Reactive Agent Technology for Real-Time, Multisensor Target Tracking

COSTAS TSATSOULIS AND EDWARD KOMP
Department of Electrical Engineering and Computer Science
Information and Telecommunication Technology Center (ITTC)
The University of Kansas
2335 Irving Hill Rd., Lawrence, KS 66045
USA

Abstract: - We describe an architecture for a reactive agent system that can control a network of autonomous sensors to track a number of vehicles in real-time. In this paper we discuss the theory behind the reactive agent architecture, and present its implementation and a set of experiments using a simulator and a set of real MTI sensors. Our work shows how reactive agents can achieve significant real-time tracking accuracy as compared to other types of deliberative agents.

Key-Words: - reactive agents, multiagent system, multisensor target tracking

1 Introduction

We describe a reactive agent approach to multisensor target tracking in a real-time environment. Each agent controls a sensor, and is motivated to use the sensor to track targets appearing in its coverage area, and also to make the sensor available to other agents in an effort to satisfy the global goal of tracking multiple targets concurrently. The act of balancing the local use of tracking resources and the global goal satisfaction increases the complexity of the problem. Each agent in the system is autonomous, monitors its environment through a sensor, and reacts to changes that it observes. There is no hierarchical organization among the agents allowing the system as a whole to react to world events more quickly. Our agents follow a simple, reactive protocol that allows them to request the aide of other agents, and to also intelligently respond to tracking requests by other agents in the multiagent system.

The driving application is to track as many targets as possible and as accurately as possible using a network of sensors. Each sensor has a set of consumable resources, such as beam-seconds (the amount of time a sensor is active), battery power, and communication channels, that each sensor desires to utilize efficiently. Each sensor is at a fixed physical location and, as a target passes through its coverage area, it has to collaborate with neighboring sensors to triangulate their measurements to obtain an accurate estimate of the position and velocity of the target. As more targets appear in the environment, the sensors need to decide which ones to track, when to track them, and when not to track them, always being aware of the status and usage of sensor resources.

Our solution to the problem is to use *reactive agents.* The agents sense their immediate environment and, if they sense a target, they contact a set of neighboring agents to form a *tracking coalition.* The agents that are contacted decide whether they are free or already engaged in tracking activities, and either immediately join the coalition or refuse to do so.

We compared our work with a set of non-cooperating agents, where tracking is done individually without forming coalitions, and showed that our system offers a significant advantage.

2 Problem Formulation

The problem we address in this work is to use a set of sessile, intelligent sensors to track any number of moving targets in a dynamic and noisy environment. The problem is further complicated by the real-time constraints of the environment and the fact that agents have to share physical resources such as communication channels and disk storage. For example, for a target moving at one foot per second, accurate tracking requires one measurement each from at least three different sensors within a time interval of less than 2 seconds. The real-time constraints force our agents to deal with issues such as CPU allocation (since speed of execution depends on it), disk space allocation, communication latency, and processing times. Finally, the environment is noisy and subject to uncertainty and error: messages may be lost, a sensor may fail to operate, or a communication channel could be jammed. Thus, in addition to improving autonomy, one is required to promote noise-resistance in agent reasoning, sensor control, and communications.

The sensors are 9.35 GHz Doppler MTI radars that communicate using a 900 MHz wireless, radio-frequency (RF) transmitter with a total of eight available channels. Each sensor can at any time scan one of three sectors, each covering a 120-degree swath. Sensors are connected to a network of CPU platforms on which the agents controlling each sensor reside. The agents (and sensors) must communicate over the eight-channel RF link, leading to potential channel jamming and lost messages. Finally, there is software (the "tracker") that, given a set of radar measurements, produces a possible location and velocity for a target; the accuracy of the location and velocity estimates depend on the quality and frequency of the radar measurements: as we mentioned, the target must be sensed by at least three radars within a two second interval for accurate tracking.

3 Problem Solution

The reactive agent architecture can be defined as follows: the agents examine their environment and based on it they establish behavioral parameters. The agents are divided into three categories: Radar Agents (RA), Node Control Agents (NCA), and Tracker Agents (TA). A Radar Agent is very simple and has the basic knowledge necessary to communicate with other agents and to control a sensor (e.g. turn the radar on and off, select beam direction, etc.). A Node Control Agent contains the intelligence required to perform tracking of targets: when to track, what to track, and for how long. There is a RA and a NCA for each sensor. In addition to the RA and NCA agents, there are tracking software modules which contain code that, given the radar returns from multiple sensors, estimate the location and velocity of a target. The tracking software is wrapped by a Tracker Agent (TA) which evaluates the quality of the radar measurements received and then forwards only reliable ones to the tracker. This agent architecture is shown on Figure 1.

