On-line Motion Planner for Multi-Agents based on Real-Time Collision Prognosis

Sang-Hoon Ji*, Ji-Min Kim*, Beom-Hee Lee**

* Department of Electrical Engineering, Seoul National University, Seoul, Korea
(Tel : +82-2-880-6485; E-mail: {robot91, tonin00}@snu.ac.kr)
** **Professor**, Department of Electrical Engineering, Seoul National University, Seoul, Korea
(Tel : +82-2-880-7311; E-mail: bhlee@asri.snu.ac.kr)

Abstract: In this paper, we propose a novel approach to decentralized motion planning and conflict-resolution for multiple mobile agents working in an environment with unexpected moving obstacles. Our proposed motion planner has two characteristics. One is a real-time collision prognosis based on modified collision map. Collision map is a famous centralized motion planner with low computation load, and the collision prognosis hands over these characteristics. And the collision prognosis is based on current robots status, maximum robot speeds, maximum robot accelerations, and path information produced from off-line path planning procedure, so it is applicable to motion planner for multiple agents in a dynamic environment. The other characteristic is that motion controller architecture is based on potential field method, which is capable of integrating robot guidance to the goals with collision avoidance. For the architecture, we define virtual obstacles making delay time for collision avoidance from the real-time collision prognosis. Finally the results obtained from realistic simulation of a multi-robot environment with unknown moving obstacles demonstrate safety and efficiency of the proposed method.

Keywords: collision map, potential field, virtual obstacle, collision prognosis

1. INTRODUCTION

Multiple small robots can accomplish the task which cannot be carried out by a single robot. So navigation of mobile agents has been an area of significant interest in robotics [1], [8]. However, collision-free motion planning of multiple robots turns out to have a NP-hard problem and not to be solved mathematically [10]. Therefore a number of researches attempted to solve the problem by numerical methods. They are classified into decentralized heuristic methods and centralized deterministic methods according to the information handling structure among robots [2].

Centralized methods define multiple agents as a complicated system where a supervisor integrates all the robots and plans their motion totally [3]. The methods alllow for complete planners, but their computational load increases according to the number of the robots [3], [6]. So many methods have been proposed to overcome the problem. PRM sacrifices completeness for computational simplicity [12]. And the collision map method uses a two-dimensional figure in which the path and trajectory information of two moving robots are incorporated [9]. But they have a disadvantage that the planners recalculate the velocities of all robots when unexpected moving obstacles come in the robot paths.

In decentralized methods each robot plans individually for itself by means of collecting information from other robots and environmental information around the robot [6]. In this method, the computational load is small, but neither the optimization of the total planning nor stability of the whole process is ensured. Modified decentralized methods were proposed to overcome the problems. Relative-distance method based on potential field method changes the path of robots by using local priorities [4]. However, it is difficult to apply this algorithm to more than three robots. The framework for negotiation that permits parallel path computation and dynamic priority assignment was proposed in [5]. And an approach based on combination of AI with real-time control techniques was presented [7]. However, these methods did not consider moving obstacles and their performance is not good when a number of robots are moving.

So we present a modified decentralized method. By combining artificial potential fields, collision map and path segments related to geometric characteristics, the method in the paper can modify easily the robot speeds when unexpected moving obstacles (UMO) came in the robot's paths.

The paper is organized as follows. In section II dynamic properties related to collisions among robots are presented. And then in section III, they are expressed as functional parameters so that they are used to modify speed of robots for collision avoidance in conflict situation. In section IV a method based on dynamic approach is proposed. The method is based on local collision map and gives a solution for making potential field that safely guide robots to their goals in dynamic environments. In section V numerical examples are presented to demonstrate the significance of the proposed method for motion planning of robots in dynamic environments with UMO's.

2. PROBLEM DEFINITION

We focus on the problem for planning multi-agents in dynamic environments and assumptions of our study are stated as follows:

[A1] All robots are assigned start and goal positions,
[A2] Robots' paths and priorities are pre-determined by global path planner and should not be changed,
[A3] Inter-robot communication is reliable and wide,
[A4] UMO's will not rush into any robot standing,

A1 and A2 are reliable when motion planning is decoupled into path planning and trajectory planning. And A3 is reasonable as most robots have Ethernet based LANs. A4 is also true when unknown moving obstacles are human beings.

