Collision-Avoidance Motion Planning Amidst Multiple Moving Objects

Kao-Shing Hwang, Hung-Jen Chao and Jy-Hsin Lin*
Department of Electrical Engineering
National Chung Cheng University
Chiayi, Taiwan 621, R.O.C.
**Department of Industrial Management*
Huafan University
Taipei County, Taiwan 223, R.O.C.

In this paper, an automated motion planner is proposed for coordinating collision-free motions among dynamic moving objects or machines in a 3D space. Based on the concept of modified path-velocity decomposition, an algorithm is proposed for evaluating the performance of different motion strategies. As a result, the motion planner will coordinate collision-free motions for multiple uncontrollable/controllable objects moving in a shared environment. A dynamic object can also be a multi-body object, such as a robot. The applicability of the proposed approach is demonstrated through two examples. The results show that multiple controllable and uncontrollable machines can work together in an industrial environment by efficiently avoiding collisions.

Keywords: collision avoidance, motion planning, space/time graph, speed alteration method, flight surveillance

1. INTRODUCTION AND PROBLEM STATEMENT

Motion planning in a 3D, time-varying environment is an on-going research problem. Research efforts are geared towards determining efficient methods for collision avoidance among static and dynamic obstacles in a shared environment. Two major areas of these works are navigation and guidance. In general, navigation is the problem of deciding on a path or route (global) to be followed. Typically, navigation or path planning in a known environment involves identifying a collision-free path through a population of static obstacles to satisfy some constraints, such as the shortest distance. Guidance, sometimes referred to as local navigation, is the problem of maneuvering around local (moving) obstacles while being directed along a global path. Therefore, classical collision avoidance can be stated as follows: Given an arbitrary rigid polyhedral moving object **P** with multiple degrees of freedom, such as a robot, find a continuous collision-free path taking **P** from an initial configuration to a desired configuration [1].

The above problem has given rise to a variety of methods. A general solution using free-space has been presented by Lozano-Perez and Wesley [2, 3]. In that approach, each obstacle in the environment is "grown" by the dimensions of a robot. That robot is then effectively reduced to a dimensionless point. The algorithm then reduces to finding a safe

Received April 25, 1997; accepted October 9, 1998.

path for moving a point among obstacles. Later, Lozano-Perez presented another simple and efficient algorithm, using configuration space, to plan collision-free motions for general manipulators with a known set of stationary obstacles [4]. Another method proposed by Brooks [5] used the concept of "Freeways" for representing an environment. This method defines space between obstacles as freeways, with a centerline (called a "*spline*") that is used to navigate along the freeway. Fujimura et al., developed a quadtree-type hierarchical structure to represent polygonal obstacles [6]. Their approach suffers from a large search space but works well in a sparsely occupied space. Gewali et al., [7] presented polynomial time algorithms for finding the shortest paths in 3D space among polyhedral obstacles resembling vertical buildings with a fixed number of distinct heights. They presented a shortest-path algorithm with complexity $\mathbf{O}(n^2)$ when all n obstacles have equal height. For path planning with moving polyhedral obstacles, Shih et al., presented another approach [8] using a proper free-space representation by means of polytypes. After a global graph search has been employed, a family of feasible trajectories is generated, and a constrained optimization problem is formulated to obtain the final near-optimal trajectory. Another very interesting approach proposed by Kant and Zucker [9] decomposes the path planning in a dynamic environment into two subproblems: (1) planning the path to avoid static obstacles, and (2) planning velocities along the path to avoid uncontrollable moving obstacles. This approach results in a significant reduction in the complexity of the problem. However, there are limitations in this approach. First, it does not permit path alterations, only velocity alterations. Therefore, under certain conditions, it may not give a solution even though one may exist. Secondly, it does not account for controllable objects, such as vehicles, robots etc.

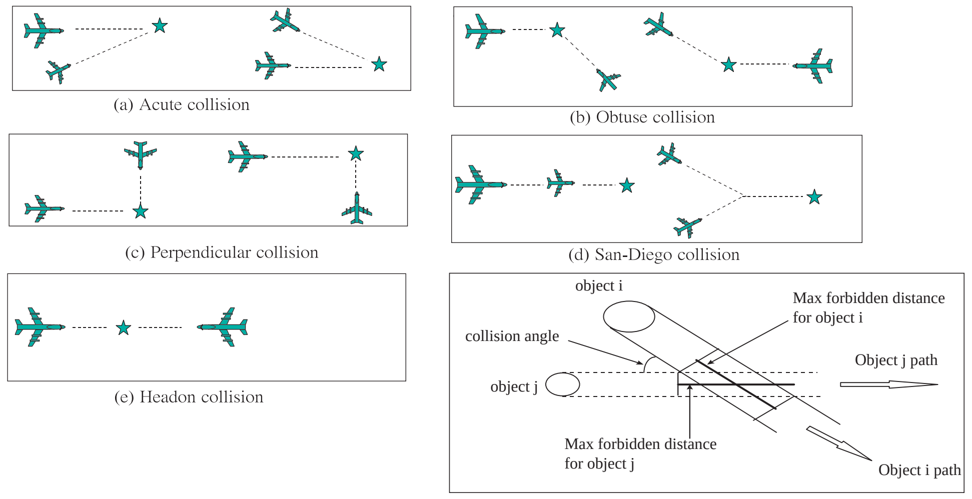
Most of the above works focused on either collision avoidance among static obstacles only or among uncontrollable moving obstacles in a shared environment. However, in an actual industrial environment, there are controllable moving obstacles, such as X-Y tables, conveyers, vehicles and other robots. Therefore, there is a need to plan collision-free motions among dynamic and static objects by adequately controlling these controllable objects or machines under some constraints, such as priorities, costs, speed limits, time delay and so on. With this type of motion planning and control, besides the traditional collision-avoidance planning, the concept of collision-free can be extended to real industrial applications, such as assembling a group of parts by controlling the motions of the surrounding machines so that the desired sequence is a collision-free motion. Therefore, in this paper, we will extend Kant and Zucker's method to plan collision-free motions for multiple objects in a shared environment, where both static and dynamic (controllable and uncontrollable) objects exist.

To solve the stated problem, objects in the environment will be categorized into two classes, i.e., *static* and dynamic. Any object that has a fixed position in the environment all the time is called a static object. As for dynamic objects, they consist of controllable and uncontrollable moving objects. *Controllable* means that the speed or the path of a moving object can be adjusted by the planner. *Uncontrollable* means that neither the speed nor the path of an object can be changed throughout the motion. Under the priority specification, the object with the highest priority will be considered as *dominant* and the other controllable ones as *subordinate*. Furthermore, when we concentrate on planning the motion of the dominant object, all the other objects will be treated as obstacles. After planning for a dominant object is finished, this object will be considered as an uncontrollable object for planning of other lower priority objects. Therefore, there are static, uncontrollable, and subordinate obstacles in the environment. Following this terminology, a method for planning collision-free motions among moving objects will be proposed.

2. COLLISIONS AMONG OBJECTS

2.1 Collision Types

There are five collision types for any two straight-line moving objects when collision occurs. Let α be the angle measured from the intersection of any two straight-line paths, shown in Fig. 1(f). The possible collision types are described below: (For illustration purposes, flight path surveillance [10] in Figs. 1(a)-(e) is used.) (a) Acute collision: they collide with $0^o < \alpha < 90^o$; (b) Obtuse collision: they collide with $90^o < \alpha < 180^o$; (c) Perpendicular collision: they collide with $\alpha = 90^o$; (d) San-Diego collision: they are moving along the same path and in the same direction, i.e., $\alpha = 0^o$ and the speed of the object behind is higher than the speed of the object ahead; and (e) Head-On collision: they are moving along the same path but in opposite directions, i.e., $\alpha = 180^o$.



(f) Enlarged graph of collision region

Fig. 1. Collision types from flight path surveillance and the collision region graph of two straight-line paths.

Collision types (a), (b) and (c) can be avoided by just changing the velocities of objects. For instance, in an assembly cell, robots are required to slow down or wait to pick up the correct parts from controllable part feeders to ensure correct sequencing; or in a military situation, a battleship can escape from the attack of a torpedo coming from its side by speeding up or slowing down. Collision types (d) and (e) belong to the consequence of two moving objects having parts of paths of two moving objects coincide. For instance, one airplane is taxiing onto the runway to prepare for takeoff and another airplane is trying to land on the same runway at the same time in the opposite direction. We cannot avoid a collision by only adjusting their speeds; the assigned path (runway) of the landing airplane must be changed, or a waiting time for the taxiing airplane should be included in order to avoid collision.

