A Game Theory Approach to the Robot Motion Tracking Problem

Carlos Rodríguez Lucatero

UIA Ciudad de México, Departamento de Ingeniería,
Prolongación Paseo de la Reforma 880, México D.F., 01210
profesor.lucatero@gmail.com

Abstract. The paper studies the problem of tracking a target robot by an observer robot. The strategy of the target robot is not known in advance. The observer robot will try to learn the target's robot strategy by keeping a model about the target robot behaviour. This will be done by modelling the tracking problem as a repeated two player game, where the robots objective is to look for strategies that maximizes their expected sum of rewards in the game. We make the assumption that the robot motion strategies can be modelled as a finite automata. First we suppose that they behave competitively and then we relax this constraint to explore the case of a more general kind of interaction between the target and the observer.

1 Introduction

Many applications require continuous monitoring of a moving *target* as is the case of movie filming of moving actors whose motions are not known in advance and is necessary to keep inside the scope of the camera the actions of the main character. Other application can be the virtual presence for trying to keep track remotely of some moving objects as vehicles, people, etc. Another application of robot tracking can be the automated surveillance of museums where there can be a robot that follows a guest from the beginning of his visit till he leaves the museum. In the last few years many research efforts have been done in the design and construction of efficient algorithms for reconstructing unknown robotic environments [12][13][14][15] and apply learning algorithms for this end [12][13] . When we are looking for efficient interaction strategies we have to take into account the reward obtained for the actions executed in that moment as well as the consequences of the taken actions on the future behaviour of the other entities. This task can become very hard given that the effectiveness of the interaction strategy depends on the strategies of the other agents. The problem is that the strategies of the other agents are private. For dealing with these problems it is necessary to provide the agents with the capability to adapt their strategies based on

© *A. Gelbukh, S. Torres, I. López (Eds.)*
Advances in Artificial Intelligence
Research in Computing Science 20, 2006, pp. 145-154

their interaction history. This implies that we have to endow the agents with learning capabilities. In the near past there have been written many excellent articles on learning models of intelligent agents as those elaborated by David Carmel & Shaul Markovitch [1] [2]. In the field of Multi-agent Systems there have been written excellent papers about Markov games as a framework for multi-agent reinforcement learning (M.L. Littman [6]). In the field of robot motion planning there has been published very interesting papers about the problem of calculating motion strategies for maintaining visibility in space cluttered by obstacles, of a moving target whose movements were partially predictable and where the movements of the *target robot* as well as the *observer robot* have uncertainties. One of the main concerns on robot tracking is to keep the visibility of the *target robot* by maintaining the target on the scope of the cone of the visual sensor of the observer robot (S. La Valle et al [4] [5]). Another very important concern that can be found in many papers on robot tracking is to keep the visibility in real time of the *target robot* whose behaviour is unpredictable by using a reactive planner (R. Murrieta et al [7]). These last to papers [4] [5] and [7] will be the starting point of the present paper where we will propose some extensions under the focus of interest of game theory. In [4] [5] and [7] they make the assumption that the strategy of the *target robot* is to evade the *observer robot* and based on that they propose geometrical and probabilistic solutions of the tracking problem which consists on trying to maximize, by the *observer,* the minimal distance of escape of the *target*. We feel that the solution lacks at least in two aspects. First the *target* don't interact with the *observer* so there is no evidence that the strategy will try to escape if it doesn't knows what are the actions taken by the *observer*. The second aspect is that even if there can take place some sort of interaction between the *target* and the *observer*, the *target* is not necessarily following an evasion strategy so this may produce a failure on the tracking task.

In Section 2 we retake the formulation given in [5] and we will make some remarks about the limitations of this approach. In section 3 we will provide a precise formulation of the problem in terms of strategies in game theory and present a modelling of the tracking problem as a repeated two game player. In Section 4 we present some concluding remarks and future works to be done by us.