2.1 Condition of collision-free motion

It is assumed that a circle mass in R^d (d=2, 3) moves from a start position to a goal position along the given path. In the motion, it has bounds on the magnitude of the acceleration and velocity. Because each robot has its radius and areas, in [10] collision-free condition was made based on occupied functions of robots. Although the condition is powerful, it is not pertinent when the search area is wide and so there are a number of the cells considered. Thus for simplicity we use path segments used in [13]. In Fig. 1, three robots have common regions, and each path is divided several path segments. For example, the path of R2 is divided to P21, … , P27. A path segment P22 is a region in which R2 may collide with R1.

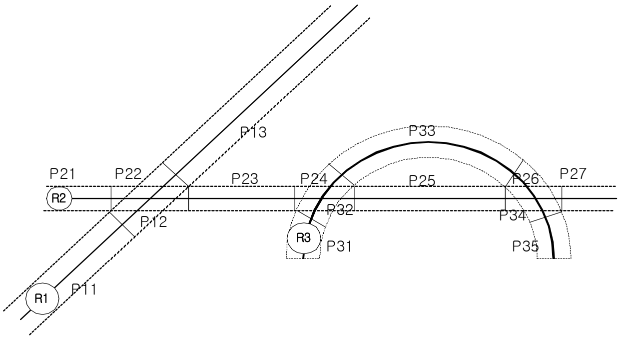


Fig. 1 Path segments.

A collision between R2 and R1 may occur when they are located on path segments related to collision. So we defined collision free condition by using path segments as follows:

[Rule 1: Collision Free Condition]

No Robot is allowed to be located on the path segments which the other robots are located the segments related to.

2.2 Collision map

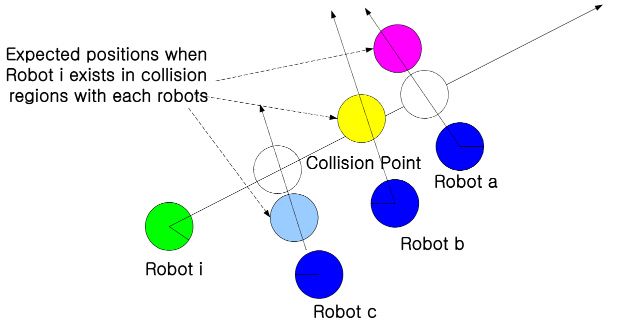
Collision map is time windows on normalized traveled length versus servo time curve. When the time windows for a robot are drawn, constraint satisfaction algorithm is used to find a feasible trajectory for the lower priority robot (LPR) [9].

In Fig. 2(a) Robot i may collide with Robot a, Robot b, and Robot c on its path. But Robot a is planned to exit the collision region before Robot i arrives there. Robot b meets Robot i in their collision region. Robot c has not yet arrived at their collision region when Robot i exits there. Fig. 2(b) shows three regions in the collision map: 'critical region', 'probable region', and 'safe region'. 'Critical region' has collisions, and the change of velocity profile of LPR needs. Neither 'probable region' nor 'safe region' has a collision due to difference in arrival time.

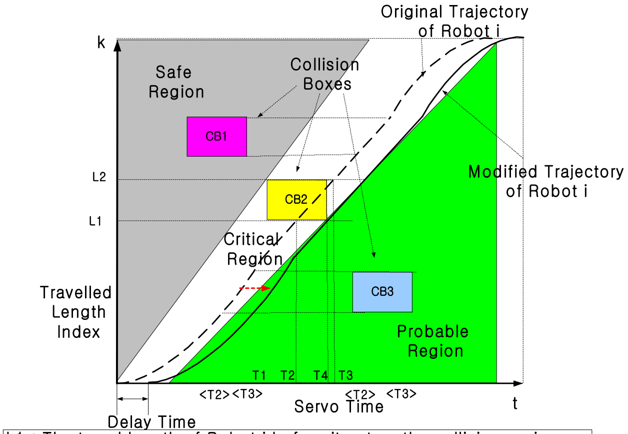
2.3 The effect of unknown moving obstacle

The effects of UMO are direct and indirect effect. Direct effect is to make new collision and generate collision boxes on the collision map. That is to say, if an UMO appears suddenly and is expected to invade the path of a robot, a collision will occur between the robot and the UMO. Therefore, the robot should change the speed to avoid the collision.

