



A Novel Resampling Method for Particle Filter for Mobile Robot Localization

Widyawardana Adiprawita¹, AdangSuwandi Ahmad², Jaka Sembiring³,
and Bambang R. Trilaksono⁴

School of Electrical Engineering and Informatics, Bandung Institute of Technology

Ganesha 10, BandungIndonesia

¹wadiprawita@stei.itb.ac.id

²asaisrg@stei.itb.ac.id

³jaka@stei.itb.ac.id

⁴briyanto@lskk.ee.itb.ac.id

Abstract: This paper present a particle filter for mobile robot localization also known as Monte Carlo Localization (MCL) to solve the localization problem of autonomous mobile robot. A new resampling mechanism is proposed. This new resampling mechanism enables the particle filter to converge quicker and more robust to kidnaping problem. This particle filter is simulated in MATLAB and also experimented physically using a simple autonomous mobile robot built with Lego Mindstorms NXT with 3 ultrasonic sonar and RWTH Mindstorms NXT Toolbox for MATLAB to connect the robot to MATLAB. The particle filter with the new resampling algorithm can perform very well in the simulation as well as in physical experiments.

Keywords: Autonomous mobile robot, Monte Carlo Localization, particle filter, resampling, LEGO Mindstorm NXR, RWTH toolbox.

1. Introduction

In the last decade, there has been so many progress in autonomous mobile robot's development in Indonesia. This progress is partly driven by the enthusiasms of the Indonesia autonomous mobile robot competition (Kontes Robot Cerdas Indonesia, KRCI), held annually in regional and national level. The winners of this competition are sponsored to compete in international competition such as Trinity College Fire Fighting Robot Contest, and in 2011, Indonesia's robots won first and second rank for both wheeled and legged robot. This results show the competitiveness of KRCI. This event is sponsored by Ministry of Education of Indonesia. Many of this development is focused on mechanical platform development and ad hoc algorithm to solve the mission in the competition, so the resulted algorithm is only suitable specifically for the competition. This ad hoc algorithm is mainly developed to solve the localization and navigation problem in the competition.

Autonomous Vehicle Research Group (AVRG), a research groups of School of Electrical Engineering and Informatics, Bandung Institute of Technology conducts several research in autonomous mobile robots field. One of the research is probabilistic robotic architecture. The motivation of this research is to provide a solid robotic framework for future research, not just ad hoc algorithm for specific mission and situation, especially for localization and navigation for autonomous mobile robot.

2. Probabilistic Robotic Architecture

Building autonomous robots has been a central objective of research in artificial intelligence. Over the past decades, researchers in AI have developed a range of methodologies for developing robotic software, ranging from model-based to purely reactive paradigms. More

than once, the discussion on what the right way might be to program robots has been accompanied with speculations concerning the very nature of intelligence per se, in animals and people (Thrun, 2000).

One of these approaches, probabilistic robotics, has led to fielded systems with unprecedented levels of autonomy and robustness. While the roots of this approach can be traced back to the early 1960s, in recent years the probabilistic approach has become the dominant paradigm in a wide array of robotic problems. Probabilistic algorithms have been at the core of a series of fielded autonomous robots, exhibiting an unprecedented level of performance and robustness in the real world. These recent successes can be attributed to at least two developments: the availability of immense computational resources even on low-end PCs and, more importantly, fundamental progress on the basic algorithmic and theoretical levels(Thrun, 2000).

So what exactly is the probabilistic approach to robotics? At its core is the idea of representing information through probability densities. In particular, probabilistic ideas can be found in perception, i.e., the way sensor data is processed, and action, i.e., the way decisions are made :

Probabilistic perception. Robots are inherently uncertain about the state of their environments. Uncertainty arises from sensor limitations, noise, and the fact that most interesting environments are - to a certain degree - unpredictable. When “guessing” a quantity from sensor data, the probabilistic approach computes a probability distribution over what might be the case in the world, instead of generating a single “best guess” only. As a result, a probabilistic robot can gracefully recover from errors, handle ambiguities, and integrate sensor data in a consistent way. Moreover, a probabilistic robot knows about its own ignorance - a key prerequisite of truly autonomous robots.

Probabilistic control. Autonomous robots must act in the face of uncertainty - a direct consequence of their inability to know what is the case. When making decisions, probabilistic approaches take the robot’s uncertainty into account. Some consider only the robot’s current uncertainty, others anticipate future uncertainty. Instead of considering the most likely situations only (current or projected), many probabilistic approaches strive to compute a decision-theoretic optimum, in which decision are based on all possible contingencies. The proposed robotic architecture will consist both probabilistic perception and probabilistic control. Here is the diagram of the architecture

