'92 KACC 1992. 10. 19~21

A Robot Motion Planning Method for Time-Varying obstacle avoidance

Nak Yong Ko+, Bum Hee Lee++, Myoung Sam Ko++

+ Department of Control and Instrumentation Eng., Chosun Univ., Gwangjoo 501-759, Korea
++ Department of Control and Instrumentation Eng., Seoul National Univ., Seoul 151-742, Korea

Abstract – An analytic solution approach to the *time-varying obstacle avoidance problem* is pursued. We formulate the problem in robot joint space(JS), and introduce the *view-time* concept to deal with the time-varying obstacles. The view-time is a set of continuous times in which a time-varying obstacle is viewed and approximated by an equivalent stationary obstacle. The equivalent stationary obstacle is transformed into the JS obstacle. In JS, the path and trajectory avoiding the JS obstacle is planned.

I. INTRODUCTION

A robot usually works in an environment with other robots, work pieces, machines, and workers. The robot should avoid collisions with these obstacles. These obstacles are usually non-stationary. They move and change their shapes, i.e., they are time-varying. The time-varying obstacle avoidance problem is to plan the robot motion from an initial to a goal configuration avoiding time-varying obstacles. The problem is one of the main issues of robot motion planning. The problem generally may not be solved by path planning or trajectory planning alone[1]. Researchers on the problem have often adopted and modified the methods for stationary obstacle avoidance. Their approaches are classified as follows:

1) The method using a *space-time* concept[2],[3]: A space is extended to a space-time by adding an extra *time* dimension to the space. In the space-time, the *motion* planning for time-varying obstacle avoidance is reduced to the *path* planning for stationary obstacle avoidance.
2) The method *adjusting the velocity* of a robot[1],[4],[5]: By varying the trajectory on an initially given path in Cartesian space(CS) or JS, the robot can avoid time-varying obstacles.
3) The method using an *artificial potential field*[6],[7],[8]: The artificial potential field is a field of forces where obstacles are repulsive surfaces against the robot, and the goal point is an attractive pole to the robot.
4) The method using a *distance function* between objects[9]: The collision is described in terms of the distances between objects. Keeping the distances between the robot and the objects above some positive value prevents the collision between them.

In this paper, a new concept, *view-time*, is introduced to solve the time-varying obstacle avoidance problem. The view-time based motion planning method is applicable to the various collision avoidance problems with stationary and time-varying obstacles.

II. PROBLEM FORMULATION

A. Nomenclature

N : degrees of freedom of a robot.
$q_i(t)$: generalized joint variable of the i-th joint.
$({}^ix, {}^iy, {}^iz)$: point represented in the i-th link coordinate system.
(x,y,z) : point represented in the base coordinate system.
t_0 : initial time of robot motion.
t_f : final time of robot motion.
$T = \{ t \mid t_0 \le t \le t_f \}$: motion time set of the robot.
$T_m = t_f - t_0$: motion time period of the robot.
${}^{i-1}A_i(q_i)$: 4x4 homogeneous transformation matrix representing the i-th link coordinate system with respect to the (i-1)-th link coordinate system with the i-th generalized joint variable q_i.
${}^0A_N(q_1,q_2,...,q_N) = {}^0A_1(q_1)\ {}^1A_2(q_2) \cdot ... \cdot {}^{N-1}A_N(q_N)$.

B. Mathematical Representations of Robot, Obstacle and the Related Concepts

In CS, it is difficult to represent the configuration and shape of a robot manipulator. In N-dimensional JS, the configuration of a robot is defined as a point.

Definition 1: The configuration of a robot at time t in JS, ${}^JM(t)$, is defined as:
$${}^JM(t) = (q_1(t), q_2(t), ..., q_N(t))$$
□

For all $t \in T$, ${}^JM(t) \in R^N$, and the function ${}^JM : T \to R^N$ describes the path and trajectory of the robot motion in JS. So, the robot motion planning means the planning of ${}^JM(t)$ for all $t \in T$.

Definition 2: The work space of a robot in JS, ${}^JWS$, is defined as:
$${}^JWS = \{ (q_1,q_2,...,q_N) \mid q_{i,min} \le q_i \le q_{i,max},\ i=1,2,...,N \}$$
where $q_{i,min}$: lower limit of the i-th joint motion
$q_{i,max}$: upper limit of the i-th joint motion
□

Let ${}^{C}OS(t)$ be the set of all points that are in an obstacle in Cartesian space.