The indirect effect is change in trajectory of LPR invoked by change in that of HPR for avoiding UMO's. Originally LPR needs to modify its speed for avoiding the HPR in the Fig. 3(a) - case 1. But there is change of speed of HPR to avoid UMO in Fig. 3(b). So LPR can pass the region before HPR comes there without speed change in LPR. This is called 'fortunate case', which shows collision map method to work inefficiently in dynamic environments. 'Unfortunate case' happens as follows. If HPR is delayed for some reasons, the collision box in the safe region moves towards the critical region. The case shows collision map method gives an unsafe solution in dynamic environments.



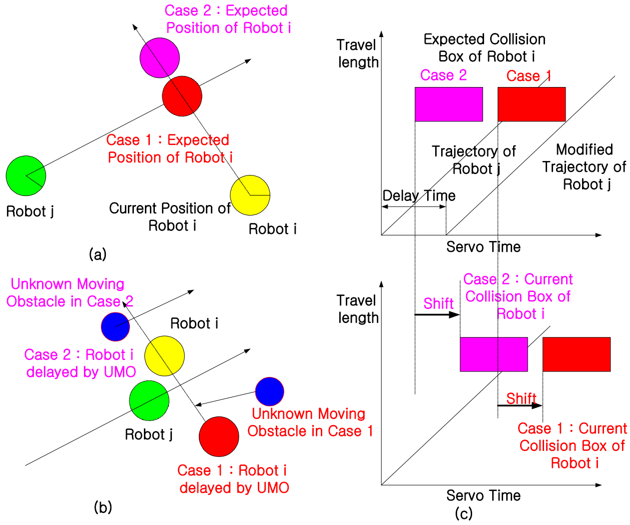
(a) Robot collision relations



L1 : The travel length of Robot i before it enters the collision region
L2 : The travel lenght of Robot i before it exits from the collision region
T1, T4 : The time when robot b enters and exits the collision region
T2, T3 : The time when robot i enters and exits the collision region

(b) The collision regions

Fig. 2 The example of collision regions of moving robots.



(a) Originally expected situations. (b) Current situations

(c) Shift of collision box invoked by UMO

Fig. 3 The example of indirect effects of UMO.

We deduce that the original methods, where LPR knows the planned speeds of HPR's in advance and is to be optimally controlled, are not applicable to the multi-agent motion planning in dynamic environments. Therefore we present a method for multi-agent in the dynamic environments, which is based on frequent update of information.

3. THE PROBLEM FORMULTAION

3.1 Collision-free motion formulation

In Fig. 4, L_1 is length from $P_{HPR}(t)$ and entrance point of LHR and L_2 is that from between $P_{HPR}(t)$ to the collision exit point. We defined variables related to dynamic characteristics of a collision box in order to formulate the problem of making an collision-free motion for multiple robots and calculate the variables in table 1 when two robots has constant speeds, where TS_{HPR} is start time of HPR and TS_{LPR} is that of LPR.

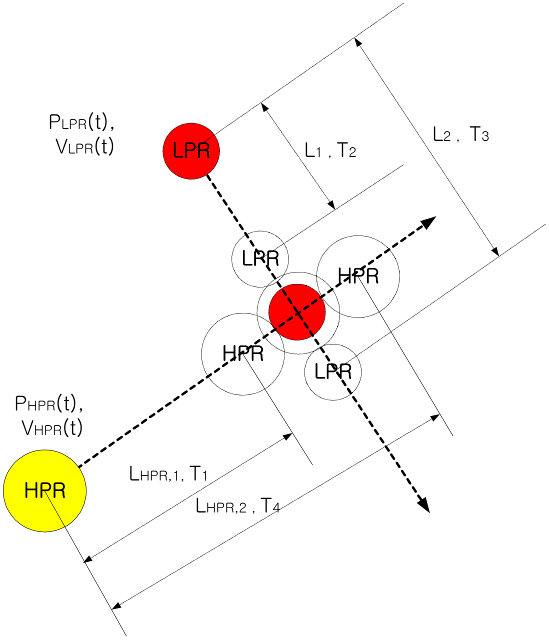


Fig. 4 The variables related to collision.

TABLE 1 Meaning of variables related to the collision map.

Variable	Meaning
T1	The time when HPR enters the region. $T1 = \frac{L_{HPR,1}}{V_{HPR}} + TS_{HPR}$
T2	The time when LPR enters the region. $T2 = \frac{L_1}{V_{LPR}} + TS_{LPR}$
T3	The time when LPR exits from the region. $T3 = \frac{L_2}{V_{LPR}} + TS_{LPR}$
T4	The time when HPR exits from the region. $T4 = \frac{L_{HPR,2}}{V_{HPR}} + TS_{HPR}$
M, D	$M = T3 - T1$, and $D = T4 - T2$.

