ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 5, No.17 , September 2015
ISSN : 2322-5157
www.ACSIJ.org

Composite Local Path Planning for Multi-Robot Obstacle Avoidance and Formation Navigation

Sung-Gil Wee[1], Yoon-Gu Kim[1#], Jinung An[1], Dong-Ha Lee[1] and Suk-Gyu Lee[2]

[1] **Wellness Convergence Research Center, DGIST**
Daegu, 711-873, South Korea
[#] ***ryankim9@dgist.ac.kr***

[2] **Department of Electrical Engineering, Yeungnam University**
Gyeongsan, 712-749, South Korea

Abstract

This paper proposes a composite local path planning method for multi-robot formation navigation with path deviation prevention using a repulsive function, A-star algorithm, and unscented Kalman filter (UKF). The repulsive function in the potential field method is used to avoid collisions among robots and obstacles. The A-star algorithm helps the robots to find an optimal local path. In addition, error estimation based on UKF guarantees small path deviation of each robot during navigation. The proposed method of composite local path planning is verified by the simulation results of the collective robot navigation because the system maintains a designated formation and performs a successful return to the assigned formation with effective obstacle avoidance under various experimental conditions.

Keywords: *Collision-free, Composite local path planning, Path deviation prevention, Path re-planning.*

1. Introduction

Recent advances in the fields of computation and intelligence are accelerating the development and practicality of multi-robot technologies, which include flexible multi-robot formations, collision-free path planning, and cooperative robots. In previous studies, most service robots were focused on enhancing their own intelligence and performing individual missions. However, recently, in order for special-purpose robots to effectively carry out complex and special assignments such as exploration, surveillance, rescue, manipulation, and other field applications, it is necessary that a multi-robot-based application scenario and strategy should be well-implemented through various types of technology integration, combining decentralized control, various sensors, robot intelligence, and so on [1, 2, 3, 4, 5, 6].

Autonomous navigation of robot intelligence has been studied in many previous works in the literature. With regard to path planning, the potential field has been employed to apply the virtual forces generated on a robot by using the energy magnitude working on the system. The robot finds a path to avoid obstacles using the potential field just like the reciprocal action of magnets [7, 8]. One of the graph search algorithms, the A-star algorithm, is also widely used in path planning [9]. However, most of the local path planning algorithms have inevitable limitations such as local minima problems. Koditschek [10] et al. developed a local minima free potential field method to overcome this type of problem; Chang [11] et al. suggested a path planning algorithm based on the potential field and the Voronoi diagram for a hybrid path planner that fulfills both map building and driving simultaneously. In addition, Carpin [12] et al. proposed a dynamic obstacle avoidance algorithm in which robots following a leader robot avoid dynamic obstacles using a decentralized control system.

In this paper, we discuss the effective movement of a cluster of multiple robots as they attempt to find the shortest path and to avoid obstacles or other robots without any collisions. We propose a method of composite local path planning for multi-robot formation navigation with path deviation prevention achieved by using a repulsive function, the A-star algorithm, and an unscented Kalman filter (UKF). The repulsive function in the potential field method is used to avoid collisions among robots and obstacles. The A-star algorithm helps the robots find an optimal local path. In addition, next step position estimation based on UKF reduces deviation from the tracking path of each robot during navigation. The proposed method of composite local path planning is verified because the system maintains a designated formation and performs a successful return to the assigned formation with effective obstacle avoidance under various experimental conditions.

2. Functional Algorithms for Composite Local Path Planning

This chapter describes the functional algorithms used for composite local path planning: a repulsive function of the

ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 5, No.17 , September 2015
ISSN : 2322-5157
www.ACSIJ.org

potential field for collision avoidance and the A-star algorithm for shortest path planning. The A-star algorithm cannot be used to avoid unexpected conflicts when multiple robots move along a determined path. Therefore, the collision problems can be resolved by applying a potential repulsive function.

