Journal of Intelligent and Robotic Systems **29:** 191–210, 2000.

3D Local Trajectory Planner for UAV

JERZY Z. SASIADEK
Department of Mechanical and Aerospace Engineering, Carleton University, 1125 Colonel By Drive, Ottawa K1S 5B6, Canada; e-mail: jsas@ccs.carleton.ca

IGNACY DULĘBA
Institute of Engineering Cybernetics, Wrocław University of Technology, Janiszewski St. 11/17, 50-362 Wrocław, Poland; e-mail: iwd@ict.pwr.wroc.pl

(Received: 22 November 1999; in final form: 9 May 2000)

Abstract. In this paper a local trajectory planner is described and applied. This planner works in three dimensional environment populated with static and passive movable obstacles. The main contribution of this paper is a proposal of a new method of autonomous navigation. Novelty of this approach relies on splitting the motion planning problem into two stages: a decision mode and a trace mode. In the decision mode, vehicle selects its current direction of motion on the basis of the current value of a performance index. In the trace mode vehicle traces boundary and edges of obstacles using its on-board sensors. Depending on vehicle's environment, the two modes follow one another many times. Another new idea is a negative velocity feedback. The feedback stabilizes velocity of the vehicle around a value considered safe in a given environment. The planner, although autonomous, may be adjusted by higher order system (strategic behavior) by changing its parameters. It is not computationally intensive and therefore can be used in real-time applications.

Key words: autonomous systems, collision avoidance, guidance, navigation and control, robotics, trajectory planning, unmanned aerial vehicles (UAV).

1. Introduction

Controlling a vehicle is not a simple task, because environment as well as dynamics of the vehicle must be taken into account while planning its motion. The vehicle can be a free-flying robot, spacecraft, aircraft, helicopter, mobile robot or any autonomous, guided vehicle. The aforementioned objects are quite different with respect to their motion abilities and ranges of environment perception. For example, fixed wing aircraft cannot fly with low velocities and their turning radius is quite large compared to mobile robots which can stop almost immediately and they can, relatively easy, reorient themselves. Moreover, just opposite to mobile robots, the aircraft can detect obstacles at long distances and the number of obstacles in their environment is usually relatively small. Variety of vehicles makes development of a general motion planning algorithm difficult. Nevertheless, in this paper, we will propose some design principles of a general motion planning algorithm for all vehicles. For each particular object, depending on its motion characteristics, some data parameterizing the algorithm should be adjusted appropriately.

Usually, the task of motion planning is divided into two stages: a path planning and a trajectory planning. The path planning has an environment description as an input with its start and goal points. It generates a geometrical path the vehicle needs to follow. The path, in sequence, is time-parameterized by a trajectory planner. The trajectory planner based on dynamic characteristics of the vehicle determines time-dependent characteristics like positions, velocities and accelerations.

The global path planning carries all information before any motion of a vehicle is performed. When a global information known in advance is neither perfect nor predictable, there is a tendency to design so-called local path planners, (Borenstein and Koren, 1991a; Elnagar and Base, 1993). Although such planners lead to the loss of a path optimality but their actions are still focused on a target reaching while avoiding obstacles. The paper addresses a problem which falls in between a path planning and a trajectory planning. It covers a situation when environment data acquired before trajectory planning are not perfectly reliable, but they still exist. A rough map of the environment is built using environment data and a path is planned. A task of local trajectory planner is to plan trajectory based on the path, and our planner must perform tactical decision in order to modify preplanned path when the path, due to lack of reliable data, may collide with obstacles. In this paper three dimensional version of the local trajectory planner will be presented developed by Sasiadek and Dulęba for the 2D case (1995) and adapted into 3D environment (1997).

Before giving a detailed description of a 3-dimensional (3D) local trajectory planner let us properly specify the system among other planners known in literature of the subject, pointing to their advantages and disadvantages. Path planners are generally divided into local and global ones and, usually, they do not take into account dynamics of the vehicle. Former group works in on-line mode while the latter in off-line mode. Known approaches to path planning might be divided into four categories: deterministic, stochastic, learning-based, reflexive (behavioral).