Definition 3: The JS Obstacle corresponding to ${}^{C}OS(t)$, ${}^{J}OS[{}^{C}OS(t)]$, is defined as:

$$ {}^{J}OS[{}^{C}OS(t)] = \bigcup_{i=1}^{N} {}^{J}OS_i[{}^{C}OS(t)] $$

where,

${}^{J}OS_i[{}^{C}OS(t)] = \{ (q_1(t),q_2(t),...,q_N(t)) \mid$ *C-1* and *C-2* $\}$

C-1: for $j = i+1,...,N$, $q_{j,min} \leq q_j(t) \leq q_{j,max}$

C-2: for $j = 1,2,...,i$, $q_j(t)$ satisfies

$$ {}^{0}A_i(q_1(t),q_2(t),...,q_i(t)) \cdot ({}^{i}x_l,{}^{i}y_l,{}^{i}z_l,1)^T = (x_{os},y_{os},z_{os},1)^T, $$

for all $(x_{os},y_{os},z_{os}) \in {}^{C}OS(t)$, and for all $({}^{i}x_l,{}^{i}y_l,{}^{i}z_l) \in$ the i-th link

□

${}^{J}OS_i[{}^{C}OS(t)]$ is the set of all the JS robot configurations that cause the collision between ${}^{C}OS(t)$ and the i-th link of the robot. The shape of ${}^{J}OS[{}^{C}OS(t)]$ depends on the kinematic characteristics of the robot and the shape of ${}^{C}OS(t)$.

The collision between the robot and the obstacle can be defined as the inclusion of ${}^{J}M(t)$ into ${}^{J}OS[{}^{C}OS(t)]$.

Definition 4: The collision between the robot and the obstacle occurs at time t, if ${}^{J}M(t) \in {}^{J}OS[{}^{C}OS(t)]$.

□

C. Constraints on Collision-Free Robot Motion Planning

The constraints on robot motion planning are classified into smoothness constraint, dynamic constraint, and collision constraint. The smoothness constraint restricts the velocity, the acceleration, and the jerk, to maintain the smoothness of the joint motion trajectory.

Definition 5: The smoothness constraint is defined as:

${}^{J}M(t) \in {}^{J}CN_S(t)$, for all $t \in T$

where, ${}^{J}CN_S(t) = \{ {}^{J}M(t) \mid |q_i^{(1)}(t)| \leq VB_i, |q_i^{(2)}(t)| \leq AB_i, |q_i^{(3)}(t)| \leq JB_i, i = 1,2,...,N \}$

VB_i, AB_i, JB_i: velocity, acceleration, and jerk bounds of the i-th joint motion

□

The dynamic constraint limits the joint torques and forces required for the robot motion to the maximum torques and forces of the joint actuators.

Definition 6: The dynamic constraint is described as:

${}^{J}M(t) \in {}^{J}CN_D(t)$, for all $t \in T$

where, ${}^{J}CN_D(t) = \{ {}^{J}M(t) \mid \tau_{i,min} \leq \tau_i \leq \tau_{i,max}, i=1,2,..,N \}$

$$ \tau(t) = (\tau_1, \tau_2, .., \tau_N)^T = \mathbf{D}(\mathbf{q}(t)) \cdot \mathbf{q}^{(2)}(t) + \mathbf{h}(\mathbf{q}(t),\mathbf{q}^{(1)}(t)) + \mathbf{c}(\mathbf{q}(t)) $$

$\tau_{i,min}$: the minimum torque or force of the i-th joint actuator

$\tau_{i,max}$: the maximum torque or force of the i-th joint actuator

□

The collision constraint confines ${}^{J}M(t)$ to the collision-free space in JS.

Definition 7: The collision constraint is described as:

${}^{J}M(t) \in {}^{J}CN_C[{}^{C}OS(t)]$, for all $t \in T$

where, ${}^{J}CN_C[{}^{C}OS(t)] = {}^{J}WS - {}^{J}OS[{}^{C}OS(t)]$

□

D. Mathematical Formulation of the Problem

Let the initial configuration and the goal configuration of the robot in JS be ${}^{J}M_s$ and ${}^{J}M_g$, respectively. The collision-free robot motion planning problem is formulated as the following.

Collision-Free Robot Motion Planning Problem:

Plan the ${}^{J}M(t)$ for all $t \in T$, satisfying the following constraints and boundary conditions

1) Constraints:

${}^{J}M(t) \in {}^{J}CN_S(t)$, ${}^{J}M(t) \in {}^{J}CN_D(t)$, ${}^{J}M(t) \in {}^{J}CN_C(t)$ for all $t \in T$

2) Boundary conditions:

${}^{J}M(t_0) = {}^{J}M_s$, ${}^{J}M(t_f) = {}^{J}M_g$

□

This formulation gives a useful view of the problem. The problem is constructed by adding the collision constraint to the general robot motion planning problem. In the time-varying obstacle avoidance problem, the ${}^{J}CN_C(t)$ varies with time.

