COOPERATIVE MOTION PLANNING FOR MULTIPLE VEHICLES IN AUTOMATED STORAGE SYSTEMS

Enrique J. Bernabeu **Josep Tornero**

Departamento de Ingeniería de Sistemas y Automática.
Universidad Politécnica de Valencia. E-46022, Valencia, POB 22012, SPAIN

Abstract: In this paper, a new geometric technique for planning collision-free motions for multiple vehicles in industrial environments is presented. This planner is executed at high speed, as frequent as new position and speed estimations are available. The proposed motion planner is based on the prediction of collisions between pairs of industrial vehicles. Such a prediction is measured as the minimum translational distance between their motions. A predicted collision is avoided by generating intermediate temporal-positions for each vehicle. These intermediate positions are restricted to the feasible path of involved vehicles. Cooperation among mobile vehicle is computed for each pair of vehicles based on priority indices. *Copyright © 2002 IFAC*

Keywords: Collision detection, Collision avoidance, Motion planning, Trajectory planning, Autonomous vehicles.

1. INTRODUCTION

Several motion-planning techniques are proposed in the literature. A randomized motion planner is presented in (Kindel, *et al.*, 2000). This planner samples the space×time state of the robot by picking control inputs at random in order to compute a roadmap that captures the connectivity of the space.

Multiple mobile robots with cooperative collision avoidance in a two-dimensional free-space environment are treated in (Fujimuri, *et al.*, 2000). Each mobile robot is modeled by a circle and collisions are predicted according to geometric aspects of their known motions (predicted crossing point, crossing angles). Then, current navigation is modified by switching the direction angle.

Avoidance behavior rules are proposed in (Mataric, 1992). When a collision is predicted, mobile robots are stopped for a fixed period of time and/or their directions are changed.

Collision prediction and avoidance is also generated by selecting robot velocities outside a set that would result in collision with a given obstacles (Fiorini and Shiller, 1998). It is assumed that the instantaneous state (position and velocity) of each mobile object is measurable. A collision-free trajectory is then obtained by searching a tree of feasible avoidance maneuvers, computed at discrete-time intervals.

The whole time span is partitioned into collision-free and collision intervals (Pérez-Francisco, *et al.*, 1998). By improving the hierarchical representation, collision intervals will become shorter until either they end up in a collision-free interval or a user-defined limit in the geometric model of the object is reached.

The cooperative-motion-planner technique presented in this paper and developed for an automated storage system is based on applying collision prediction tests among the motions of several industrial forklift vehicles as the one shown in Figure 1.

Vehicles are constrained to follow predetermined and invariable paths defined by the shelves location. Each vehicle is equipped with a laser sensor LMS200 from Sick Optic-Electronic with working range of 180º and 150 meters and a precision of 15 mm from 1 to 8 meters. A Kalman filter is used in order to provide to the motion planner positions and speeds of visible vehicles.

Vehicles and other objects in the automated storage system are modeled by spherically extended polytopes (s-topes) (Hamlin *et al.*, 1992). An s-tope is the convex hull containing an infinite set of swept

Fig. 1. Automatic forklift vehicle

spheres. In (Bernabeu and Tornero, 2000a) a method for generating automatically the geometric models of the involved objects is shown.

In our proposed technique, each vehicle motion is represented by the s-tope that contains the volume swept from its current position to its estimated final position. As speed along a given motion is assumed to be constant, the corresponding final position is obtained by translating the current position in a temporal horizon Δt.

Collision prediction is based on the computation of the minimum translational distance between the s-topes representing respectively two given motions. When a collision is predicted, two collision-free intermediate configurations are generated (one for each mobile vehicle) just by changing speed of the involved vehicles.

The remainder of the paper is organized as follows. In Section 2, the geometric-modeling structure is briefly described. Collision prediction and avoidance between two mobile vehicles is shown in Section 3. Collision avoidance constrained to keep the direction of the provided motions is show in Section 4. Trajectory planning technique for cooperative multiple mobile robots is presented in Section 5. Additionally, the application to an automated storage system has been considered. Finally, some conclusions are listed in Section 6.