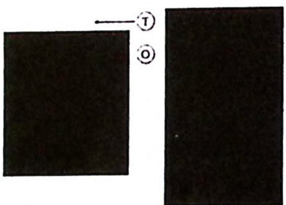
2 Formulation of the Tracking Problem as a Robot Motion Planning Problem

The problem can be posed in a worst case situation where the *target robot* try to evade the *observer robot*. Under this assumption the main goal of the strategy of the *observer robot* or *pursuer* will be to guarantee that the *target robot* or *evader* will be found for all possible motions. We can initially assume that the *pursuer* is equipped with vision or range sensing and that both robots are modelled as 2D points so each robot has a configuration space of dimension 2. This is a generalization of the *evasion-pursuit* that have been studied and formalized as a general decision problem where two agents have diametrically opposed interests. So the *pursuer* and the *evader* are modelled as points on the plane that is a bounded 2D Euclidean space cluttered by polygonal obstacles such that the task of keeping in a visibility cone the *evader* makes the problem become harder and by consequence, more appealing from the standpoint of motion planning. We will start with some definitions given in [5] [4] and [7] for the sake of clarity and the purpose of giving context to our work. In [5] are formulated two interesting research questions 1) What bounds can be established on the number of pursuers needed to solve the problem in terms of geometrical and topological complexity of the free space ? and 2) can a successful solution strategy be efficiently calculated for a given problem ? . We will try to give an answer to the second question from a game theory point of view and show that we can extend the scope of [4] using this formalism to more general *target* strategies than the *evasion* strategy. The problem of tracking have been defined in [5] as follows. Let F denote the closure of the collision free space. All *pursuer* and *evader* positions must lie in F. Let $e(t) \in F$ be the *evader* position at time $t \geq 0$. It is assumed that $e:[0,\infty) \rightarrow F$ is a continuous function, that the *evader* can execute arbitrarily fast motions and that the initial evader position $e(0)$ and path e are not known by the *pursuers*. In our case we will deal with only one *pursuer*. Let $\gamma(t)$ the position of the *pursuer* at time $t \geq 0$ and $\gamma:[0,\infty) \rightarrow F$ a continuous function representing his strategy. For any point $q \in F$ let $V(q)$ be the set of all point (i.e. linear segments joining q and any point in $V(q)$ lies in F). A strategy γ is considered as a *solution strategy* if for any $e:[0,\infty) \rightarrow F$ there exist a time $t \in [0,\infty)$ such that $e(t) \in V(\gamma(t))$. That means that the *evader* will eventually be see by the *pursuer* regardless its path. Given that the *evader's* position is unknown, one don't have access to the state at a given time, and that motivates the use the notion of *information space* that identifies all unique situations that can occur during motion strategies. In [4] are studied the motion strategies for maintaining visibility of a moving target. In this paper it is studied the problem of maintaining the visibility of the *target* with a camera mounted in an *observer* robot as motion planning problem, and assumed the following conditions 1) an *observer* nust maintain visibility of a moving *target*; 2) the workspace contains static obstacles that prohibit certain configurations of both the *observer* and *target*; 3) the workspace also contains static obstacles that occlude the target from the observer; 4) a (possibly partial) model is known for the *target*. In [4] they formulate the problem of one *pursuer* and one *evader* in the combined configuration space $X = C^{o}_{free} \times C^{t}_{free}$. They use a discrete time representation for facilitating the expressions of the uncertainty of the *target* for instance k is an index that refers to the time step that occurs at $(k-1)\Delta t$, and Δt is a fixed sampling rate. The *observer* is controlled through actions u_k from some space of actions U. The discrete time trajectory of the *observer* and the *target* were given by the transition equations $q^{o}_{k+1} = f^{o}(q^{o}_{k}, u_k)$ and $q^{t}_{k+1} = f^{t}(q^{t}_{k}, \theta_k)$ respectively where θ_k represents the unknown actions of the *target* from a space of actions Θ. In the case of a predictable *target* the transition equations is simply $q^{t}_{k+1} = f^{t}(q^{t}_{k})$. Together f^{o} and f^{t} define a state transition equation of the form