III. VIEW-TIME AND ITS PROPERTIES

The position, orientation, and shape of a time-varying obstacle vary with time. All these data at every number of instances completely describe the motion of the obstacle. However, it is impossible to deal with the infinite number of data for time-varying obstacle avoidance. The view-time concept can remove this difficulty.

The definitions of the view-time and the related concepts are as follows.

Definition 8: The i-th view-time, vt_i, is a set of time defined as:

$vt_i = \{ t \mid t_i \leq t \leq t_{i+1} \}$, $i = 0,1,2,...$

where, t_0 is the initial time, and $t_i < t_j$ if $i < j$.

□

Definition 9 (view-time period):

1) The i-th view-time period, vp_i, is defined as:

$vp_i = t_{i+1} - t_i$

2) If $vp_i = vp_0$, for all $i = 1,2,...$, then the view-time period is *fixed*.

3) If there exists some j, such that $vp_j \neq vp_0$, then the view-time period is *varying*.

□

Definition 10: The swept volume of ${}^{C}OS(t)$ in vt_i, ${}^{C}OS(vt_i)$, is defined as:

${}^{C}OS(vt_i) = \{ (x,y,z) \mid (x,y,z) \in {}^{C}OS(t), t \in vt_i \}$

□

Planning the robot motion avoiding ${}^{C}OS(vt_i)$ guarantees collision avoidance for the i-th view time vt_i. So, in vt_i, ${}^{C}OS(vt_i)$ is regarded as the stationary obstacle corresponding to the time-varying obstacle ${}^{C}OS(t)$. If $T \subseteq vt_0$, ${}^{C}OS(vt_0)$ is the entire volume swept by the obstacle during the motion.

From *Definition 3*, the JS obstacle corresponding to ${}^{C}OS(vt_i)$ is ${}^{J}OS[{}^{C}OS(vt_i)]$. The shape of ${}^{J}OS[{}^{C}OS(vt_i)]$ is dependent on the kinematic characteristics of the robot and the shape of ${}^{C}OS(vt_i)$. The shape of ${}^{C}OS(vt_i)$ is dependent on vt_i and the motion of the obstacle. These two facts suggest that the collision-free path and trajectory be affected by vt_i and the motion of the obstacle.

With these definitions, the following is the property to be taken into account for collision-free motion planning using the view-time.

Property 1:

1) If ${}^{J}M(t) \notin {}^{J}OS[{}^{C}OS(vt_i)]$, the robot will not collide with the obstacle ${}^{C}OS(t)$, for all $t \in \{ t \mid t_i < t < t_{i+1} \}$.
2) If ${}^{J}M(t_{i+1}) \notin \{ {}^{J}OS[{}^{C}OS(vt_i)] \cup {}^{J}OS[{}^{C}OS(vt_{i+1})] \}$, the robot will not collide with the obstacle ${}^{C}OS(t)$, at time $t=t_{i+1}$.
3) If ${}^{J}M_g \in {}^{J}OS[{}^{C}OS(vt_j)]$ for all j such that $j \geq i$, there exists no path and trajectory from ${}^{J}M(t_i)$ to ${}^{J}M_g$, avoiding the swept volumes ${}^{C}OS(vt_j)$, $j \geq i$.

□

Although, ${}^{J}M(t)$ is such that ${}^{J}M(t) \notin {}^{J}OS[{}^{C}OS(vt_i)]$ for all $t \in vt_i$, it may happen that ${}^{J}M(t_{i+1}) \in {}^{J}OS[{}^{C}OS(vt_{i+1})]$. In this case, the robot collides with the swept volume ${}^{C}OS(vt_{i+1})$ at time $t = t_{i+1}$. Therefore, the condition ${}^{J}M(t) \notin {}^{J}OS[{}^{C}OS(vt_i)]$ doesn't guarantee the collision avoidance at time $t = t_{i+1}$. As stated in 2) of *Property 1*, the sufficient condition for collision avoidance at time $t = t_{i+1}$ is that ${}^{J}M(t_{i+1}) \notin \{ {}^{J}OS[{}^{C}OS(vt_i)] \cup {}^{J}OS[{}^{C}OS(vt_{i+1})] \}$.

IV. COLLISION-FREE MOTION PLANNING METHOD

The conditions 1) and 2) in *Property 1* satisfy the collision constraint in the collision-free motion planning problem. The smoothness constraint and the dynamic constraint are satisfied through trajectory planning. Based on the properties of the view-time, we propose a procedure to plan the collision-free motion.

Collision-Free Motion Planning Procedure:

Initialization: ${}^{J}M(t_0) = {}^{J}M_s$, $i = 0$

Step 1 (Path Planning: Sec. IV.A):
Plan a collision-free path in JS avoiding ${}^{J}OS[{}^{C}OS(vt_i)]$ from ${}^{J}M(t_i)$ to ${}^{J}M_g$.