In this work, each mobile robot consists of two wheels and one auxiliary wheel; the state equation is modelled by Eq.(1) [13, 14]. Robots can prevent a collision between themselves by using the obstacle avoidance algorithm when one robot detects obstacles with other robots within the detection range of the robot. The robot system model is based on the differential wheeled mobile robot. r_L and r_R denote the radius of the left and right wheels, respectively, of each robot. ω_R and ω_L are the rotation velocities of each wheel, and D denotes the distance between the centers of both the wheels.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \frac{\Delta t}{2}(r_R\omega_{R,k+1} + r_L\omega_{L,k+1})\cos\theta_k \\ \frac{\Delta t}{2}(r_R\omega_{R,k+1} + r_L\omega_{L,k+1})\sin\theta_k \\ \frac{\Delta t}{D}(r_R\omega_{R,k+1} - r_L\omega_{L,k+1}) \end{bmatrix} \quad (1)$$

2.1 Collision prevention

In general, by using an attractive force (F_{att}) and a repulsive force (F_{rep}) according to the pose of the mobile robot, the potential field for a collision-free path finding algorithm is applied to move the robot from a current position to the target position while avoiding collision between a robot and an obstacle. The forces are proportional to the gradient of the potential function. The potential field function with the attractive and repulsive forces is represented as Eq.(2) [15, 16, 17].

$$\begin{aligned} F(q) &= -\nabla U(q) \\ U(q) &= U_{att}(q) + U_{rep}(q) \end{aligned} \quad (2)$$

The proposed system employs a potential repulsive function (PRF) part (U_{rep}) to avoid collisions among multiple robots while those robots are tracking the same path at the same time. When the robots approach within the sensing range of each other, the potential energy (U_{rep}) increases, as shown in Eq.(5); a repulsive force (F_{rep}) occurs in the opposite direction of the robot heading, as shown in Eq.(6). Meanwhile, when the robots get far away from each other, the potential energy (U_{rep}) decreases to 0 and the repulsive force (F_{rep}) disappears. The expressions for the repulsive function and force are defined as Eqs. (3, 4, 5, 6).

$$d(q) = \min_{p \in R} d(q,p) \quad (3)$$

$$\rho(q) = d(q) - Rob_{rad} \quad (4)$$

$$U_{rep}(q) = \frac{1}{2}\eta_{rep}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, \ if \quad \rho(q) \le \rho_o \quad (5)$$

$$U_{rep}(q) = 0, \ if \quad \rho(q) > \rho_o$$

$$F_{rep}(q) = \eta_{rep} \cdot \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right) \cdot \frac{1}{\rho^2(q)} \cdot \frac{q - q_{rob1}}{\rho(q)}, \ if \quad \rho(q) \le \rho_o \quad (6)$$

$$F_{rep}(q) = 0, \ if \quad \rho(q) > \rho_o$$

where $d(q)$ denotes the distance between the coordinates q and p of robots R_1 and R_2; q is the current coordinate of R_1 for avoiding an obstacle, and p is the coordinate of R_2, which is recognized as the obstacle. η_{rep} is an adjustment constant for the repulsive function and Rob_{rad} is the robot radius. ρ_0 is a positive integer reflecting the distance within the range of R_2. $\rho(q)$ prevents collisions between the robots. In particular, the dimensions of the robot should also be considered in this collision problem. These factors can easily prevent collision by keeping a constant distance difference using the correlation of $\rho(q)$ and ρ_0 when the robots are on the same route at the same time.

2.2 Path planning

The potential field function can be one of several ways to find a path to a target position by using the attractive force and the repulsive force. This method is often used in a real-time route searching or for unknown maps. However, this method leads to a great deal of calculation time and even to a falling back to the local minimum. To deal with these problems, therefore, this study applies the A-star algorithm. The A-star algorithm is represented by $f(x) = g(x) + h(x)$ [18, 19]. A robot creates its shortest path, which is determined by calculating the cost $g(x)$ needed to reach from the start point to the target point and $h(x)$ from the target point to the start point. The expected distance cost is represented empirically by $f(x)$. In the case of multiple robots of this work, each robot with its different start point can reach the target point through its shortest path as determined by the lowest cost based on the A-star algorithm.

The advantage of the A-star algorithm is that it allows the robot to find the shortest path that allows obstacle avoidance. However, the disadvantage is that the robots cannot overcome collision amongst themselves when they are attempting to reach the same point at the same time. As a result, we propose a method of composite local path planning. This method, in order to maintain the shortest path and to avoid collisions among the robots during multi-robot navigation, mutually compensates for multi-robot navigation by combining the repulsive function and the A-star algorithm.

2.3 Path deviation prevention

UKF, unlike the extended Kalman filter (EKF), is applied to a nonlinear system using the sigma point (sp) according to mean and covariance without linearization. This algorithm can more accurately estimate coordinates than can EKF [20, 21, 22], which requires linearization by the mean and covariance of the state variables. The UKF is applied to our system by iterating the following steps (1)–(6): initialization, calculation of sp, time update, and measurement update. The recursive prediction and observation processes reduce estimation error between the real coordinate and the ideal one of a robot. While the current position and steering angle of each robot are used as the observation data, the next planned pose for each robot to move is used as the reference for prediction. UKF compensates for the position estimation error of the next time step so that it can prevent each robot from deviating from the planned path.

