

Cleaning Task Planning for an Autonomous Robot in Indoor Places with Multiples Rooms

Paulo Pinheiro, Eleri Cardozo, Jacques Wainer, and Eric Rohmer

Abstract—This paper presents a mobile robot planning approach for solving the problem of indoor cleaning tasks. In general, the robot must find its pose first and only then move to the final destination to cleaning out. Our model works with a multi-level planning approach where the mission is treated online only. The user sets up the cleaning or the robot uses an agenda with predefined set of missions. POMDP plans are created for the localization, using the map and the robot specifications. The plans are created offline only once and used indefinitely regardless of missions. We will show the multi-level planning process where the robot finds its pose in the high level of representative rooms and then moves to the lower level to finding its precise pose. We demonstrated the approach with experiments on both simulator and real robot. The multi-level planning allowed the robot to find its pose and fulfill the tasks of the agenda faster while keeping the precision.

Index Terms—Cleaning robot tasking, localization, robotic planning, POMDP.

I. INTRODUCTION

An autonomous robot must know its pose to accomplish a specific cleaning task, such as heading out to a certain room and cleaning out spots on the floor. Robotic vacuum cleaners generally suffer from the lack of more accurate sensors such as laser rangefinders, turning the localization a more challenging process. The other challenge pops up from the limitation of some embedded processors, which possibly might not be able to treat a large set of states of real indoor scenarios.

In this paper, we are interested in solving the localization problem and tasking for this group of robot and their limitations. For this we will present a multi-level planning model that can be used by small-embedded processors to helping the robot to find its pose faster while using cheaper sensors, like ultrasound sonars.

We will also present a support for user's agenda in which the robot can choose the destination based on a predefined schedule. The agenda does not have to contain only the rooms and the hours to clean out them, but the residents' information, their rooms and the time at each resident is away from home. Thus the robot can clean out their rooms while they are not in

the site. As described in our previous work [1], we are also considering that the robot can head out towards its goals at the same time it is trying to find its pose. The localization plan is precomputed and embedded into the robot's processor, then it is mixed at execution time with the goal mission, executing both task and localization. The robot performing the cleaning tasks using the multi-level approach could accomplish the task faster than other robots using the comparative models. Fig. 1 shows the improved path of the proposed model to reach the destination. In this experiment, the start point is the room on the left; the room to be cleaned is the one on the upper-left corner; and the final destination is at the right room. Once the robot is at the desired room, the cleaning algorithm can be executed. In this work we will not discuss cleaning algorithms.

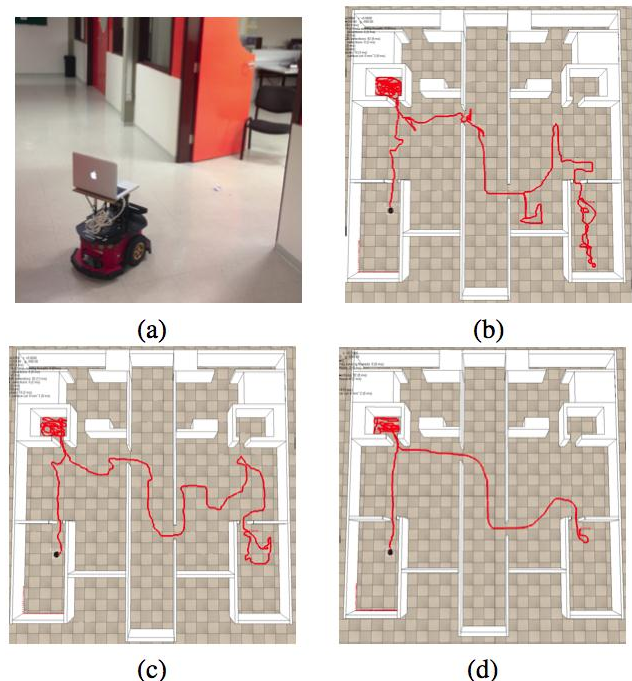


Fig. 1. (a) The Pioneer 3DX robot playing as robotic vacuum cleaner. (b) Localization using active Markov Localization. (c) Multi-level Bayesian network and particle filter approach. (d) POMDP proposed model.

II. RELATED WORK

In several works on localization problem, the robot performs actions in a random manner with no planning at all, since the mission of the robot is taken up as priority [2], [3], [4].