Step 2 (Trajectory Planning: Sec. IV.B):
On the JS collision-free path, plan the trajectory of the view-time vt_i, subject to the smoothness and dynamic constraints.

Step 3 (Path and Trajectory Modification: Sec. IV.C):
If ${}^{J}M(t_{i+1}) \in {}^{J}OS[{}^{C}OS(vt_{i+1})]$, i.e., collision occurs at time $t=t_{i+1}$, modify the path and trajectory ${}^{J}M(t)$ to avoid ${}^{J}OS[{}^{C}OS(vt_{i+1})]$ at the view-time vt_i.

Step 4 (Test of completion):
If ${}^{J}M(t_{i+1}) = {}^{J}M_g$, then
end the motion planning procedure.
If ${}^{J}M(t_{i+1}) \neq {}^{J}M_g$,
then increase i by one, and return to *Step 1*.

□

The overall structure of the collision-free motion planning method is shown in Fig. 1. From now on, we explain each step of the procedure in some detail.

A. Path Planning (Step 1)

The collision-free path in JS is planned using the V-graph search method[5][7]. Since the V-graph search method is applicable in polyhedral obstacle case, it is necessary to approximate non-polyhedral obstacles by polyhedral obstacles. In 2-dimensional JS, the V-graph search method yields the *shortest* collision-free path from ${}^{J}M(t_i)$ to ${}^{J}M_g$ for the view-time vt_i. The generated path consists of *piecewise straight line segments.* If there is no path connecting ${}^{J}M(t_i)$ to ${}^{J}M_g$ in the V-graph, robot should wait for the view-time vt_i, that is,

$$ {}^{J}M(t) = {}^{J}M(t_i), \quad \text{for all } t \in vt_i \tag{1} $$

In 3-dimensional space, the V-graph search method not necessarily yields the shortest collision-free path. A near optimal path is generated in 3-dimensional space by adding vertices on the edges of the obstacle and searching the V-graph.

B. Trajectory Planning (Step 2)

Many trajectory planning methods often ignore the dynamic constraint[10]. If the velocity and the acceleration of the joints are confined to small bounds, the joint force or torque for the robot motion can be kept within the maximum force or torque of the joint actuator almost all the times[10]. With this assumption, VB_i and AB_i ($i = 1,2,\dots,N$) are assumed to be low values in our development.

Let ${}^{J}M_k(vt_i) = (q_{1,k}(vt_i), q_{2,k}(vt_i), \dots, q_{N,k}(vt_i))$ be the JS coordinate of the k-th vertex of the collision free path from ${}^{J}M(t_i)$ to ${}^{J}M(t_{i+1})$, and ${}^{J}M_0(vt_i) = {}^{J}M(t_i)$. A point in the path segment between ${}^{J}M_k(vt_i)$ and ${}^{J}M_{k+1}(vt_i)$ is represented as,

$$ \begin{aligned} {}^{J}M(t) &= (q_1(t), q_2(t), \dots, q_N(t)) \\ &= \alpha(t) \cdot \{ {}^{J}M_{k+1}(vt_i) - {}^{J}M_k(vt_i) \} + {}^{J}M_k(vt_i) \end{aligned} \tag{2} $$

Thus,

$$ q_i(t) = \alpha(t) \cdot \{ q_{i,k+1}(vt_i) - q_{i,k}(vt_i) \} + q_{i,k}(vt_i), \quad i = 1, 2, \dots, N \tag{3} $$

The trajectory on the path segment from ${}^{J}M_k(vt_i)$ to ${}^{J}M_{k+1}(vt_i)$ is determined by $\alpha(t)$. For (3) and the *definition 5*, the smoothness constraint is represented as,

Smoothness Constraint:

$$ | \alpha^{(1)}(t) | \leq \operatorname*{Min}_{i=1}^{N} \{ VB_i / (q_{i,k+1}(vt_i) - q_{i,k}(vt_i)) \} $$

$$ | \alpha^{(2)}(t) | \leq \operatorname*{Min}_{i=1}^{N} \{ AB_i / (q_{i,k+1}(vt_i) - q_{i,k}(vt_i)) \} $$

$$ | \alpha^{(3)}(t) | \leq \operatorname*{Min}_{i=1}^{N} \{ JB_i / (q_{i,k+1}(vt_i) - q_{i,k}(vt_i)) \} $$

□

C. Path and Trajectory Modification (Step 3)

In the view-time vt_i, the path and trajectory is planned to avoid the JS obstacle ${}^{J}OS[{}^{C}OS(vt_i)]$. Therefore, it may happen that ${}^{J}M(t_{i+1}) \in {}^{J}OS[{}^{C}OS(vt_{i+1})]$, i.e., collision occurs at time $t=t_{i+1}$. In this case, we modify the JS obstacle ${}^{J}OS[{}^{C}OS(vt_i)]$; and we replan the motion for the view-time vt_i. The rule for JS obstacle modification is as follows.

