Navigation Controllability of a Mobile Robot Population

Francisco A. Melo, M. Isabel Ribeiro, and Pedro Lima

Institute for Systems and Robotics,
Instituto Superior Técnico, Lisboa, Portugal
{fmelo, mir, pal}@isr.ist.utl.pt

Abstract. In this paper, the problem of determining if a population of mobile robots is able to travel from an initial configuration to a target configuration is addressed. This problem is related with the controllability of the automaton describing the system. To solve the problem, the concept of navigation automaton is introduced, allowing a simplification in the analysis of controllability. A set of illustrative examples is presented.

1 Introduction

Robotic navigation is a central topic of research in robotics, since the ability that a robot has to accomplish a given task may greatly depend on its capability to navigate in the environment. The related literature presents numerous works on the subject, and proposes different navigation strategies, such as Markov Models [1], dynamic behaviours [2] or Petri Nets [3].

In the last decades a great effort has been addressed to the subject of multi-robot systems. A common approach to the multi-robot navigation problem is the extension of known strategies for single robot navigation to the multi-robot case, for which there are several examples presented in the literature [4]. However, in the multi-robot navigation framework, new topics of investigation emerged, such as cooperation and formation control or flocking [5].

In this paper, the problem of multi-robot navigation is addressed. We analyze the problem of driving a robot population moving in a discrete environment from some initial configuration to a target configuration. We develop analysis strategies in order to determine under which situations the target configuration becomes non-achievable, in order to prevent those situations. In a previous work [6], this analysis has been conducted for a set of homogeneous robots([1]). This paper extends those results to a set of heterogeneous robots, i.e., where robots with different capabilities may intervene.

The robot population is modeled as a finite-state automaton (FSA) and the main contribution of this paper is the analysis of the blocking and controllabili-

[1] We consider a set of robots to be homogeneous when all robots are alike, i.e., they have the same capabilities.

D. Nardi et al. (Eds.): RoboCup 2004, LNAI 3276, pp. 499–507, 2005.
© Springer-Verlag Berlin Heidelberg 2005

ty properties of this automaton. Since it models the movement of the complete robot population in the environment, from a start configuration to given goal configuration, properties such as blocking and controllability have direct correspondence with the successful completion of this objective. For example, a blocking state corresponds to a distribution of the robots from which the desired goal configuration is not achievable (because one of the robots has reached a site from where it cannot leave, for example). Controllability means that such blocking states are avoidable: it is possible to disable some actions to prevent the robots from reaching blocking configurations.

The results presented in this paper relate the blocking and controllability properties of the automaton modeling the multi-robot system (which can be a large-dimension automaton, for complex systems) with the blocking and controllability properties of smaller automata, named as *navigation automata*, that model the navigation of each individual robot in the population.

The paper is organized as follows. In Section 2, some basic concepts are introduced and the problem under study is described. Section 3 approaches the problem of determining the blocking properties of the automaton describing the system. In Section 4, the results regarding controllability are presented for generic systems. Section 5 presents a set of illustrative examples. Finally, Section 6 concludes the paper and presents directions for future work. The proofs of all results in Sections 3 and 4 can be found in [7].

2 Navigation Automata and the Multi-robot system

In this section, some basic concepts regarding automata are introduced and the notation used throughout this paper is described.

Notation Regarding Automata [8]
An automaton Q is a six-tuple $Q = (X, E, f, \Gamma, x_0, X_m)$, where

- X is the state space;
- E is the set of possible events;
- $f : X \times E \longrightarrow X$ is the transition function;
- $\Gamma : X \longrightarrow 2^E$ is the active event function;
- x_0 is the initial state;
- X_m is the set of marked states.

The languages generated and marked by Q are denoted, respectively $\mathcal{L}(Q)$ and $\mathcal{L}_m(Q)$. An automaton is called *unmarked* if $X_m = \emptyset$.

An automaton Q is said to be *non-blocking* if $\overline{\mathcal{L}_m(Q)} = \mathcal{L}(Q)$ and *blocking* if $\overline{\mathcal{L}_m(Q)} \subsetneq \mathcal{L}(Q)$.

A set $X_C \subset X$ of states is said to be closed if $f(x, s) \in X_C$, for any $s \in \mathcal{L}(Q)$ and $x \in X_C$. A blocking automaton verifies $X_C \cap X_m = \emptyset$.

The Problem. Consider a system of N robots, moving in a discrete environment consisting of M distinct sites. This is referred as a N-R-M-S situation (N robots and M sites) or a N-R-M-S system. The set of sites is denoted by $\mathcal{S} = \{1, \ldots, M\}$.