The observations and the movements of the robot over the environment are not always deterministic due to the uncertainties on the actions and observations. To handle with uncertainties problems, probabilistic localization approach

Manuscript received September 23, 2014; revised December 10, 2014. This work was supported in part by the FAPESP under Grant 2011/091137 and CNPq/MCT/FINEP under the Grant 385024/2013-4.

P. Pinheiro, E. Cardozo, and E. Rohmer are with the School of Electrical and Computer Engineering, University of Campinas, SP 6101 Brazil (e-mail: pinheiro@ic.unicamp.br, eleri@dca.fee.unicamp.br, erohmer@dca.fee.unicamp.br).

J. Wainer is with the Institute of Computing, University of Campinas, SP 13083-852 Brazil (e-mail: wainer@ic.unicamp.br).

has been heavily used [5], [2]. Extended Kalman Filter (EKF) is one of the most used probabilistic techniques for these problems. The authors in [6] applying EKF, they use sonars to find geometric landmarks over the map improving the estimate of pose. Reference [7] uses it to improve the accuracy on dead reckoning and odometry problems. Those two cases are example of the use of probabilistic techniques for both dead reckoning and landmark application. Bayesian networks have also been used to find solution for the robot localization problems. In [8], a Bayesian network is manually designed to use local information in order to find out the robot's pose in an indoor office environment.

The authors in [5] proposed an active model based on Markov Localization that uses the entropy of the probability distribution on a grid-based map to determine the pose of the robot. The entropy variance defines what action must be taken. A good action is the one that can reduce the entropy of the system, popping up a single or few peaks over the map where the robot could be. The advantage of this method is the accuracy, but not feasible for large-size maps.

For large-size environments, the computational complexity to deal with all states is a challenge itself, since the model represents the entire map using a single probability distribution. Also, in order to choose the best action, the equations must consider all states and using small cells, the number of states grows fast. If the precision is the major advantage of it, the drawback is to deal with this computational complexity.

To bypass the computational cost, the robot performs random wandering until the belief state is concentrated at few points of local maxima, then after converging, the model starts to use Markov Localization to find out the pose. Depending on the size and shape of the map, the local maxima points can take a long time to converge.

The work presented in [9] proposed a hierarchical approach for global localization problem using particle filter and Bayesian networks. On their model, the gross localization is based on large cells that are decomposed based on geometric features the robot can find over the map. On the lower level, the model uses the particle filters to find the posterior probability for localization. When these particles converge into clusters, the higher layer starts the particle clustering and sensor planning. The high layer uses Bayesian network to represent the decomposed cells, their relationships and the belief state. The sensor planning creates an optimal sequence for localization.

Both approaches apply several random actions at the beginning of the process that is not feasible to be used by limited-battery robotic vacuum cleaners. Even for the cases where the robot is not required to find its pose faster, the power consumption from the high number of steps performed could be an issue due the battery limitation of these robots.

We will propose a multi-level planning, which allows small-embedded processor to handle with the uncertainties of the environment and to planning for cleaning tasks while the robot finds its pose. The more planned and less random steps, the less the distance traveled and the power consumption. We will compare our proposed model to the model of [9] and to the precise module presented by [5].

III. MULTI-LEVEL APPROACH

The proposed approach has been organized in 3 distinct processes: 1) the map treatment process, 2) the multi-level planning, and 3) the combining localization and agenda process.

A. Map Treatment Process

Robotic vacuum cleaners are mostly used at homes and offices. These real home-like environments generally feature multiple rooms, which hampering the localization planned process due the high number of states. In this paper we will use the map compression proposed in our previous work [10]. Based on architectural features (walls, doors, and room area) the original map is divided into rooms. The similar rooms are clustered into groups by their similarity. Each group has its rooms overlapped creating only one representative room. Those new rooms (one representative room for each cluster) contain useful and compressed information to represent each of the original rooms. The new representative rooms are connected by a connection graph, creating the final topological map.

Let's assume f is a floor of a building, where f is made of the rooms $a_1, a_2, c_1, b_1, b_2, a_3$. The decomposition process will break the map up into rooms using the architectural features. The clustering process uses number of doors and total area of the room to find similar rooms and clustering them into different groups g_n . For the floor f , we would have the groups $g_a = \{a_1, a_2, a_3\}$, $g_b = \{b_1, b_2\}$, $g_c = \{c_1\}$. For each group g , a new representative room rg_n is created based on the combination of the group's elements. Thus, we will have rg_a, rg_b, rg_c , where rg_a is the representative room that represents all rooms of the group g_a , and the rg_b and rg_c , the representative rooms for the group g_b and g_c , respectively. Finally, a graph G is used to connect all the representative rooms. G represents a topological map, where the vertex set $V(G)$ is composed of the representative rooms and the edge set $E(G)$ is made of the transition vectors between the rooms, representing by the doors, that we will call "breakout points". Fig. 2 shows the Coffman Building inspired map with 11 rooms compressed into 6 representative rooms generating the final graph.