Rule 1 (JS Obstacle Modification):

If ${}^{J}M(t_i)$ in the view-time vt_{i-1} is such that ${}^{J}M(t_i) \in {}^{J}OS[{}^{C}OS(vt_i)]$, then replace ${}^{J}OS[{}^{C}OS(vt_{i-1})]$ with ${}^{J}OS'[{}^{C}OS(vt_{i-1})] = {}^{J}OS[{}^{C}OS(vt_{i-1})] \cup {}^{J}OS[{}^{C}OS(vt_i)]$.

□

Avoiding the modified JS obstacle ${}^{J}OS'[{}^{C}OS(vt_{i-1})]$ in vt_{i-1} means that the robot avoids ${}^{J}OS[{}^{C}OS(vt_i)]$ in vt_{i-1}, i.e., one view-time ahead of vt_i. This method guarantees ${}^{J}M(t_i) \notin {}^{J}OS[{}^{C}OS(vt_i)]$, i.e., collision avoidance at time $t = t_i$. But it may happen that ${}^{J}M(t_{i-1}) \in {}^{J}OS'[{}^{C}OS(vt_{i-1})]$, i.e., collision between the robot and the modified obstacle may occur at time $t = t_{i-1}$. So, *Rule 1* is generalized to *Rule 2*.

Rule 2 (r View-Time Look Ahead Scheme):

If ${}^{J}M(t_i)$ in the view-time vt_{i-1} is such that ${}^{J}M(t_i) \in {}^{J}OS[{}^{C}OS(vt_i)]$, then

1) Determine r:
   $r = \mathrm{Min}\ \{ k \mid {}^{J}M(t_{i-k}) \notin {}^{J}OS[{}^{C}OS(vt_i)], k = 1,2,..,i \}$
2) Modify the JS obstacles:
   Replace ${}^{J}OS[{}^{C}OS(vt_j)]$ ($j = i-r, i-r+1, .., i-1$) with ${}^{J}OS'[{}^{C}OS(vt_j)] = {}^{J}OS[{}^{C}OS(vt_j)] \cup {}^{J}OS[{}^{C}OS(vt_i)]$.
3) Replan the paths and trajectories:
   Replan the paths and trajectories for the view-times vt_j, for all $j = i-r, i-r+1, .., i-1$.

□

V. AN APPLICATION AND SIMULATION RESULTS

We apply the view-time based motion planning method to the collision-free motion planning of an articulated robot with a time-varying obstacle in the workspace.

A. Simulation Model

Fig.2 shows a two-link planar robot with revolute joints and a rectangular obstacle. We plan and simulate collision-free motion of the robot under the following conditions.

1) The conditions on the robot are as follows.
   (a) dimensions: $l_1 = l_2 = 100$ cm
   (b) angular velocity bounds of the joint motions: $VB_i = 400$ deg/sec, i=1,2
   (c) angular acceleration bounds of the joint motions: $AB_i = 100$ deg/sec^2, i=1,2
   (d) starting point of the motion in JS: ${}^{J}M_s$ = (80 deg, -10 deg)
   (e) goal point of the motion in JS: ${}^{J}M_g$ = (-40 deg, 25 deg)
2) The conditions on the obstacle are as follows.
   (a) dimensions: $r_1 = r_2 = 20$ cm
   (b) starting point of the motion in CS: (130 cm, -20 cm)
   (c) goal point of the motion in CS: (40 cm, 160 cm)
   (d) path of the motion in CS: straight line path
   (e) maximum linear velocity of the motion in CS: 20 cm/sec
   (f) maximum linear acceleration of the motion in CS: 5 cm/sec^2

In (b) and (c) of 2), the reference point of the obstacle motion is assumed to be the lower left vertex of it.

B. Computation of the JS Obstacle

The *Definition 3* implies that JS obstacle can be obtained by solving the inverse kinematic problems wherever a point of the robot body coincides with a point of the obstacle. Since there are infinite number of points in the robot body and in the obstacle, there are infinite number of inverse kinematic problems to be solved. Instead of solving the infinite number of inverse kinematic problems, we obtain the JS obstacle ${}^{J}OS[{}^{C}OS(vt_i)]$ from the JS collision-free space ${}^{J}CN_C[{}^{C}OS(vt_i)]$. We compute the JS collision-free space ${}^{J}CN_C[{}^{C}OS(vt_i)]$ using the slice projection method[11]. Then, we find the JS obstacle ${}^{J}OS[{}^{C}OS(vt_i)]$ from ${}^{J}CN_C[{}^{C}OS(vt_i)]$ by

$$ {}^{J}OS[{}^{C}OS(vt_i)] = {}^{J}WS - {}^{J}CN_C[{}^{C}OS(vt_i)] \quad (4) $$

The equation (4) is derived from *Definition 7*.