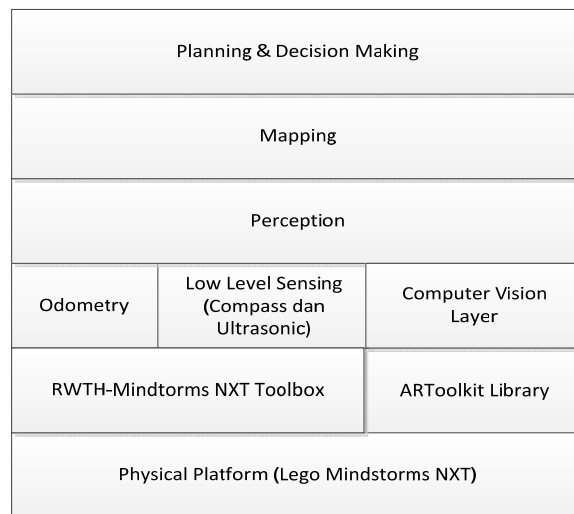


Figure 1. The proposed robotic architecture

The focus of this paper is the perception layer, which will be implemented as particle filter localization or Monte Carlo Localization.

3. Particle Filter Localization

A. Particle Filter Concept

Throughout this paper, we consider the problem of estimating the pose x of a robot relative to a given map m using a particle filter. The key idea of this approach is to maintain a probability density $p(x_t|z_{1:t}, u_{1:t})$ of the pose x_t of the robot at time t given all observations $z_{1:t}$ and control inputs $u_{1:t}$ up to time t . This probability is calculated recursively using the Bayesian filtering scheme (Rekleitis, 2003).

$$p(x_t|z_{1:t}, u_{1:t}) = \eta \cdot p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) \cdot p(x_{t-1}) dx_{t-1} \quad (1)$$

Here, η is a normalizer that ensures that $p(x_t|z_{1:t}, u_{1:t})$ sums up to 1 over all x_t . The term $p(x_t|u_t, x_{t-1})$ is the motion model and $p(z_t|x_t)$ the sensor model, respectively.

For the implementation of the described filtering scheme, we use a sample based approach which is commonly known as Monte Carlo localization. Monte Carlo localization is a variant of particle filtering where a set M of weighted particles represents the current belief. Each particle corresponds to a possible robot pose and has an assigned weight w_i . The belief update from the above equation (1) is performed according to the following three alternating steps:

1. In the prediction step, we draw for each particle a new particle according to the motion model $p(x_t|u_t, x_{t-1})$ given the action u_t .
2. In the correction step, we integrate a new observation z_t by assigning a new weight w_i to each particle according to the sensor model $p(z_t|x_t)$.
3. In the resampling step, we draw a new generation of particles from M (with replacement) such that each sample in M is selected with a probability that is proportional to its weight.

B. Particle Filter Implementation

The prediction step of the particle filter will utilize the motion model of the mobile robot platform. This motion model is involving an odometry calculation which is proved to be quite accurate. The correction step of the particle filter will utilize raw measurement from 3 ultrasonic sensor. This approach is quite uncommon for particle filter localization, usually sensors with more spatial coverage is used, such as laser scanner with 180 degree of coverage (identical with approximately 180 ultrasonic sensors). There is no option to add more sensor, since the used platform (Lego Mindstorms) has only limited sensor port.

From several experiments conducted using standard particle filter, the localization filter is either diverge or takes too long to converge. This standard particle filter utilize Select with Replacement (SWR) resampling and Linear Time Resampling (LTR). The SWR and LTR resampling algorithms are two commonly used resampling algorithm for particle filter (Rekleitis, 2003). The particle filter utilizing LTR or SWR resampling with 3 ultrasonic sensor will only converge if the initial state of the particles are correct (Bravo et.al, 2007), which usually not possible in many cases.

So, in this research we try to cope the problem by modifying the particle filter. There are two approaches taken :

- Developing new resampling method, and
- Implementing a correction step utilizing single hypothesis measurement (on heading and position state).

This paper will only discuss the first approach.

4. New Resampling Algorithm

From particle filter localization experiments with SWR and LTR resampling conducted early in this experiment, for mobile robots that has only 3 ultrasonic sensor, we found several problems :

- Initial state of the robot has to be known prior to particle filtering. Fail to fulfil this requirement will result in state divergence.
- There is steady state error which is sometimes not consistent. The running time of the particle filter does not guarantee that the steady state error will be reduced.
- The particle filter needs very long time to survive from robot kidnaping problem, and sometimes does not survive at all. Robot kidnaping problem is a special situation when a robot is moved from its current position, intentionally or unintentionally.

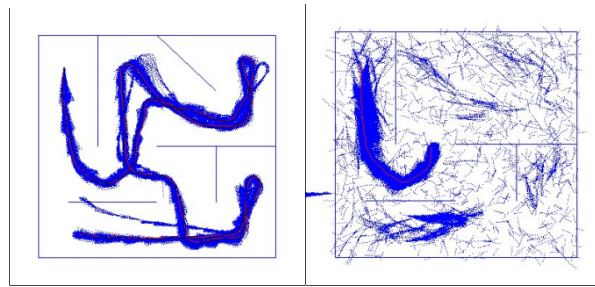


Figure 2. (a) on the left is the result of particle filter experiment with correct position state initialization (b) on the right is the result of particle filter experiments with random position initialization (using LTR resampling)

There are several problem related to resampling algorithm that has been identified as source of these problems :

- Particles are only duplicated from the previous step, this is suspected to cause the steady state error, because the particles with large weight are not changed
- There is no systematic mechanism to acquire new particles which has the potentials to help in robot kidnaping problem.