Deterministic approaches are used most often in global path planners. Two deterministic techniques: Voronoi diagrams, (Takahashi and Schilling, 1989), and Visibility Graph (Ilari and Torras, 1990) are described below. A Voronoi diagram represents a collection of locations equidistant from static obstacles. Naturally, the locations are described by a graph with edges and nodes placed at intersection of edges. Each edge is weighted with real (positive) number denoting difficulty of motion between two nodes it connects. Motion planning using Voronoi diagrams relies on finding a path connecting an initial and a final position of the vehicle with nodes of the Voronoi diagram. Then, planning (searching) on the graph between the two nodes is performed (e.g., with the use of the $A^\star$ algorithm). The technique works for static obstacles and is rather computationally involved, especially when obstacles are irregular.

Complementary features to Voronoi diagrams technique has Visibility Graph technique. It works for polygon obstacles. A motion plan is designed by searching through a graph with nodes located at nodes of obstacles-polygons. Coincidence of the nodes is established by a visibility criterion. Planning is fast but resulting path touches boundary of obstacles, therefore it is not safe at all. Due to its efficiency, this technique also works in environments with movable obstacles.

A clear disadvantage of all global motion planners discussed above is the re-planning needed each time environment changes. It happens frequently with movable obstacles. A local path planners do not suffer from this disadvantage. Let us look at an exemplary local path planner based on a potential field principle. At a current position of vehicle, attractive and repulsive forces influence the vehicle. The attractive force is produced by the goal state, while repulsive forces come from obstacles. The resulting (net) force drives the vehicle through small time interval. The potential field method of motion planning works in unstructured environments with many irregular obstacles. The main disadvantage of the method is possibility of getting stuck at a local optimum when resulting force takes the value of zero and there is no direction shown to move towards the goal.

When movable obstacles are considered and their characteristics are known in advance, they may be incorporated in the global path/trajectory planner by adding to spatial variables a time variable and solving the task problem in the such an augmented space (Kyriakopoulos and Saridis, 1992). More often, to represent dynamic environments, i.e., with movable obstacles and unknown or partly unknown characteristics, a stochastic approach is used (Burlina et al., 1992; Zhu, 1991). In stochastic approaches, planners usually work as local planners and a decision of moving vehicle in a particular direction is made on the basis of minimizing probability of collisions. Usefulness of stochastic planners decreases significantly when a single mission in unknown terrain is planned.

The machine-learning techniques are applied (Chuang and Ahuja, 1992) in unknown environments. These techniques are based on an assumption that although the world is not known in advance, but it may be acquired. There are some regularities in the environment and vehicle can gain knowledge about it while navigating.

All methods mentioned above built a model of the world and planned actions consistent with this model. Also, there are other methods which do not employ any world-model building procedure and they act behaviorally (reflexively). Instead of building models, reflexive planners act quickly to avoid nearest obstacles. Unfortunately, they do not exhibit “intelligent” behavior, so usually they are implemented as a bottom level of control system and are controlled by higher levels. Hierarchical architecture organized in layers has been proposed by Brooks (1986).

A task for a local trajectory planner is as follows: given data, geometrical path a vehicle (mobile/free-flying robot) needs to follow (environment is populated by static and movable obstacles) as an input, find a collision-free path and a velocity profile along it. In our case, vehicle is assumed to be a point and its dynamic behavior is incorporated into maximal allowable velocity and acceleration. When the vehicle occupies some volume, it can be reduced to a point by growing obstacles.

Orientation changes of the vehicle are constrained and the constraints are velocity dependent. The vehicle is equipped with sonar sensors. The system, we introduce, builds at first a histographic map (*physical map*) of an environment which in turn is translated into *a control map*. The histogram techniques have been used previously by Borenstein and Koren (1991a; 1991b). It allows to condense all sonar-range information into a sectorial data (distances to the nearest obstacle in the sectors). The data condensed in sectors are virtually sensor independent. In our case the physical map of obstacles is supplemented by a *physical goal*, i.e., the goal the vehicle would direct to from its current position if there is no obstacles around it. The control map, absent in the original histogram approach, modifies the physical map as follows: a *real goal* supplements the physical goal and movable obstacles influence the map. The real goal shows direction the vehicle will direct to in obstacle-populated environment. When there is no obstacles, both goals are the same. Movable obstacles are either active ones, when they may cause collision with the vehicle or passive ones, when, although visible, may not cause any collision. A passive obstacle, when detected, should be filtered out from the control map. In the 3D local trajectory planner only static obstacles and passive movable obstacles will be considered. To deal with active movable obstacles, higher order system should be implemented which having some global environment perception and knowledge may develop a right decision. Nevertheless, some heuristics even at the local trajectory planner can be put forward. For instance: the active obstacle is put at the control map at the place(s) the obstacle may collide with the path of the vehicle, or an extra factor in the criterion function (cf. Equation (11) of Section 2) is added to penalize a path for approaching to a movable obstacle. The two above heuristics do not advise how velocity should be controlled. Should it be increased to avoid obstacles by running it out, or decreased to wait for being passed by a movable obstacle. Obviously, one can find examples when all strategies of avoiding active obstacles may fail, mostly due to the lack of global information. Therefore, active obstacles are not included in our considerations, and have been put as a task for higher order system.