Step 1: initialize the robot state
- set up the current robot's coordinates and steering angle

Step 2: specify the controls and uncertainties
- set up the prediction uncertainty by velocity and steering angle
- set up the observation uncertainty by range and bearing

Step 3: design the model
- the process model: V is a linear velocity. θ is a steering angle. x', y', and θ' are the predicted elements of the robot pose.

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V \times \Delta t \times \mathbf{cos}(G+\theta) \\ y + V \times \Delta t \times \mathbf{sin}(G+\theta) \\ \theta + V \times \Delta t \times \mathbf{sin}(G)/2 \end{bmatrix} \tag{7}$$

- the observation model: the robot performs a range-bearing measurement from the origin. $\mathbf{z}$ is the observer model's range and bearing.

$$z = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \tan^{-1}(y/x) \end{bmatrix} \tag{8}$$

Step 4: decide the sigma point set
- use multivariate Gaussian distribution

Step 5: predict
- perform the unscented transform
- calculate the predicted observation mean
- calculate a new unscented covariance

Step 6: update
- create the predicted observation samples according to the observed model
- calculate the observation covariance and the state-observation correlation
- update the mean and covariance
- compute the Kalman gain
- perform the unscented update

3. Composite Local Path Planning

For the purpose of collision-free path planning and formation control technology for multi-robot cooperative applications, the proposed method of composite local path planning is implemented through the flow chart shown in Fig. 1. The detailed process is summarized below.

A. Initialization
- given the start and target positions of each robot
- given map information
- determine an initial formation among robots
- initialize the navigation conditions (distance between wheels, detecting range, robot size, warning range, etc.)

B. Path planning
- find the shortest path of each robot based on A-star
- release each robot from the initial formation

C. Collision avoidance strategy
- PRF for avoiding collisions between robots
- A-star for avoiding obstacles

D. Rebuilding path
- rebuild a path to the target after avoidance

E. Path deviation prevention
- UKF to compensate for position errors of deviation from the path

F. Exit CLPP
- finish obstacle avoidance
- restore the initial formation

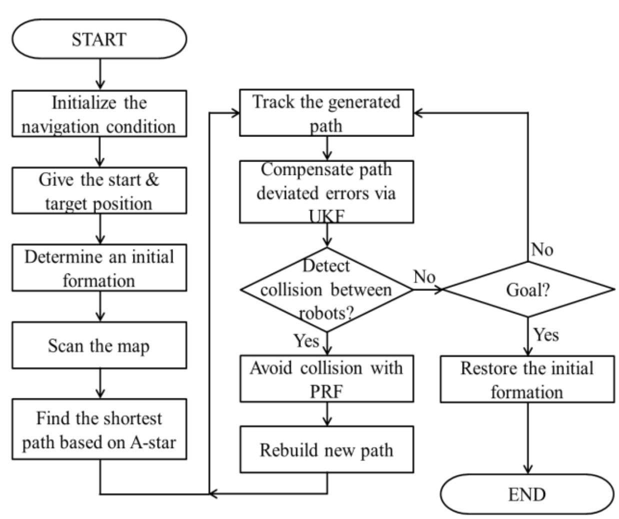


Fig. 1 Flow chart of the composite local path planning.

4. Simulation and Analysis

The effectiveness of the proposed algorithm has been verified through MATLAB simulation. This simulation models a 128 × 128 grid map, using which each robot can analyze the configuration of obstacles on the map and determine the formation control by performing path planning and collision avoidance with the other robots.

Fig. 2 shows that two robots each at different positions start navigating toward the target position; each robot avoids obstacles and controls the formation change by applying the proposed algorithm. Finally, the two robots reinstate the start formation at the destination. The collision avoidance technology is based on the PRF between the robots. The solid line (blue) and the dotted line (magenta) show the original path of each robot as generated by the A-star algorithm; the solid line (red) indicates the path of the second robot, which is newly generated to avoid collision between the robots. The mark (*) denotes the starting position of the second robot when the robots recognize a collision. The simulation results show that the robots maintain their initial formation at the destination after passing the collision area.

