fact that it is designed as an autonomous system able to move freely in complex environments with a sub-optimum energy consumption. The first experiments using the ROMA robot confirm its advantages in executing the inspection operations in complex. This type of robots could replace the human operators in performing dangerous tasks in the near future.

ACKNOWLEDGEMENTS

This work has been funded by Spanish goverment agency CICYT under project TAP95-0088

REFERENCES

[1] Bevan, N.R., Collie, et al., (1994). A robotic manipulator for inspection and maintenance of tall structures. *Symposium on Robotics and Automation in Construction (ISARC'94), Brighton (UK).*

[2] Bach, F.W., et al., (1995). High tractive power wall-climbing robot. *Automation in Construction, vol. 4, n°3.*

[3] Balaguer, C., Gambao, E., et al., (1997). Computer-aided methodology for design of robots in construction industry applications. *14th International Symposium on Robotics and Automation in Construction (ISARC'97), Pittsburgh (USA).*

[4] Balaguer, C., et al., (1998). ROMA: A multifunctional autonomous self-supported climbing robot for inspection application. *3rd IFAC Symposium on Intelligent Autonomous Vehicles (IAV'98), Madrid (Spain).*

[5] Gambao, E., Balaguer, C., et al., (1997). Robot assembly system for the construction process automation. *IEEE International Conference on Robotics and Automation (ICRA'97), Albuquerque (USA).*

[6] Glover, F. and Punnen, A.P. (1997). The travelling salesman problem: new solvable cases and linkage with the development of approximation algorithms. *Journal of Operational Research, vol. 48.*

[7] Golden, B.L., Bodin, L.D., Doyle, T., Stewart, W. Jr., (1980). Approximate travelling salesman algorithm. *Oper. Res.* 28.

[9] Hurwitz, C. (1992). TSD: performance and description of heuristic. *Ph.D. Thesis.*

[10] Kamei, R. et al., (1994). Development of multi-purpouse mobile robot capable of travelling along columns and beams. *11th International Symposium on Robotics and Automation in Construction (ISARC'94), Brighton (UK).*

[11] Kerley, J.J., et al., (1992). The climbing crawling robot. *Industrial Robots, vol. 19, n°4.*

[12] Lawler, E.L., et al., (1985). The travelling salesman problem. *John Wiley & Sons.*

[13] Lin, S. and Kernighan, B. W., (1973). An effective heuristic algorithm for TSP. *Operational Research, vol. 21.*

[14] Luk, B.L., et al., (1995). An arthropods robot for walking in hazardous environment. *2nd IFAC International Conference on Intelligent Autonomous Vehicles (IAV'95), Espoo (Finland).*

[15] McHugh, J.A., (1990). Algorithmic graph theory. *Prentice.Hall.*

[16] Or, I. (1976) Travelling salesman-type combinatorial problems and their relation to the logistics of regional blood banking" Ph. D. Dissertation, Northwestern University, Evanston, IL.

[18] Rosenkratz, D.J. et al., (1977). An analysis of several heuristics for the TSP", *SIAM Journal of Computing,vol.6.*

[19] Sahni, S., Gonzalez , T., (1976). "P-complete approximation problems" *J. Assoc. Comput. Mach.* n° 23.