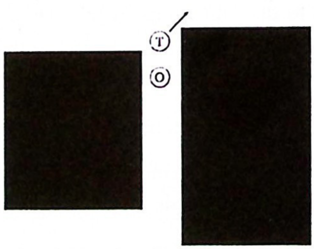
$x_{k+1} = f(x_k, u_k, \theta_k)$ where the state x_k represents the pair of configurations q_k^o and q_k^t. The visibility can be defined in many ways as for instance an omnidirectional field of view or as a fixed cone, etc. but in a more general setting it can be defined in terms of a binary relation between a visibility subspace and the space of states or more formally as $X_o \subset X$. Based on that the state trajectories can be evaluated in such a way that the *observer*'s goal is to stay in a state belonging to X_o. This control of the trajectory can be performed by applying a cost to the sequence of control inputs as follows: $L(x_1,\ldots,x_{K+1},u_1,\ldots,u_K) = \sum_{k=1}^{K} l_k(x_k,u_k) + L_{K+1}(x_{K+1})$ where K represents the time increment for issuing a action and $l_k(x_k,u_k) = \{0 \ \mathit{if} \ \ x_k \in X_o; 1 \ \mathit{otherwise}\}$ is a loss accumulated in a single time step with which enable to measure the amount of time that the *target* is not visible and evaluate a given trajectory. If the movements of the *target* are predictable this means that q_k^t is known $\forall k \in \{1,\ldots,K+1\}$ and the transition equation is simplified to $x_{k+1} = f(x_k, u_k)$ and as consequence the state trajectory $\{x_2,\ldots,x_{K+1}\}$ can be know if once we know x_1 and the inputs $\{u_1,\ldots,u_K\}$. So for problems that don't involve the optimization of the robot trajectory, the motions of the *observer robot* can be computed by a recursive calculation of the visibility and reachability sets from stage K down to stage 1 , or telling it in words, by back-chaining. Besides that the loss functional can be minimized by dynamic programming using the relationship between the *cost-to-go* functions $L_k^* = \min_{u_k}\{l_k(x_k,u_k) + L_k^*(x_{k+1})\}$ and this can be utilized iteratively for calculating the optimal actions. The visibility polygon using omnidirectional visibility can be calculated in $O(n \lg n)$ using standard sweep algorithms. Another very interesting case appears when the *target* is partially predictable in the sense that it is know the velocity bound of the *target* in which case the dynamic programming method can be used to determine optimal strategies, but even for the very simple planar space the dimension becomes four. Due to this growth of complexity it have to be used alternative approaches that make a tradeoff between computational cost and quality of the solutions obtained. So the notions of optimal strategy become more interesting due to the fact about the uncertainty on the prediction of the *target* movements calculated as $q_{k+1}^t = f^t(q_k^t, \theta_k)$ where $\theta_k \in \Theta$ are the unknown actions. These unknown actions can be modeled in two ways. The fisrt as *nondeterministic uncertainty* and the second as *probabilistic uncertainty*. In first case one design the *observer* strategy that performs the best given the worst-case choices for θ_k. In the second case it can be assumed $p(\theta_k)$ where $p(\cdot)$ denotes a probability density function, and in that case the designed strategy of the *observer* will try to minimize the expected loss. Due to the unpredictability of the *target* strategy, it has to be designed an *observer* state-feedback strategy that respond to on-line changes. Let $\gamma_k : X \rightarrow U$ denote the strategy at stage k, $\{\gamma_1, \gamma_2, \ldots, \gamma_K\}$ the strategy, and Γ the space of possible strategies. For the case of nondeterministic uncertainty, a strategy , can be selected that yields the smallest worst-case loss: $\check{L}(x_1, \gamma^*) = \inf_{\gamma \in \Gamma} \check{L}(x_1, \gamma) = \inf_{\gamma \in \Gamma} \sup_{\gamma^\theta \in \Gamma^\theta} L(x_1, \gamma, \gamma^\theta)$ for all $x_1 \in X$, and γ^θ represents a choice of θ_k for every stage. The strategy obtained by this method guarantee the least possible loss given the worst-case action of nature. Using this formalism it can be proposed as strategy to maximize the probability of future visibility over the next m steps but the computational cost increases dramatically as a function of m. Because of that, in practice, the number of steps is limited to the case $m = 1$ and select the action u_k that maximize the probability that the target stay in the scope of the *observer* at the stage $k+1$. As a extension of the preceding approach in [7] R. Murrieta et al. use nondeterministic uncertainty and worst-case analysis for trying to solve the tracking problem for obtaining the *observer* strategy by maximizing the minimal distance to escape of the *target*, that is the shortest distance the target needs to move in order to escape the *observer's* visibility region. In this work they implement a reactive planner, that means a short term planner. They do that because of the need of a rapid response time that a normal planner cannot give. Due to the limitations of the vision sensing they introduce the notion of view frustrum or angular field of view. That is defined in terms of edges that borders either an obstacle , denoted as E_{fs}, or free space, denoted as E_{fr}. The free edge of the visibility region $V(q)$ nearest to the *target* is denoted as E^*_{fr}. To maximize the distance between the target and the boundary of the visibility region of the observer it is necessary to compute de distance between q^t_k and E^*_{fr} which is denoted $D_{q^t_k / E^*_{fr}}$. For obtaining it the distance among q^t_k and every edge of $V(q)$ must be calculated, and it can be easily done using Euclidean metric when when the E_{fr} is visible from the *observer* position, otherwise it is used a geodesic metric. So the distance between q^t_k and each free edge in $V(q)$ is the solution to the equation $D_{q^t_k / E^*_{fr}} = \min_{\{v_o, v_f\}} \{D_{q^t_k / v_o} + D_{v_o / v_f} + D_{v_f / E_{fr}}\}$ where v_o are the vertex in the *targets's* visibility region, and v_f is any vertex in $V(q)$ that sees the free edge E_{fr}.

