

A Mixed Autonomy Coordination Methodology For Multi-Robotic Traffic Control

Aditya Teja V, D. V. Karthikeya Viswanath, K. Madhava Krishna

Robotics Research Center

International Institute of Information Technology

Hyderabad, India

{adityateja,viswanath}@research.iit.ac.in, mkrishna@iit.ac.in

Abstract—We present a method for coordinating multi-robotic/multi-agent traffic control at intersections. The robotic agents (RA) move guided by a potential field along the lanes. At the intersections an intersection agent (IA) controls the flow of traffic by assigning priorities to the agents that are about to enter the intersection. The priorities are computed based on the density of RA in a lane and the flow rate of traffic in those lanes. The RAs integrate these assigned priorities into their potential field computations. The modified potential field computations help the RAs to move through the intersection avoiding collisions. An elegant mixed autonomy scheme is thereby achieved where the IAs decide upon the priorities at the intersection while the low level collision avoidance maneuvers are left with the individual RAs. This scheme preserves the distributed nature and the autonomy of potential field maneuvers while simultaneously balancing the computation load between the IA and RAs. We compare this method with a method where the RAs navigate the intersection without a superior direction from the IAs through priorities or when IAs direct the RAs based on priorities computed on a first come first served basis. We show performance gain over both these methods in simulations.

Index Terms—Artificial potential field, Multi agent Systems, Intelligent Transportation System.

I. INTRODUCTION

In design of multi-agent systems one of the criteria of concern is the autonomy that is to be invested in the individual agents. Often it is unwieldy for an individual agent to possess all decision making capabilities required for achieving its goals. Decision making is entailed to be shared for efficient operation in an environment between agents at the same hierarchy or category and between different categories or hierarchies. In this paper we come up with what we feel is a practical application of mixed autonomy in coordinating traffic in an intersection. By mixed autonomy we mean that the intelligence of both the IA and RA are utilized in navigating the intersection. The autonomous decision making ability of the RA is shared with the ability of IA for intelligent navigation at the intersection.

We envisage a scenario where several autonomous vehicles called robotic agents (RA) are required to cross an intersection. The RAs are guided by the intersection agents

(IA) at the intersections. The IAs decide upon the priority of the RAs when they reach the intersection based on the density of RAs and the traffic flow rate in the lanes. The RAs respect the guidelines of the IA by incorporating them into their potential field framework while preserving their autonomy at the lower collision avoidance level. In other words the immediate decisions of where and how to move next to avoid collisions with other agents at the intersection is decided by the RAs while leaving the higher level decision of how to organize the traffic at the intersection to the IA. This results in smooth and graceful collision avoidance maneuvers at the intersection. Simulation results portray that the average time spent at the intersection is less through this mixed autonomy model where the IA and RAs share decision making than when all the decision making capability is left with the IA. In the later case the RAs move purely based on their potential field strategy without any scope of a priority based guidance through the intersection. This is because that the RAs are not in a position to gather data about the traffic in various lanes that approach the intersection while the IA is specifically suited for such data gathering role. Secondly we also show that the method of allotting priorities based on lane traffic and densities [1] weaves well with the potential field method than priorities assigned by a first come first served policy. This result is an extension of the previous results of traffic coordination [1] where we had shown priorities assigned based on traffic densities results in faster traffic flow at an intersection than a method based on reserving spaces at the intersection [2].

The essential novelty of this effort lies in an efficient integration of priorities to a collision avoidance framework. Previous efforts [1], [2] had divested most of the traffic control computations and navigation decisions with the IA. For example in [1] the IA computed the path of an RA through the intersection avoiding collisions with the other RAs at the intersection, whereas in [2] the IA reserved paths for an RA through the intersection making sure that the reserved path does not have a space-time collision with reserved paths of other RAs. In this effort same performance is achieved by reducing the computational burden of the IA

to that of assigning priorities alone, the collision avoidance is achieved through a distributed potential field framework. This we argue is more realistic where the IA provides only the guidelines and the RA integrates these guidelines within their navigation framework. The second novelty is to position the need for an IA. An IA with access to traffic densities and rates in different lanes can better guide a set of robots through an intersection than RAs who are required to navigate through the intersection without any guidance from an overseer of the intersection. The performance gain due to the guiding role of an IA is vividly portrayed in the simulation section.