Based on those identified problems, a new resampling algorithm for particle filter is proposed. The basic idea of this new resampling algorithms is that in each resampling step, a new set of particles are always drawn from the distribution, so there are no exactly the same particles. This approach is expected to solve the steady state error problem. There are two rules of this new resampling algorithm :

- P number of the new particles are drawn around the old particles with the highest particle weight. The number of the new particles is proportional with the weight of the old particles. One way to do this is to normalize the weight of the old particles by multiply the weight with particles number. The state value of the new particles are not exactly the same as the old particles state, but close. This approach is to solve the steady state error problem.
- The rest of the $100\%-P$ new particles are drawn randomly with uniform distribution from the robot's configuration space. This approach is expected to solve the robot kidnaping problem.

We have to choose P , and from the experiment, P equals to 90% of total particles population gives a very good result. Here the new resampling algorithm is presented (W is an array which contains the weight of the particles, X is an array which contains the state of each particles, and the outputs of this algorithms are W_{new} and X_{new}).

Table 1
New Resampling Algorithms

```

Require  $\sum_{i=1}^N W_i = 1$ 

{ resampling of P = 90% new particles around old particles with largest weight }
[W_descIdx] = sort(round(partikels_weight*N), 'descend');
W_desc = W_desc * N; { normalized to total particle count }
i = 1;
j = 1;
N10 = N / 10; { 1-P = 10% of population }

while (i <= (N-N10)) && (j <= N)
for k = 1 to W_desc(j)
X_new(i) = draw new state around old state X(j);
{ X_new(i) = X(j) + small random noise }
W_new(i) = W(Idx(j));

    i = i + 1;

    if i > (N-N10)
        break; { 90% of particle population }
    end
end;
j = j + 1;
end

{ add 10% of particles, uniformly distributed }
for i=(N-N10+1) to N
X_new(i) = draw new random particle from config space;
W_new(i) = 1/N;
end
W_new = W_new / sum(W_new); { weight normalization}

return [W_newX_new];

```

5. Experiment Platform

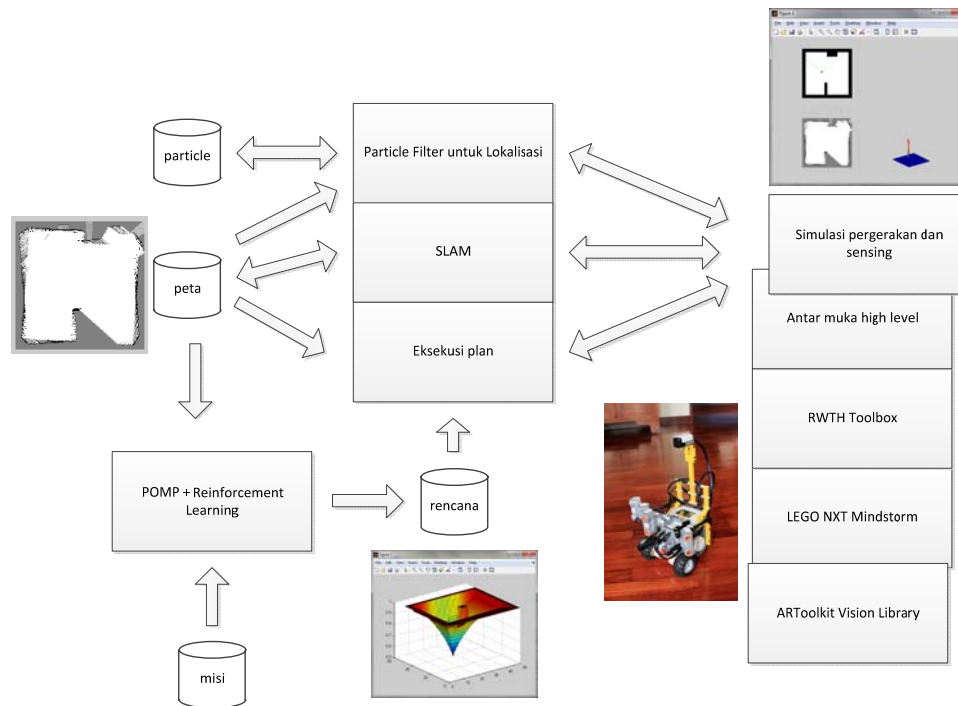


Figure 3. Architecture of experiment platform

The experiment platform for this research is built using LEGO Mindstorms NXT. LEGO Mindstorms NXT is a modular low cost hobby-educational robotic platform. The main reason of this selection is low cost, simplicity, and modularity of this platform. There are 3 kind of sensors used in this platform for this research :

- 3 ultrasonic range sensors for raw range measurement. This sensor is arranged to face left, forward and right direction, with exactly 90 degrees angular gap between them. The main purpose of these 3 ultrasonic is for correction step in the particle filter
- Compass sensor for heading measurement. This sensor is placed in far top position, to avoid any interference from motor actuator.
- CMOS color camera with wireless analog transmitter for vision sensing. This sensor is placed in front of the robot facing forward.