When the RA has no measurement tasks to carry out on behalf of any NCA, it searches in all of its sectors to detect targets in round-robin fashion, and it sends these measurements to the appropriate TA (TAs are automatically created for each target sensed). We call this mode the "Search & Detect" mode. In this mode, the TA continually checks if the measurements it receives from the RAs it is collaborating with have high-enough confidence for the TA to believe there is a target on a given sector such that it should instruct the appropriate NCA to switch to its tracking mode.

The second mode a Radar Agent works in is called the "Measure" mode. When the TA believes that there is a target visible from a given radar sector, it asks the appropriate Node Control Agent to instruct its Radar Agent to measure. So, as soon as the RA has a specific measurement task to perform, it suspends all searching and detecting, and executes its specific measurement tasks only. Since multiple TAs may ask for measurements from the same sector, the NCA keeps track of which TAs have active measure tasks to perform on each of the sectors of the radar it controls, and uses negotiation-based techniques to switch between these requests, trying to balance the use of resources.

Tracker Agents provide NCAs with information about targets to allow the NCAs to reason about which tracking task to schedule, for how long, and how to switch between tracking tasks.

One addition we have made to our methodology is that we have introduced domain-specific reasoning in the Tracker Agents (TAs). The TAs use heuristics to evaluate the quality of measurements received from the sensors, and only high quality measurements are sent to the tracking software which then establishes the location and velocity of the target. Each measurement received by an TA is assigned a certainty factor (CF). Measurements are not sent to the tracker unless their CF is above a threshold. The goal is to avoid confusing the tracker software with noise, and to also dynamically track an unknown number of multiple, simultaneous targets. Currently, our heuristics integrate the amplitude value of a measurement, its support by multiple sensor sectors, and the expected location of a target compared to where the measurement says the target is. Each heuristic is assigned a different weight, with the amplitude weighted by 5 points, and the support by other sensors and proximity to the actual target location both weighted by 2 points. The proximity of a measurement to the previous estimated target location is also used to disqualify a measurement, since a target cannot move randomly.

If a target appears too far away from its previous position, the measurement is considered to be noise and is dismissed. The domain heuristics are combined in the traditional way of combining probabilities, namely:

$$c_1 + c_2 + c_3 - c_1 c_2 - c_1 c_3 - c_2 c_3 + c_1 c_2 c_3$$

where c_1, c_2, and c_3 are the certainty factors of the three heuristics.

The Node Control Agents automatically instantiate Tracking Agents when a potential new target is detected, and the TA is eventually destroyed when additional measurements cannot be collected to confirm that the target remains active in the scenario-defined "room". The MTI sensors and the vehicles were simulated by the Radsim simulator version 2.08 [1]. Agents connect to Radsim and use a standard API to send control signals to their respective sensor platforms. Radsim, in turn, uses a model of the platforms' behavior to return, in the case of taking a measurement, a hypothetical value of amplitude and frequency based on the current locations of the targets.

We designed and ran a number of experiments in Radsim, constantly increasing the number of sensors and targets. We have examples of experiments with 3, 5 and 6 targets for which we have been able to generate respectable simulation results using up to 20 active nodes. Figure 2 shows one of our two largest experiments to date. Sensors controlled by reactive agents are indicated by *Sn*, where *n* is an integer (e.g. S1, S2, S3, etc.). The tracks of the targets are indicated by the figure 8's. Figure 3 shows the tracking of these 6 targets by 18 sensors, using dynamic tracker generation. The numbers indicate the trackers generated; "b" indicates the beginning of the tracking (when a tracker is generated) and "e" the end (when a tracker loses a target and is destroyed). In figure 2 the agents lost the target indicated by 3b-3e and reacquired it later (7b-7e).

In addition to simulation experiments, we performed experiments with real MTI sensors and targets (eight sensors and two targets moving in an oval). Figure 4 shows the physical set-up of the experiment. The results of the experiment were similar to the ones using the Radsim radar simulation environment.