Fig. 3 (a) shows the steering angles of the first and second robots at each time step. The steering angle of each robot is appropriately determined according to the variation in the movement of the robots. Fig. 3 (b) shows the distance difference between both robots at each time step. The distance gap between time index 10 and 90 is 0. This means that two robots are on the same path and will have a collision. The PRF algorithm is applied to solve this collision problem. The second robot produces a new path for collision avoidance. As a result, the robots avoid the mutual collision by using the PRF, as shown in the results of (c); they go on to re-form the initial formation. The distance difference is 7 at the start position and approximately 11 at the goal position. By comparing Fig. 3 (b) and Fig. 3 (c), it can be observed that the distance gap around the goal position is the same; however, Fig. 3 (c) shows that the robots can avoid collision in the time index from 10 to 90 by following the newly generated path.

In the next simulation, we verify the results of the simulation for three robots by applying the proposed algorithm. Figs. 4, 5, and 6 and Table 1 show the simulation results of formation maintenance and shortest path planning among the three robots. Each robot uses the A-star algorithm for its own obstacle avoidance and shortest path finding. As shown in Fig. 4, when some of the clustered robots search out obstacles on the way to the target position, the relevant robots make detours.

A collision occurs between robots when some of the robots attempt to pass through a particular area at the same time. In order to prevent such collisions, the simulation results shown in Fig. 4 (a) deal with this problem by applying the PRF definition to perform avoidance and path re-planning. Further, by employing UKF, the robots shown in Fig. 4 (b) follow their paths without path deviation.

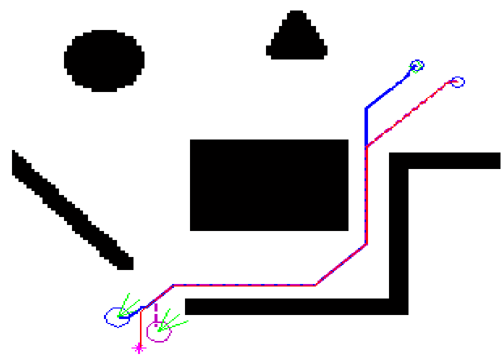
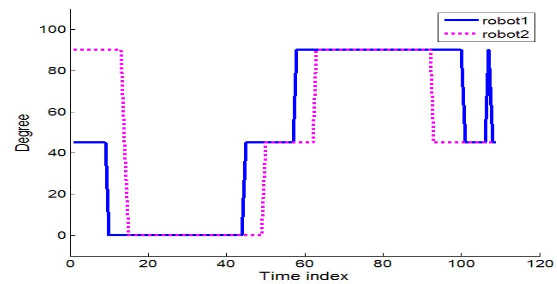
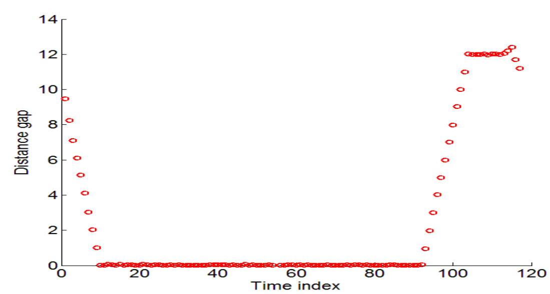


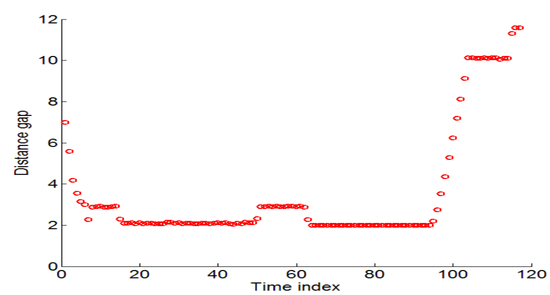
Fig. 2 Navigation for two robots to pass a collision area.



(a) Steering angles of two robots.

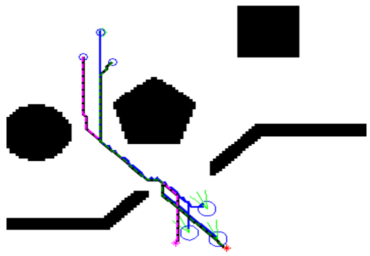


(b) Distance gap between two robots along the original path.

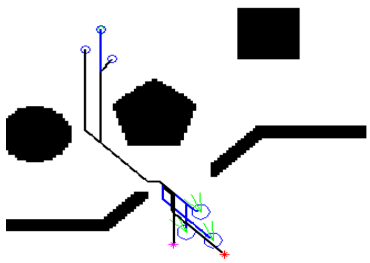


(c) Distance gap between two robots based on the newly generated path of the second robot.

Fig. 3 Navigation results of two robots.