The actuator used in this are two servo motors with differential drive configuration.

For controlling the mobile robot, RWTH Mindstorms NXT Toolbox is utilized. This software toolbox enables us to control the LEGO Mindstorms NXT from MATLAB environment running on PC. PC running MATLAB and the toolbox can communicate to the LEGO Mindstorms NXT using bluetooth wireless connection, with 115200 baud rate connection speed. By using MATLAB in conjunction with RWTH Mindstorms NXT Toolbox and bluetooth wireless connection to control the robot, the robot can move freely in real time manners while complex algorithm can be developed using PC along with many MATLAB readily available functions and toolboxes.

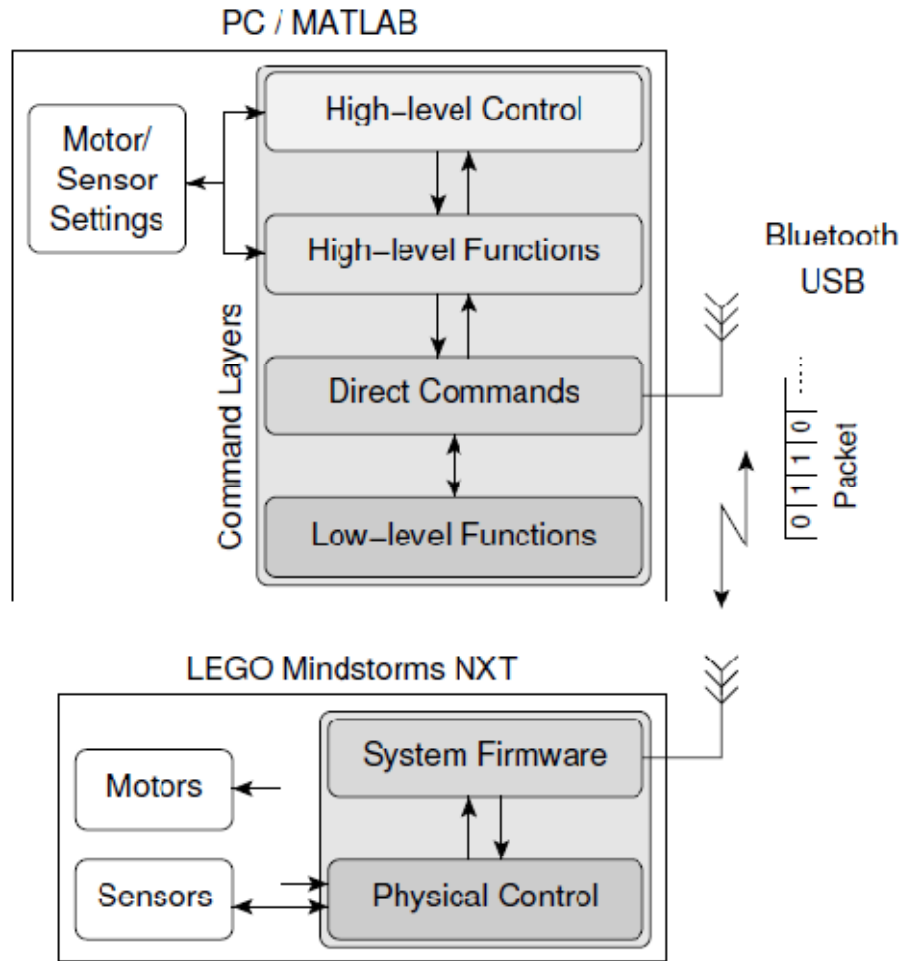


Figure 4. RWTH Mindstorms NXT Toolbox architecture

Beside MATLAB and RWTH Mindstorms NXT Toolbox there is another software running on the PC for computer vision functionality. This software is utilizing ARToolkit library, an augmented library to identify augmented reality tag. This library is usually used for augmented reality purpose in game and education application. But in this research this library is used for landmark identification and positioning in an improved environment for autonomous mobile robot research. By using augmented reality visual tag, landmark identification and positioning can be done in efficient and quick manner. This ARToolkit library is developed by Hirokazu Kato from Human Interface Technology Laboratory (HITLAB), University of Washington. In future research this mechanism can be replaced by more general purpose computer vision algorithm for landmark identification such as SIFT or SURF algorithm. The compass and landmark identification will be used for single hypothesis measurement in particles correction step (flexible particle concept). This approach is not presented in this paper.