2. GEOMETRIC MODELING

An spherically extended polytope is the convex hull of a finite set of spheres. A sphere is denoted $s=(c,r)$ where c is the center and r is its radius. Given the set of spheres $\mathrm{S}=\{s_0,s_1,...,s_n\}$, the convex hull of such a set, S_S, contains an infinite set of swept spheres expressed by

$$S_\mathrm{S} = \left\{ s : s = s_0 + \sum_{\mathrm{i}=1}^{n} \lambda_\mathrm{i}(s_\mathrm{i} - s_0), s_\mathrm{i} \in \mathrm{S}, \lambda_\mathrm{i} \geq 0, \sum_{\mathrm{i}=1}^{n} \lambda_\mathrm{i} \leq 1 \right\} \quad (1)$$

The convex hull of spheres does not include all possible spheres that can fit inside the s-tope, only those generated by (1). Spheres in S are called spherical vertices which number determines the order of a S-tope. Simplest s-topes (those formed up to four spheres) are called respectively: sphere, bi-sphere, tri-sphere and tetra-sphere. Graphical examples of s-topes are shown along the paper.

3. COLLISION PREDICTION AND AVOIDANCE FOR MOBILE VEHICLES

The proposed method for distance computation between s-topes is based on an extension of the well-known GJK algorithm (Gilbert *et al.*, 1988). This algorithm computes the distance between two polytopes as the separation between the origin point O and their Minkowski difference set.

In this sense, the Minkowski difference s-tope of two s-topes S_A, S_B, defined respectively by the sets of spheres A and B, is given in (Hamlin *et al.*, 1992),

$$\begin{aligned} S_{AB} = S_A - S_B = \{ s = (c,r) : c = c_A - c_B, r = r_A + r_B, \\ (c_A, r_A) \in A, (c_B, r_B) \in B \} \end{aligned} \quad (2)$$

where c_A, c_B and r_A, r_B are respectively the centers and radii of spheres in A and B.

In the same way that a polytope with four or more points is a polyhedral object with triangular facets (Gilbert, *et al.*, 1988), tetra-spheres (or greater-order s-topes) are objects composed of tri-sphere facets.

The original GJK algorithm is just applied in order to compute distances between polytopes described by the sets of the centers of the spherical vertices.

$$\begin{aligned} C' = \{c'_0, c'_1, ... c'_l\};\ l \leq 3;\ \mathrm{O}^\perp = c'_0 + \sum_{\mathrm{j}=1}^{l} \lambda_{\mathrm{Oj}}(c'_\mathrm{j} - c'_0) \\ \lambda_{\mathrm{Oj}} \in [0,1];\ \sum_{\mathrm{j}=1}^{l} \lambda_{\mathrm{Oj}} \leq 1\ ;\ d_\mathrm{O} = \| \mathrm{O}^\perp \| \end{aligned} \quad (3)$$

where C' is a set, with four centers (spheres) as maximum, taken from the set defining S_{AB}. $\mathrm{O}^\perp$ is the projection of the origin point onto the structure defined by the centers C' and d_O is the resulting distance.

Minimum translational distance (MTD) between two s-topes is then computed according to l.

a) If l=0, i.e. $C' = \{c'_0\}$ with radius r'_0, then

$$\mathrm{MTD}(\mathrm{O}, S_A - S_B) = d_\mathrm{O} - r'_0\ ; \qquad \hat{\boldsymbol{v}}_\mathrm{MTD} = \frac{\mathrm{O}^\perp}{d_\mathrm{O}} \quad (4)$$

$\hat{\boldsymbol{v}}_\mathrm{MTD}$ states the translational vector of the MTD. Sign of MTD codifies the relationship between s-topes

$$sign(\mathrm{MTD}) = \begin{cases} 1, & \mathrm{MTD} > 0 \rightarrow \text{Separation} \\ 0, & \mathrm{MTD} = 0 \rightarrow \text{Contact} \\ -1, & \mathrm{MTD} < 0 \rightarrow \text{Penetration} \end{cases} \quad (5)$$

b) If l=1,2 then

$$\mathrm{MTD}(\mathrm{O}, S_A - S_B) = d_{\mathrm{O}} - \left(r'_0 + \sum_{\mathrm{j}=1}^{l} \lambda_{\mathrm{Oj}} (r'_{\mathrm{j}} - r'_0) \right) \qquad (6)$$

r'_{j}, j=0,1,2 are respectively the radii of the spheres whose centers has been returned in C'. $\hat{\boldsymbol{v}}_{\mathrm{MTD}}$ is obtained as (4). MTD algorithm finishes returning as maximum two parameters λ_{Oj}.