On table 2 we find out Rule 1 is false only when both M and D are positive, so define collision-free motion as follows:

[Rule 2: Collision Free Motion]

LPR has to keep the point (M,D) on probable region or safe region while it is located in collision regions.

Table 2 Classification of collision regions.

Regions	M	D	Collision	Time Variables
Safe	+	-	No	T1 < T4 < T2 < T3
Critical	+	+	Yes	T1 < T2 < T4 < T3
Probable	-	+	No	T2 < T3 < T1 < T4

Here we show how (M, D, t) moves, when HPR sticks to planned speed and 'time delay' is applied to LHR. In 'time delay', the point moves on specific surface f(M,D,t) = M+D = const. until it arrives at M-t plane as shown in Fig. 5. As a result, (M,D,t) moves to the safe region before LHR arrives the collision region. So we can say the method do good for collision avoidance within the environment without UMO's.

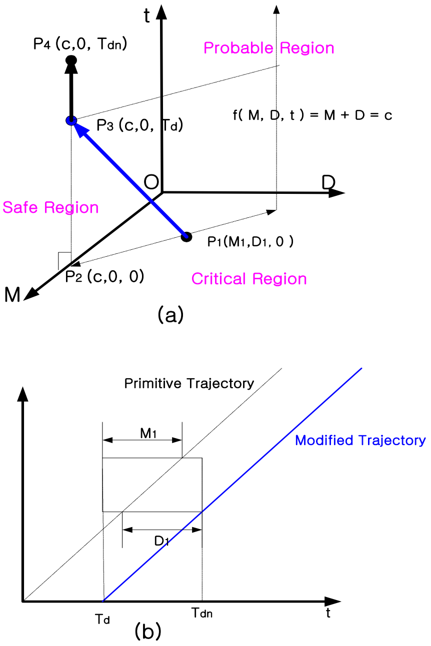


Fig. 5 (M,D,t) representation and collision map analysis.

3.2 Analysis of effect of a moving obstacle on a robot

Direct effect is new collision boxes. If we assume the moving obstacle stick to current velocity, it is solved by applying collision map method. So we skip the effect here.

Indirect effect is invoked by the shift of collision boxes of the HPR's on the collision map of an LPR. We explains the effect based on the variables on table 1.

$$M = T3 - T1 = \frac{L_2}{V_{LPR}} + TS_{LPR} - \frac{L_{HPR,1}}{V_{HPR}} - TS_{HPR} \quad (1)$$

$$D = T4 - T2 = \frac{L_{HPR,2}}{V_{HPR}} - TS_{HPR} - \frac{L_1}{V_{LPR}} - TS_{LPR} \quad (2)$$

In order to analysis indirect effect we categorize the effects by using gradients as follows:

$$M' = M - \left(\frac{L_2}{V_{LPR}^{\;2}}\right) \bullet \Delta V_{LPR} + \left(\frac{L_{HPR,1}}{V_{HPR}^{\;2}}\right) \bullet \Delta V_{HPR} + \Delta TS_{LPR} - \Delta TS_{LPR} \quad (3)$$

$$D' = D + \left(\frac{L_1}{V_{LPR}^{\;2}}\right) \bullet \Delta V_{LPR} - \left(\frac{L_{HPR,2}}{V_{HPR}^{\;2}}\right) \bullet \Delta V_{HPR} - \Delta TS_{LPR} + \Delta TS_{LPR} \quad (4)$$

Case 1 : LPR has only relatively delayed start (D_0)

We call ($TS_{LPR} - TS_{HPR}$) ‘relatively delayed start’ and notate D_0. If there is only D_0 and no change in speeds, M and D is replaced as follows:

$M' = M + D_0$, $D' = D - D_0$ and $M' + D' = M + D$.

So (M,D) moves along the line : M + D = constant

If **D_0** is positive, the point moves to safe region and ‘fortunate case’ may be invoked. And if **D_0** is negative, the point moves to probable region and ‘unfortunate case’ may be invoked

Case 2 : HPR has only change in its speed,

In case M and D is replaced as follows:

$$M' = M + \left(\frac{L_{HPR,1}}{V_{HPR}{}^2}\right) \bullet \Delta V_{HPR}, D' = D - \left(\frac{L_{HPR,2}}{V_{HPR}{}^2}\right) \bullet \Delta V_{HPR}$$

So (M,D) moves to probable region if ΔV_{HPR} is positive and because $L_{HPR,2}$ is bigger than $L_{HPR,1}$, slope of movement is smaller than -1.