The results of this effort find applications in the areas of Intelligent Transportation Systems, traffic scheduling systems and especially in futuristic scenario where many autonomous vehicles navigate the roads.

II. LITERATURE REVIEW

In the areas of intelligent transportation systems a lot of literature deal with reducing congestion in lanes and efficient routing of vehicles along the city arteries [3], [4], [5], [6]. However work relating to routing traffic through an intersection has been rare bearing [1], [2], [7]. However many of these approaches work on human guided vehicles and there does not seem to be an effort that integrates these approaches to a robotic navigation framework. Similarities of this effort are also with network mediated robot navigation [8], [9], [10]. Here the robotic agent takes help of a sensor network to decide where and how to move. As per the survey of authors there has not been an effort where the sensor network routes multi robotic traffic in some optimal fashion.

III. PROBLEM FORMULATION

Given: An intersection in a planar world with 2 or more roads leading into it. Robotic agents, moving at different velocities, crisscross at the intersection. An intersection agent monitors the intersection and is capable of communicating with the robotic agents.

Objective: To guide the robots across the intersection such that robots do not collide and the sum over the time spent by each robotic agent in crossing is reduced.

Assumptions:

- a Number of robotic agents is not fixed and they can be introduced in any pathway till such time there is no place to spawn any further robotic agents due to lack of space or congestion in that pathway.
- b Each robot knows the rough coordinates of the area it has to reach after crossing the intersection.

Assumption ‘a’ is often used in agent community [2], [7]. It serves as a yardstick for evaluating the control mechanism. It is a welcome assumption more than anything.

Assumption ‘b’ serves in giving each robot a destination at

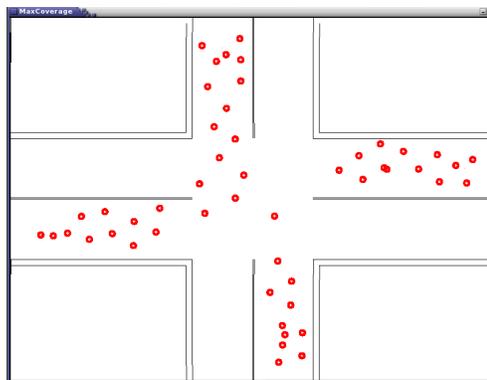


Fig. 1. Robots approaching the intersection

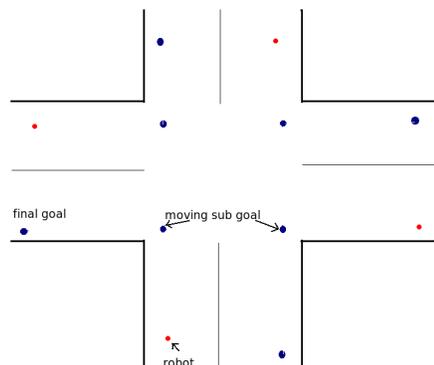


Fig. 2. Robots approaching the intersection

the intersection. Coordinates can be known either through GPS or by querying the Intersection agent.

IV. MOTIVATION

The figure 1 shows a typical intersection, with four lanes leading into it, and multiple robots approaching it from all sides. Clearly, A collision avoidance mechanism has to be employed at the intersection in order for the free flow of the traffic. This can be done by planning the paths of the robots with a look-ahead and making sure no two robots occupy the same space-time at the intersection. However, this scheme, puts more burden on the Intersection Agent as the computation complexity and the number of messages that have to be passed between the IA and the RAs are large. Hence, there is a need for a mixed autonomy where the IA is not overwhelmed as the robots themselves avoid collisions. We use the potential field method because of its ease of implementation and its practical applicability. But a pure potential field method would just ensure that robots would not collide with each other. For a better throughput of traffic we integrated a priorities based robot-movement mechanism into the potential field method.