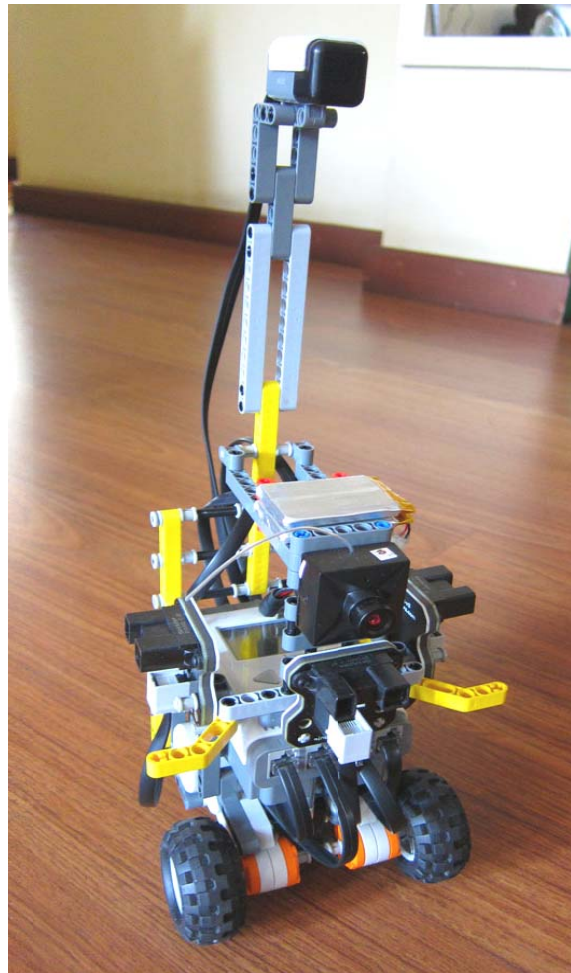


Figure 5. Robot platform built using LEGO Mindstorms NXT

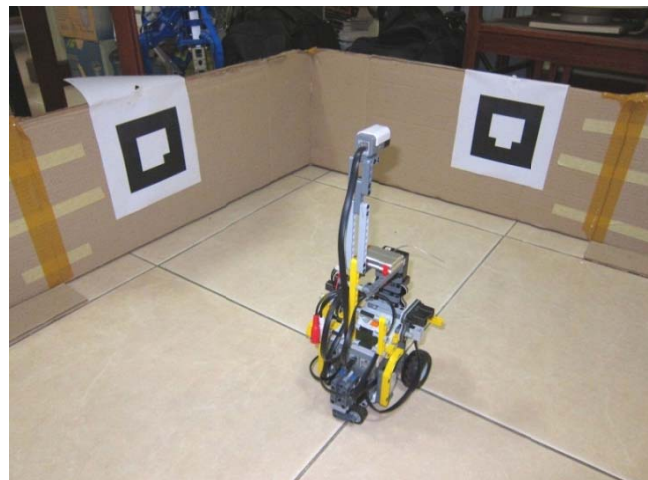


Figure 6. Experimentenvironment with augmented reality tags as landmarks

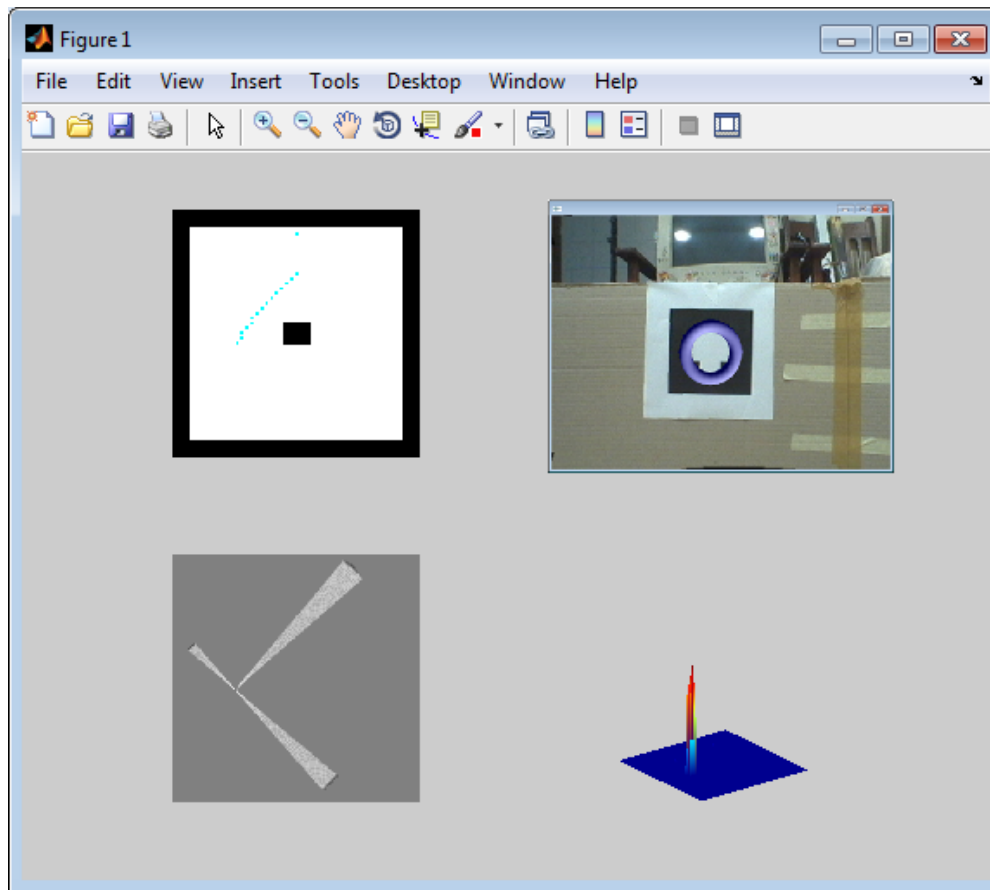


Figure 7. Graphical user interface of the robot platform

6. Experiments and analysis

In the first experiment of this research, three resampling algorithms of particles filter are tested : LTR resampling, SWR resampling and new resampling proposed in this paper. In this research a particle filter for localization is used for testing purpose. In this particle filter, beside 3 ultrasonic range finder for raw measurement, the compass sensor is also used to bound the heading error (this will be discussed in future paper). CMOS Camera for vision is not used in this experiment. In this experiment the robot mission is just wandering around in the operating environment with reactive obstacle avoidance, and the particle filter is used to localize the robots. Three experiments are conducted with each resampling algorithm. The exact ground truth position of the robots are measured and will be compared with the position estimation resulted from the particle filter. The norm used for comparison between three resampling algorithms is Euclidean distance between ground truth position and estimated position. Certain number of steps are chose to represent the transient and steady state condition.