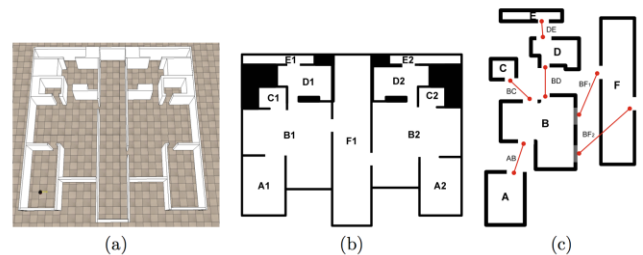


Fig. 2. (a) The Coffman Building inspired map running on the V-REP simulator. (b) The grid representation for the map. (c) The compressed graph representation.

B. Multi-Level Planning

The Multi-Level planning is based on two layers, as shown in Fig. 3. The higher layer deals with the gross localization at the level of representative rooms (described in the previous section). The lower level is used to find the robot's precise pose at level of cells of the original room.

On the higher layer, the localization is treated in the level of representative rooms, where the robot must find what

representative room rr it is, and its pose $(x_{rr}, y_{rr}, \theta_{rr})$ in this representative room.

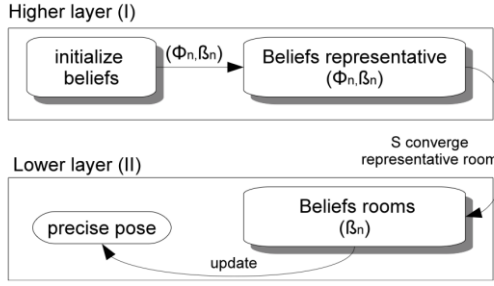


Fig. 3. Multi-level layers, where β is the belief, ϕ the current plan and the higher layers informs the lower layer about the representative room.

For the high layer we have the set of representative rooms $R_p = \{rr_1, \dots, rr_n\}$, and the list $P_1 = \{\phi_1, \dots, \phi_n\}$ where ϕ is a plan for each rr . Also, for each representative room, the belief state β is updated during the localization process for the n representative rooms, given by $B_1 = \{\beta_{rr_1}, \dots, \beta_{rr_n}\}$.

After initializing the beliefs, the process of localization picks out a plan ϕ_i , as the starting plan. The chosen plan ϕ_i is the correct one, when the robot is in the room r_o , the representative room of r_o is rr_i , and the plan for rr_i is ϕ_i . The initial plan is randomly chosen. The robot starts up to move over the environment based on the actions informed by the planner regardless whether it is using the correct plan or not. The entropy S for each representative room is updated and decreases converging to a point on the final representative room rr_c (where the robot might be at). Fig. 4 shows the multi-level approach, where rr_c is the representative room where the point converged.

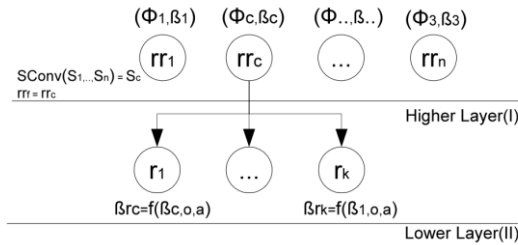


Fig. 4. Multi-level approach where the entropy S converges into the representative room rr_c exploding it to the original rooms r_1, \dots, r_k .

On the higher layer, the entropy will converge for one of the representative rooms, rr_c . At this point the robot is aware of which representative room it is, consequently the robot is in one of the similar rooms r_1, \dots, r_k .

On the lower layer, the process decomposes the representative room rr_c into its k original rooms r . Given the belief state β_c , the observations o and actions a taken, we can plot the belief state for all original room r , based on the backtracking given by $\beta_{r_n} \leftarrow f(\beta_c, o, a)$, where n is a room. The robot uses the higher probability point of the belief state as its starting point and heads out toward the breakout point to increase the pose accuracy, using Markov Localization or POMDP localization model (if the plan for the original room is available).