Case 3 : LPR has only change in its speed

In case M and D is replaced as follows:

$$M' = M - \left(\frac{L_2}{V_{LPR}{}^2}\right) \bullet \Delta V_{LPR}, D' = D + \left(\frac{L_1}{V_{LPR}{}^2}\right) \bullet \Delta V_{LPR}$$

So (M,D) moves to probable region if ΔV_{LPR} is positive, and because L_2 is bigger than L_1, slope of movement is between -1 and 0.

In Fig. 6, we assume HPR has constant speed and LPR speed is modified using original collision map method. If there is no UMO, (M,D) is expected to move from P1 to P3. But when HPR speed is bigger than expected or LPR has relative delay start, P3 shifts to P3’ whose M’ and D’ are negative. This implies over speed reduction or over time delay of LPR and performance is poor. And when HPR speed is smaller than expected or LHR has negative delay start, P3 shifts to P3’’ whose M’’ and D’’ are positive. This implies instability of LPR at time t_{dn}.

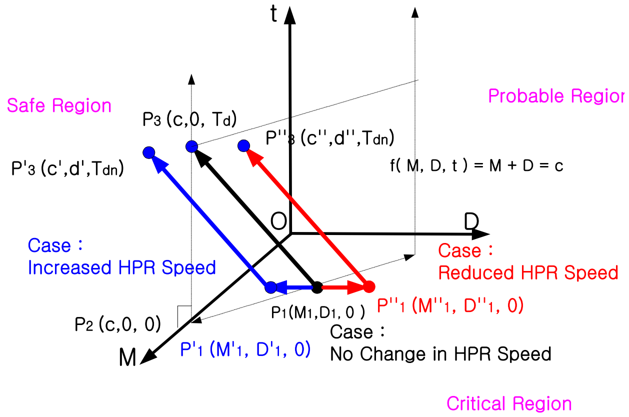


Fig. 6 Indirect Effects of UMO.

3.3 Generation of potential field for collision avoidance

When two robots have collision regions, LPR has collision box on its collision map and modify trajectory for collision avoidance. From the physical meaning, HPR makes delay of LPR like an obstacle in electromagnetic field. Two robots acts like virtual obstacles to each others as shown in Fig. 7. Thus if we determine intensity of the virtual obstacle based on collision map, we can achieve safe guidance of multiple robots. We use general potential field method, which is known for safe guiding single robots to the goal. So we define potential field as follows:

$$\begin{aligned} &F_{art}(P_i(t), P_j(t), P_{gj}, \dot{P}_i(t), \dot{P}_j(t)) \\ &= F_{g,j}(P_i(t), P_{gj}) + F_{vj,i}(P_i(t), P_j(t), \dot{P}_i(t), \dot{P}_j(t)) \end{aligned} \tag{5}$$

Where, F_v is repulsive force by virtual obstacles and F_g is attractive force by the goal.

Hereafter the control problem to guiding multiple robots to their goals is converted to the problem to determining intensity of virtual obstacle for collision avoidance by using collision prognosis based on collision map.

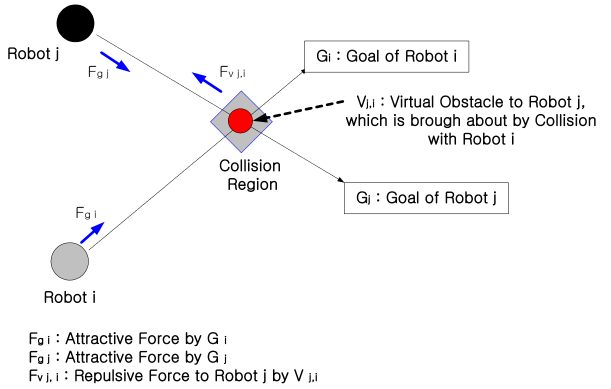


Fig. 7 Collision prognosis based on artificial potential field.

4. THE SOLUTION TO THE PROBLEM

In section III collision map determines delay time for collision avoidance. But its performance was degraded when it was applied to a dynamic environment. We propose an algorithm for preventing performance of collision map degrading. In this algorithm a robot makes not a global but a local collision map which makes potential field for navigation. So we determine force sources generating potential field first as virtual obstacles. And then, we determine artificial force field made by force sources and a target point. In this paper force sources will not modify paths of robots but adjust speed of robots. Attractive force generated by a goal point has constant magnitude and it directs to a goal point. Repulsive forces are generated by a moving obstacles and higher priority robots. Their direction is contrary to attractive force, and magnitude is determined by delay time for collision avoidance.