2.2 Forbidden Regions

For a possible collision between two objects, *i* and *j*, there are forbidden regions along the path of each object that may collide with the other object. Each region consists of two parameters: the Maximum Forbidden Distance (*MFD*) and the Forbidden Time Interval (*FTI*). *MFD*(*i*, *j*) represents the possible contact distance of object *i*, with object *j* measured along the path of object *i*. *FTI*(*i*, *j*) represents the maximum possible contact time for object *j* passing through the corresponding forbidden distance *MFD*(*j*, *i*) on the path of object *j*. V_j represents the speed of object *j*, i.e.,

$$FTI(i,j) = \frac{MFD(j,i)}{V_j}. \tag{1}$$

For the sake of generality and simplicity, the forbidden region parameters can be approximated by setting a safety distance *R* for each dynamic object. This distance forms a bound along the motion of the object, representing the margin for avoiding collisions. Based upon this concept of a safety distance, the *MFD* of a multi-body object, such as a robot, can be estimated through simulation. We will demonstrate it using examples in a later section. To illustrate this concept using two colliding single-body objects, the forbidden region parameters are displayed in Fig. 1(f). For collision types (a), (b) or (c), their *MFD*s can be easily estimated from the geometry of collisions as follows:

$$MFD(i,j) = 2R_i \cot(\alpha) + 2R_j \csc(\alpha), \tag{2}$$

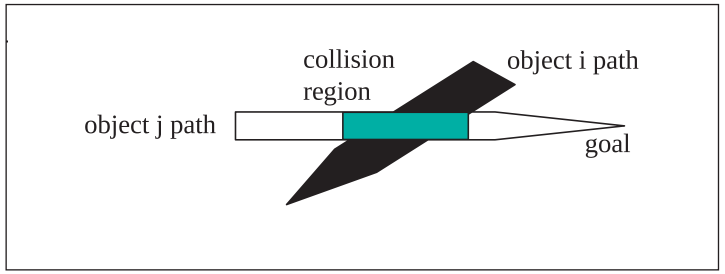
$$MFD(j,i) = 2R_j \cot(\alpha) + 2R_i \csc(\alpha), \tag{3}$$

where angle α will be replaced by $180^{\circ} - \alpha$ if the collision type is obtuse.

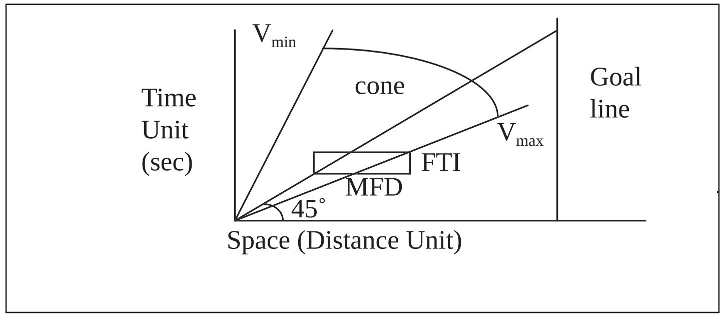
2.3 Space/Time Graph Properties

The technology of the space/time graph(STG) for path (speed) planning is implemented in this paper. This graph is a two-dimensional figure with time versus distance. The object to be planned is treated as a point in the STG. Within this graph, time and distance are both normalized in such a way that if the object is moving at a constant speed, a speed path for the object is a straight line 45 degrees from the distance axis (i.e., the slope of this line is 1) originating at the origin of the graph (see Fig. 2(b)). Therefore, the units of this graph are expressed by a scaled Distant Unit (DU) and a corresponding Time Unit (TU) of the object to be planned. As for the goal of the motion, it is represented by a vertical line (parallel to the time axis, i.e., Y-axis) at the desired final position on the distance axis of the STG. Therefore, if the object's speed is adjusted, it will be very easy to determine the time discrepancy for reaching the desired position.

By estimating the positions of the other moving objects at various times, all the information about the possible collisions within the environment can be incorporated into the STG of the planning object, which we call the dominant object. Based on this STG, the defined forbidden time and distance can then be mapped into its STG. For example, Fig. 2(a) shows an obtuse collision between objects *i* and *j*. The collision condition of object *j* due to the motion of object *i* can be estimated by *MDF*(*j*, *i*) and *FTI*(*j*, *i*) as shown in Eqs. (1-3). These two values can be easily represented as a rectangle in the space/time graph shown in Fig. 2(b). Since in real applications, the speed of any controllable moving object will have a limit range, the above STG can be modified by adding a speed-zone cone as shown in Fig. 2(b). Each controllable moving object has its own given speed-zone cone with its own upper speed limit (V_{max}) and lower speed (V_{min}) limit as the edges of the cone. To avoid forbidden regions, the range of the cone will be used to determine adequate speeds for the dominant object and the subordinates. The newly derived speed shall not exceed its maximum or minimum. The slope of the newly derived line in the graph will be less than 1 if it is faster than the initial speed. The slope of the line will be greater than 1 if it is slower.



(a) Obsture collision example



(b) Space/Time Graph of object j

Fig. 2. Space/Time graph properties.

Based on the collision information represented in the STG, the motions of all controllable objects can be adjusted to avoid possible collisions. Since there are many possible adjustments from speed changes, an optimization planning algorithm will be needed to

3. MOTION AND PATH PLANNING AMONG OBSTACLES

3.1 Path Planning Among Static Obstacles

Before discussing dynamic obstacles, we will discuss obstacle avoidance planning among static obstacles. This path planning problem is to find the shortest path between two points *s* and *t* in a 2D/3D space with a set of static obstacles. Geometrically, we may view it in two complementary ways: planning the path so as to avoid obstacles or planning the path so that it lies in the free space, i.e., the physical space minus obstacles. Path planning among static obstacles is the initial step in controlling an object in order to fix a navigation path. In practice, it is assumed that this route, i.e., global navigation, has been chosen based upon a criterion and the locations of static obstacles. The criterion chosen may be a minimum distance path, a minimum fuel consumption path as a function of velocity, or another meaningful distance measure. Many research results have been obtained in studying this topic; therefore, in this paper, we will concentrate on planning among dynamic objects. In other words, we will assume that there are no static obstacles along the desired path of any dynamic moving object.

Not only can the global navigation path be determined through planning among static objects, but the same principle can also be applied to searching the optimal speed path on a STG. From the space/time graph, a proper collision-free motion of the dominant object can be obtained by avoiding forbidden regions formed by all objects. It is, thus, a 2D motion planning problem among static objects formed by forbidden regions, and the motion of the dominant object can be determined by minimizing a selected evaluation function. Therefore, this 2D static problem becomes a subset of planning motions among dynamic objects. There are many approaches to solving this problem. However, due to the generality of the problem, in this paper, an algorithm of the iterative approach [11, 12] will be used. The evaluation function will be defined in section 4.

3.2 Motion Planning Among Moving Obstacles

As discussed in section 2.1, there are five types of collisions. However, in practical industrial applications, only the first three types occur frequently. It is essential to understand that the control strategy for the first three types of collisions is constrained to only changing the speeds of objects along their paths. For example, when a mobile robot moves within the confines of a passageway, it can not and should not move from the planned route when another vehicle crosses its path. Typically, any traversal of the road map is continually adjusted in terms of stopping, slowing down, or accelerating in an attempt to avoid obstacles. Therefore, collision-free motions can be planned among moving objects by determining appropriate intermediate velocities for all controllable objects moving along their paths. However, the above strategy will not work for Head-On and San-Diego type collisions. To avoid these two types of collisions, path detours are necessary. These two types occur more frequently in civil or military applications. In this paper, we will consider only the first three collision types.