C. Combining Localization and Agenda Missions

In this work we are interested in extending the work of [1] where robot uses a plan to find its pose and also considers an

ordinary mission, combining both in an online process. Only one POMDP plan is generated for the map regardless the mission. We will use the same process described in the mentioned work, but this time applied to a low-cost robots with ultrasound sonars and odometry only. Here the mission might not be given by a manager, but by an agenda with predefined tasks. The robot consults the current time of the day and performs the mission. Fig. 5 shows the framework of the POMDP planning model for localization and the fusion with the agenda data, time of the day and the vector \vec{C} . This vector is a result of the combination of the localization vector \vec{L} with the mission vector \vec{M} .

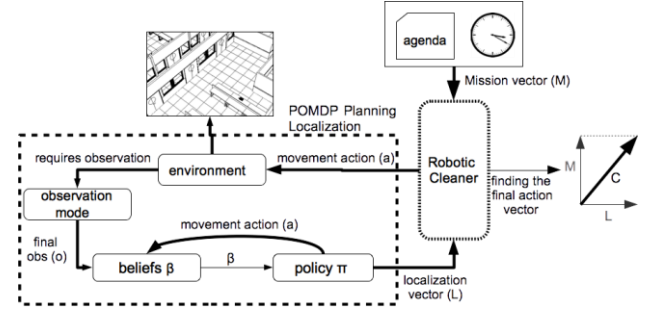


Fig. 5. Framework with the POMDP planning localization model giving the best action for localization (vector L), and the agenda and time of the day building up the mission vector.

The vector \vec{L} defines the optimal action for localization calculated by the POMDP planner. This action tries to maximize the robot's localization. The belief state β can point out the cell with the highest probability, i.e., the cell where the robot most likely to be at. Based on this, we assume the robot is really there, and calculate the vector \vec{M} towards the goal point using the A* algorithm.

The vector \vec{L} must be inversely proportional to the localization probability, $\vec{L} = (1 - P(l))$. The mission vector is defined as $\vec{M} = P(l)$. If the robot is aware about its pose, $P(l)$ is high, then $\vec{M} = P(l)$ as well. Otherwise, if $P(l)$ is low, \vec{L} will be high and \vec{M} is low, prioritizing the action for localization. The resulting vector \vec{C} defines the best action to be taken. It may happen a vector cancel out each other. In this case, the tasking vector turns null and the localization process takes the priority.

IV. EXPERIMENTS

For the experiments we will use the Pioneer 3-DX robot equipped with 12 ultrasonic sonars. For the simulation rounds we will use the V-REP simulator with a 3D model of the same robot.

Differently from the work of [1], the robot here does not have expensive laser rangefinders available, since the motivation is to use low-cost robots, such as iRobot Roomba robot.

We will compare the multi-level model to the comparative models described on the related work section, the F model [5] and to the Z model [9]. The cleaning mission are predefined and set up into an agenda. For the experiments we will count the number of steps for each mission. A single mission is a trip

from a room to other. The robot does not know your pose. Fig. 6 shows the maps used on the experiments: the Coffman building inspired map (University of Minnesota) and the Automation Lab (University of Campinas).

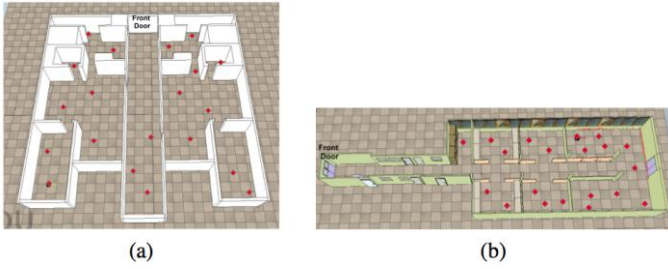


Fig. 6. (a) Initial points and the front door in the Coffman Building. (b) Automation Lab points and its front door.

The experiments with the real robot were held in the Coffman building inspired map, and the simulations experiments in the Automation Lab map.

We have run 20 rounds of 3 simulations, one simulation for each one of the 3 models (F, Z and M models). Every round places the robot at a different initial pose, but the initial poses of the same round are the same. We have repeated the experiment for each map and analyzed the results according to the student's t-test, simulation samples, and average gain. To apply the student's t test for paired samples, we have run 20 rounds of 3 simulations, totalizing 60 simulations total per map. Given the number of steps n_x (until the robot brings off the mission) of the most efficient model X, and the number of steps n_p of the proposed model M on a single round r , we calculate the ω_r factor, where $\omega_r = (n_p) / (n_x)$. For all k rounds r , we also calculate the average of the factors ω , given by: $\eta = (\sum \omega_r) / k$, and the average gain, $g = (1 - \eta)$.