The graphic below illustrate the comparison between three resampling algorithms

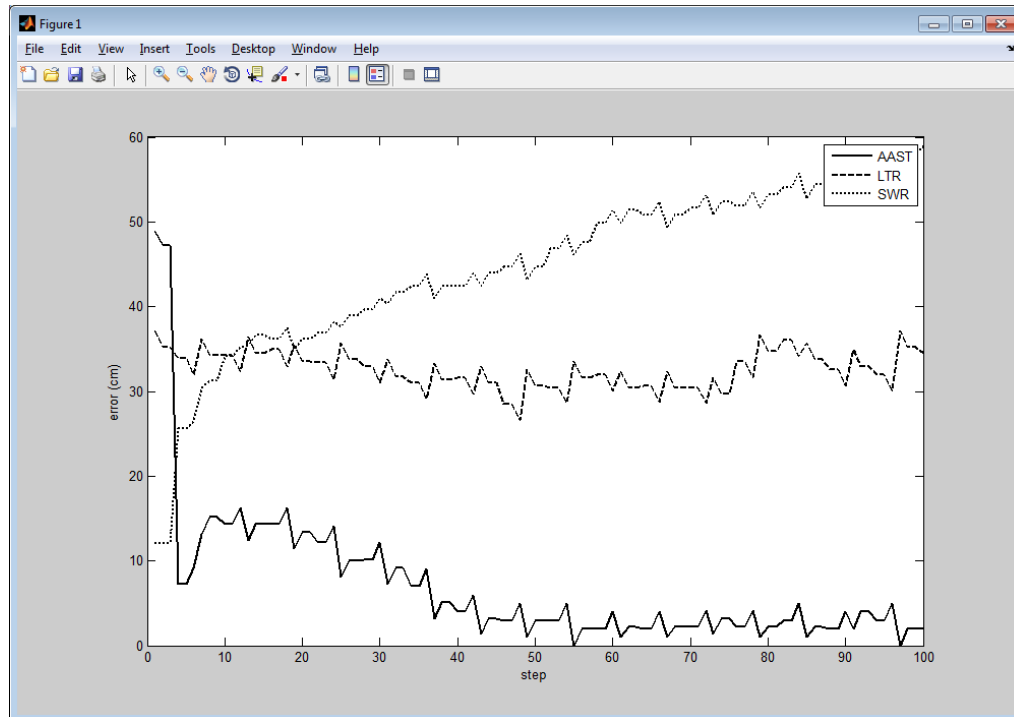


Figure 8. Error comparison between three resampling algorithms

From the above graphics, it can be seen that the particle filter with new resampling algorithms has a better error correction trend compared to particle filter with SWR or LTR resampling algorithms. The particle filter with new resampling algorithms reach the steady state condition after around 60 steps and the position error is less than 5 cm. This new resampling algorithms give significant performance enhancement compared to the commonly used resampling algorithms.

In the second experiment of this research, the performance of the new resampling algorithms and SWR resampling algorithms are compared in robot kidnaping situation. LTR resampling algorithms is not tested in this experiment, because it performs significantly worse than SWR resampling algorithms. In this experiment, the particle filter localization has to be considerably responsive to new estimation convergence after the robot is suddenly moved to new random location. The same as the previous experiment, Euclidean distance between ground truth position and estimated position is used as performance indicator.