The proposed approach to motion planning among dynamic obstacles is based on the properties presented by the modified space/time graph. The planning process consists of three phases. The first phase identifies all sub-tasks from a desired global navigation path for each controllable or uncontrollable object. Since we are not dealing with static obstacles in this paper, we will assume that all global navigation paths are already planned without any static obstacles in their paths. Each sub-task along the global path of each object contains a desired goal position and desired arrival time for reaching the position. Based on the information gathered about each sub-task in the first phase, the second phase first creates STGs for objects and then finds an appropriate speed path for each controllable object from the corresponding STG. In order to create simultaneous safe motions for all moving objects, the speeds of the controllable objects are controlled by shifting the corresponding forbidden regions on the STG to avoid potential collisions. As a result, a set of optimal speed paths for all controllable objects are determined. As for the third phase, the planning information of STGs for the next sub-task is updated based on the results obtained from phase two. By repeatedly executing phases two and three until all the sub-tasks along the global paths are finished, a set of near-optimal motion planning for all the dynamic objects can be obtained. In the following, these three phases are described:

(A) Phase one

During the first phase, we assume that the desired global navigation path for each controllable moving object is planned by means of discrete path segments. Along the global path of each object, different intermediate positions can be identified as sub-tasks. According to these discrete positions, the desired arrival time for each sub-task can then be determined from the desired speed of each moving object. The set of the desired speed, goal position and arrival time will then be the initial parameters for constructing STGs corresponding to different sub-tasks for different controllable objects along their global paths. Therefore, the first phase of planning will determine a set of sub-tasks, including desired positions and speeds with desired arrival times, for all objects moving along their global paths.

(B) Phase two

For each sub-task along the global path of a controllable object, we have obtained the desired goal position and arrival time at a desired speed. In this phase, three steps are involved in determining a set of optimal speed paths from space/time graphs according to the priorities of all the controllable objects. Step 1 considers uncontrollable obstacles only. As for step 2, it includes the effect of controllable objects. Step 3 determines a set of optimal speed paths for all the controllable objects so that they can accomplish their sub-tasks. The details of these three steps are described in the following:

Step 1: This step considers only uncontrollable objects in the environment. The controllable moving object with the highest priority will be chosen as the dominant object. Based on the current motions of all the uncontrollable objects, all possible collisions are located along the planned path of the dominant object. Then, the initial forbidden region parameters are calculated and mapped onto the corresponding

STG of the dominate object. From the desired speed and end-point of this subtask, a speed path and goal line of the STG can be created. Based on the STG, many possible collision-free speed paths can be found by avoiding the forbidden regions. Using these paths, the object can reach the goal paths without any collision. Each possible speed path generated will be assigned a local cost C^L. Due to modification of the speed, the goal position may not be reached at the desired arrival time. Any time discrepancy will be considered as a cost factor for searching for the best path. Since there are many feasible speed paths that can be generated, a number of piece-wise speed paths can be planned for the dominant object within this STG. The optimal one will be selected in the following two steps.

Step 2: This step considers controllable objects in the environment. For each speed path of the dominant object generated in step 1, which considers only uncontrollable objects, this step calculates the forbidden region parameters from all the controllable objects and then maps them onto the generated STG. For each speed path, if any possible collision will occur within the range of this sub-path, this collision will be avoided by changing the speed of the subordinates or of the dominant if necessary. The speed change of each subordinate object or of the dominant object incurs additional cost of this speed path. For each sub-path that may be involved in selecting the best path, a local cost C^{L} for this change will also be incurred.

Step 3: In this step, the best speed path among all the feasible speed paths generated in step 2 is determined. The criterion is used to minimize an evaluation function that may include the time discrepancy, the number of modifications etc. The detailed model for optimization will be given in the next section.

The above three steps will are performed sequentially for all controllable objects based on their priorities. After the motions of higher priority objects are planned, these objects become uncontrollable objects for the lower priority objects. The algorithm terminates when all the controllable objects have been considered and their solutions have been generated.

(C) Phase three

During phase two, for each sub-task of a controllable object along its global navigation path, an optimal speed path is generated. If within this sub-task, the desired states, such as the positions and arrival times, of all the controllable objects can be maintained, this set of speed paths is an optimal solution. On the other hand, if the desired states of any controllable object cannot be maintained, e.g. delaying arrival time, the speed of any delayed object has to be adjusted for the next sub-task to maintain the desired global performance obtained in phase one. Therefore, this phase tries to compensate for the modifications made during execution of the previous sub-tasks. The planning information of the STGs for next sub-task are then updated based on the results obtained in phase two. By repeatedly executing phases two and three until all the sub-tasks along the global paths of all the objects are finished, a set of near-optimal motion planning for all the dynamic objects can be obtained.

4. PROBLEM MODELING AND PERFORMANCE EVALUATION

On the basis of the described approach, in this section, we formalize a model for motion planning among dynamic objects. As the motion of an object in an environment changes over time, it can be considered as a dynamic system with states (e.g., the object's position and velocity along its path) that change over time. The behaviors of the *states* are governed by the equation of motion, i.e., the *state equation* or *state transition equation*, of the system. This equation enables us to predict and calculate future states of the system at a particular time based on the knowledge of the present and past states of the system. Therefore, a dynamic system will be proposed for the stated motion problem. The set of speeds for changing the current states to the next feasible states without any collisions while satisfying some performance evaluation criteria will then be a solution for the defined motion planning problem.

To evaluate the performance of a dynamic system, some cost functions must be defined. The selected cost factors must reflect the amount of work involved in making the motion modifications required to complete the task. A good cost assignment will reduce the number of different motions to be pursued during the process. For different problems, the selected cost factors may be different. However, some general cost factors are involved in any application related to motion planning. For example, the number of objects that require speed modification during the process is a cost factor. If there are more modifications, the cost will be higher. In considering the next motion from current states, the priorities and fuel consumption of objects are other types of cost factors. In the scheduling problem, the time deviation from the desired finishing time will also be another cost factor. To evaluate the total cost for the whole planning process, each cost factor aforementioned will be assigned a cost weight. Although some general cost factors can be identified as described above, additional cost factors should also be added, depending on the system and environment restrictions.

On the basis of the above discussion, the problem of planning the motion for the dominant object among static and dynamic objects in space is an N-stage decision (selecting speed) process, where N depends on the number of possible collisions along the object's path. If there are m uncontrollable and n controllable objects, the position and velocity at the k^{th} stage can be represented as two $(m + n) \times 1$ vectors, X_k and V_k, respectively. Furthermore, the position state can be separated into two groups, controllable and uncontrollable, denoted as X_k^c and $X_k^{''}$. Similarly, the velocity state also contains two groups, V_k^c and $V_k^{''}$. Since the velocities of uncontrollable objects cannot be adjusted, the adjustable group will be only V_k^c. For each task, after determining the initial state, position X_0 and velocity V_0, of all the objects, and the initial time at which the process starts, t_0, the dynamic equation of the proposed system during the time interval $[t_k, t_{k+1}]$ can be defined by the following discrete system:

$$V_{k+1}^{c} = {}^{v}F_{k+1}^{c}(X_k, V_k) \tag{4}$$

$$X_{k+1} = {}^{P}F_{k+1}(X_k, V_{k+1}), \tag{5}$$

where $k = 0,\dots, N-1$. Based on the current states of all the objects, the velocity $V_{k+1}{}^{c}$ in Eq. (4) represents the decision made for all the controllable objects, where ${}^{v}F_{k+1}{}^{c}(\cdot)$ represents the decision functions for all the controllable objects at stage $k+1$. $V_{k+1}{}^{c}$ combined with $V_{k+1}{}^{u}$ has the effect of moving all the objects from state X_k to state X_{k+1} as represented in Eq. (5), where ${}^{p}F_{k+1}(\cdot)$ represents the system function at this decision stage $k+1$.

Modeling the above system, for an N-stage decision process with the initial states, X_0 and V_0, a set of speeds, a set of states, and a set of costs from the decision will be generated as follows:

(1) $(V_1{}^{c}, \dots, V_N{}^{c})$, the *speeds* of the controllable objects employed during the process;
(2) $(X_1, \dots, X_N)$, the *positions* of the process to which the speeds $(V_1{}^{c}, \dots, V_N{}^{c})$ give rise; and
(3) $(C_1, \dots, C_N)$, the *decision costs* for controllable objects, corresponding to different states at different stages, where C_i is an n by 1 vector representing n controllable objects. At ith stage, the cost for a controllable object o, $o = 1, \dots, n$, is defined as

$$C_i(o) = \sum_{k=1}^{p(o)} \sum_{j=1}^{q} W_j^L C_j^k \tag{6}$$

where $p(o)$ represents the number of sub-paths on a piece-wise speed path on the STG of object o; q represents the number of local cost factors; C_j^k represents the jth local cost factor for the sub-path k; and W_j^L it represents an assigned weight for the corresponding local cost factor j. Each local cost factor represents the consequence of a modification. For example, changing speeds on STGs to avoid collision with forbidden regions is a cost factor, the time delay is another cost factor for the problem of generating an efficient assembly sequence. In the problem of optimal fuel, the fuel consumption rate is an important cost factor. If the modification involves moving more than one object, the priority of an object is also considered as another cost factor. To determine an optimal solution, performance evaluation is then needed.