A control map is a base to work out a control rule determining next velocity and attitude of the vehicle based on the map and current values of the parameters. The most demanding task in the system is a real goal determination. The most popular approach to determine a real goal is based on a potential field approach (Hwang and Ahuja, 1992; Ratering and Gini, 1993), although the approach has its own disadvantages (Khatib, 1993). Taking advantage of a preplanned, but not necessarily collision-free path, we need not use the potential field approach. A real goal is determined in two different modes. The first one (*decision mode*), is applied when the vehicle starts its motion. In this mode a criterion function is defined on a set of possible directions of motion. The function should take into account the following factors: distance to the physical goal, difficulty in reorienting the vehicle, and obstacle avoidance. The function is minimized to give the best direction for the next step of control. Unfortunately, the mode cannot be applied alone due to the fact that the vehicle can be trapped in a local minimum of the criterion function, which means that being far from the goal the vehicle may be stopped permanently. To avoid this setback *a trace mode* is added. This mode is switched on when due to obstacles a decision mode cannot determine the approaching direction to the goal. In the trace mode the vehicle follows boundary of the obstacle on the way to the physical goal, until conditions are met to switch on to the decision mode again. In the trace mode, the vehicle can increase temporarily its distance to the goal.

The vehicle selects not only its direction of motion, but also its velocity. The velocity is stabilized at the value considered safe in a given environment. In fact, a negative velocity feedback is implemented. In obstacle-cluttered environment the speed decreases while in obstacle-free terrains its value grows to the maximal allowable value.

The paper is organized as follows: Section 2 describes a local trajectory planner working in 3D environment populated with static and passive movable obstacles. Section 3 presents numerical simulations of the system and in Section 4 the paper is concluded.

2. A Local Trajectory Planner – Modules

In this section, after introducing nomenclature, the 3D local trajectory planner scheme is given and its components discussed.

2.1. NOMENCLATURE

$|\cdot|$, $\|\cdot\|$ absolute value and Euclidean norm, respectively.
$\langle\cdot,\cdot\rangle$, $\cdot\times\cdot$ scalar and cross products, respectively.
$\|\cdot,\cdot\|_{\text{ang}}$ denotes an angle distance between normalized vectors

$$\|x, y\|_{\text{ang}} = \arccos(\langle x, y\rangle), \quad \|x\| = \|y\| = 1. \tag{1}$$

Trian: triangularization of the sphere in $\mathcal{R}^3$ (S^2), is a set of vectors (directions) uniformly distributed on the sphere.

$$Trian(S^2) = \left\{x \in \mathcal{R}^3,\ \forall z\ \|x - z\| < \epsilon, \|x\| = \|z\| = 1\right\} \tag{2}$$

with a small positive constant ϵ determining the number of directions constituting the triangulating set. Triangulation is a domain of a criterion function evaluating possible directions of motion. Each of the direction defines a *sector* centered around the vector.

Strip: subset of a triangulation set, defined as follows: let be given non collinear unit vectors x, y, $z = (x \times y)/\|x \times y\|$, and an strip-angle parameter ξ. The strip groups all directions close to the plane spanned by vectors x, y.

$$Strip(z, \xi) = \left\{r \in Trian\left(S^2\right), \|r, z\|_{\text{ang}} - \pi/2| < \xi\right\}. \tag{3}$$

R_s is a constant denoting a sonar range, i.e., a radius of a ball observed by sensors.