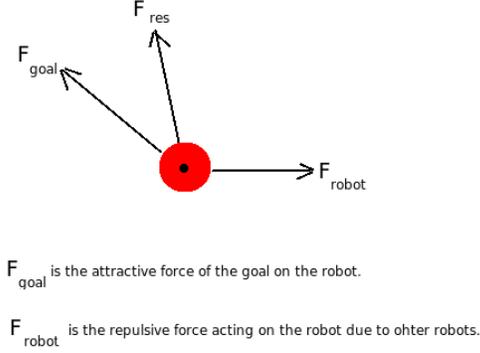


Fig. 3. The forces acting on the Robot

V. THE METHODOLOGY

Potential field-based techniques have been used extensively to solve navigation problems in mobile robotics. In these methods, virtual potential fields are used to represent goals and constraints and the control law for the robots motion is formulated such that it moves from a high potential state to a low potential state similar to the way in which a charged particle would move in an electrostatic field. The robots can be guided to their destination using the potential field method. The main idea is to define potential zones in the environment and to guide the robots to the least potential zone. The interactions between the various objects of interest decide the next move of the robots. The various forces on the robots are calculated and the motion of the robot is decided. We explain the different types of forces acting on the robot in the next subsection. In our approach we define moving sub-goals which start at various points in the intersection and progress towards the final goal which is the destination lane. The figure 2 shows the respective sub-goals of the robots. This moving sub-goal ensures a curve like path which is a more realistic path for a car like robot. The sub-goal creates a low potential zone which traverses towards the final destination lane. The robots get attracted into this moving low potential zone and finally reach the desired lane. The following section details the various forces that act on a robot.

A. Forces

The forces acting on the robot are of two basic types, the attractive force of the (sub)goal and the repulsive force of the obstacle. The figure 3 shows the different forces acting on the robot. The goal here is the respective lanes into which the robots want to enter. The obstacles in this case would be the road walls and the other robots traveling in the lane. The attractive force of the sub-goal creates a low potential zone into which the robot which is at a higher potential gets attracted to. The obstacle's repulsive force creates a high potential zone which repels the robot which is at a lower potential. Both the forces are inversely proportional to the

square of the distance between the two interacting bodies. The upper limit of the magnitude of the forces are clipped to avoid the forces approaching infinity when the distance tends to zero. The sub-goal moves towards the final goal at a constant velocity which is equal to the velocity of the robot following the sub-goal. Once the sub-goal is in the close proximity of the final goal, it merges with the final goal.

$F_{robot-robot}$ is the force between two robots. This force is always repulsive.

$F_{robot-obstacle}$ is the force between the robot and the lane walls. This force is always repulsive.

$F_{robot-subgoal}$ is the force between the robot and the sub-goal. This force is always attractive.

$$F_{robot-robot} = \frac{K_1}{d^2} \times \frac{(r_1 - r_2)}{\|r_1 - r_2\|} \quad (1)$$

Here the d is the distance between the two robots with the position vectors r_1 and r_2 . The force is a repulsive force which acts on both the robots in the direction of the line joining the two position vectors.

$$F_{robot-obstacle} = \frac{K_2}{d^2} \times \frac{(r_1 - r_o)}{\|r_1 - r_o\|} \quad (2)$$

This is a repulsive force which acts on the robot in the direction of the perpendicular line joining the robot and the obstacle. Here d is the perpendicular distance of the robot from the obstacle. r_1, r_o are the positions of the robot and the obstacle respectively.

$$F_{robot-subgoal} = \frac{K_3}{(maxD - dGoal)^2} \times \frac{(r_{Sgoal} - r_1)}{\|r_{Sgoal} - r_1\|} \quad (3)$$

A parameter $maxD$ is introduced to put an upper limit to the force as the force approaches infinity as the robot approaches the goal. $dGoal$ is the current distance between the sub-goal and the robot. r_{Sgoal} is the position vector of the sub-goal and r_1 is the position vector of the robot. This is an attractive force which acts on the robot in the direction of the line joining the position vectors of the robot and the sub-goal.

K_1, K_2, K_3 are suitable constants which act as weights to the force functions.

In this sub-section we defined the different forces acting on the robots. In the subsequent section we detail how priorities are assigned to the robots.