c) If l=3, i.e. $C' = \{c'_0, c'_1, c'_2, c'_3\}$ implies that the O is inside the structure defined by the centers and then a collision is presented (Gilbert *et al.*, 1988). Consequently, MTD is computed as

$$\mathrm{MTD}(\mathrm{O}, S_A - S_B) = -min\{\mathrm{MTD}^+(\mathrm{O}, S'_{012}), \qquad (7)$$
$$\mathrm{MTD}^+(\mathrm{O}, S'_{013}), \mathrm{MTD}^+(\mathrm{O}, S'_{023}), \mathrm{MTD}^+(\mathrm{O}, S'_{123})\}$$

MTD^+, with S'_{ijk} being an extern facet of s-tope S_A-S_B, is obtained applying (6) but considering the spheres $\{s'_{\mathrm{i}}, s'_{\mathrm{j}}, s'_{\mathrm{k}}\}$ and adding the radii expression instead of subtracting it. $\hat{\boldsymbol{v}}_{\mathrm{MTD}}$ is obtained as (4) but its sign is changed because MTD is computed in the opposite direction.

Minimum translational distance between two mobile s-topes is obtained as follows. On the one hand, let A be a n-ordered s-tope whose current position at t_0 is given by s-tope A_{S}, defined by the set of spheres $\{s_0^A, s_1^A, ..., s_{\mathrm{n-1}}^A\}$, where $sc_{\mathrm{i}}^A \in \Re^3$ and $sr_{\mathrm{i}}^A \in \Re$ are respectively the centers and radii of spheres s_{i}^A, i=0,1,...,n-1, and A_{G}, defined by $\{g_0^A, g_1^A, ..., g_{\mathrm{n-1}}^A\}$ with $g_{\mathrm{i}}^A = (gc_{\mathrm{i}}^A, gr_{\mathrm{i}}^A)$, is the estimated future position of s-tope A at t_0+Δt by following path $\hat{\boldsymbol{p}}_{\mathrm{A}}$ with constant speed v_A. Future position is estimated as follows

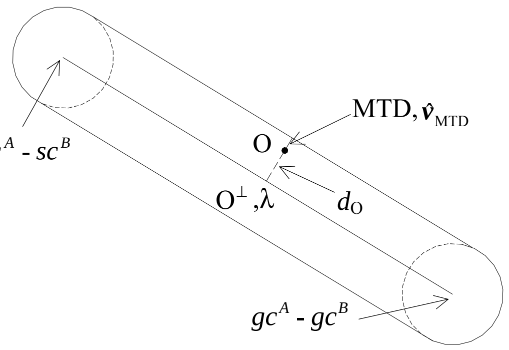
$$gc_{\mathrm{i}}^A = sc_{\mathrm{i}}^A + v_{\mathrm{A}} \cdot \Delta t \cdot \hat{\boldsymbol{p}}_{\mathrm{A}} \quad \forall \mathrm{i} = 0,1,...,\mathrm{n-1} \qquad (8)$$

where $\| \hat{\boldsymbol{p}}_{\mathrm{A}} \|$=1, v_A is speed of A at t_0. On the other hand, let B be a m-ordered s-tope whose current B_{S} and estimated future B_{G} positions are respectively given by the sets of spheres $\{s_0^B, s_1^B, ..., s_{\mathrm{m-1}}^B\}$ and $\{g_0^B, g_1^B, ..., g_{\mathrm{m-1}}^B\}$. Time span is $t\in[t_0,t_0+\Delta t]$. v_B is its speed at t_0.