The graphic below illustrates the comparison between two resampling algorithms

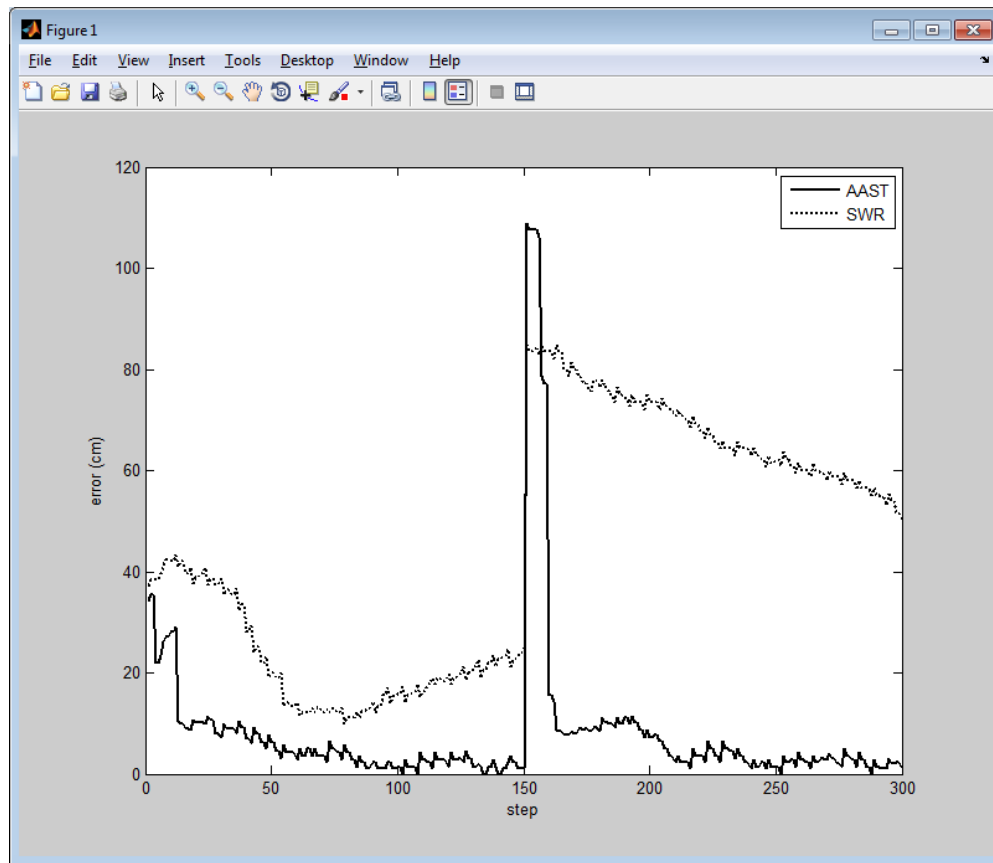


Figure 9. Error comparison between new resampling algorithms and SWR resampling algorithms in robot kidnaping situation

From the above graphics, it can be seen that the new resampling algorithms enable the particle filter to recover more quickly in robot kidnaping situation in the 150th step. In less than 15 filtering steps after the kidnaping, the particle filter with new resampling algorithms can already be able to reduce the error to less than 20 cm, and at the 60th step it has already been able to reach steady state condition with less than 5 cm steady state error. Meanwhile the particle filter with SWR resampling algorithms has tendency to converge, but the transient condition is significantly longer than the particle filter with new resampling algorithms. In this experiment it can be seen that the particle filter with new resampling algorithms performs much better than the particle filter with SWR resampling algorithms.

7. Conclusions

In this paper, a new resampling algorithm for particle filter localization is proposed. The basic ideas of the new resampling algorithms are : (1) the new samples are drawn around the old particles with highest weight. The number of new particles is proportional with the weight. (2) The states of the new particles are not just duplicated from the states of the old particle but, drawn near the states of the old particle, so the new particles are slightly different from the old particle. (3) certain amount of new particles are drawn randomly with uniform distribution from the robot's configuration space to accommodate kidnaping situation. A robotic

experiment platform is also built using LEGO Mindstorms NXT, and to enable the flexibility of programming using MATLAB, RWTH Mindstorms NXT Toolbox and bluetooth wireless communication between PC and LEGO Mindstorms NXT is used. This scheme enables high flexibility of autonomous mobile robot programming using MATLAB.

Two experiments are conducted to test the performance of the new resampling algorithms compared to other commonly used resampling algorithms in particle filter, i.e. SWR and LTE resampling algorithms. The first experiment is only localization mission with reactive obstacle avoidance. The second experiment is the same with the first but with added robot kidnaping situation. From both experiments, it can be concluded that the new resampling algorithms performs much better for robot with 3 ultrasonic range finder.