Generally, when in site i, a robot will not be able to reach all other sites in a single movement. The function $\Omega_k : \mathcal{S} \rightarrow 2^{\mathcal{S}}$ establishes a correspondence between a site i and a set $\mathcal{S}_i \subset \mathcal{S}$ of sites reachable from i in one movement of robot k. If $j \in \Omega_k(i)$, then, for robot k, site j is *adjacent* to site i. Function Ω_k is called the *adjacency function* for robot k.

This paper addresses the problem of driving the robots from an initial configuration C_I to a final or target configuration C_F. The set of sites containing at least one robot in the final configuration is denoted by $\mathcal{S}_T$. The sites in $\mathcal{S}_T$ are called *target sites*. From the point of view of final configuration, no distinction is made among the robots, i.e., it is not important which robot is in each target site.

Navigation Automata. A robot k moves in the environment defined by the topological map according to its own adjacency function and is described by an unmarked automaton $G_k = (Y_k, E_k, f_k, \Gamma_k, y_{0k})$. Y_k is the set of all possible positions of robot k, verifying $Y_k = \mathcal{S}$. E_k is the set of all possible actions for robot k. Actions consist of commands leading to the next site for the robot to move to. Since all robots have the same event space, to avoid ambiguity, action i issued to robot k is denoted by $Go_k(i)$, where i is the next site for the robot k. Therefore, all events in the system correspond to movements of the robots. It is assumed that only one robot moves at a time. The active event function Γ_k when robot is in state i corresponds to the sites reachable from i in one movement. This means that $\Gamma_k = \Omega_k$.

Definition 1 (Navigation Automaton). *Given a robot k moving in a discrete environment consisting of M distinct sites, the* navigation automata *for this robot are the marked automata* $G_k(Y_m) = (Y_k, E_k, f_k, \Gamma_k, y_{0k}, Y_m)$, *where:*

- Y_k, E_k, f_k *and* Γ_k *are defined as above;*
- Y_m *is a set of target states,* $Y_m \subset \mathcal{S}_T$.

In the case of a homogeneous set of robots, in which $\Omega_1 = \ldots = \Omega_N$, all G_i are alike, except for the initial condition y_{0k}. In this situation, when the initial condition is clear from the context or not important, a navigation automaton will simply be denoted by $G(Y_m) = (Y, E_m, f_m, \Gamma_m, y_0, Y_m)$.

2.1 The Multi-robot System

If there are no constrains on the number of robots present at each site, each of the robots can be in M different positions, and there are M^N different possible configurations.

The system of all robots can be described by a FSA, $G = (X, E, f, \Gamma, x_0, X_m)$, where X is the set of all possible robot configurations, yielding, in the most general situation, $|X| = M^N$. Each state $x \in X$ is a N-tuple $(x_1, x_2, ..., x_N)$ and x_i is the site where robot i is.

Also, there is a set $X_F \subset X$ of states corresponding to the target configurations. Notice that, in automaton $G, X_m = X_F$.

As seen before, robot k has an available set of actions E_k, denoted by $Go_k(i)$. Therefore, the multi-robot system has a set of actions $E = \bigcup_k E_k$, all consisting of $Go_k(i)$ actions.

3 Blocking

Let G be the automaton modeling a N-R-M-S system.

If G is blocking, there is a closed set of states C, called *blocking set*, such that $C \cap X_m = \emptyset$. This, in turn, means that whenever the robots reach a configuration corresponding to a state $x \in C$ it is not possible to drive them to the desired configuration anymore.

Usually, blocking is checked by verifying $\overline{\mathcal{L}_m(G)} \subsetneq \mathcal{L}(G)$ exhaustively. In the present case, as the system can lead to relatively large automata for not so large M and N, a more effective way to check the blocking properties of G is desirable. This section addresses this problem.

3.1 Blocking and Navigation Automata

Let G be the automaton modeling a N-R-M-S system. If the system is homogeneous, Result 2 of [6] holds.

For a generic (non-homogeneous) system, if G is non-blocking, then so is each of the navigation automaton G_i, with all target states as marked states. However, the converse is not true. Theorem 1 follows.