Motion of s-topes A and B are given by two news s-topes defined respectively by the sets of spheres:

$$\begin{array}{l} \text{From s-tope } A\text{: } \{s_0^A, s_1^A, ..., s_{\mathrm{n-1}}^A, g_0^A, g_1^A, ..., g_{\mathrm{n-1}}^A\} \\ \text{From s-tope } B\text{: } \{s_0^B, s_1^B, ..., s_{\mathrm{m-1}}^B, g_0^B, g_1^B, ..., g_{\mathrm{m-1}}^B\} \end{array} \qquad (9)$$

Each one of the intermediate temporal-position of s-tope A and B, $\{x_0^A, x_1^A, ..., x_{\mathrm{n-1}}^A\}$, $\{x_0^B, x_1^B, ..., x_{\mathrm{m-1}}^B\}$, at $t_x\in[t_0,t_0+\Delta t]$, is characterized by the parameter $\lambda \in]0,1[$ as follows



MTD, $\hat{\boldsymbol{v}}_{\mathrm{MTD}}$
$sc^A - sc^B$
O
$\mathrm{O}^{\perp}, \lambda$
d_{O}
$gc^A - gc^B$

Fig. 2. Computation of the MTD from the Minkowski difference of two mobile spheres

$$\begin{array}{ll} xc_{\mathrm{i}}^A = sc_{\mathrm{i}}^A + \lambda(gc_{\mathrm{i}}^A - sc_{\mathrm{i}}^A) & \forall \mathrm{i} = 0,1,...,\mathrm{n-1} \\ xc_{\mathrm{j}}^B = sc_{\mathrm{j}}^B + \lambda(gc_{\mathrm{j}}^B - sc_{\mathrm{j}}^B) & \forall \mathrm{j} = 0,1,...,\mathrm{m-1} \\ t_x = t_0 + \lambda \cdot \Delta t & \end{array} ; \ \lambda \in]0,1[\qquad (10)$$

Let consider the following Minkowski difference s-topes

$$\begin{array}{l} S_S^{A-B} = A_{\mathrm{S}} - B_{\mathrm{S}} \\ S_G^{A-B} = A_{\mathrm{G}} - B_{\mathrm{G}} \end{array} \qquad (11)$$

S_S^{A-B} is an s-tope formed from s-topes A and B at t_0 and S_G^{A-B} is defined from s-topes A and B at t_0+Δt. Such new s-topes defines a new motion, each one of the infinite intermediate positions S^{A-B} is characterized by parameter $\lambda \in]0,1[$ as follows

$$S^{A-B} = S_S^{A-B} + \lambda\left(S_G^{A-B} - S_S^{A-B}\right) \qquad (12)$$

Substituting in (12) expressions in (11)

$$S^{A-B} = A_{\mathrm{S}} - B_{\mathrm{S}} + \lambda(A_{\mathrm{G}} - B_{\mathrm{G}} - (A_{\mathrm{S}} - B_{\mathrm{S}})) \qquad (13)$$

Operating

$$S^{A-B} = (A_{\mathrm{S}} + \lambda(A_{\mathrm{G}} - A_{\mathrm{S}})) - (B_{\mathrm{S}} + \lambda(B_{\mathrm{G}} - B_{\mathrm{S}})) \qquad (14)$$

Note that $(A_{\mathrm{S}} + \lambda(A_{\mathrm{G}} - A_{\mathrm{S}}))$ represent the motion of s-tope from A_{S} to A_{G} at time span $t\in[t_0,t_0+\Delta t]$. And the same with s-tope B. Therefore, (14) holds the distance between two mobile s-topes at continuous time. Consequently, distance between two mobile s-topes (vehicle) is determined by computing the following minimum translational distance

$$\mathrm{MTD}\left(\mathrm{O}, S^{A-B}\right) \qquad (15)$$

Finally, after computing the mentioned distance, the following parameters are available

$$\mathrm{MTD}, \lambda = \sum_{\mathrm{j}=1}^{l} \lambda_{\mathrm{Oj}}, \ l \le 2, \ \lambda \in]0,1[, \ \hat{v}_{\mathrm{MTD}} = \frac{\mathrm{O}^{\perp}}{d_{\mathrm{O}}} \qquad (16)$$

Figure 2 shows graphically the computation of the MTD of two mobile spheres A and B. Notice that MTD is negative and, consequently, a collision between the provided motions is predicted. For clearness reasons, the corresponding Minkowski difference s-tope has been represented in 2D.