B. Priority Assignment

The intersection agent knows the density of robots approaching it in each of the pathways that lead to it. The intersection agent maintains the list of robots corresponding to a pathway; the list is updated every time a new robot comes its way. The intersection agent also calculates the rate

of change in densities from the list of robots it has. Having aggregated info from all the pathways it assigns priorities to them. First the pathways are clustered based on the density values as high density and low density clusters. Among the clusters with high density the pathways are ranked on increasing order of rate of change of density. This process is repeated for clusters classified as low density clusters. Thus the pathway with highest density and lowest rate of change of it gets the top most rank or priority since this is a situation corresponding to congestion. Within a pathway the agents are ranked based on their closeness to the intersection. The first n_a number of them are ranked and then ranking of agent in the next pathway is proceeded. All the agents in a lower ranked pathway have ranks lower than those in a higher priority pathway. We say agent a_x has a lower rank than agent a_y if the value of the rank of a_x , $r(a_x)$ is actually higher than the value of rank of a_y , $r(a_y)$. This process of priority assignment is repeated every t samples by the intersection agent so that there is no starvation at the intersection. Algorithm 1 gives the sequence of steps that are implemented by the intersection agent to assign priorities.

Algorithm1: Priority Assignment

- 1) Let $d_1, d_2 \dots d_m$ be the number of robots approaching in the m pathways leading to the intersection respectively currently, at time $t = t_1$.
- 2) Let $dp_1, dp_2 \dots dp_m$ be the number of robots in the pathways at time $t = t_1 - t$.
- 3) Let β be the parameter used for segregating high density pathways from low density pathways.
- 4) Repeat for every t time samples.
- 5) Calculate the rate of change of density, $rd_i = d_i \cap dp_i$, for $i = 1, \dots, m$.
- 6) Group all pathways that have robot densities higher than β and group the rest.
- 7) Assign Priorities to each of the pathways in the high density group based on their corresponding rate of change of density, rd_i . The pathways with the lowest rate of change of density gets the highest priority. Repeat the process for the low density group such that the priorities of the low density group are lower than those of the higher density group.
- 8) Assign ranks to robots according to the priority of the pathways in which they are.

In this subsection we described how priorities are assigned to the robot based on which the motion of robots is modified. This procedure, of integrating priorities into the potential field forces resulting in path modification, is detailed in the following subsection.

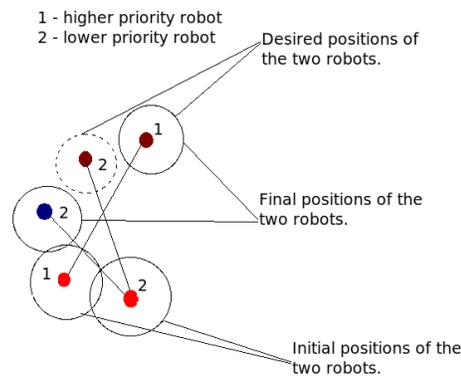


Fig. 4. The modified motion of the robots due to priorities

C. The Algorithm: Integration of priorities in Potential Field Forces

The robots are assigned priorities based on the above mentioned policy as they are about to enter the intersection. Once the priorities are assigned, the robots continue navigating across the intersection. If a higher priority robot encounters a lower priority robot at time t , the former's next move will be computed and carried out while the latter's next move is not computed and its motion will remain the same as that was in its previous step. The net force on the robot with the lower priority will remain same for this time instant. For the robot with the lower priority,

$$F_{net}(t) = F_{net}(t + 1) \tag{4}$$

This is shown in the figure 4. The figure shows a black circle circumscribing the robot which is the sensor range of the robot. In the figure, robots 1 and 2 are in conflict with robot-1 having higher priority than that of robot-2. Now, as indicated in the figure, robot-1, with higher priority moves to its desired position or the position that has been computed for it to move to in the next simulation step. But robot-2 with lesser priority is unable to move to its desired position as its next move has not been computed and it continues on its previously computed path.

D. Other Priority Assignment Strategies

Robots approaching the intersection could be prioritized based on the time at which they arrive at the intersection i.e. robots are given priorities based on a first-come-first-served policy. Robots which arrive at the intersection first are given higher priorities than those which come after wards. This policy was compared against the priority assignment policy we mentioned in the previous sections. Also the proposed potential field based coordination can be carried out without assigning any priorities. The results section shows the statistics of the comparison between the three methods.