Theorem 1. *In a generic N-R-M-S system, for its automaton G to be non-blocking, all the navigation automata $G_i(Y_m)$ must non-blocking, with $Y_m = \mathcal{S}_T$. Similarly, if G is blocking, there is at least one i and one target state $y_m \in Y_m$ such that $G_i(y_m)$ is blocking.*

Proof. See in [7]. □

3.2 Blocking Information Matrix

From section 3.1 one can conclude that, generally, it is not possible to determine the blocking properties of G simply by taking into account the blocking properties of each navigation automaton individually, since these properties depend on the relation between them. It is, however, possible to determine whether or not G is blocking, by *comparing* the blocking properties of each automaton $G_i(y_m)$.

In an N-R-M-S system, let $K(m)$ be the number of robots in target site m in the final configuration C_F. If $U \subset \mathcal{S}_T$ is a set of target sites, define $K(U) = \sum_{m \in U} K(m)$. Define as $B(m)$ the number of robots that block with respect to target site m. If $U \subset \mathcal{S}_T$ is a set of target sites, define $B(U)$ as the number of robots *simultaneously* blocking the sites in U. In general, $B(U) \neq \sum_{m \in U} B(m)$.

If the number of robots blocking simultaneously the sites in some set $U \subset \mathcal{S}_T$ is such that $N - B(U) < K(U)$, then G blocks, and therefore, $N - B(U) < K(U)$. This means that there are not enough "free" robots to go to the sites in M. This condition may be easily verified using the *blocking information matrix*.

Definition 2 (Blocking Information Matrix). *Given a generic N-R-M-S system, the* blocking information matrix *(BIM)* $\mathbf{B}_N$ *is a* $N \times N$ *matrix such that element* (k, m) *is* 0 *if* $G_k(y_m)$ *is blocking and* 1 *otherwise.*

Each of the N lines of matrix $\mathbf{B}_N$ corresponds to a different robot. On the other hand, if a site m has $K(m)$ robots in the target configuration, matrix $\mathbf{B}_N$ will have $K(m)$ columns corresponding to this site. Matrix $\mathbf{B}_N$ is easily computed from the analysis of the navigation automata $G_k(y_m)$, and the following result can be proved.

Theorem 2. *Given the Blocking Information Matrix* $\mathbf{B}_N$ *for a* N-R-M-S *system, the automaton* G *describing the overall system is blocking if and only if there is a permutation matrix* $\mathbf{P}$ *such that* $\mathbf{PB}_N$ *has only ones in the main diagonal.*

Proof. See in [7]. □

4 Supervisory Control

In this section, the problem of controllability of G is addressed. The controllability problem is related to the design of a supervisor S, such that, when applied to the original system, the resulting system marks some desired language $\mathcal{K}$.

Although the automaton G describing the system already marks the desired language, in a situation where the automaton is blocking, it is not desirable that the system reaches a blocking state, since this will prevent the final configuration to be reached. The presence of a supervisor S in the system under study will necessarily relate to this situation where blocking must be prevented. It is important to determine the existence of such a supervisor, i.e., it is important to determine if the system is controllable.

In the following analysis, the existence of unobservable events is disregarded even if they make sense from a modeling point of view, as described in [6].

4.1 System Controllability

Consider the automaton G describing a N-R-M-S system, which is assumed blocking, and suppose that there is a non-empty set of uncontrollable events $E_{uc} \subset E$. These events may correspond to accidental movements of the robots which cannot be avoided. If the system is homogeneous, Result 3 of [6] holds.

Consider, now, a heterogeneous system. As stated before, blocking in G is related to the number of robots "available" to fill each target site, when considering blocking sets. Controllability relates with the ability of a supervisor to disable strings of events driving a robot to a blocking set.

Let $E_{uk} \subset E_k$ be the set of uncontrollable events for robot k. It is possible to include controllability information in matrix $\mathbf{B}_N$ in order to conclude about the controllability of G. If $G_k(y_m)$ is blocking but controllable with respect to the language $\mathcal{K} = \mathcal{L}_m(G_k(y_m))$, then the element (k, m) of matrix $\mathbf{B}_N$ is set to -1. This motivates the following and most general form of Theorem 2.

Theorem 3. *Given the Blocking Information Matrix* $\mathbf{B}_N$ *for a generic* N*-*R*-*M*-*S *system, the automaton* G *describing the overall system is blocking if and only if there is a permutation matrix* $\mathbf{P}$ *such that* $\mathbf{PB}_N$ *has only ones in the main diagonal.*

If G *is blocking, but there is a permutation matrix* $\mathbf{P}_1$ *such that* $\mathbf{P}_1\mathbf{B}_N$ *has only non-zero elements in the main diagonal, then* G *is controllable with respect to the language* $\mathcal{K} = \mathcal{L}_m(G)$.