References

- [1] Alpaydin, E. (2004) : Introduction to Machine Learning, MIT Press, Massachusetts.
- [2] Behrens, A., Atorf, L., Aach, T. (2010) : Teaching Practical Engineering for Freshman Students using the RWTH-Mindstorms NXT Toolbos for MATLAB, Institute of Imaging and Computer Vision, RWTH Aachen University, Germany.
- [3] Bravo, F. G., Vale, A., Ribeiro, M.I. (2007) : Particle-Filter Approach and Motion Strategy for Cooperative Localization, Departamento de IngenieriaElecrcnica, SistemasInformaticos y Automatica, Univ. de Huelva, Spain
- [4] Burguera, A., Gonzales, Y., Oliver, G. (2009) : Mobile Robot Localization using Particle Filters and Sonar Sensors, Universitat de les IllesBalears, Spain
- [5] Carpenter, J., Clifford, P., Fearnhead, P. (1999) : An Improved particle filter for non-linear problems, *IEEE proceedings - Radar, Sonar and Navigation*, 146:2-7
- [6] Foka, A., Trahanias, P. (2007) : Real-time hierarchical POMDPs for autonomous robot navigation, Institute of Computer Science, University of Crete, Greece, *Robotics and Autonomous Systems Journal*
- [7] Guez, A., Pineau, J. (2010) : Multi Tasking SLAM, School of Computer Science, Mcgill University, Montreal, Canada, 2010 *IEEE International Conference on Robotics and Automation, Anchorage Convention District, May 2-8 2010, Anchorage, Alaska, USA*
- [8] Hahnel, D., Burgard, W., Fox, D., Thrun, S. (2003) : An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements, Departement of Computer Science University of Freiburg, *Computer Science and Engineering University of Washington, Computer Science Departement Stanford University*.
- [9] Hu, W., Downs, T., Gordon, W., Milford, M., David, P. (2004) : A Modified Particle Filter for Simultaneous Robot Localization and Landmark Tracking in an Indoor Environment, School of Information Technology and Electrical Engineering, University of Quensland, Brisbane
- [10] Montemerlo, M., Thrun, S. (2003) : Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM
- [11] Muller, J., Rottman, A., Rendl, L.M., Burgard, W. (2009), A Probabilistic Sonar Sensor Model for Robust Localization of a Small-size Blimp in Indoor Environments using a Filter partikel, *ICRA Conference 2009*
- [12] Rekleitis, M. R. (2003) : A Particle Filter Tutorial for Mobile Robot Localization, Centre for Intelligent Machines, McGill University, Canada
- [13] Riisgard, S., and Blas, M. SLAM For Dummies (2004): A Tutorial Approach To Simultaneous Localizing and Mapping, MIT
- [14] Sutton, R.S., Barto, A.G. (1998) : Reinforcement Learning An Introdtution, MIT Press, Cambridge, MA.
- [15] Thrun, S. (2000) : Probabilistic Algorithms in Robotics, CMU-CS-00-126, School of Computer Science, Carnegie Mellon University, Pittsburgh.
- [16] Thrun, S., Burgard W. dan Fox, D. (2005) : Probabilistic Robotics, The MIT Press, Cambridge, England.



Widyardana Adiprawita, lecturer at STEI-ITB. Receiver his Electrical Engineering degree at Electrical Engineering ITB with honor in 1997. Finished master degree at Informatics Engineering ITB in 2000. Now still pursuing doctoral degree at Electrical Engineering ITB. His research interests are embedded system, robotics, and intelligent agent autonomy. He has written more than 20 papers published in international and national publication.



Adang Suwandi Ahmad (ASA) received his engineering degree (Ir.) in Electrical Engineering from ITB in 1976, Diplome Etude Approfondi Signaux et Bruits (DEA) option Electronique, and Docteur Ingenieur Signaux et Bruits option Electronique (Dr.-ing) from Universite des Sciences du Languedoc Montpellier, France in 1978 and July 1980 respectively. He became Institut Teknologi Bandung's Professor in Intelligent Electronics Instrumentation System in 2000. ASA's past researches were in Electronics Instrumentation Systems (Devices and Systems) and Intelligent Electronics Systems/Artificial Intelligence. Cooperation with Navy R&D Service (1992) had yielded a War Game Simulator. He founded Intelligent System Research Group (ISRG) ITB in 1993. ASA now focuses in bioinformatics computation, information sciences, intelligent computations and intelligent-based systems. He was Dean of the School of Electrical Engineering and Informatics ITB.



Jaka Sembiring received his engineering degree (Ir.) in Electrical Engineering from ITB in 1990. He also received Master of Engineering (M.Eng) degree and Doctorate (Dr.) degree in Electrical Engineering from Waseda University, Japan in 1997 and 2000 respectively. His research interests are information theory and stochastic signal processing. He has written more than 50 technical papers published in national and international publications. Besides performing his duty as a lecturer in Information Technology at the School of Electrical Engineering and Informatics ITB, he was also working at Ministry of Communication and Information of the Republic of Indonesia as Division Head of Application at Information and Communication Technology Infrastructure Center. Currently, he is Vice Dean of Resources of School of Electrical Engineering and Informatics ITB.



Bambang Riyanto Trilaksono received his engineering degree in Electrical Engineering from ITB in 1986, his master degree in Electrical Engineering from Waseda University, Tokyo Japan in 1991 and his Doctor of Engineering from Waseda University, Tokyo Japan in 1994. He became ITB's Professor in Robust Control in 2009. He is a well known Researcher in Control Engineering in Indonesia as well as in international control engineering society.