Map is a function which assigns a distance for each direction x from *Trian* to the nearest obstacle in this direction

$$Map(x) = \begin{cases} \mathrm{dist}(x, Obst) & \text{if } \mathrm{dist}(x, Obst) \leqslant R_s, \\ \infty & \text{otherwise}. \end{cases} \tag{4}$$

Sectors with $\mathrm{dist}(x, Obst) = \infty$ are called *obstacle-free* and form *OFree* set, those sectors with $\mathrm{dist}(x, Obst) < \infty$ are called *obstacle-populated* and they are collected in *OPop* set.

$\mathrm{dist}_F(x, OPop)$ is a distance for $x \in OFree$ sector to the nearest obstacle-populated sector

$$\mathrm{dist}_F(x \in OFree, OPop) = \min_{y \in OPop} \|x, y\|_{\mathrm{ang}} \in (0, \pi), \tag{5}$$

$\mathrm{dist}_P(x, OFree)$ is a distance for $x \in OPop$ sector to the nearest obstacle-free sector

$$\mathrm{dist}_P(x \in OPop, OFree) = \min_{y \in OFree} \|x, y\|_{\mathrm{ang}} \in (0, \pi), \tag{6}$$

$v_{\max}$ is a amplitude of a maximal allowable linear velocity of the vehicle, and

$a_{\max}$ is a amplitude of a maximal allowable acceleration/deceleration of the vehicle,

$\mathrm{pos}(t), v(t), o(t)$ denote current position, velocity, attitude (orientation) of a vehicle respectively. Let us assume $\|o(t)\| = 1$. For visualization purposes the attitude is expressed in spherical coordinates.

$\mathrm{knot}_{\mathrm{curr}}$ is a current (sub)goal (knot point),

R_e is an emergency distance allowing to stop the vehicle moving with maximal velocity

$$R_e = \frac{v_{\max}^2}{2a_{\max}}.$$

All the vector-based functions have one more argument. This argument is a position in $\mathcal{R}^3$ when the vectors have their initial point, i.e., the current position of the vehicle.

The task of a local trajectory planner relies on determining a path, velocity and orientation for a vehicle following a prescribed path, possibly populated by

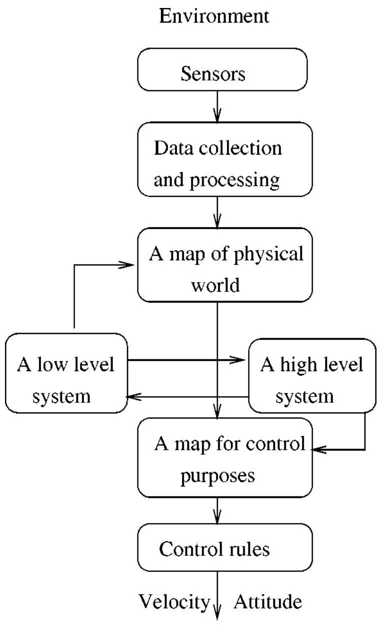

Figure 1. Scheme of a local trajectory planner.

static or/and passive movable obstacles. The scheme of a local trajectory planner is depicted in Figure 1.

A path to follow is supplied to a local trajectory planner by a higher level system. It takes a form of a set of knot-points (*knots*) connected by straight lines in Cartesian space (*segments*). An *active segment* is the segment where the trajectory is being planned. The first active segment has its left boundary at the starting point. Two knots are distinguished. The initial knot where the vehicle begins its motion and the final one which is a goal the vehicle is planning to reach. A neighborhood of each knot, called a transition area, is a circle with a specific radius. When the vehicle reaches the transition area around knot point within the radius, the active segment is changed. Segment changes have been introduced to make the resulting trajectory smooth. Otherwise, with segment changes, the trajectory would be sharp at knots. Formally, the path can be described as follows:

$$\begin{aligned}&\mathrm{PATH} = \left\{\mathrm{knot}_i \in \mathcal{R}^3,\ i = 0, \dots, N\right\},\\&\mathrm{RADII} = \{r_i \in \mathcal{R},\ i = 1, \dots, N-1\},\end{aligned} \tag{7}$$

where $N+1$ is the number of knots (knot_0 is the the initial, and knot_N is the final knot). The r_i denotes a radius of segment change for the ith knot.

Vehicle equipped with sonar sensors obtains information about its environment. After preliminary processing the data, it creates a physical map of the environment. The map is just a sphere with unit radius divided into sectors. In each sector (x) a resultant obstacle is assigned which includes obstacles within the sector. The obstacle is characterized by a distance $Map(x)$ from the current position of the vehicle.