In general, ${}^{J}CN_C[{}^{C}OS(vt_i)]$ is not a polygon, and neither is ${}^{J}OS[{}^{C}OS(vt_i)]$. To find a collision-free path using the V-graph search method, ${}^{J}OS[{}^{C}OS(vt_i)]$ is approximated by a convex hexagon ${}^{J}\underline{OS}[{}^{C}OS(vt_i)]$. Fig. 3 shows a JS obstacle ${}^{J}OS[{}^{C}OS(vt_3)]$ and its approximation ${}^{J}\underline{OS}[{}^{C}OS(vt_3)]$, for the case of $vp_i = 1.0$ sec.

C. Simulation Results

To investigate the effects of selection for the various view-time periods, the simulations are done for the following four cases.

1) *case 1:* $vp_i = 0.5$ sec, i = 1,2,...
2) *case 2:* $vp_i = 1.0$ sec, i = 1,2,...
3) *case 3:* $vp_i = 1.5$ sec, i = 1,2,...
4) *case 4:* $vp_i = 2.0$ sec, i = 1,2,...

The *number of view-time periods(NVP)*, the *robot motion time period* T_m, and the *number of applications of LAS(NLAS)* for collision-free motion planning are shown in Table 1. It is found that they are dependent on the view-time period. The robot motion time period T_m is calculated by multiplying the view-time period with the NVP required for collision-free motion.

Fig. 4 shows the JS paths and trajectories for the cases. Figs. 5 and 6 show the collision-free motions in CS for the *case 1* and *4*, respectively.

From these results, we summarize the effects of the view-time period on the planned motion as the followings.

1) The robot motion time period becomes shorter as the view-time period becomes longer.
2) The computational burden for motion planning becomes less significant as the view-time period becomes longer.
3) The smoothness of the robot motion increases as the view-time period becomes shorter. That is, the shorter view-time period produces smoother and less

roundabout path and trajectory.

Thus, there are tradeoffs among the computation time, the robot motion time period, and the smoothness of the robot motion.

VI. CONCLUSIONS

View-time is introduced to solve the time-varying obstacle avoidance problem. The view-time based motion planning method has the following characteristics.

1) It is applicable to the various collision-free robot motion planning problems with time-varying obstacles.
2) In a view-time vt_i, it uses the method of stationary obstacle avoidance scheme.
3) The number of computations for motion planning, the motion time period T_m, and the amount of excessive roundabout motion are dependent on the view-time period vp_i.
4) It is impossible to reduce simultaneously the number of computations for motion planning, the motion time period T_m, and the amount of excessive roundabout motion. To determine the optimal view-time period vp_i, it is necessary to take account of the tradeoffs among these three factors.

The subjects for further investigation are as follows.

1) Research on the varying view-time period to adapt the view-time period to the change of obstacle motion.
2) Improvement of the view-time based motion planning method for real-time application.
3) Efficient method to compute the JS obstacle $^JOS[^COS(vt_i)]$ from the swept volume $^COS(vt_i)$.
4) Trajectory planning method satisfying the dynamic constraint as well as the smoothness constraint.

REFERENCES

[1] B.H.Lee, "Constraint identification in time-varying obstacle avoidance for mechanical manipulators," *IEEE Trans. Syst., Man, Cybern.*, Vol.19, No.1, pp.140-143, Jan./Feb. 1989.

[2] K.Fujimura and H.Samet, "A hierarchical strategy for path planning among moving obstacles," *IEEE Trans. Robotics Automat.*, Vol.5, No.1, pp.61-69, Feb. 1989.

[3] C.L.Shin, T.T.Lee, and W.A.Gruver, "A unified approach for robot motion planning with moving polyhedral obstacles," *IEEE Trans. Syst., Man, Cybern.*, Vol.20, No.4, pp.903-915, Jul./Aug. 1990.

[4] B.H.Lee and C.S.G.Lee, "Collision-free motion planning of two robots," *IEEE Trans. Syst., Man, Cybern.*, Vol.17, No.1, pp.21-32, Jan./Feb. 1987.

[5] K.Kant and S.W.Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robotics Res.*, Vol.5, No.3, pp.72-89, Fall 1986.

[6] O.Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robotics Res.*, pp.90-98, Spring 1986.

[7] K.Kant and S.Zucker, "Planning collision-free trajectories in time-varying environments: A two-level hierarchy," in *Proc. 1988 IEEE Int. Conf. Robotics Automat.*, pp.1644-1649, 1988.