V. RESULTS

Fig. 7(a) shows the number of steps the robot performed to find its pose (only localization) in the Coffman Building Map. The robot using the proposed model could perform the localization task using fewer steps than the comparative models. The model of [9] obtained the best results on the simulations #10, #14 and #20.

Fig. 7(b) shows the results for both, localization and mission in the same map. In this case, 20 more rounds were executed selecting the missions available on the robot's agenda. Considering both, localization and cleaning mission, the line of the proposed model moves away from the line of the other models. Using the proposed model, when the robot finds its pose, it is closer to the final destination. Similar behavior was found in a previous work [1] for robots fulfilling ordinary tasks.

By conventional criteria, the paired two-tailed P less than 0.0001 is considered extremely statistically significant. The robot using the proposed model could find its pose with an average gain of 26.3% comparing to the robot using the Z model. When considering both, localization and mission, the average gain was even better, 32.5%, i.e., the robot took 32.5% fewer steps to get the mission done than the robot using

the Z model. Fig. 8 shows the result for the Automation Lab map.

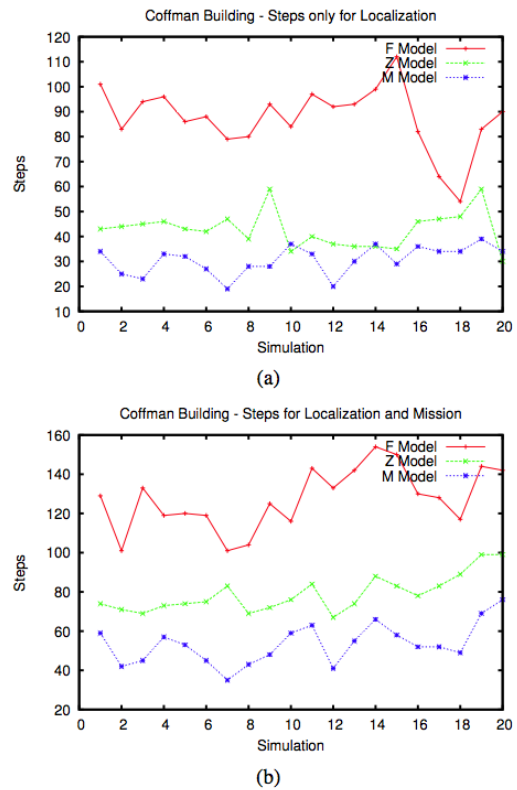


Fig. 7. (a) Steps taken only for localization in the Coffman Building. (b) Steps for both localization and mission.

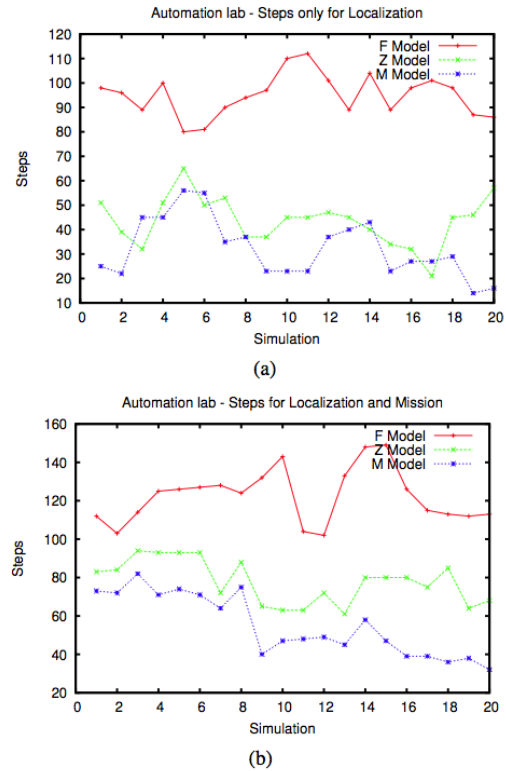


Fig. 8 (a) Steps taken only for localization in the Automation Lab. (b) Steps for both localization and mission.