When a collision is predicted between two given motions (9), such a collision is avoided by two intermediate temporal-positions, one for each mobile vehicle. These intermediate configurations are generated as follows:

$$\forall i=0,1,...,n-1;\quad \forall j=0,1,...,m-1$$
$$xc_i^A = sc_i^A + \lambda(gc_i^A - sc_i^A) - \mu\cdot\delta\cdot \mathrm{MTD}\cdot\hat{v}_{\mathrm{MTD}} \qquad (17)$$
$$xc_j^B = sc_j^B + \lambda(gc_j^B - sc_j^B) - (1-\mu)\cdot\delta\cdot \mathrm{MTD}\cdot(-\hat{v}_{\mathrm{MTD}})$$
$$t_x = t_0 + \lambda\cdot\Delta t$$

where $\delta\geq 1$ states a safety threshold. Radii of s-topes do not change. $\mu\in[0,1]$ quantifies the degree of avoidance desired (priority) for each mobile vehicle.

The low computational cost of the minimum-translational-distance algorithm is shown in (Bernabeu and Tornero, 2000b). This low cost allows one to execute the motion-planner algorithm so frequent as new information (speed and position estimations of the involved vehicles) from the environment is received.

4. COLLISION AVOIDANCE BY CHANGING SPEED

The motion planner, applied in its general way, as it is deduced from equation (17), generates the minimum translations of the involved mobile vehicles for avoiding predicted collisions. That is, both updated directions (paths) and updated speeds are generated.

However, in the cooperative-motion planning assumed in the paper, paths are constrained to pre-established lines. As a consequence, only speeds of the involved mobile vehicles are modified in order to avoid predicted collisions. Therefore, the translational motions computed in (17) have to be projected onto the feasible lines in order to generate intermediate temporal-positions.

In this section, the computation of the minimum translational distance constrained to a given direction is described.

Given two motions, firstly, it is applied the computation of the MTD between them. Let $\mathrm{MTD}_{\mathrm{dir}}$ be the minimum translational distance constrained to the pre-established path direction of one of the involved vehicles and represented by the vectors $\hat{p}_A, -\hat{p}_A$. Then MTD, when negative (collision), is updated in order to obtain $\mathrm{MTD}_{\mathrm{dir}}$.

$\mathrm{MTD}_{\mathrm{dir}}$ is obtained (see figures 3 and 4) as follows,

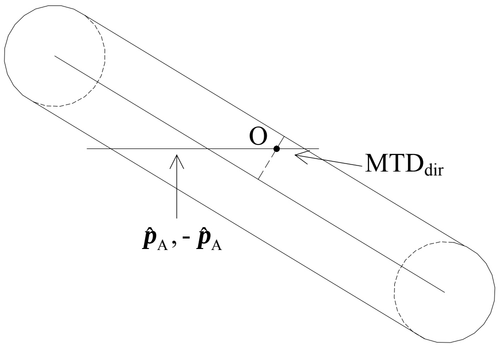
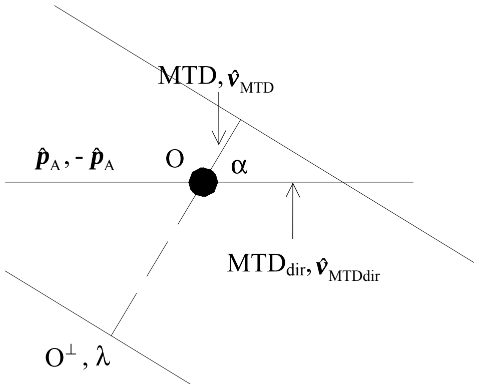


Fig. 3. Computation of the $\mathrm{MTD}_{\mathrm{dir}}$ from the MTD obtained



$\mathrm{MTD}, \hat{v}_{\mathrm{MTD}}$
$\hat{p}_A, -\hat{p}_A$
O
α
$\mathrm{MTD}_{\mathrm{dir}}, \hat{v}_{\mathrm{MTDdir}}$
$O^{\perp}, \lambda$

Fig. 4. Zoom of the involved area for the $\mathrm{MTD}_{\mathrm{dir}}$ computation

$$\alpha = \mathrm{acos}(\hat{p}_A\cdot\hat{v}_{\mathrm{MTD}})$$
$$\left.\begin{array}{ll} \mathrm{MTD}_{\mathrm{dir}} = \dfrac{\mathrm{MTD}}{\cos\alpha};\ \hat{v}_{\mathrm{MTDdir}} = \hat{p}_A, & \text{if } \alpha < \dfrac{\pi}{2} \\ \mathrm{MTD}_{\mathrm{dir}} = \dfrac{\mathrm{MTD}}{-\cos\alpha};\ \hat{v}_{\mathrm{MTDdir}} = -\hat{p}_A, & \text{if } \alpha > \dfrac{\pi}{2} \end{array}\right\} \quad (18)$$