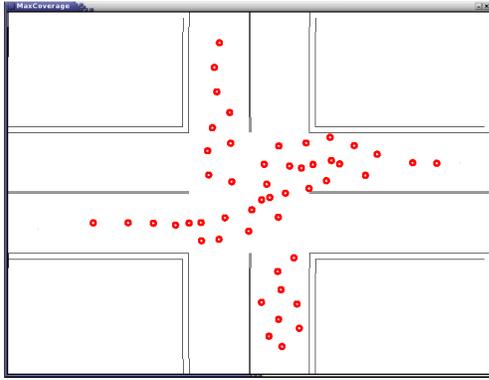


Fig. 5. The simulator in action

E. Discussion

In this section we have described all the algorithms that are required to implement the proposed methodology. We also described the various forces that would be guiding the robots across the intersection. The potential field method gives a realistic way of collision avoidance while the priorities make sure that traffic throughput is high. Our method of priority assignment would make sure that robots that come from high density lanes move faster, thus, reducing congestion. The graphs and statistics shown in the results section substantiate our claim. We have also included a section on the simulator that we have built to test the proposed method.

VI. SIMULATOR

To verify our algorithm we developed a simulator using OpenGL that could simulate the various forces described in the earlier sections. The simulator shows the four main lanes entering into the intersection. The four main lanes are sub divided into two one way lanes, the outgoing lane and the incoming lane. The figure 5 shows a snapshot of our simulator in action. The resolution of the map we used for the simulation was 800×600 . The size of the intersection is 200×200 . Each robot is circular with a radius of 10 pixels. The basic unit of time in the simulator is one simulation step and we define the acceleration, velocity in terms of the pixels covered in one simulation step. We calculate the accelerations of the robots based on the *Force* functions defined in the previous section. We limit the acceleration to 4 pixels per unit time square. We limit the velocity to 7 pixels per unit time. The angular acceleration of the robot is limited to $\pi/12$ radians per unit time square. The following actions take place in the simulation:

- 1) Robots are continuously spawned in the lanes leading to the intersection and are randomly assigned the direction in which they have to turn. The robots are spawned at a safe distance (which is defined as 5 pixels) between each other.

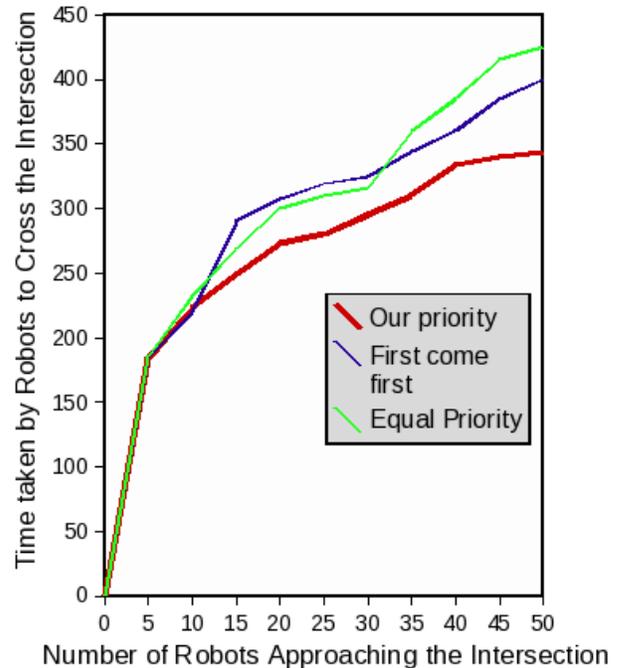


Fig. 6. Graph comparing the average time taken by robots in crossing the intersection

- 2) The robot knows the coordinates of the lane it needs to enter. This information can be got by using GPS or interacting with the IA.
- 3) The robot detects the obstacles based on the on board sensors and acts accordingly to avoid collisions and navigation. The actions taken are accelerate, decelerate, change direction, or continue going in the same direction.
- 4) The position, velocity and acceleration of the robot are updated after every simulation step.