4.1 Collision prognosis based virtual obstacles

In section III we can check whether there is a collision and determine delay time for collision avoidance by using two variables M and D. And we showed the M grew and D shrank when start of LPR was delayed. So we determine an intensity of virtual obstacles as a function of M and D as follows:

$$\begin{aligned} &C(p_j(t), p_i(t), v_i(t), v_j(t)) \\ &= Kc \bullet (1+sgn(M)) \bullet (1+sgn(D) \bullet exp(D \bullet (L2-L1)-1) \end{aligned} \tag{6}$$

L1, L2, M, and D is variables defined on collision maps, and C is an intensity of virtual obstacle, and ***sgn()*** is a sign function. We use an exponential function in order to consider a real continuous robot model with acceleration and deceleration intervals. And for the simplicity we use (L2-L1), the travel length of LPR in collision region, instead of M in Eq(6). M and D is calculated as follow:

[step1] Calculate acceleration of LPR.
When LPR is located far from the goal, it has predefined maximum acceleration, and it is located near the goal, it has deceleration.
[step2] Calculate its speed profile
LPR uses acceleration in step1 to speed profile and its speed is limited by maximum speed.
[step3] Determine velocity of HPR.
It is updated using communication between robots or a supervisory system.
[step4] Calculate (M,D) using collision map method.

If no collision occurs, either M or D is smaller than 0, so an intensity of virtual obstacle is zero. As robot speed decreases, robot trajectory shifts right and D decreases, which results in weakness of the intensity of virtual obstacle. So an intensity of virtual obstacle is compatible to the concept of original collision map. This estimation has advantages over 'relative-velocity method' in that this method is applicable to more than three robots and planner stability is increased due to using not speed but acceleration data.

4.2 Virtual obstacle based artificial potential field

Artificial potential field pulls a robot to the goal and pushes the robot from obstacles. It is made by attractive force and repulsive force as shown Fig. 7. Attractive force pulls the robot to the goal established on the global static path-plan. Its magnitude is constant and it is directed to a goal point. We define magnitude as follows:

$$F_{g,j}(P_j(t),\ P_{j,goal}) = \max(\min(MaxF_j,\ \xi_j \bullet \| P_j(t) - P_{j,goal} \| - \psi_j),\ -MaxF_j) \quad (7)$$

Where,
Pj(t), Pj,goal : the current point and goal point of Robot j,
MaxFj : the force of Robot j invoking maximum acceleration
ξj : the proportional constant
ψj : the force constant for deceleration

When the robot approaches the obstacle or moving robots, the repulsion potential field works to reduce the speed of the robot. We define repulsive force as follows:

$$F_{v,j,i}(P_j(t), P_i(t), V_j(t), V_i(t)) = \begin{cases} \eta \bullet C(P_j(t), P_i(t), V_j(t), V_i(t)) / \| P_j(t) - CP_{i,j} \|^2, & \text{if} \| P_j(t) - CP_{i,j} \| < D_{s,j} \\ 0 & \text{,otherwises} \end{cases} \quad (8)$$

Where,
CP i,j : the intersection point between two paths
C : the intensity of moving obstacles or robots
η : the proportional constant
Ds,j : the constant for safe margin distance around the obstacle.

Since robot motion planning is executed at specific time interval, Ds,j is necessary for safe guiding robots. And we assume robot cannot run against predefined motion direction, so robots aren't allowed to have negative speeds. Thus we adjust proportional values and $D_{s,j}$. We use the value by the two times value as the maximum deceleration distance.

4.3 System control architecture

Our planner comprises three control loops as shown in Fig. 8. : an inner loop for linearization of the robot dynamics, outer loops for external forces (repulsive and attractive). It is similar the architecture in [11], which integrates robot guidance to a moving goal and collision avoidance in dynamic environments, but is hard to apply multi-agents. Because robot position control is assumed to do well, the inner loop is not explained here. Because we aim at guiding robots to static goals, attractive forces are related to only distance between robot positions and goal points. We calculate the magnitude of forces by using Eq(7). Repulsive forces are invoked by virtual obstacles. We first make collision map and determine two variables related to collision map, M and D based on the procedures in 4.1. Here after we determine the intensities of virtual obstacles by using Eq(6). Finally repulsive forces are calculated by Eq(8).