(a)



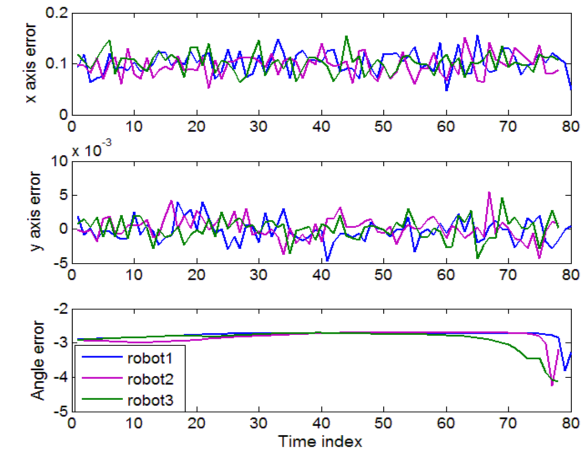
(b)

Fig. 4 Collision avoidance, path re-planning, and deviation prevention in the case of three robots.

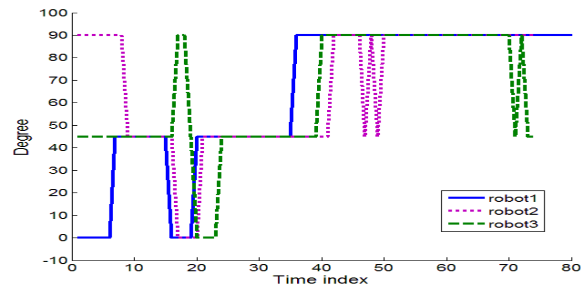
Fig. 5 (a) shows the deviation of the coordinates x and y and the steering angle θ when comparing the ideal navigation path with the real one for the three robots. Fig. 5 (b) shows the steering angle variation of each robot. Fig. 5 (c) shows the distance difference between robots 1 and 2, robots 2 and 3, and robots 1 and 3, along the initial path. Fig. 5(d) shows the results of the distance difference for the newly generated path. These figures also show that the robots recover their initial formation at the goal position. Table 1 summarizes the path cost of each robot for cases of generating new paths. The newly generated paths have a slightly greater error difference compared to that of the real path because of additional robot movement for collision avoidance. Furthermore, the entire data processing can become slow during path re-planning after obstacle avoidance. This slowness is caused by the calculation load required for map rebuilding. However, since the system does not apply path re-planning at every time step, the periodic operation of the proposed method is carried out.

Table 1: Comparison of path costs.

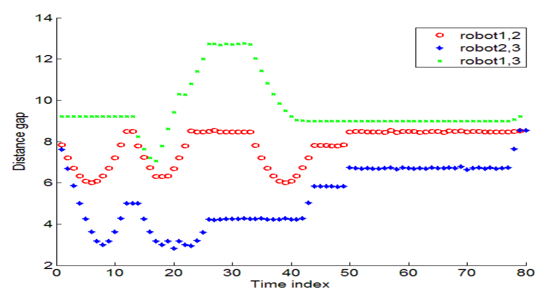
	Ideal path cost	Real path cost	New path cost
Robot1	90.6123	91.5980	91.5980
Robot2	89.4661	90.4264	93.7696
Robot3	92.3222	93.7401	99.9828



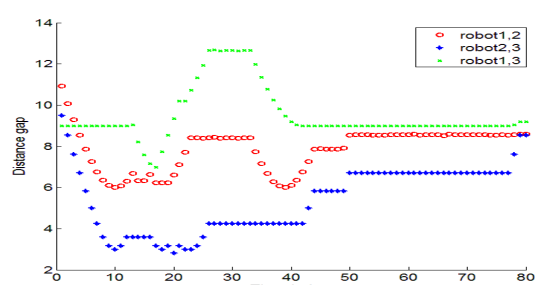
(a) Deviation of coordinates x and y and steering angle of three robots.



(b) Steering angles of three robots.



(c) Distance gap between two robots along the original path.



(d) Distance gap between two robots based on the newly generated path of the second robot

Fig. 5 Navigation results of three robots.