$\hat{v}_{\mathrm{MTDdir}}$ states the vector where the minimum translational distance has been constrained. When α=0, it implies that $\hat{v}_{\mathrm{MTD}} = \hat{p}_A$ or $\hat{v}_{\mathrm{MTD}} = -\hat{p}_A$, and so, it is not needed to apply (18). Note that when $\alpha = \pi/2$, it is because vectors $\hat{p}_A$ and $\hat{p}_B$, representing paths of the involved vehicles, are equals.

In this paper, it is assumed that if vehicles move along the same paths, the translational vector, where a collision is to be avoided, is exceptionally different from their paths vectors, i.e., it can not be computed by (18).

After computing the $\mathrm{MTD}_{\mathrm{dir}}$ and $\hat{v}_{\mathrm{MTDdir}}$ between two vehicles *A* and *B*, it is stated that, on the one hand, when $\hat{v}_{\mathrm{MTDdir}} = \hat{p}_A$, it is needed to speed up vehicle *A* and decelerate *B* in order to avoid the predicted collision. On the other hand, when $\hat{v}_{\mathrm{MTDdir}} = -\hat{p}_A$, it implies that is required a deceleration in vehicle *A* and an acceleration in *B* in order to avoid such a collision.

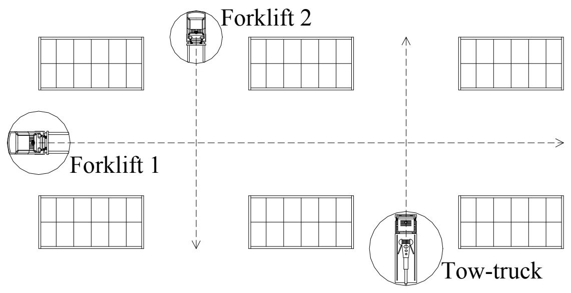


Fig. 5. Automated Storage System

Generally speaking, the technique for collision avoidance shown in this section can be used for computing the minimum translational distance in any given desired direction.

5. COOPERATIVE MOTION PLANNING

In accordance with the previously proposed method for collision prediction and avoidance, an exact algorithm for generating cooperative collision-free motions for multiple industrial vehicles is now introduced.

The algorithm assumes that the instantaneous state (position and speed) of each mobile object is known (measured or estimated).

Mobile robots are modeled by 3D s-topes (2D is enough in most of the situations). Motion of each vehicle is represented by the s-tope joining its current position at t_0 with its future estimated position at $t_0+\Delta t$, and computed as (8) indicates.

The cooperative motion planner runs very fast, so it could be executed at the sampling rate of the position and speeds sensors. That is, the motion planner constitutes a closed-loop approach.

A collision-free motion for each involved vehicle is obtained by computing the minimum directional distance from the motion of a given vehicle *A* to each one of the s-topes representing the motions of the rest of visible vehicles.

After computing the minimum directional distance between the motions of vehicles *A* and *B*, the following parameter are available λ, $\mathrm{MTD}_{\mathrm{dir}}$, $\hat{\boldsymbol{v}}_{\mathrm{MTDdir}}$, $\hat{\boldsymbol{v}}^{B}_{\mathrm{MTDdir}}$. This last parameter is obtained as

$$\hat{\boldsymbol{v}}^{\mathrm{B}}_{\mathrm{MTDdir}} = \begin{cases} \hat{\boldsymbol{p}}_{\mathrm{B}}, & \textit{if } \hat{\boldsymbol{v}}_{\mathrm{MTDdir}} = -\hat{\boldsymbol{p}}_{\mathrm{A}} \\ -\hat{\boldsymbol{p}}_{\mathrm{B}}, & \textit{if } \hat{\boldsymbol{v}}_{\mathrm{MTDdir}} = \hat{\boldsymbol{p}}_{\mathrm{A}} \end{cases} \tag{19}$$

After predicting a collision, two intermediate temporal-positions, that are free of collision, are obtained by applying