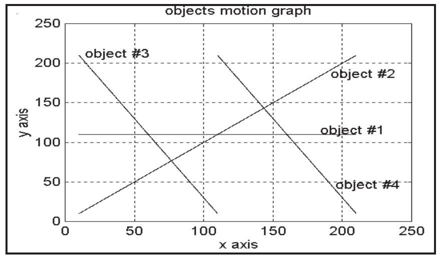
To evaluate the performance for each decision mode, the *criterion function* is a function of states (position and velocity) and the incurred cost, which is defined as $g(X_0, X_1, \dots, X_N; V_1, \dots, V_N; C_1, \dots, C_N)$. The objective of this multi-stage process is to determine: (a) ($V_1{}^{c*}, \dots, V_N{}^{c*}$), an *optimal solution*; and (b) $(C_1{}^{*}, \dots, C_N{}^{*})$, the *optimal performance* resulting from the decision. Based on this formulation, the proposed problem is then to find an optimal solution so that the criterion function, or the *cost function*, can be minimized for a certain task. The optimal solution can be obtained as follows:

$$\begin{aligned} & g(X_0, X_1^*, \dots, X_N^*; V_1^*, \dots, V_N^*; C_1^*, \dots, C_N^*) \\ & = \min_{V_1^c, \dots, V_N^c} [g(X_0, X_1, \dots, X_N; V_1, \dots, V_N; C_1, \dots, C_N)]. \end{aligned} \tag{7}$$

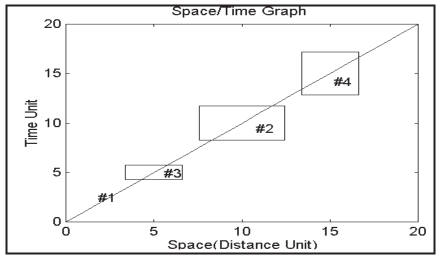
Using an iterative approach [2], an optimal solution for each sub-task along the global path can then be found. According to phase three of the proposed approach, if the modification degrades the global performance, the desired states for the next sub-task will be modified accordingly to compensate for the degraded performance. For example, a delayed airplane will have to speed up to keep up with its schedule. In the next section, two examples will be given to demonstrate the proposed approach.

5. SIMULATIONS OF OFF-LINE EXAMPLES

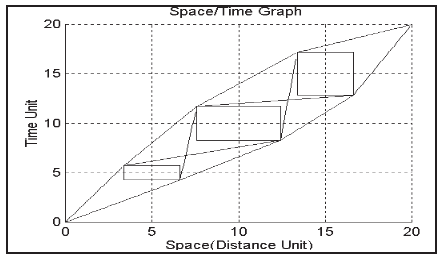
In our simulated environment, three conditions constrain our task. (1) Uncontrollable moving objects do not collide with one another, since there is no way to avoid it. (2) The ending positions of all desired motions are collision free. (3) The priority, weighting factor and cost of the performance evaluation function are predetermined for each controllable object. In the two examples depicted in Fig. 3(a) and 5(a), we assume that the path of every object is a simple straight-line. In example 1, there are four objects in an environment, two controllable objects (objects 1 and 2) and two uncontrollable objects (objects 3 and 4). In



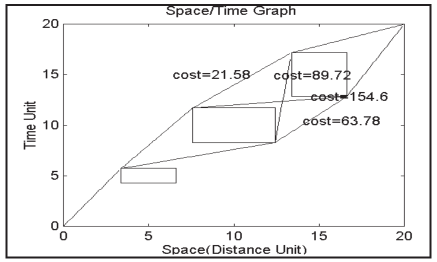
(a) The simulation environment



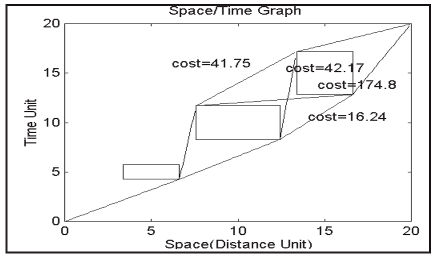
(b) STG of object 1



(c) All speed paths generated



(d) Some of the speed paths and costs generated



(e) Some of the speed paths and costs generated example 2, there are five objects in an environment, two controllable objects, as in example 1, and three uncontrollable objects. In both examples, the starting and ending Cartesian positions of object 1 are described, respectively, by the following homogeneous transformations:

$$POS_1^S = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 110 \\ 0 & 0 & 1 \end{bmatrix}, \qquad POS_1^e = \begin{bmatrix} 1 & 0 & 210 \\ 0 & 1 & 110 \\ 0 & 0 & 1 \end{bmatrix}.$$

As for the second object, its starting and ending Cartesian positions are as follows:

$$POS_2^S = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}, \qquad POS_2^e = \begin{bmatrix} 1 & 0 & 210 \\ 0 & 1 & 210 \\ 0 & 0 & 1 \end{bmatrix}.$$

The length unit in our examples is one millimeter (mm). To simulate the possible collisions in our examples, we assume that no controllable object will move backwards, but that it can stop. In addition, the Cartesian speed limits of objects 1 and 2 are restrained to within 200 mm/sec and 400 mm/sec, respectively. To simplify this 3D detection, the collision condition is determined by checking forbidden regions derived from the paths of each object with respect to all other objects. As described above, the forbidden region is determined by a safety distance with respect to the object. In our simulation, the safety distance for both controllable objects is set to be 10 mm. Therefore, at any point along an object path, a potential collision can be detected by checking three 2D projections. If any 2D projection of any object path intersects with a 2D path projection of any object during a particular time period, the shortest distance between the objects at the intersected points of the projections will be computed. If this distance is less than its corresponding safety distance, then it is considered to be a potential collision point. By checking every distance unit along the path, the forbidden regions for potential collisions involving each object can be determined

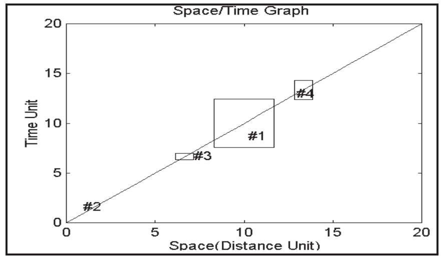
Example 1: The actual environment of example 1 is depicted in Fig. 3(a). In this example, two controllable objects (object 1 and 2) are desired to move initially with constant speeds of 10 mm/sec (V_1^0) and 14.14 mm/sec (V_2^0), respectively, from their starting positions to their ending positions. There are two other uncontrollable objects (object 3 and 4) moving in the space. To simulate the effect of collision, we assume that the safety distances of objects 3 and 4 are 5mm and 8mm, respectively. Object 3 moves from (10, 210) to (110, 10) with a constant speed of 22.36 mm/sec. Object 4 moves from (110, 210) to (210, 10) with a constant speed of 7.45 mm/sec. If there are no collisions, the desired arrival times for these four objects are 20 sec., 20 sec., 10 sec. and 30 sec., respectively. If there are possible collisions for the planned motions, the speeds of the two controllable objects will be controlled within their specified speed limits to avoid collision between them and the two uncontrollable moving objects. In this example, two local cost factors are considered during the performance evaluation process. The first one, C_1^L, is the speed change from the previous subpath, which is computed from $|V' - V| / \text{TU}$, where V' and V denote the speeds of the new subpath and the previous subpath, respectively. TU is the time unit of the STG. The corresponding weighting factor is set to 1 for this example. The second cost factor, C_2^L

is the delay of the arrival time of the object. The corresponding weighting factor is set to $10^1{}_0$, representing no time delay. As this is a one stage decision in this example, according to Eq. (6), the total cost of each generated STG path i for object j can be calculated as follows: $C^i(j) = C_1^i + 10^1{}_0 \times C_2^i$.