The vehicle occupies a center of the physical map and is characterized by its current velocity and orientation. Vehicle's motion is constrained by parameters reflecting its dynamic properties. Vehicle's velocities, accelerations and reorientation abilities are constrained

$$0 \leqslant v(t) \leqslant v_{\max}, \qquad a(t) \leqslant a_{\max}, \tag{8}$$

$$\Delta\alpha = f(v(t)), \tag{9}$$

where $v(t),\ a(t)$ are amplitudes of velocity and acceleration of the vehicle, respectively; $\Delta\alpha$ is a maximal allowed orientation change in a unit interval of time. It is reasonable to assume that the last quantity is velocity dependent. Actually, it is computed according to the formula

$$\Delta\alpha\bigl(v(t)\bigr) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\left(1 - \frac{v(t)}{v_{\max}}\right), \tag{10}$$

where $\alpha_{\min}, \alpha_{\max}$ are prescribed values of minimal and maximal orientation changes.

The physical map is supplemented by a physical goal (PG). PG is a unit vector pointing at a current segment point towards which the vehicle would direct itself in the absence of obstacles. In Figure 2 three different cases of determining PG are shown:

(a) if a current segment lies within vehicle's sonar range, then PG is one of the two points where sonar range intersects with the segment line. It is always the nearest point to the given segment end-point;
(b) if a goal (final knot) is within the sonar range, then PG coincides with the goal-point;
(c) if a current segment is out of vehicle's sonar range, then PG is a point where a bisector of an angle between a line perpendicular to the segment from current vehicle location and a line to the segment's end point crosses the segment;
(d) if the distance from the current location of the vehicle to the ith knot (segment end-point) is smaller than r_i then the current segment is changed for the next in sequence of segments. Then, one of the (a)–(c) cases applies.

The map of physical world can be viewed as a picture taken from the current position of the vehicle with a physical goal added. Real obstacles are lumped together to form a virtual obstacle which populates any given sector. Unfortunately,

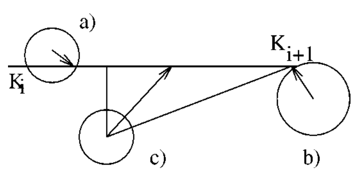

Figure 2. Three cases of determining the physical goal (PG).

such a map cannot be used directly for control purposes because the physical goal may be obscured. One of the problems is related to movable obstacles. If obstacles are passive, they should be filtered out from the map. If obstacles are active, they must be transferred to locations which are most dangerous to the vehicle and their previous positions (should be) filtered out as well. These modifications are the basis for designing a control map (Map_{CTRL}) which is a modified physical map. The physical goal (PG) has an equivalent in the control map. In general, the vehicle will be directed to the real goal (RG). RG may be just PG, if there is no obstacles, but this is not true in obstacle-crowded environment. RG is determined in two possible modes:

Decision mode. In this mode a direction of motion at a current location is determined basing of three factors: safety, goal reaching and difficulties in reorienting a vehicle.

Trace mode. In this mode decisions worked out by the decision mode are executed. The vehicle follows obstacle boundaries until the decision mode is invoked again.

Below details of the two modes are presented.

2.1.1. *Decision Mode*

In the decision mode all obstacle-free ($x \in OFree$) sectors are evaluated according to the following performance index

$$J(x) = k_1 J_1(x) + k_2 J_2(x) + k_3 J_3(x), \quad \|x\| = 1, \tag{11}$$

and the optimal sector becomes the real goal

$$J(\mathrm{RG}) = \min_{x \in OFree} J(x). \tag{12}$$

Components of the criterion function $J(x)$ are the following:

- $J_1(x)$ represents the angle distance from the physical goal to the sector x

$$J_1(x) = \|x, \mathrm{PG}\|_{\mathrm{ang}}, \qquad \|\mathrm{PG}\| = \|x\| = 1, \tag{13}$$

- $J_2(x)$ represents safety measures in maneuvering among obstacles

$$J_2(x) = \pi - \mathrm{dist}_F(x, OPop). \tag{14}$$

  The π term has been introduced to reflect small values of this component for the safest sectors. Note, that in case of $J_2(x)$ sectors with smaller distance from PG than the prescribed one, J_2 assumes value 0. This design allows for not to punish sectors which are fairly safe.