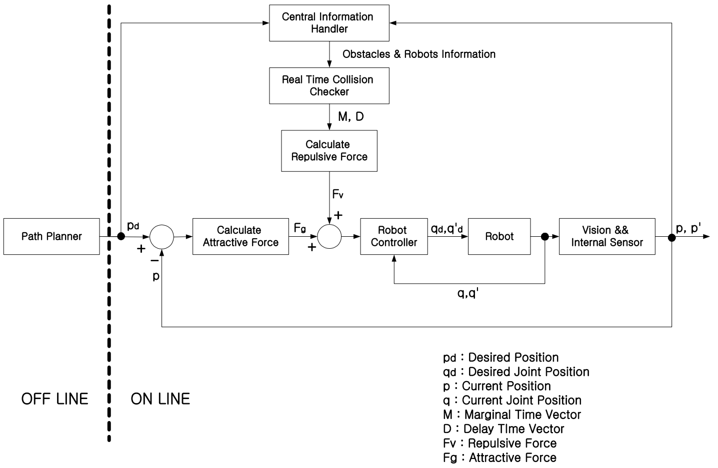


Fig. 8 Architecture of motion planner.

In Fig. 8, we design multi-agent control architecture as distributed agents supported by supervisory mentor. The supervisor has global sensor network (we used a global vision system) and obtain global environmental information including robots, and gives the information to all the robots via wireless communication with a specified interval frequency. In this paper we neglect time delay of network communication and will consider the instability problem invoked by network time delay.

5. SIMULATION RESULTS AND DISCUSSION

[Experiment 1] Two robots and no UMO

We executed this case to show that the proposed method can give same performance and stability to original collision method. Maximum speed of LPR is 1m/sec and that of HPR is 0.8 m/sec, and maximum acceleration of LPR and HPR is 0.5m/sec^2. The arrival time of LPR is 9.8sec in original collision map and 9.0sec in the proposed method. And the result in Fig. 9 showed the proposed method gives as same stability as original method in environments with no UMO. In Fig. 9 Y-axis is relative distance value, which is the distance between two robot centers minus sum of radii of them. Thus when relative distance is below zero, two robots collide. In this case the proposed method gave bigger relative distance, so it gave much more stability. We concluded that the proposed method is as good as an original collision map.

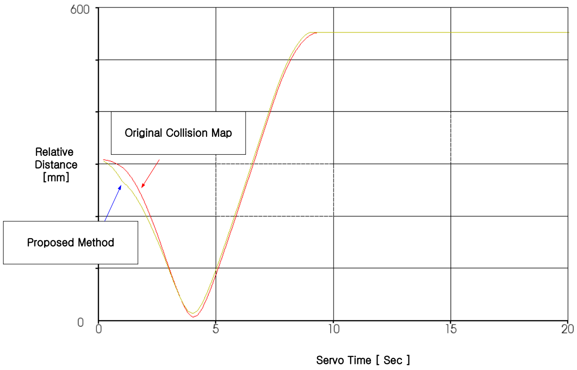


Fig. 9 Simulation result: Two robots and no UMO.

[Experiment 2] One UMO and fortunate case

We executed this case to show that the robot guided by the proposed method can move safely and more efficiently than original collision method. Moving obstacles removed one collision box on the local collision map, so the robot ran as efficiently as with the case of one collision box. The arrival time of LPR is 9.8sec in original collision map and 8.0sec in the proposed method. We concluded that the proposed method is much efficient than the original method to the problem of motion planning for multiple robots with moving obstacles.

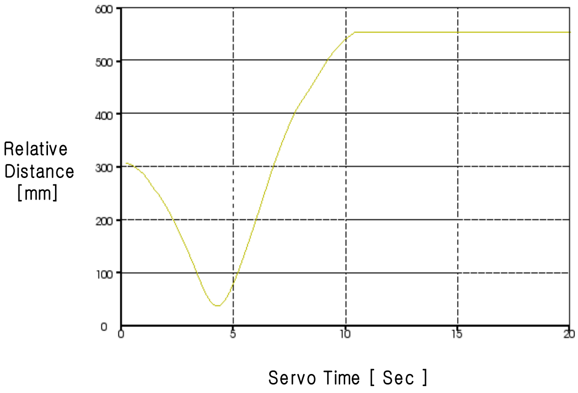


Fig. 10 Simulation result: One UMO. Fortunate case

[Experiment 3] One UMO and unfortunate case

We executed this case to show that the robot guided by the proposed method can move more safely. Moving obstacles delayed HPR, so LPR had to avoid collision with HPR. In figure 11 the distance between two robots was always bigger than zero. We concluded that the proposed method can give safety to multiple robots with moving obstacles.