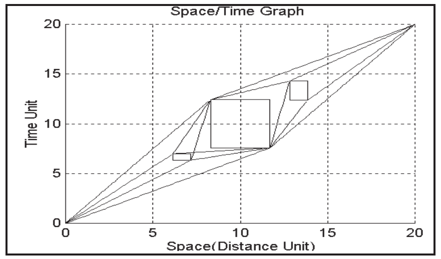
In this example, we assume that object 1 has higher priority than object 2; therefore, object 1 is the dominant object, and its initial STG is shown in Fig. 3(b). The optimal collision-free STG path of object 1 is generated first, followed by the corresponding optimal path of object 2. The 45° line shows that object 1 moves with a speed of 10 mm/sec from the starting position to its goal, which is at 20 DU. In Fig. 3, the DU (Distance Unit) is 10 mm, and the TU (Time Unit) is 1 sec. Throughout our simulations, three forbidden regions are displayed on the STG of object 1 as shown in Fig. 3(b), where the number 2 is the region colliding with object 2, number 3 is that which collides with object 3, and number 4 is that which collides with object 4. Forbidden region 2 represents the collision between two controllable objects, and the lines are dotted to show that this region can be moved because of the controllability of object 2. Region 3 shows that controllable object 1 will collide with object 3. Region 4 shows that controllable object 1 will collide with object 4. The initial STG for object 2 can also be generated, as shown in Fig. 4(a), with a DU of 14.14 mm and a TU of 1 sec. In this example, all the objects collide with object 2, and the area is represented by regions 1, 3, and 4, respectively.

The first step of phase two based on the proposed approach considers uncontrollable objects only. Therefore, forbidden regions of uncontrollable objects 3 and 4 are considered first. Examination of the desired speed of object 1 reveals that the 45° line will collide with regions 2, 3 and 4 in Fig. 3(b). By avoiding collisions, the best modified speeds for object 1 can be found and are shown in Figs. 3(d) and 3(e), respectively. All of the speed paths are represented in Fig. 3(c). In Fig. 3(d), the first speed set generated for object 1 is $(V_1^1, V_1^2, V_1^3, V_1^4)^1 = (5.91, 7.03, 10.61, 23.40)^1$. The superscript outside the parentheses denotes the STG path number. That is, object 1 changes speed from $V_1{}^o = 10$ mm/sec to $V_1{}^1 =$ 5.91 mm/sec and travels for 5.72 sec, then with a speed of 7.03 mm/sec travels for 5.99 sec, then with a speed of 10.61 mm/sec travels for 5.46 sec, and finally arrives its goal on time at a speed of 23.40 mm/sec. The speed change cost along the path can be calculated as $(V_1^1, V_1^2, V_1^3, V_1^4)^2(1) = (4.09, 1.12, 3.58, 12.79)^1$. The superscript inside denotes the number of subpath number while the subscript distinguishes the local cost. In Fig. 3(e), the second speed set generated for object 1 is $(V_1^1, V_1^2, V_1^3, V_1^4)^2 = (15.48, 14.43, 9.27, 4.72)^2$. That is, object 1 changes speed from 10 mm/sec to 15.48 mm/sec and travels 4.28 sec, then with a speed of 14.43 mm/sec for 4.02 sec, then with a speed of 9.27mm/sec travels for 4.54 sec, and finally arrives its goal on time with a speed of 4.72 mm/sec. Its local cost is $(C_1^1, C_1^2, C_1^3, C_1^4)^2(1) = (5.48, 1.05, 5.16, 4.55)$. Since object 1 arrives at it goal on time in both cases, the delay cost factor has no effect.

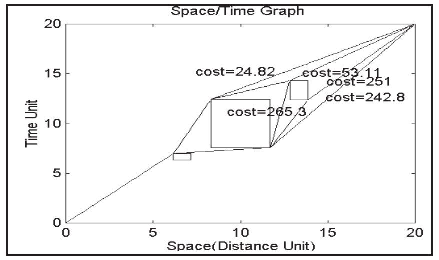
To avoid collisions with uncontrollable objects, different speed paths for object 1 are obtained as shown in Figs. 3(d) and 3(e). According to these modified paths, the corresponding STGs of object 2 are then generated and are shown in Fig. 4(a). To avoid collisions with object 1, 3 and 4 shown in Fig. 4(b), the speed path of object 2 must be modified as shown in Fig. 4(c), with $(V_2^1, V_2^2, V_2^3)^1 = (12.46, 5.57, 21.82)^1$, or as shown in Fig. 4(d), with $(V_2^1, V_2^2)^2 = (9.45, 21.82)^2$. If these two paths are allowable, the corresponding STGs of object 1 after the speed changes of object 2 will be as shown in Figs. 3(d) and 3(e), respectively. Fig. 4(c), as for speed path 2, it is a possible path. Based on a time unit of 1 sec,



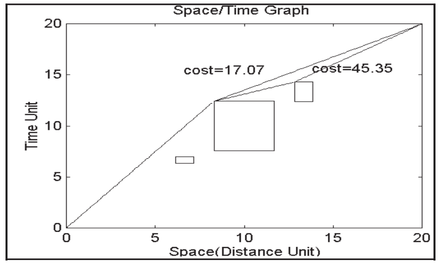
(a) STG of object 1



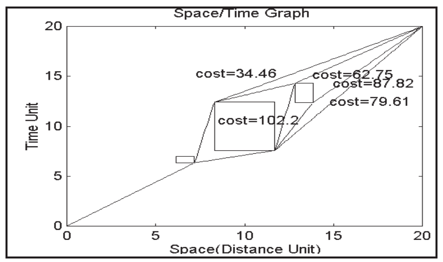
(b) All speed paths generated



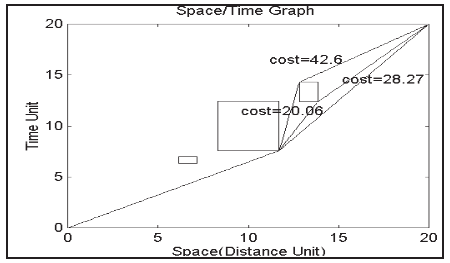
(c) Some of the speed paths and costs generated



(d) Some of the speed paths and costs generated



(e) Some of the speed paths and costs generated



(f) Some of the speed paths and costs generated

Fig. 4. The environment of simulation example 1 (the STGs of object 2 for example 1).

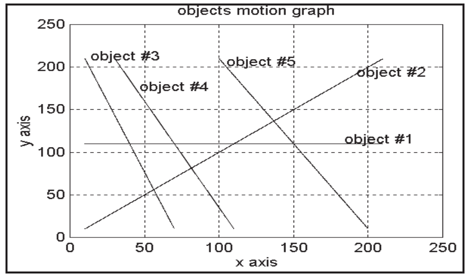
the local cost for this speed path is $(C_1^1, C_1^2, C_1^3)^1(2) = (1.68, 6.89, 16.25)^1$, or Fig. 4(d), with $(C_1^1, C_1^2)^2(2) = (4.70, 12.37)^2$. The superscript inside denotes the subpath number while the subscript distinguishes the local cost. Therefore, the STG of object 1 in Fig. 3(e) are a possible motion planning.

At this point, all the possible collision-free paths have been generated. According to Eq. (6), the cost for STG path 1 of object 1, shown in Fig. 3(d), can be calculated as $C^1(1) = (C_1^1)^1 + (C_1^2)^1 + (C_1^3)^1 + (C_1^4)^1 = 21.58$. For this new path, there is no additional cost for object 2. Similarly, the cost for path 2 of object 1, shown in Fig. 3(e), is $C^2(1) = (C_1^1)^2 + (C_1^2)^2 + (C_1^3)^2 + (C_1^4)^2 = 16.24$. In generating this path, an extra cost for modifying the speed of object 2 as shown in Figs. 4(c)-(f) is needed. This cost is evaluated as $(C_1, C_1, C_1, C_1)^2(2) = (24.82, 17.07, 34.46, 20.06)^2$. Obviously, the first plan in Fig. 3(e) is an optimal solution. The solution set for object 1 can be found as the follows: $(V_1^*, V_2^*, V_3^*, V_4^*)=$ (15.48 mm/sec, 14.43 mm/sec, 9.27 mm/sec, 4.72 mm/sec). As for the motions of object 2 and uncontrollable objects 3 and 4, they will not be affected. They will reach their goals at the scheduled time.