[8] Y.Koren and J.Borenstein, "Potential field method and their inherent limitations for mobile robot navigation," in *Proc. 1991 IEEE Int. Conf. Robotics Automat.*, pp.1398-1404, April 1991.

[9] E.G. Gilbert and D.W.Johnson, "Distance function and their application to robot path planning in the presence of obstacles," *IEEE J. Robotics Automat.*, Vol.1, No.1, pp.21-30, Mar.1985.

[10] C.S.Lin, P.R.Chang, and J.Y.S.Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial manipulators," *IEEE Trans. Automat. Contr.*, Vol.AC-28, No.12, pp.1066-1074, Dec.1983.

[11] T.Lozano-Pérez, "A simple motion planning algorithm for general robot manipulators," *IEEE Trans. Robotics Automat.*, Vol.3, No.3, pp.224-238, June 1987.

Table 1. The NVP, T_m, and NLAS applied for collision-free motion.

vp_i(sec)	NVP[1)]	T_m(sec)	NLAS[2)]
0.5	32	16.0	11
1.0	11	11.0	3
1.5	6	9.0	2
2.0	4	8.0	1

Figures

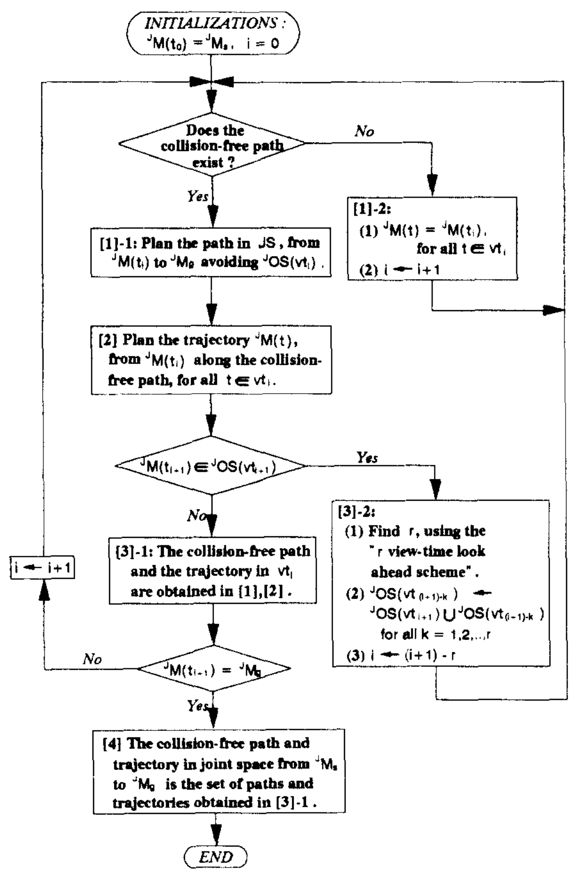


Fig. 1. Flow diagram of the motion planning algorithm.

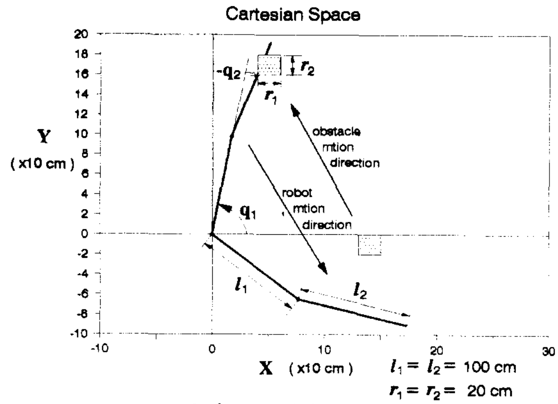


Fig. 2. Simulation model.

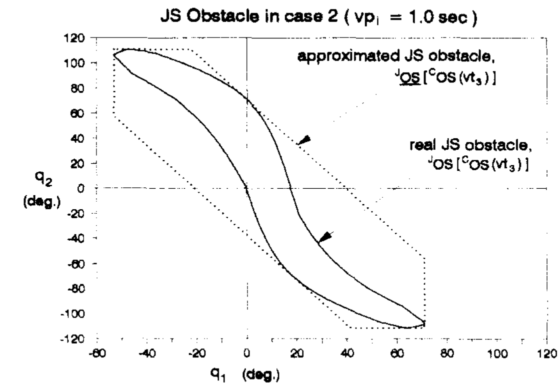


Fig. 3. Approximation of ${}^J OS[{}^C OS(vt_3)]$ with $vp_i = 1.0$ sec.