2.1 Some Limitations of the Precedent Approach

The work done in [4] and [7] was very interesting but is limited to the case where the *target robot* is assumed to follow an evading strategy. That is due to the worst-case approach and the fact they were concerned to solve the tracking problem in real time tacking in account the uncertainty on the behaviour of a partially predictable *target*. The *evasion* strategy behaviour assumption is a very natural one but this entails the interaction between the *target* and the *observer* in such a way that the *evader* could take the best decision for evading the *observer*. This was completely ignored in [4], [5] and [7]. Lets give an schematic example. In the following image we denote with **T** and **O** the *target* and *observer* respectively. The black boxes represent obstacles in 2D and the arrow represents the movement decision taken by **T.** If **T** is following an evasion strategy it will try to minimize his distance to escape and **O** will try to maximize the minimal distance to escape of **T.**



In the case that **T** is not trying to evade **O** it can moves as it is shown in the following image.



If **O** is trying to maximize the minimal distance to escape of **T** it can lose the tracking. Another aspect that has not been considered in [4], [5] and [7] was the case where the *target* behaviour be different from the evasion one. In that case the actions or strategies calculated by the *observer* such that the *target* remains on the visibility region of the *observer* may fail if the actions of the *target* were to move to a position out of the visibility region and not necessarily to one related with the shortest distance to escape. Because of the reasons exposed above we propose as an extension of the precedent work done in [4], [5] and [7] to endow the *target* and the *observer* with a learning capacity in such a way that the *observer* can predict the behaviour of the *target* in a case different from the *evasion* strategy assumed in [4], [5] and [7].