Proof. See in [7]. □

Observe the relation between Theorem 3 and Result 3 of [6]. In fact, the latter can be derived from Theorem 3 when the robot set is homogeneous.

5 Examples

We present three examples of the application of Theorem 3 in a simple indoor rescue situation. Consider a non-homogeneous set of three robots in which:

The Crawler. (Cr) has tracker wheels and is capable of climbing and descending stairs. It is able to open doors only by pushing;

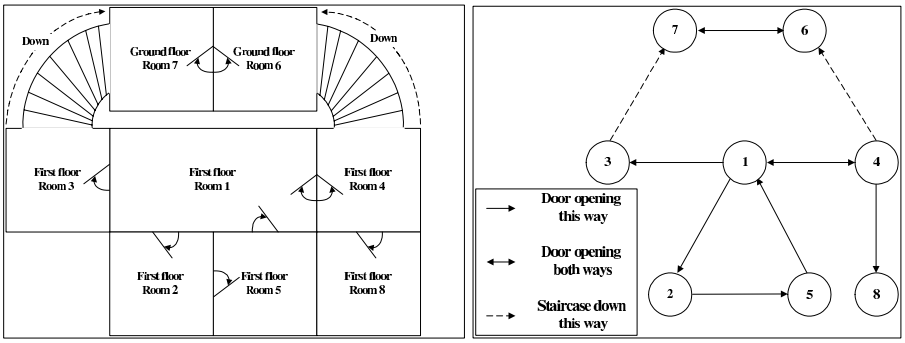

Fig. 1. Map of the environment

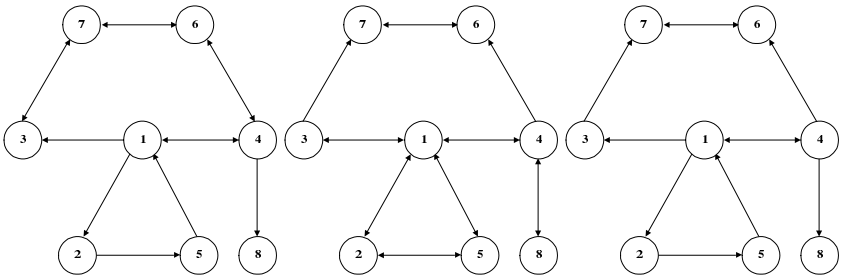

Fig. 2. Automata for the robots

The Puller. (Pl) is a wheeled mobile manipulator, able to open doors either by pushing or pulling. However, it is not able to climb stairs;

The Pusher. (Ps) is a wheeled robot, able to open doors only by pushing. It cannot climb stairs.

The rescue operation takes place in the indoor environment depicted in Figure 1 (e.g., a fire scenario). On the left is the physical map of the place, and on the right is the corresponding topological map. Each of the robots is described by a different automaton, as represented in Figure 2.

The robots will leave Room 1 to assist three different victims, somewhere in the building. The doors open as shown in Figure 1 which limits the robots access to the different rooms. When in Rooms 6 or 7, only the Crawler can go upstairs. Finally, when in Rooms 3 and 4, all the robots may fall downstairs, i.e., events $Go_k(6)$ and $Go_k(7)$ are uncontrollable for all k. The following examples illustrate the practical use of Theorem 3. We determine if there are configurations that prevent the success of a given rescue operation which, in terms of the framework proposed in this paper, correspond to blocking configurations. The situation where there are victims in sites a, b and c is referred to as the $a - b - c$ Rescue.

5.1 $6 - 7 - 8$ Rescue

In this situation, the BIM for the system is:

$$\mathbf{B}_3 = \begin{bmatrix} -1 & -1 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}, \tag{1}$$

where the lines correspond to Pusher, Puller and Crawler and the columns correspond to target sites 6, 7 and 8, respectively.

Note that both Ps and Cr, once inside Room 8, are not able to leave. Then, $G_{Ps}(6)$, $G_{Ps}(7)$, $G_{Cr}(6)$ and $G_{Cr}(7)$ are blocking. However, by disabling the events $Go_{Ps}(8)$ and $Go_{Cr}(8)$, blocking can be prevented and $\mathbf{B}_3(1,1) = \mathbf{B}_3(1,2) = \mathbf{B}_3(3,1) = \mathbf{B}_3(3,2) = -1$.