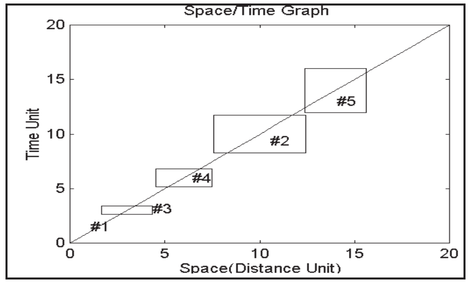
Example 2: In the second example, five objects with different paths are in an environment as shown in Fig. 5(a). The same two controllable objects (object 1 and 2) used in the first example move with desired constant speeds of 10 mm/sec and 14.14 mm/sec, respectively, in the environment with three other uncontrollable objects (objects 3, 4 and 5) moving in the space. Object 3 moves from (10, 210) to (70, 10) with a speed of 34.8 mm/sec. Object 4 moves from (30, 210) to (110, 10) with a speed of 17.95 mm/sec. As for object 5, it moves from (100, 210) to (200, 10) with a speed of 7.99 mm/sec. The safety distances of these three objects are assumed to be 5 mm, respectively. Based on the desired speeds of all five objects, the corresponding desired arrival times are 20 sec, 20 sec, 5 sec, 12 sec and 28 sec. The two controllable objects will be kept from colliding with any other moving object. The same cost function defined in example 1 is used in this example.

As in the first example, object 1 is assumed with a higher priority and its initial STG shown in Fig. 5(b) with 10 mm/DU and 1 sec/TU. Throughout our simulation, four forbidden regions are plotted on the STG of object 1, representing potential collisions with object 2, object 3 , object 4 and object 5 on Fig. 5(b). As for the initial STG for object 2, it is shown in Fig. 6(a) with 14.14 mm/DU and 1 sec/TU. As shown in Fig. 5(b), the first collision occurs between object 1 and object 3. Since objects 3, 4 and 5 are not controllable, collision avoidance can be accomplished by modifying object 1. Examining Fig. 5(b), the solution is found to speed up object 1 so that the path will go to the objects's non-collision field. Fig. 5(c) shows the first possible collision-free speed path combination, $(V_1^1, V_1^2, V_1^3, V_1^4, V_1^5)^1=$ (4.89 mm/sec, 8.34 mm/sec, 6.27 mm/sec, 11.11 mm/sec, 19.17 mm/sec)^1, for object 1. After this modification, object 1 can reach its destination without any collision or time delay. For this modification, the total cost is $(C_1^1, C_1^2, C_1^3, C_1^4, C_1^5)^1(1) =$ (5.11, 3.45, 2.07, 4.84, 8.06)^1. However, there is an extra cost for slowing down object 2. The corresponding STG of object 2 after the speed changes of object 1 is shown in Fig. 6(a). Fig. 6(b) shows the speed path for object 2. To avoid collision with object 1, the speed of object 2 must be changed. However, as indicated in Fig. 6(d), object 2 must be modified with $(V_2^1, V_2^2)^1 =$ (9.45 mm/sec, 21.82 mm/sec)^1. For this modification of object 2, the total cost is $(C_1^1, C_1^2)^1(2) =$ (4.69, 12.37)^1. After these modifications are made, there are no additional collisions among all the objects. Different speed paths and costs generated for object 1 are obtained as shown in Figs. 5(d)-5(h).

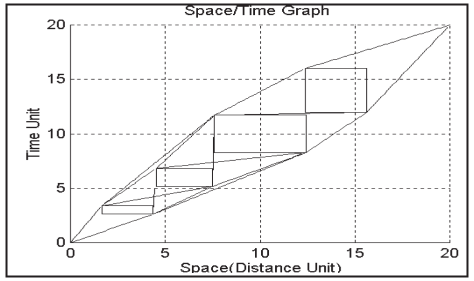
Different generated speed paths and costs for object 2 are obtained as shown in Figs. 6(c)-6(h). According to the iterative approach for evaluating different possible paths, the paths in Figs. 5(g) and 6(d) are the optimal solutions for avoiding collision. The optimal solution for object 1 is as follows: $(V_1^*, V_2^*, V_3^*, V_4^*)$ = (16.62 mm/sec, 14.21 mm/sec, 8.70 mm/sec, 5.46 mm/sec), and $C(1)^* = C_1^1(1) + C_1^2(1) + C_1^3(1) + C_1^4(1)$ = 6.62 + 2.41 + 5.51 + 3.24 = 17.78. As for the optimal solution for object 2, it can be found as follows: $(V_1^*, V_2^*)=$ (9.45 mm/sec, 21.82 mm/sec), and $C(2)^* = C_1^1(2) + C_1^2(2) =$ 4.70 + 12.37 = 17.07. Based on the above planning strategy, all the objects in the environment can reach their respective goals without any collisions.



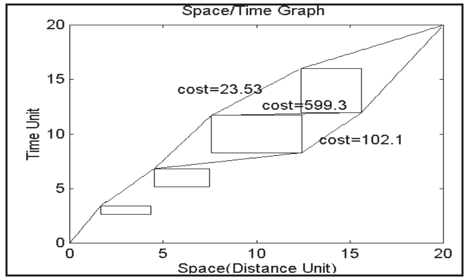
(a) The simulation environment



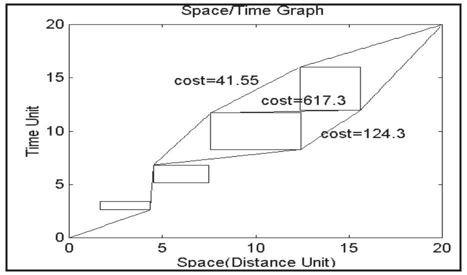
(b) STG of object 1



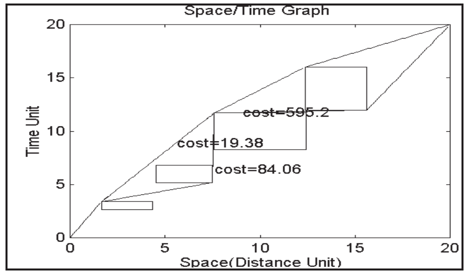
(c) All speed paths generated



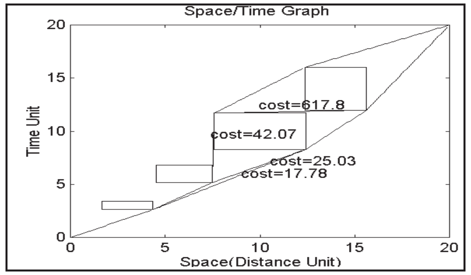
(d) Some of the speed paths and costs generated



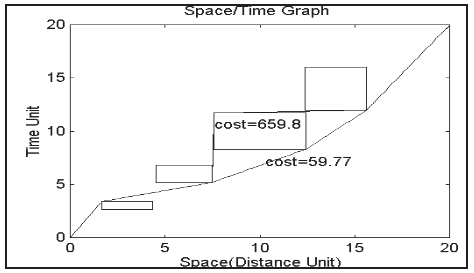
(e) Some of the speed paths and costs generated



(f) Some of the speed paths and costs generated

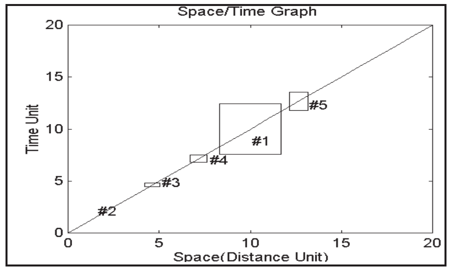


(g) Some of the speed paths and costs generated

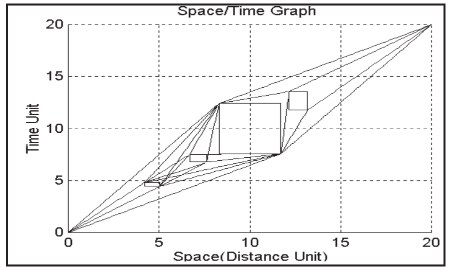


(h) Some of the speed paths and costs generated

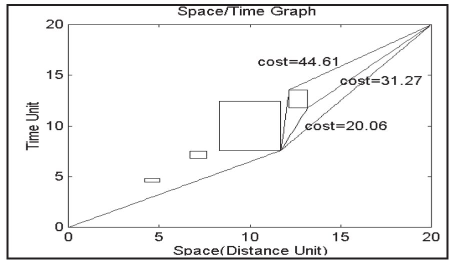
Fig. 5. The environment of simulation example 2 (the STGs of object 1 for example 2).