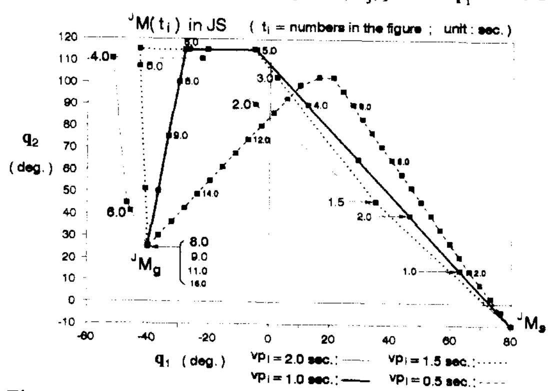
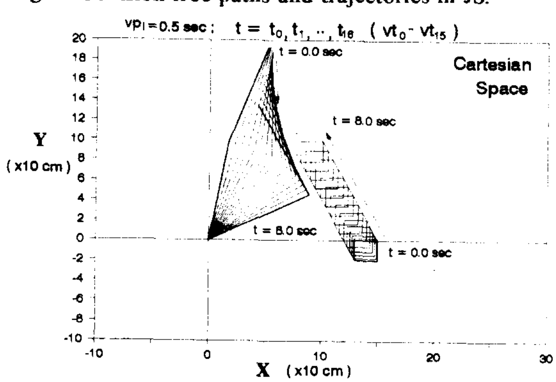
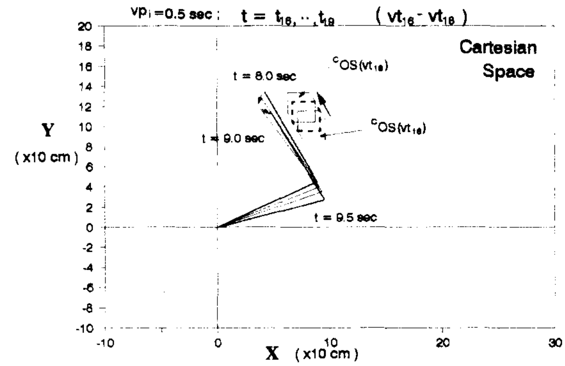


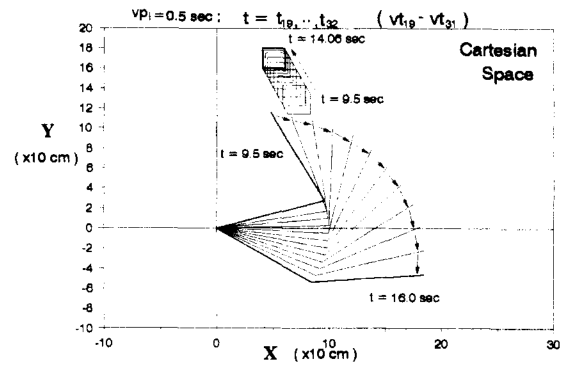
Fig. 4. Collision-free paths and trajectories in JS.



(a) 0.0 sec $\le t \le$ 8.0 sec

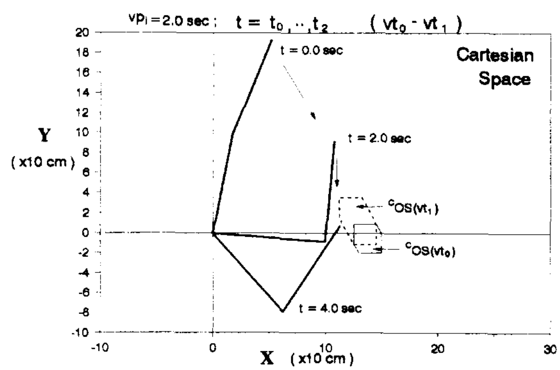


(b) 8.0 sec $\le t \le$ 9.5 sec

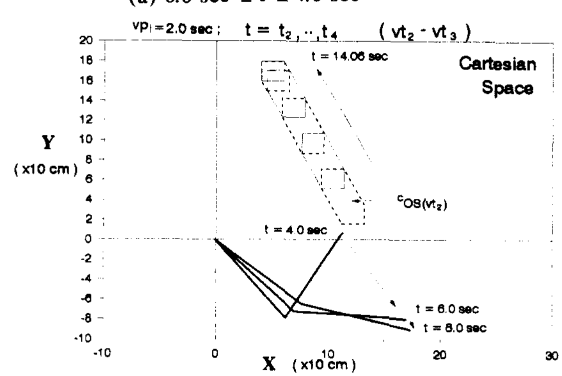


(c) 9.5 sec $\le t \le$ 16.0 sec

Fig. 5. Collision-free motion in CS for the *case 1*.



(a) 0.0 sec $\le t \le$ 4.0 sec



(b) 4.0 sec $\le t \le$ 8.0 sec

Fig. 6. Collision-free motion in CS for the *case 4*.