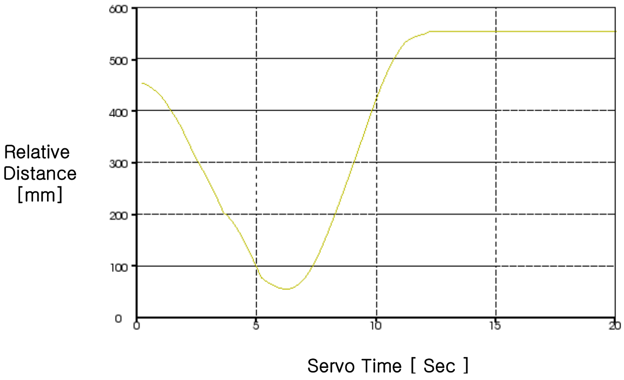


Fig. 11 Simulation result: One UMO. Unfortunate case

6. CONCLUSION

To guide robots safely in dynamic environments, dynamic characteristics related to conflict among robots were analyzed. And we proposed a motion planner based on virtual obstacles, which is made from dynamic characteristics related to collisions. This method succeeded in safe guiding robots in environments with one UMO. In future dynamic priority selection will be researched to increase performance in the special cases including ‘Follower-leader’ problem.

ACKNOWLEDGMENTS

This research was supported by ‘the Brain Korea 21 Project in 2005’, by ‘Automation and Systems Research Institute’ in Seoul National University, and by ‘the Intelligent Robotics Development Program’, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

REFERENCES

[1] D. V. Dimarogonas, S. G. Loizou and K. J. Kyriakopoulos, “Decentralized motion control of multiple holonomic agents under input constraints”, Proc. IEEE Conference on Decision and Control, vol.4, pp. 3390-3395, 2003.

[2] Y. Guo and L. E. Parker, “A distributed and optimal motion planning approach for multiple mobile robots”, Proc. of IEEE Int. Conference on Robotics and Automation, 2002.

[3] T. Arai and J. Ota, “Motion planning of multiple mobile robots”, Proc. of IEEE Int. Conference on Intelligent Robots and Systems, 1992.

[4] J. Hasegawa, K. Kurihara and N. Nishiuchi, “Collision-free path planning method for mobile robot”, IEEE Int. Conference on Systems, Man and Cybernetics, vol.3, 2002

[5] K. Azarm and G. Schmidt, “Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation”, Proc. of IEEE Int. Conference on Robotics and Automation, Albuquerque, New Mexico, 1997.

[6] B. Aronov, M. de Berg, A. Frank van der Stappen, P. Švestka and J. Vleugels, “Motion planning for multiple robots”, ACM, 1998.

[7] L. Chun, Z. Zheng and W. Chang, “A decentralized approach to the conflict-free motion planning for multiple mobile robots”, Proc. of IEEE Int. Conference of Robotics and Automation, Detroit, Michigan, 1999.

[8] Z. Feng, T. Dalong and W. Yingzi, “Obstacle avoidance for mobile robots on relative coordinates”, Proc. of IEEE Int. Conference on Robotics, Intelligent Systems and Signal Processing, 2003.

[9] B. H. Lee and C. S. G. Lee, “Collision-free motion planning of two robots”, IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-17, no.1, 1987.

[10] A. Akella and S. Hutchinson, “Coordinating the motions of multiple robots with specified trajectories”, Proc. of IEEE Int. Conference on Robotics and Automation, pp.624-631, 2003.

[11] R. pence and S. Hutchinson, “An integrated architecture for robot motion planning and control in the presence of obstacles with unknown trajectories”, IEEE Transactions on Systems, Man. and Cybernetics, vol. 25, no. 1, 1995.

[12] Christoper M. C., Stephen M. R., J. C. Latombe, “Motion planning for multiple mobile robots using dynamic networks”, Proc. of IEEE Int. Conference on Robotics and Automation, pp.4222-4227.

[13] Elzbieta Roszkowska, “Supervisory control for deadlock avoidance in compound processes”, IEEE Trans. on Systems, Man, and Cybernetics, vol. 34, no. 1