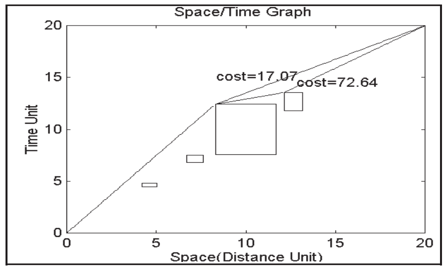
(a) STG of object 1



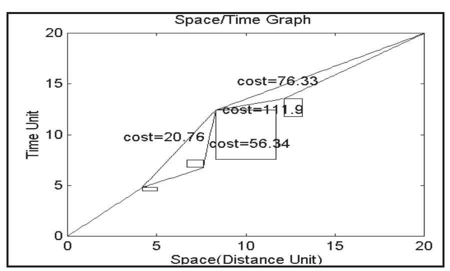
(b) All speed paths generated



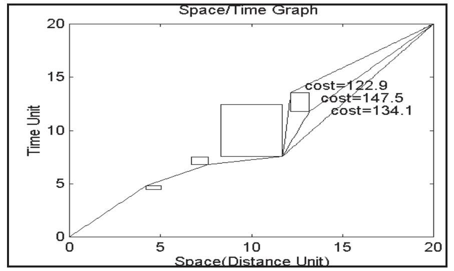
(c) Some of the speed paths and costs generated



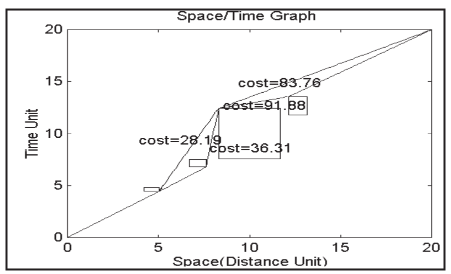
(d) Some of the speed paths and costs generated



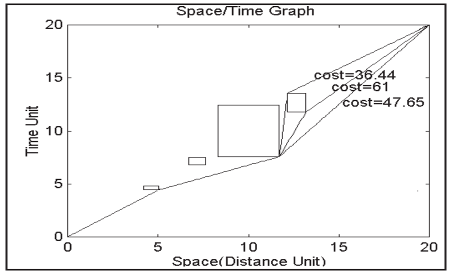
(e) Some of the speed paths and costs generate



(f) Some of the speed paths and costs generated



(g) Some of the speed paths and costs generated



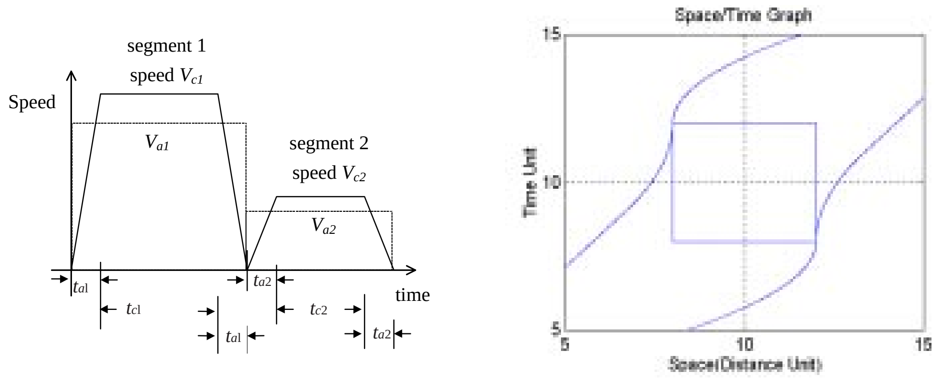
(h) Some of the speed paths and costs generated

Fig. 6. The environment of simulation example 2 (the STGs of object 2 for example 2).

6. TRANSITION BETWEEN SPEED PROFILES

The goal is to drive the robot so that it will transit between the different STG speed segments without experiencing infinite acceleration/deceleration. On the otherhand, the algorithm introduced only provides the associated average speeds for each STG segment along the trajectory. A continuous velocity profile should be maintained for each segment and the whole trajectory for smooth motion. The trapezoidal speed profile has been adopted in many related works [13]. Fig. 7(a) shows two speed profiles for two adjacent STG segments. Let t_{a1} and t_{a2} denote the time of acceleration/deceleration in STG segments 1 and 2; let t_{c1} and t_{c2} be the periods during which the robot moves at constant speed. In this approach, the relationship between the constant speed, V_c, and the average speed, V_a, resolved from the STG graph is represented as

$$V_d = \frac{V_a(t_c + 2t_a)}{t_c + t_a}.$$

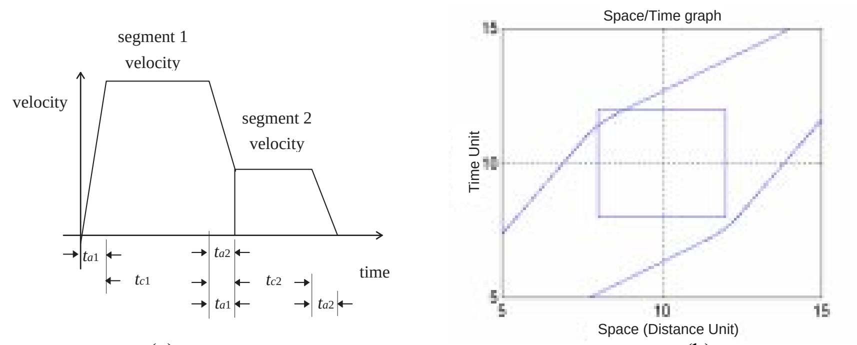


(a) (b)

Fig. 7. (a) Speed profiles for two adjacent STG segments, (b) STG segments in trapezoidal speed profiles.

The real STG segments obtained in this approach are plotted in Fig. 7(b). There is a curve between segments 1 and 2 because of the deceleration/acceleration in the two segments. As mentioned above, to ensure that the robot is able to pass through the forbidden region safely, the region should be drawn a little larger than the calculation indicates. Similarly, to keep the curve from entering the forbidden region, the calculated one should be enlarged by expanding the bottom and top boundaries by a time unit of t_a, for instant.

Nevertheless, as shown in Fig. 7(a), the robot stops moving transiently at the reflexive point while it transits between two STG segments in this approach. One way of preventing the robot from stopping as it goes from segment 1 to segment 2 is to add the two speed profiles togethe; i. e., when segment 1 is decelerating, segment 2 should be accelerating simultaneously. The combined speed profile is obtained by adding the slopes of the individual speed profiles of each segment as shown in Fig. 8(a). Fig. 8(b) shows the modified STG segments obtained using this method. This approach can be extended to a series



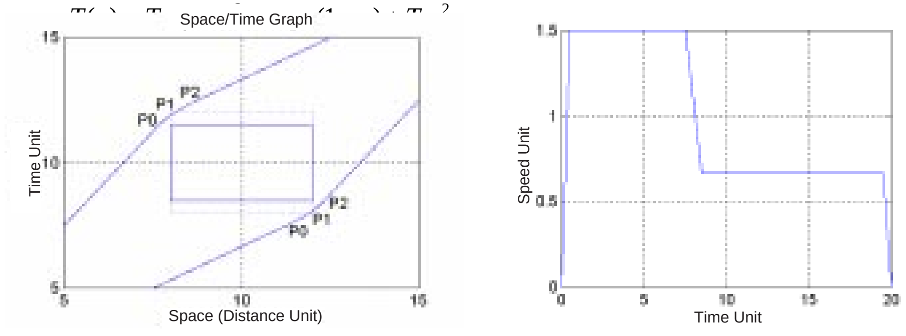
(a) (b)

Fig. 8. (a) The merged profiles of two adjacent segments, (b) STG segments in a merged speed profile.

of n-line segments by simply starting the $(n-1)$th segment's motion at the deceleration point of the nth segment. However, the drawback of this method is that the transversing time of each segment shrinks by one unit of acceleration/deceleration time. This can result in a need to modify the subsequent speeds to maintain the same arrival time at each maneuver point.