5. Experiment Results

To validate the proposed algorithm, experiments were carried out in two sets of conditions: single-robot path planning and multi-robot path planning in an environment (450cm x 450cm) and in an extended map (1620cm x 1620cm). The mobile robots for the experiments have differential wheeled driving systems that have a motor controller based on ATmega 128 and an upper controller based on a PC and LabVIEW. To demonstrate the performance of the proposed algorithm, the robots were set at a linear velocity of 0.1 m/s and an angular velocity of 0.5 rad/s in normal navigation; a velocity of 0.005 m/s and angular velocity of 0.02 rad/s in deceleration navigation were used for accurate position and attitude of the robots. The whole system operation flow is illustrated in Fig. 6.
Fig. 7 shows the navigation experiment of multiple robots following a planned path in a 450cm x 450cm space. Fig. 8 shows the experiment for path deviation prevention. Figs. 8 (a) and (b) show the results for robots without and with UFK, respectively. While the robots cannot reach the target point without UKF, the robots can reach the target point with small error, as shown in Fig. 8 (b). Every time a robot detects a path deviation error, it compensates for the error using UKF.

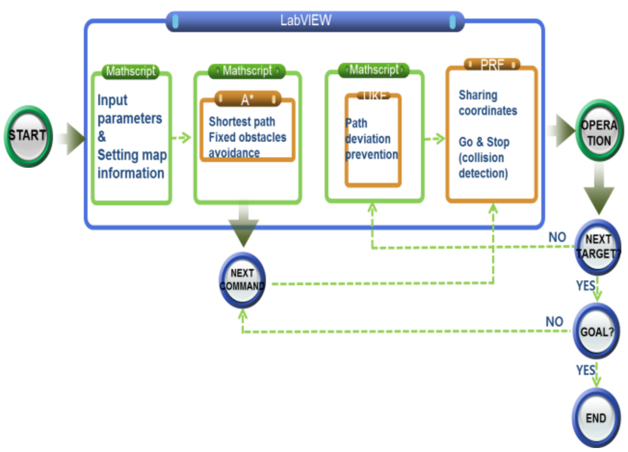


Fig. 6 Operation flow of whole system for experiments.

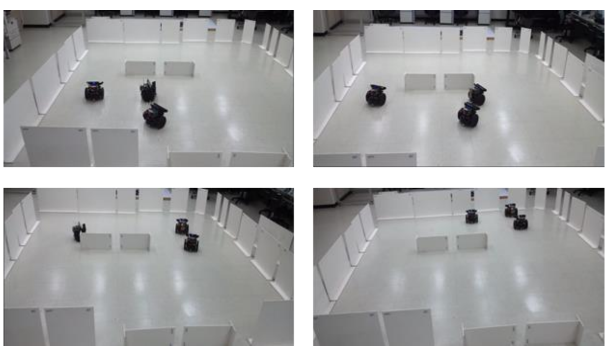
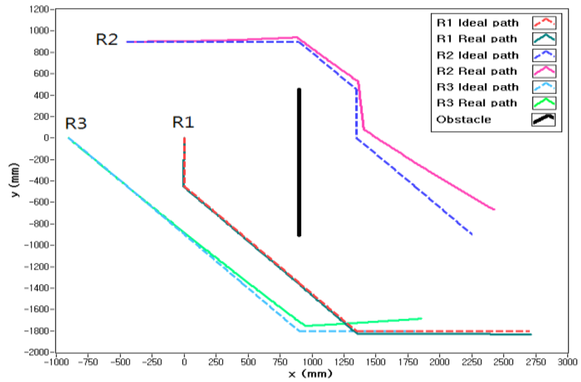
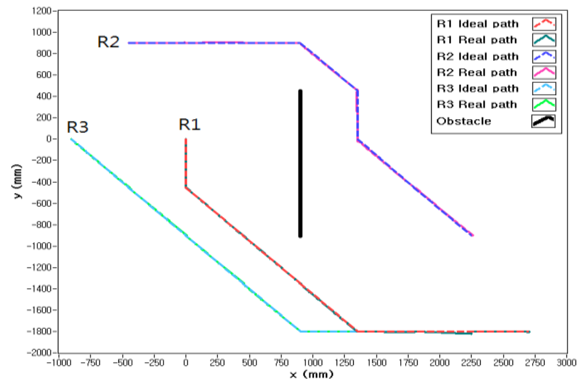


Fig. 7 Real navigation of multiple robots following a planned path.



(a) Multi-robot navigation without UKF.



(b) Multi-robot navigation with UKF.

Fig. 8 Results of path deviation prevention without and with UKF.

The composite local path planning of multiple robots in an expanded experiment with a 1620cm x 1620cm map is shown in Fig. 9. The obstacle complexity in this space is increased. As can be seen in Fig. 10, which was drawn using real-time data acquisition according to multi-robot movement, the multi-robot navigation results for the expanded experiment illustrate that the proposed composite local path planning works well even in the expanded real space.