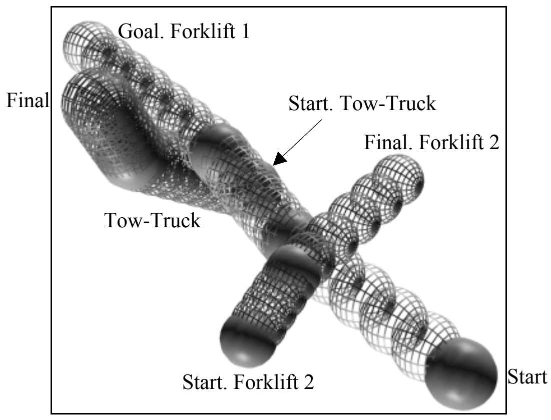


Fig. 6. Cooperative motion for the industrial vehicles with the same priority

$$\forall \mathrm{i}=0,1,\ldots,\mathrm{n}-1; \quad \forall \mathrm{j}=0,1,\ldots,\mathrm{m}-1$$
$$xc^{A}_{\mathrm{i}} = sc^{A}_{\mathrm{i}} + \lambda(gc^{A}_{\mathrm{i}} - sc^{A}_{\mathrm{i}}) - \mu\cdot\delta\cdot\mathrm{MTD}_{\mathrm{dir}}\cdot\hat{v}_{\mathrm{MTDdir}} \tag{20}$$
$$xc^{B}_{\mathrm{j}} = sc^{B}_{\mathrm{j}} + \lambda(gc^{B}_{\mathrm{j}} - sc^{B}_{\mathrm{j}}) - (1-\mu)\cdot\delta\cdot\mathrm{MTD}_{\mathrm{dir}}\cdot\hat{v}^{\mathrm{B}}_{\mathrm{MTDdir}}$$
$$t_x = t_0 + \lambda\cdot\Delta t$$

Notice that, according with (19) and (20), one vehicle is accelerated and the other is decelerated to avoid a predicted collision between them. The degree of acceleration or deceleration is handled by parameter $\mu\in[0,1]$, that states the priority of each vehicle.

This collision-free trajectory planner has been applied to an automated storage system. The scenario, depicted in figure 5, includes two forklifts and a tow-truck with their respective pre-established paths shown by dashed lines.

For that scenario, many situations have been tested. In particular, original speeds of forklift 1, forklift 2 and tow-truck are 2.77 m/sec., 2.22 m/sec. and 1.39 m/sec respectively. Speeds of these vehicles do not change unless a predicted collision has to be avoided. Arrowheads in the dashed lines in figure 5 are taken as final positions for the vehicles.

The motion planner has been implemented in C code running at a sampling rate of 1 sec. In the paper, there are three different situations. For the first one, all the vehicles have the same priority showing the motion depicted in figure 6 with the z coordinate representing the time. Rendered spheres represent the intermediate temporal-position generated. Spheres have been depicted at a sampling interval of 10 sec. Total computational cost of motion-planner algorithm for this cooperative motions is 108 μsec. on a Pentium-II 350MHz

Motions shown in figure 7 result of considering priority zero both the forklift 2 and the tow-truck. Spheres have been represented with the same criteria than in figure 6. The overall motions are computed in 208 μsec. on a Pentium-II 350MHz.

Finally, figure 8 shows the case where the priority of forklift 2 is double than forklift 1. Tow-truck priority is set to zero. Motions are computed in 332 μsec.

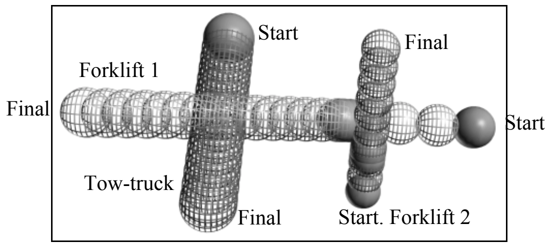


Fig. 7. Priority is set to zero for forklift 2 and the tow-truck

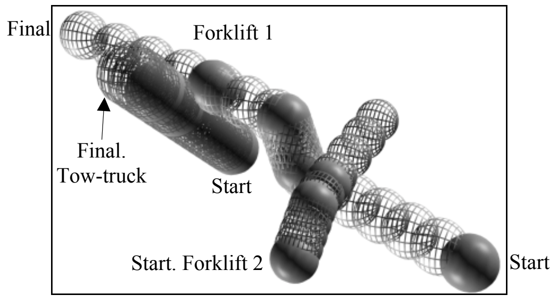


Fig. 8. Forklift 2 priority is double than Forklift 1. Tow-truck priority is set to zero.