To overcome this problem, the spline approximation approach can be used. In this method, the acceleration/deceleration times, T_{acc}s, of all the segments are assumed to be the same. As shown in Fig. 9(a), the connection point, P_1, of segments 1 and 2 can be obtained once the forbidden region is determined. Therefore, the acceleration/deceleration points, P_0 and P_2, for segment 1 and segment 2 can be evaluated by $P_0 = (D_1 - V_{a1} * T_{acc}, T_1 - T_{acc})$ and $P_2 = (D_1 + V_{a2} * T_{acc}, T_1 + T_{acc})$, and the curve between P_0 and P_1 can be obtained using the notable Bezier curve interpolation method [14] :

$$D(u) = D_0(1-u)^2 + 2D_1u(1-u) + D_2u^2,$$



(a) (b)

Fig. 9. (a) STG segments obtained using the Bezier curve interpolation method, (b) The speed profile of two adjacent segments obtained using the interpolation method.

where u is the interpolating index and $u \in [0, 1]$. The transition speed can be derived from

Similar to the previous approach, the performance of this approach in terms of its ability to get close to the connection point between two segments depends on the average speed of each segment. In general, the smaller the speed difference between the two segments, the nearer the curve comes to the point. However, no matter how far the curve is from the point, it never touches the region due to the expansion of the region if the acceleration/ deceleration time is the same for each segment.

7. DISCUSSION AND CONCLUSIONS

This study has focused on collision-free paths among dynamic moving objects. A review of static objects provided us with background and the basic strategies for dealing with collision-avoidance paths. Our study on dynamically moving objects led to a modified path-velocity decomposition method, with the space/time graph allowing for 2D analysis of dynamic objects moving in a 3D environment. Based on the space/time graph, a model for planning motion among moving objects has been outlined. For demonstration purposes, two examples have also been given.

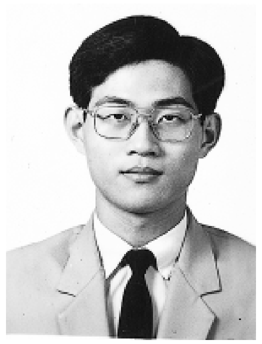
The main focus of this paper has been a collision-free motion among dynamically moving objects. However, as motion planning strategies for avoiding collisions are imperative to problems of sequencing and scheduling, the proposed research can also be applied to these two problems. Sequencing concerns arrangements and permutations in which a set of tasks under different conditions are performed on all machines. As for scheduling, it concerns specification of starting or completion times of certain jobs on all machines. An obvious example is aviation flight traffic control. Sequencing demonstrates very well the applicability of our proposed model to planning of motions among moving objects. One possible application is planning an efficient assembly sequence for an automated assembly center. In manufacturing, robots are often used to assemble many parts, obtained from controllable part feeders, on boards or production plates on some controllable x-y tables. In other words, the locations of different parts on an assembled board are predetermined based on product requirements. The problem is to efficiently use the surrounding machines to assemble these parts. These desired locations of different parts on the board are the intermediate positions of a global path to be determined. The problem is to visit these desired points with the greatest efficiency. Different arrangements for assembly different parts to their desired locations on a board will result in various assembling sequences. Finding the most efficient sequence under certain measures from all possible sequences is an optimization problem. When there exists a large variety of part types, an efficient sequence can really increase productivity. We will continue to study this problem.

Another problem that needs to be studied is the situation of Head-On or San-Diego collision, which requires path modification. These types of collisions cannot be avoided by only adjusting their speeds because of the overlapping navigation path segments. The detour path must be considered and selected in order to avoid collision with an intruder coming from behind or ahead. In some cases, a waiting time should be added between the motions of multiple objects to avoid collision. Path deviation and waiting time will be the main factors in constructing cost functions for performance evaluation. Therefore, the strategy for detouring path under different conditions will be another research topic in the study the general collision-free motion among moving objects or machines.

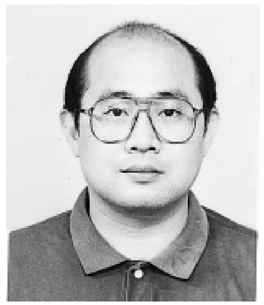
REFERENCES

1. B. Donald, “On motion planning with 6 degrees of freedom: solving the interaction problems in configuration apace,” *IEEE International Conference on Robotics and Automation*, 1985, pp. 782-787.
2. T. Lozano-Perez and M. A. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacles,” *Communication of ACM*, Vol. 22, No. 10, 1979, pp. 560-570.
3. T. Lozano-Peres, “Spatial planning: a configuration space approach,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, 1981, pp. 681-698.
4. T. Lozano-Peres, “A simple motion-planning algorithm for general robot manipulators,” *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3, 1987, pp. 224-238.
5. R. A. Brooks, “Solving the find-path problem by good representation of free pace,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, 1983, pp. 190-197.
6. K. Fujimura and H. Samet, “A hierarchical strategy for path planning among moving obstacles,” *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 1, 1989, pp. 61-69.
7. L. P. Gewali, S. Ntafos and I. G. Tollis, “Path planning in the presence of vertical obstacles,” *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 3, 1990, pp. 331-341.
8. C. L. Shih, T. Lee and W. A. Gruver, “A unified approach for robot motion planning with moving polyhedral obstacles,” *IEEE Transactions on System, Man, and Cybernetics*, Vol. 20, No. 4, 1990, pp. 903-915.
9. K. Kant and S. W. Zucker, “Planning collision-free trajectories in time-varying environments: a two-level hierarchy,” in *Proceedings of IEEE International Conference on Robotics and Automation*, 1988, pp. 1644-1649.
10. W. H. Harman, “Active BCAS surveillance performance,” in *Proceedings of Active Beacon Collision Avoidance System (BCAS) Conference*, 1981, pp. F27-F46.
11. R. Boudarel, J. Delmas and P. Guichet, *Dynamic Programming and its Application to Optimal Control*, Academic Press, N.Y. 1971.
12. K. S. Hwang, M. D. Tsai and M. Y. Ju, “Constant speed trajectory planning of two-link mobile robots,” *Journal of Control Systems and Technology*, Vol. 4, No. 1, 1996, pp. 13-22.
13. R. D. Klafter, et. al., *Robotics Engineering: an Integral Approach*, Prentice-Hall, 1989, pp. 645-648.
14. D. Hearn and M. P. Baker, *Computer Graphics*, 2nd Edition, Prentice-Hall, 1994, pp. 327-355.

Kao-Shing Hwang (黃國勝) is an associate professor in the Department of Electrical Engineering at National Chung Cheng University, Taiwan. He received the B.S. degree in Industrial Design from National Cheng Kung University, Taiwan, in 1981, and the M.M.E. and Ph.D. degrees in EECS from Northwestern University, Evanston, I.L., in 1989 and 1993, respectively. He was a design engineer at the SANYO Electric Co., Taipei, Taiwan, during 1983-1985. Between 1987 and 1988, he joined the C & D Microsystem Co., Plastow, N. H., as a system programmer. His job involves designing PC drivers and graphic animation. Since August 1993, he has been with National Chung Cheng University, Taiwan. He is also the director of the Information Management Division at the university computer center. His areas of interest are neural networks and learning control, robotic compliance, and collision avoidance.



Hung-Jen Chao (趙宏仁) received the B.S. degree in 1996 from I-Sou University, Kaoshiung, and the M.S. degree in 1998 from National Chung-Cheng University, Chiayi, both in electrical engineering. In addition to the modeling and simulation of robotic collision avoidance, his research interests include control theories, neural fuzzy systems, genetic algorithms and machine learning.



Jy-Hsin Lin (林知行) is the chairman of and an associate professor in the Department of Industrial Management at Huafan University. He received the Ph.D. degree in Industrial Engineering & Operations Research from the Polytechnic University of New York. His research interests are scheduling, automated manufacturing systems, and simulation. He is a member of IIE and INFORMS.