3 Stating the Tracking Problem as a Repeated Game

One of our objections in way that the tracking problem has been formulated in [4], [5] and [7] was that it does not consider the fact that if the *target* is going to have an *evasion* behaviour it must take place an interaction between the *target* and *observer* such that the first can observe the second and take the best decision for an *evader.* It is well known that searching for an optimal interactive strategy is a hard problem because it depends on the behaviour of the others. Given that the agents involved in an interaction are autonomous their strategies are private, as is the case of the *observer* and the *target*. For dealing with this problem we propose to endow the interacting agents with a learning ability such that they can adapt their behaviours or strategies based on their interaction experience. We propose to use *model-based* approach for learning an efficient interactive strategy between the *observer* and the *target* inspired on what has been proposed in [1]. The agents keep a model of the opponent's strategy that is modified or adapted during the interaction, exploiting the current model to predict the other's behaviour and choose its own action according to the prediction. In case of failure in the prediction the agent updates the opponent's model to make it consistent with the new information. This approach give rise to two important questions 1) given a model of another agent, how an agent react optimally against it ? 2) when there is a prediction failure how an agent can adapt his opponent's model ?. To give an answer to these questions it can be used some tools of the *game theory*. The interaction between the *observer* and the *target* can be modelled as a *repeated two-player game* where the goal of each agent is to compute interaction strategies that maximizes its expected sum of rewards. This can be done efficiently under some assumptions about the kind of strategies followed by the agents as well as about the type utility functions for the *repeated games*. The first assumption is that each agent follows a regular strategy, i.e., a strategy that can be represented by a deterministic finite automata [10] and [9]. This assumption is based on the fact that in the case of having a *live complete sample* and a knowledgeable teacher that answers *membership queries* posed by the learner, it can be obtained a incremental polynomial learning DFA algorithm [9]. A second assumption is about the form of the utility functions, i.e.*discounted-sum* and *limit-of-the-means*, because it has been proved in [1] based on the work done in [8] that the best response strategy can be obtained efficiently given these common utilities functions.

Definition 1. *The tracking problem posed as a two-player-game is a tuple* $G=\langle f^o, f^t, u_1, u_2\rangle$, *where* f^o, f^t, *are the finite set of alternative moves for the observer and the target and* $u_1, u_2 : f^o \times f^t \rightarrow \Re$ *are utility functions that define the utility joint move* (q^o, q^t) *for the players (i.e. observer and target).*

We propose that the tracking problem can be formulated as a sequence of encounters between the *observer robot* and the *target robot* and that situation can be described as a repeated game G', that is a repetition of G an indefinite number of times. At any stage k of the game, the players decide their actions $(q_k^o, q_k^t) \in f^o \times f^t$, simultaneously. A history $h(k)$ of G' is a finite sequence of joint moves chosen by the *observer* and the *target* until the current stage of the game. $h(k)=\langle (q_0^o, q_0^t), (q_1^o, q_1^t), \ldots, (q_{k-1}^o, q_{k-1}^t)\rangle$ denotes the history of movements of each robot. The empty history is denoted by ε. The set of finite histories is denoted as $H(G')$. The strategies are functions from the set of histories of games to the set of robots actions or moves $\sigma^o : H(G') \rightarrow f^o$ and $\sigma' : H(G') \rightarrow f'$ for the observer and the target respectively. An infinite sequence of joint moves during a repeated game G' between the *observer's* and the *target's* strategies is denoted by (σ^o, σ'). The repeated game G' played by σ^o and σ' defines the history $h(k)$ as follows:

$$g_{\sigma^o,\sigma'}(0) = \varepsilon$$

$$g_{\sigma^o,\sigma'}(k+1) = g_{\sigma^o,\sigma'}(k) \| (\sigma^o(g_{\sigma^o,\sigma'}(k)), \sigma'(g_{\sigma^o,\sigma'}(k)))$$