Fig. 9 Composite local path planning of multiple robots in the expanded experiment.

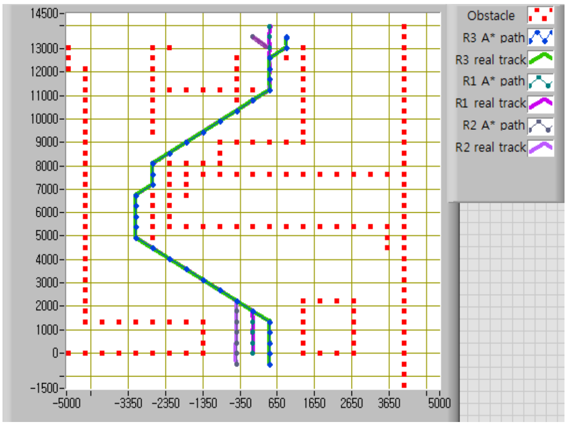


Fig. 10 Multi-robot navigation experiment results.

6. Conclusion

This paper proposes a method of enhanced local path planning for multi-robot formation navigation with path deviation prevention by compositely using a repulsive function, the A-star algorithm, and UKF. The repulsive function in the potential field method is used to avoid collisions among the robots and obstacles. The A-star algorithm helps the robots find an optimal local path. In addition, error estimation based on UKF guarantees minimum path deviation for each robot during navigation. The proposed composite local path planning has been verified by simulation and experimental results for collective robot navigation: robots maintained a designated formation and performed a successful return to the assigned formation with effective obstacle avoidance under various experimental conditions.

Our future work will focus on system enlargement by adding a number of robots to the experiment and by developing more advanced technology. The swarm robot system must be optimized to enhance the proposed algorithm and guarantee real-time obstacle avoidance, local path planning, and path tracking control.

Acknowledgments

This work was supported by the DGIST R&D Program of the Ministry of Science, ICT and Technology of Korea (15-BD-01).

References

[1] L. E. Parker., “Path Planning and Motion Coordination in Multiple Mobile Robot Teams,” In: Meyers, R.A., Encyclopedia of Complexity and System Science, Springer, 2009.

[2] I. Kassabalidis, M. A. El-Sharkawi, R. J. Marks Ⅱ, P. Arabshahi and A. A. , “Swarm Intelligence for Routing in Communication Networks,” Global Telecommunication Conference, GLOBECOM’01. IEEE, 2001, Vol. 6, pp. 3613-3617.

[3] C. Blum and X. Li, “Swarm Intelligence in Optimization,” Natural Computing Series. Springer, 2008, pp. 43-85.

[4] P. Valckenaers, M. Kollingbaum, H. Van Brussel, O. Bochmann and C. Zamfirescu, “The Design of Multi-Agent Coordination and Control Systems using Stigmergy,” Proceedings of the IWES'01 Conference, 2001.

[5] L. Leonardi, M. Mamei and F. Zambonelli, “Co-Fields: Towards a Unifying Model for Swarm Intelligence,” DISMI-UNIMO-3-2002, University of Modena and Reggio Emilia, 2002.

[6] S. P. Hou, C. C. Cheah and J. J. E. Slotine, “Dynamic Region Following Formation Control for a Swarm of Robots,” 2009 IEEE International Conference on Robotics and Automation Kobe International Conference Center, 2009, pp. 1929-1934.

[7] L. Barnes, W. Alvis, M. Fields, A. K. Valavanis and W. Moreno, “Swarm Formation Control with Potential Fields Formed by Bivariate Normal Functions,” Control and Automation, MED '06. 14th Mediterranean Conference on Control and Automation, 2006, pp. 1-7.

[8] L. Yin and Y. Yin, “An Improved Potential Field Method for Mobile Robot Path Planning in Dynamic Environments,” Proceedings of the 7th World Congress on Intelligent Control and Automation, 2008, pp. 4847-4852.

[9] R. Inam, D. Cederman and P. Tsigas, “A* Algorithm for Graphics Processors,” THIRD SWEDISH WORKSHOP ON MULTI-CORE COMPUTING-MCC’10, 2010.

[10] D. E. Kodischek, and E. Rimon, “Robot Navigation Functions on Manifolds with Boundary,” Advances in Applied Mathematics, 1990, Vol. 11, No. 4, pp. 412-442.

[11] Y. Chang and Y. Yamamoto, “Path Planning of Wheeled Mobile Robot with Simultaneous Free Space Locating Capability,” Intelligent Service Robotics, 2009, Vol. 2, pp. 9-22.