4 Experimental Results

We ran a set of experiments with our reactive agent architecture versus a set of BDI agents that employ multiagent negotiation for allocation of radar resources [2]. Our results indicate that reactive agents perform better and achieve more accurate results than negotiating BDI agents. This is probably due to the real-time nature of the environment, where agents must make decisions, form coalitions, and perform radar measurements in at least 2 seconds, which handicaps BDI agents that spend too much time deliberating. Also, the MTI radars can only offer an approximate position and velocity estimates for the target. Our architecture reacts to the current target position estimate and does not attempt to project its future positions. Since BDI agents must guess where the target will be in the future in order to generate tracking agent coalitions, they do not perform well in the multi-target tracking environment.

The following table shows the example performance of reactive versus BDI agents for a simple eight sensor, two target experiment. The superior performance of the reactive agents is obvious (all error measurements in feet):

	Error in X direction	Error in Y direction	Total MSE location error
BDI Minimum	0.002	0.03	0.27
BDI Maximum	27.81	24.52	31.07
BDI Average	5.87	6.14	8.98
BDI Std. Dev.	5.06	4.72	6.11
Reactive Min.	0.00	0.00	0.01
Reactive Max.	6.78	4.13	7.72
Reactive Ave.	1.08	0.88	1.51
React. Std. Dev.	1.31	0.89	1.47

The previous table shows that the reactive system achieves average accuracy six times better than that of the BDI agents. Also, the performance is much more consistent, with standard deviation of 1.47 feet, compared to 6.11 feet.

5 Conclusions

We described a reactive agent system for real-time multisensor target tracking. Our agents submit collaboration requests to neighboring agents and use domain heuristics to minimize the effect of noise in the equipment and the environment. We compared our agents to BDI negotiating agents, and showed that the performance of our agents is superior, probably due to the real-time nature of the environment.

Acknowledgements:

The work described in this paper is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-99-2-0502. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

References:

[1] Lawton, J.,The Radsim Simulator, in *Distributed Sensor Networks: A Multiagent Perspective*, Lesser, V., C. L. Ortiz, Jr., and M. Tambe (eds.), The Netherlands: Kluwer Academic Publishers, 2003, pp. 11-20.

[2] Soh, L-K. and C. Tsatsoulis, Reflective Negotiating Agents for Real-Time Multisensor Target Tracking, *Int. J. Conf. On Artificial Intelligence (IJCAI-01),* Seattle, WA, 2001, pp. 1121-1127.

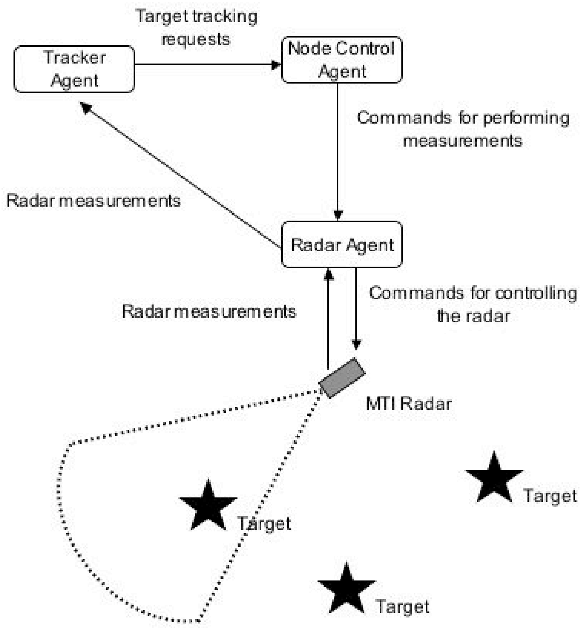

Figure 1: Architecture and information flow of our agents. The Tracker Agent wraps the software that estimates the position of the target and also evaluates the quality of the radar measurements. The Radar Agent controls the low-level functionality of the MTI radar. The Node Control Agent decides which tracking requsts to honor in order to balance the use of the radar to track multiple trargets.