***Definition** 2. Tracking as a two-player repeated-game over a stage game G is a tuple $G' = \langle \Sigma^o, \Sigma', U^o, U' \rangle$ where Σ^o and Σ' are sets of strategies for the observer and the target respectively and $U^o, U' : \Sigma^o \times \Sigma' \rightarrow \Re$ are the utility functions. U^o defines the utility of the infinite sequence $g_{\sigma^o,\sigma'}$ for the observer and U' for the target.*

***Definition** 3. $\sigma^o_{opt}(\sigma', U^o)$ is called the optimal strategy for the observer w.r.t. σ' and utility U^o, iff $\forall \sigma \in \Sigma^o, [U^o(\sigma^o_{opt}(\sigma', U^o), \sigma') \geq U^o(\sigma, \sigma')]$.*

***Definition** 4. $\sigma'_{opt}(\sigma^o, U')$ is called the optimal strategy for the target w.r.t. σ^o and utility U', iff $\forall \sigma \in \Sigma', [U'(\sigma'_{opt}(\sigma^o, U'), \sigma^o) \geq U'(\sigma, \sigma^o)]$.*

In the present work are considered two common utility functions for each robot, the first is the discount factor $U^o_{ds}(\sigma^o, \sigma') = (1-\gamma^o)\sum_{k=0}^{\infty} \gamma^o_k (\sigma^o(g_{\sigma^o,\sigma'}(k)), \sigma'(g_{\sigma^o,\sigma'}(k)))$ for the observer and respectively $U'_{ds}(\sigma^o, \sigma') = (1-\gamma')\sum_{k=0}^{\infty} \gamma'_k (\sigma'(g_{\sigma^o,\sigma'}(k)), \sigma^o(g_{\sigma^o,\sigma'}(k)))$ for the target for $0 \leq \gamma^o < 1$ and $0 \leq \gamma' < 1$, the second kind of utility function is *limit-of-the-means* $U^o_{lm}(\sigma^o, \sigma') = \liminf_{k \to \infty} \frac{1}{k} \sum_{k=0}^{\infty} u^o(\sigma^o(g_{\sigma^o,\sigma'}(k)), \sigma'(g_{\sigma^o,\sigma'}(k)))$ for the observer and $U'_{lm}(\sigma^o, \sigma') = \liminf_{k \to \infty} \frac{1}{k} \sum_{k=0}^{\infty} u'(\sigma^o(g_{\sigma^o,\sigma'}(k)), \sigma'(g_{\sigma^o,\sigma'}(k)))$ for the target. So taking into account the repeated game formalism combined with robot motion planning tracking problem formulation we can make for instance

$u' = D_{q_k'/E_{fr}^{\bullet}} = \min_{\{v_o, v_f\}} \{D_{q_k'/v_o} + D_{v_o/v_f} + D_{v_f/E_{fr}}\}$ and

$u^o = \check{L}(x_1, \gamma^{\bullet}) = \inf_{\gamma \in \Gamma} \check{L}(x_1, \gamma) = \inf_{\gamma \in \Gamma} \sup_{\gamma^o \in \Gamma^o} L(x_1, \gamma, \gamma^o)$ and replace them into the respective *limited-of-the-means* common utility functions then we can model the tracking problem of the kind *evader/pursuer* as a two-player repeated game. As can be seen the utility functions are more general and the *evader/pursuer* case is just an instance of the possible behaviors, so with these theoretic game model we can deal with more general behaviors depending on the utility functions u^o, u' chosen.