For this map also the two-tailed P value was less than 0.0001 - extremely statistically significant difference between the proposed model and the Z model. The robot using the proposed model in the Automation lab map had an average

gain of 23.0% only for localization, when compared to the best comparative model, again the Z model. For both, the localization and mission, the average gain was 29.8%.

VI. CONCLUSIONS

In this paper we have proposed a multi-level planning for cleaning tasks to be performed by low-cost robotic vacuum cleaners. So far, these robots have in general low precision sensors, which generate a large amount of uncertainties over actions and observations. We proposed a POMDP model as a solution to deal with these uncertainties and to point out the best action for localization and cleaning mission. The original map is broken up into rooms and similar rooms are clustered generating representative rooms in a topological map. The planning model has two layers. In the higher layer the robot finds the representative room where it is. Then, in the lower level the robot find its precise pose. The results shown that even only for localization, the proposed model obtained better results, precisely because the POMDP model itself is capable to deal with the uncertainty of the real world. When considering both localization and mission the results were better also for low-cost robots, especially due the combination of the mission and localization vectors, similar approach used in other works. In a future work we will investigate the performance of the cleaning algorithm themselves when pair with our algorithms of reaching destinations and finding the robot's pose.

REFERENCES

- [1] P. Pinheiro, E. Cardozo, J. Wainer, and E. Rohmer, "Arrangement map for task planning and localization for an autonomous robot in a large-scale environment," in *Proc. IEEE International Conference on Control Science and Systems Engineering (CCSSE 2014)*, to be published, 2014.
- [2] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, pp. 99–141, 2001.
- [3] J. Castellanos, J. Neira, and J. Tardos, "Multisensor fusion for simultaneous localization and map building," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 908–914, 2001.
- [4] S. Koenig and R. G. Simmons, "Xavier: A robot navigation architecture based on partially observable Markov decision process models," *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, pp. 91–122, MIT Press, 1998.
- [5] D. Fox, W. Burgard, and S. Thrun, "Active Markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.
- [6] J. Leonard and H. F. Durrant-White, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.
- [7] S. Roumeliotis and G. Bekey, "Localization for mobile robot teams using maximum likelihood estimation," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 781–795, 2002.
- [8] H. Asoh, Y. Motomura, I. Hara, S. Akaho, S. Hayamizu, and T. Matsui, "Combining probabilistic map and dialog for robust life-long office

navigation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996, vol. 2, pp. 807–812.

- [9] H. Zhou and S. Sakane, "Sensor planning for mobile robot localization – A hierarchical approach using Bayesian network and particle filter," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 350–356.
- [10] P. Pinheiro and J. Wainer, "A mixed map representation approach for mobile robot localization planning," in *Proc. International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2013, pp. 401–408.



Paulo Pinheiro was born in Brazil, who received his Ph.D. in computer science in 2013 from University of Campinas (BRA)/University of Minnesota (USA), in a sandwich doctorate program. At the University of Minnesota, Dr. Paulo has worked on detection and estimation theory for mobile robot localization. Most of the works were developed at the Multiple Autonomous Robotic Systems (MARS) Laboratory. In his current research at University of Campinas, he is developing mobile robot solutions for a low-cost powered robotic wheelchair.



Eleri Cardozo was born in Brazil, who is a professor at the State University of Campinas. In 1987, he received his Ph.D. in electrical and computer engineering at the Carnegie Mellon University. He has wide experience in robotics, computer networks, distributed systems and software engineering. Currently, Dr. Cardozo is developing researching on assistive robotics and solutions for autonomous or semi-autonomous powered wheelchairs focusing on navigation problems to be used by Patients with neurological disorders.



Jacques Wainer was born in Brazil, who is an associate professor in computer science at State University of Campinas. He has finished his Ph.D. in 1991 at the Computer Science Department at the Pennsylvania State University. He has published several works on artificial intelligence and machine learning in the last decade and he has been working on planning for mobile robot localization.



Eric Rohmer is a French researcher who is working at the Faculty of Electrical and Computation Engineering of the State University of Campinas in Brazil. He concluded his Ph.D. in 2005 in the Tohoku University in Japan, where he worked as a robotic researcher until 2011. His field of interest concerns dynamic simulation based telerobotic platforms, mobile robots locomotion for space exploration and search and rescue operations, and more recently assistive robotics. He is one of the researchers who designed and developed Quince robot, the first Japanese robot in use inside the Fukushima crippled nuclear reactor.