If Ps or Pl get downstairs, they cannot go back upstairs. However, they cannot get to Room 8 without going through Room 4 and eventually falling to Room 6, which cannot be avoided, since $Go_k(6)$ is uncontrollable. Then, $G_{Ps}(8)$ and $G_{Ps}(8)$ are blocking and uncontrollable, and $\mathbf{B}_3(1,3) = \mathbf{B}_3(2,3) = 0$.

Finally, Pl can always reach Rooms 6 and 7, and Cr can always reach Room 8. $G_{Pl}(6)$, $G_{Pl}(7)$ and $G_{Cr}(8)$ are non-blocking, and $\mathbf{B}_3(2,1) = \mathbf{B}_3(2,2) = \mathbf{B}_3(3,3) = 1$.

From Theorem 3 the system is blocking but controllable. For example in the configuration where Crawler and Pusher are in room 8, it is impossible to reach the target configuration. However, this can be prevented, by disabling, for example, $Go_{Ps}(8)$, which is a controllable event.

5.2 $2-8-8$ Rescue

In this situation, the BIM for the system is:

$$\mathbf{B}_3 = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 1 & 1 \end{bmatrix}, \tag{2}$$

with the columns corresponding to target sites 2, 8 and 8, respectively. It becomes evident that the system is blocking but, unlike the previous example, it is uncontrollable. Since two robots are required for site 8 and the only way to reach Room 8 is through Room 4, they will eventually move to Room 6 instead of moving to Room 8 (since $Go_k(6)$ is uncontrollable), once they get to Room 4. In this situation, it may be impossible to assist both victims in site 8. In fact, as long as there is more than one victim in site 8 ($K(8) > 1$), this problem will always exist. This happens because there are two robots which "helplessly" fall downstairs, blocking site 8 ($B(8) = 2$). Then, $N - B(8) = 3 - 2 = 1 < K(8)$, and the system is blocking. Since the only way to Room 8 is through Room 4, this situation cannot be prevented.

6 Conclusions and Future Work

The problem of analyzing the navigation of a set of mobile robots operating in a discrete environment was approached. Relevant results have been derived, that allow the use of small dimension automata (navigation automata) to infer about the blocking and controllability properties of the automaton that describes the complete system. In a situation where a specific configuration is aimed for a set of robots, the presented results allow to determine, using global information, if the global objective is achievable, and if blocking configurations are avoidable.

An important extension of the present work is the determination of the relation between the blocking properties of the navigation automata and the ergodicity of the Markov Chain which can be used to model the complete system, when a probabilistic uncertainty is associated to the events representing the movements of the robots. Other interesting issue is the use of this local information in an optimal decision process, when a decentralized system is considered.

Acknowledgements

Work partially supported by Programa Operacional Sociedade de Informação (POSI) in the frame of QCA III and by the FCT Project *Rescue—Cooperative Navigation for Rescue Robots* (SRI/32546/99-00). The first author acknowledges the PhD grant SFRH/BD/3074/2000.

References

1. Simmons, R., Koenig, S.: Probabilistic Robot Navigation in Partially Observable Environments. Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Canada (1995) 1080–1087
2. Steinhage, A.: Dynamical Systems for the Generation of Navigation Behaviour. PhD thesis, Institut für Neuroinformatik, Ruhr-Universität, Bochum Germany (1997)
3. Hale, R.D., Rokonuzzaman, M., Gosine, R.G.: Control of Mobile Robots in Unstructured Environments Using Discrete Event Modeling. SPIE International Symposium on Intelligent Systems and Advanced Manufacturing, Boston, USA (1999)
4. Balch, T., Hybinette, M.: Social Potentials for Scalable Multirobot Formations. IEEE International Conference on Robotics and Automation (ICRA-2000), San Francisco, USA (2000)
5. Balch, T., Arkin, R.C.: Behavior-based formation control for multi-robot teams. IEEE Transactions on Robotics and Automation **14** (1998) 926–939
6. Melo, F.A., Lima, P., Ribeiro, M.I.: Event-driven modelling and control of a mobile robot population. Proceedings of the 8th Conference on Intelligent Autonomous Systems, Amsterdam Netherlands (2004)
7. Melo, F.A., Ribeiro, M.I., Lima, P.: Blocking controllability of a mobile robot population. Technical Report RT-601-04, Institute for Systems and Robotics (2004)
8. Cassandras, C.G., Lafortune, S.: Introduction to Discrete Event Systems. The Kluwer International Series On Discrete Event Dynamic Systems. Kluwer Academic Publishers (1999)