[12] S. Carpin and L. E. Parker, “Cooperative Leader Following in a Distributed Multi-Robot System,” Proceedings of the 2002 IEEE International Conference on Robotics & Automation, 2002, pp. 2994-3001.

[13] C. Jung and W. Chung, “Accurate Calibration of Two Wheel Differential Mobile Robots by Using Experimental Heading Errors,” Robotics and Automation (ICRA), 2012, pp. 4533-4538.

[14] D. Chwa, “Tracking Control of Differential-Drive Wheeled Mobile Robots Using a Backstepping-Like Feedback Linearization,” Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions, 2010, Vol. 40, pp. 1285-1295.

[15] Q. Tang, Y. Shen, C. Hu and J. Zeng, “Multi-Swarm Cooperation Optimization for Multi-Modal Functions in Repulsive Potential Field,” 2011 Fourth International Workshop on Advanced Computational Intelligence (IWACI), 2011, pp. 70-74.

[16] D. H. Kim, H. O. Wang, G. Ye and S. Shin, “Decentralized Control of Autonomous Swarm Systems Using Artificial Potential Functions: Analytical Design Guidelines,” Decision and Control, 43rd IEEE Conference, 2004, Vol.1, pp.159-164.

[17] P. Shi and Y. Zhao: Automation and Logistics, “Global Path Planning for Mobile Robot Based on Improved

Artificial Potential Function," Automation and Logistics, ICAL'09, 2009, pp.1900-1904.
[18] M. S. H. Nizami, S. M. M. R. Al-Arif, A. H. M. I. Ferdous, M. M. S. Riyadh and F. R. Faridi, "Efficient Algorithm for Automated Rescue Boats," Computer Supported Cooperative Work in Design, 2012, pp. 698-702.
[19] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions of Systems Science and Cybernetics, 1968, Vol. Ssc-4, No. 2.
[20] G. Li, L. Wang and Y. Tan, "Based on Quantum Particle Swarm Optimization and Unscented Kalman Filter Orbit State Prediction," Digital Manufacturing and Automation, 2012, pp. 257-260.
[21] S. Qi and H. Jian-Da, "An Adaptive UKF Algorithm for the State and Parameter Estimations of a Mobile Robot," Acta Automatica Sinica, 2008, Vol. 34, No. 1, pp. 72-79.
[22] Reddy J. B. V., Dash P. K., Samantary R. and Moharana A. K., "Fast Tracking of Power Quality Disturbance Signals Using an Optimized Unscented Filter," Instrumentation and Measurement, IEEE Transactions, 2009, Vol. 58, pp. 3943-3952.

Sung-Gil Wee received B.S. and M.S. degrees in Electrical Engineering from Yeungnam University in 2006 and 2008, respectively. During the period of 2008-2011, he worked at the Gyeongbuk Research Institute of Vehicle Embedded Technology as a researcher. He is now with the Daegu Gyeongbuk Institute of Science & Technology of the Rep. of Korea.

Yoon-Gu Kim received M.S. and Ph.D. degrees in Computer Engineering from Yeungnam University in 2005 and 2008, respectively. From 2008 to the present, he has been with the Daegu Gyeongbuk Institute of Science & Technology of the Rep. of Korea. His current research interests include robotics, SLAM, and cooperative control.

Jinung An received M.S. and Ph.D. degrees in Mechanical Engineering from the Korea Advanced Institute of Science and Technology in 1997 and 2005, respectively. From 2008 to the present, he has been with the Daegu Gyeongbuk Institute of Science & Technology of the Rep. of Korea. His research interests include robotics, BCI, and Haptics.

Dong-Ha Lee received B. E., M.S., and Ph. D. degrees from the Dept. of Electronics, Gyeongbuk National University, Korea, in 1985, 2001, and 2005, respectively. He received T-MBA certification from the Dept. of Business, Gyeongbuk National University, in 2006. He was a director of S/W engineering at LG Electronics from 1987 to 2005. His research interests include issues related to Cyber Physical Systems (CPS). He focuses on image processing, DTV systems, Embedded S/W, and Bio-inspired robotics for IT convergence in smart environment applications. Currently, he is a principal investigator at the BIRRE lab at DGIST, Daegu, Korea.

Suk-Gyu Lee received B.S. and M.S. degrees in Electrical Engineering from Seoul National University in 1979 and 1981, respectively. He received his Ph.D. degree from UCLA in 1990. From 1982 to the present, he has been with Yeungnam University. His research interests include robotics, SLAM, nonlinear control and adaptive control.