3.1 Learning the Target's Automata

In this paper we assume that each robot is aware of the other robot actions, i.e. Σ^o, Σ' are common knowledge while the preferences u^o, u' are private. It is assumed too that each robot keeps a model of the behavior of the other robot. The strategy of each robot is adaptive in the sense that a robot modifies his model about the other robot such that the first should look for the best response strategy w.r.t. its utility function. Given that the search of optimal strategies in the strategy space is very complex when the agents have *bounded rationality* it has been proved in [10] that this task can be simplified if we assume that each agent follow a DFA strategy. In [8] has been proven that given a DFA opponent model, there exist a best response DFA that can be calculated in polynomial time. In the field of computational learning theory it has been proved by E.M. Gold [11] in that the problem of learning minimum state DFA equivalent to an unknown target is *NP*-hard. Nevertheless D. Angluin has proposed in [3] a supervised learning algorithm called *ID* which learns a target DFA given a *live-complete* sample and a knowledgeable teacher to answer *membership* queries posed by the learner. Later Rajesh Parekh, Codrin Nichitiu and Vasant Honavar proposed in [9] a polynomial time incremental algorithm for learning DFA. That algorithm seems to us well adapted to the tracking problem because the robots have to learn incrementally the strategy of the other taking as source of examples the visibility information as well as the history about the actions performed by each agent.

4 Conclusions and Future Work

As we have exposed in the present work, the *one-observer-robot/one-target-robot* tracking problem can be formulated as a two-player game and enable us to analyse it in a more general setting than the *evader/pursuer* case. The prediction of the *target* movements can be done for more general *target* behaviours than the *evasion* one, endowing the agents with learning DFA's abilities. The next step will be to elaborate the associated algorithms and implement them in a computer program. Another interesting issue is to apply the algorithms to the case of many *evaders* and many *pursuers*.

References

1. David Carmel & Shaul Markovitch, "Learning Models of Intelligent Agents," Technical Report CIS9606 Technion, 1996.
2. David Carmel & Shaul Markovitch, "How to explore your oponent's strategy (almost) optimally," Proceedings ICMAS98 Paris France, 1998.
3. Dana Angluin, "A note on the number of queries needed to identify regular languages", Information and Control, 51, 76-87, 1981.
4. S.M. La Valle, H.H. González Baños, Craig Becker, & J.C. Latombe, "Motion Strategies for Maintaining Visibility of a Moving Target", Proceedings of the IEEE International Conference on Robotics and Automation, 731-736,April 1997.
5. S.M. La Valle, David Lin, Leonidas J. Guibas, J.C. Latombe & Rajeev Motwani, "Finding an Unpredictable Target in a Workspace with Obstacles," Proceedings of the IEEE International Conference on Robotics and Automation, April 1997.
6. M.L. Littman, "Markov games as a framework for multiagent reinforcement learning," Proceedings of the eleventh International Conference on Machine Learning, 157-163 ,1994.
7. R. Murrieta-Cid, H.H. González-Baños & B. Tovar, "A Reactive Motion Planner to Maintain Visibility of Unpredictable Targets," Proceedings of the IEEE International Conference on Robotics and Automation , 2002.
8. Christos H. Papadimitriou & John N.Tsitsiklis, "The complexity of Markov decision processes", Mathematics of Operations Research, Vol. 12, No. 3, August 1987.
9. Rajesh Parekh, Codrin Nichitiu & Vasant Honavar, "A polynomial time incremental algorithm for learning DFA", Proceedings of the Fourth International Colloquium on Grammatical Inference (ICGI'98), Ames, IA. Lecture Notes in Computer Science vol. 1433 pp. 37-49. Berlin: Springer-Verlag..
10. Ariel Rubinstein, "Finite Automata Play the Repeated Prisioner's Dilemma", Journal of Economic Theory, 39, 83-96, 1986.
11. E.M. Gold, "Complexity of automaton identification from given data", Information and Control, 37, 302-320, 1978.
12. Dana Angluin, Jeffery Westbrook, & Wenhong Zhu, "Robot Navigation with range Queries", ACM STOC 96, 469-478, 1996.
13. Ronald L. Rivest, & Robert E. Schapire, "Inference of Finite Automata using Homing Sequences", Information and Computation, 103, 299-347, 1993.
14. A. Blum, P. Raghavan & B. Schieber, "Navigating in unfamiliar geometric terrain", ACM STOC 91, 494-504, 1991.
15. V. Lumelsky & A. Stepanov, "Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape", Algoritmica, 2, 403-430, 1987.