Figure 9 shows the vehicles' acceleration and deceleration of the cooperative motions in figure 8. Note that speed of the tow-truck does not change (its priority is zero).

6. CONCLUSIONS

A fast algorithm for generating collision-free trajectories for multiple mobile vehicles has been presented in this paper.

The cooperative-motion-planning technique is based on the computation of translational distances between pairs of mobile vehicles. Additionally, motions have been constrained in order to keep prescribed paths. Therefore, a predicted collision is just avoided by accelerating and decelerating cooperatively pairs of involved vehicles.

The criterion for forcing accelerations or decelerations is established by a set of priorities.

The cooperative motion-planner has been applied to an automated storage system under several different situations. Few of them are described in the paper.

Compared with other motion planners in the literature, main advantages of our proposed planner come from a double aspect. First one, the introduced continuous-time approach for predicting and avoiding collisions among mobile vehicles. In other words, no discretization of the solution space is required. As a second aspect, the motion planner, based on geometrical considerations, is executed at high sampling rate, so frequent as new information from the environment is received.

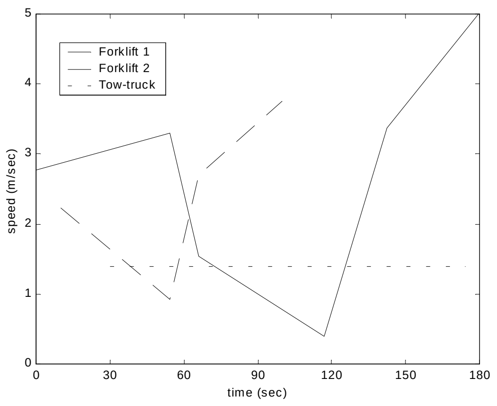


Fig 9. Speed up and deceleration from the example in figure 8.

ACKNOWLEDGEMENTS

This research has been partially funded by the Spanish Government: Plan Nacional de I+D (1FD97-1136) and Ministerio de Fomento, MIMICS Project (BOE 66, 17/3/00)

7. REFERENCES

Bernabeu, E.J., and J. Tornero (2000a). Optimal geometric modeler for robot motion planning. *Journal of Robotic System* **17**(11), 593-608.

Bernabeu, E.J., and J. Tornero (2000b). Real-time generation of collision-free paths for a mobile sphere. In: *IEEE Int. Conf. on Robotics & Automation*, pp 2278-2283

Fiorini, P., Z. Shiller (1998). Motion planning in dynamic enviroments using velocity obstacles. *Int. Journal of Robotics Research* **17**(7), 760-772.

Fujimuri A., M. Teramoto, P.N. Nikiforuk, and M.M. Gupta (2000). Cooperative Collision Avoidance between Multiple Mobile Robots. *Journal of Robotic Systems* **17**(7), 347-363.

Gilbert E.G., D.W. Johnson, and S.S. Keerthi (1988). A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics & Automation*, **4**(2), 193-203.

Hamlin, G.J., R.B. Kelley, and J. Tornero (1992). Efficient distance calculation using spherically-extended polytope (s-tope) model. In: *IEEE Int. Conf. on Robotics & Automation*, pp. 2502-2507.

Kindel R., D. Hsu, J-C Latombe, and S. Rock (2000). Kinodynamic Motion Planning Admist Moving Obstacles. In: *IEEE Int. Conf. on Robotics & Automation,* pp. 537-543.

Mataric M.J (1992). Minimizing Complexity in Controlling a Collection of Mobile Robots. In: *IEEE Int. Conf. on Robotics & Automation,* pp. 830-835.

Pérez-Francisco M., and A.P. del Pobil (1998). Efficient Collision Detection for Real-World Motion Planning. *Practical Motion Planning in Robotics* (K. Gupta, A.P. del Pobil) pp. 243-257, John Wiley & Sons, NY, USA .