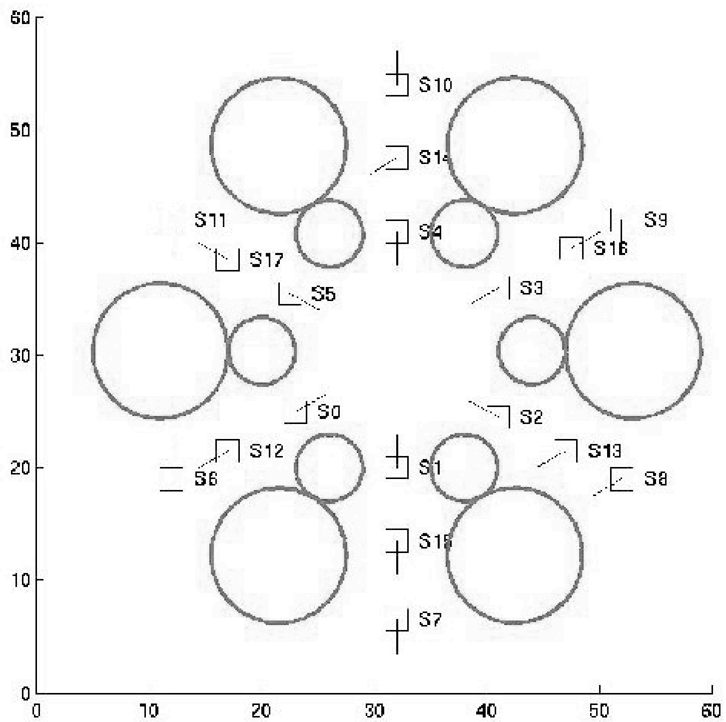

Figure 2: Sensor location and target trajectory for a 6-target, 18-sensor tracking experiment.

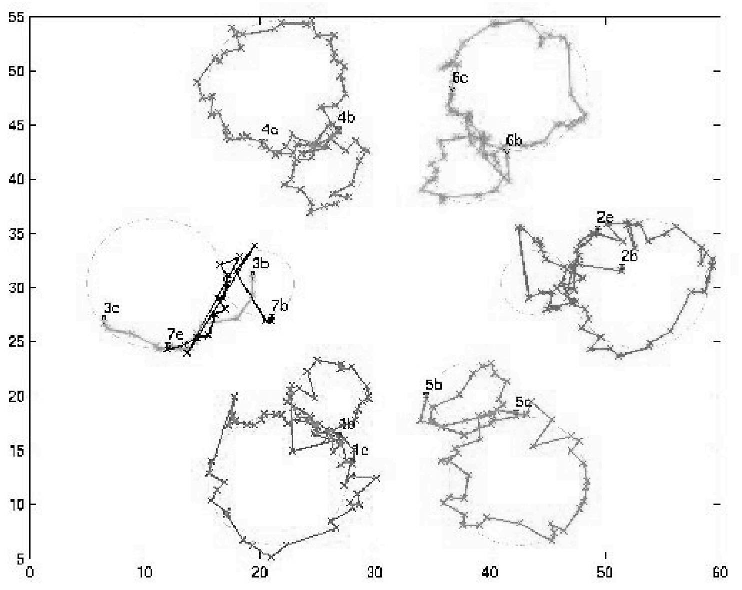

Figure 3: Tracking of the six targets of Figure 2 by sensors controlled by our reactive agents. «b» indicates the beginning of a track and «e» indicates its end.

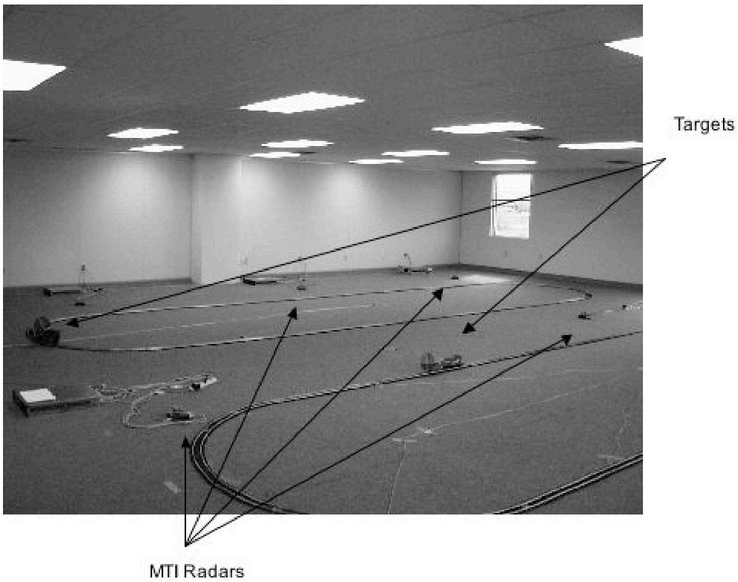

Figure 4: Physical set-up of eight MTI sensors and two target vehicles moving on oval tracks.