VII. RESULTS

For showing the advantages gained by using our method we used the average time taken by the robots to move across the intersection. The simulation was run for 30 times to arrive at the average times taken by the robots to cross the intersection. The simulation was run spawning different number of robots to infer the trend of the average time taken with the increase in the number of robots. Figure 6 shows the comparison between all the three mentioned methodologies namely, priority based coordination, coordination on first come first served basis and the last one when there are no priorities assigned to the robots. The graph plots the average time taken by the robots to cross the intersection as the number of robots that are approaching the intersection keeps increasing. The graph clearly shows that our method fares significantly better than the other two methods.

| | FCFS | Our Priority |
|-------------------------------------|-------|--------------|
| No. of Robots Observed | 500 | 500 |
| Mean time to cross the intersection | 400 | 340 |
| Standard Deviation | 28.10 | 19.49 |
| Variance | 790 | 380 |

TABLE I
 $t = 39.2, df = 998, p < 0.001$

A. Statistical Significance

We used the Student's t distribution to verify the statistical significance of our results. We compared the two policies by using the mean time taken by a robot to cross the intersection. We used the Research hypothesis, 'The time taken by a robot to cross the intersection when FCFS based policy is used, is more than the time taken when our method of priority assignment is employed' and thus the null hypothesis H_0 : 'There is no difference between the two policies'. Table I shows the results used for testing the statistical significance. 500 robots were observed and the time each robot took to cross the intersection was recorded. This was done for both the policies. The table shows the mean, standard deviation and variance of the recorded data. Based on the data, the degrees of freedom, df , were $998(500 + 500 - 2)$ and t was 39.2. For the calculated df & t , we found that we could reject H_0 at a 0.001 level of significance i.e. the probability of the null hypothesis, that there is no difference between the two policies, being true is less than 0.001 or, in other words, the probability of the research hypothesis, that our priority based policy works better than FCFS based policy, being true is more than 0.999. Thus, we conclude, the advantage gained by using our method is statistically significant.

VIII. CONCLUSION

The management of traffic at the intersections has become a very important problem to tackle. A smart solution for this problem is needed as never before due to the ever increasing traffic. A new strategy, based on potential field methods, has been proposed for a realistic coordination of multiple robotic agents at an intersection. The methodology integrates priorities, assigned to the robots based on current traffic density, with potential field to make the robots move across the intersection without collisions and faster. The algorithm is much more realistic compared to the work done earlier in the same field - the bottom level collision avoidance routine is left to the robots and the Intersection Agent computes the traffic density and assigns priorities to the robots which they respect and follow allowing a smooth coordinated traffic flow. Multiple simulation runs have shown the advantage gained by the current methodology over previous ones.

IX. FUTURE WORK

We could extend our work to intelligent lane changing of the robots in the lanes which would enable a better

intersection management. Assuming there are three sub lanes, the left going robots should eventually end up in the left most lane before entering the intersection and the straight going robots in the middle lane and the right going robots in the right most lane. As soon as the robots decide on the direction they are going to make at the approaching intersection, they should navigate to their respective sub lanes.

REFERENCES

- [1] D V Karthikeya Viswanath, K Madhava Krishna, "Sensor Network Mediated Multi Robotic Traffic Control in Indoor Environment", Proc. of ICAR, Jeju, Korea, 2007
- [2] Kurt Dresner, Peter Stone, "Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism", The Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004
- [3] A L C Bazzan, "A Distributed Approach for Coordination of Traffic Signal Agents", Autonomous Agents and Multi-Agent Systems, 10(2), pp. 131-164, 2005
- [4] D A Roozmond, "Using intelligent agents for urban traffic control systems", Proc. of the International Conference on Artificial Intelligence in Transportation Systems and Science, pp. 69-79, 1999
- [5] Bertini R L, "Toward Optimal Sensor Density for Improved Freeway Travel Time Estimation and Traveler Information", Intelligent Transportation Systems Conference, 2007.
- [6] Kuriyama H, Murata Y, Shibata N, Yasumoto K, Ito M, "Congestion Alleviation Scheduling Technique for Car Drivers Based on Prediction of Future Congestion on Roads and Spots", Intelligent Transportation Systems Conference, 2007.
- [7] Kurt Dresner, Peter Stone, "Multiagent Traffic Management: An Improved Intersection Control Mechanism", The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, 2005
- [8] Q. Li, M. DeRosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," in The 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03), 2003.
- [9] Maxim Batalin, Gaurav S. Sukhatme, and Myron Hattig, "Mobile Robot Navigation using a Sensor Network," In IEEE International Conference on Robotics and Automation, Apr 2004.
- [10] Peter Corke, Ron Peterson and Daniela Rus, Localization and Navigation Assisted by Cooperating Networked Sensors and Robots In International Journal of Robotics Research, 2005.