- $J_3(x)$ represents difficulties in reorienting the vehicle

$$J_3(x) = \|x, o\|_{\text{ang}}, \quad \|o\| = \|x\| = 1, \tag{15}$$

where o is a current attitude of the vehicle.

Positive coefficients k_1, k_2, k_3 are introduced to allow for a comparison among different quantities and, more importantly, to weight components of the performance index $J(x)$.

The algorithm is designed in such a manner that after selecting the real goal, a distance from the current position to the next knot point is saved as $\text{dist}_{\text{PG}} = \|\text{pos} - \text{knot}_{\text{curr}}\|$. This distance is important in switching from a trace to a decision mode.

2.1.2. *Trace Mode*

This mode has been introduced to prevent the vehicle from stopping at a local minimum of the criterion function defined in the decision mode. The real condition to switch from the decision to trace mode does not require stopping the vehicle. To switch modes, the following condition must be satisfied:

$$\text{dist}_P(\text{PG}, \textit{OFree}) \geqslant \text{dist}_{D2T}, \tag{16}$$

where dist_{D2T} is a given constant. The switching condition states that the physical goal and its neighborhood is populated with obstacles. The condition (16) prevents the vehicle from getting too close to an obstacle, just to minimize the distance to the goal-point. It invokes a trace mode when the vehicle is relatively far from obstacles and on its way to the goal. In the trace mode the vehicle traces obstacle boundaries (at some distance). The tracing procedure is as follows:

1. When trace mode is called, dist_{PG}, PG_{mem} and RG_{mem} are recorded. PG_{mem} and RG_{mem} are values of PG and RG in the last moment of the decision mode, respectively. Moreover, the point A is determined in the $\text{PG}_{\text{mem}}, \text{RG}_{\text{mem}}$ plane which shows boundary of the obstacle to be traced (Figure 3).
2. All directions evaluated in the trace mode to determine a real goal should belong to the strip,

$$\text{RG} \in \textit{Strip}\left(\frac{\text{RG}_{\text{mem}} \times \text{PG}_{\text{mem}}}{\|\text{RG}_{\text{mem}} \times \text{PG}_{\text{mem}}\|}, \eta\right).$$

   The condition imposed on RG states that RG should be kept in proximity of the plane $\text{PG}_{\text{mem}}, \text{RG}_{\text{mem}}$ within a prescribed angle η.
3. In the successive iterations, when new maps are built, obstacle populated sectors from the strip are evaluated to determine a "successor" of the boundary point A. The "successor" is simply a new coordinate of the point A in the

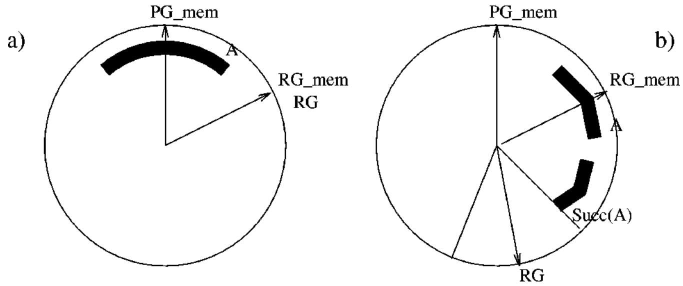

Figure 3. Determining RG in trace mode: (a) immediately after switching from decision mode, (b) couple of iterations later.

next iteration. The point A indicates the boundary of the currently traced obstacle (edge of the obstacle). Let us assume β as a safety-range parameter. If the successor has no obstacles in the range of 2β, measured clockwise in the $\mathrm{PG}_{\mathrm{mem}}, \mathrm{RG}_{\mathrm{mem}}$ plane, then a real goal is determined according to the formula

$$\mathrm{RG} = Rot\left(\frac{\mathrm{RG}_{\mathrm{mem}} \times \mathrm{PG}_{\mathrm{mem}}}{\|\mathrm{RG}_{\mathrm{mem}} \times \mathrm{PG}_{\mathrm{mem}}\|}, -\beta\right) \cdot Succ(A). \tag{17}$$

$Rot(w, \theta)$ is a rotation transformation about a given vector $w = (w_1, w_2, w_3)^{\mathrm{T}}$ by the angle θ (Spong and Vidyasagar, 1989):

$$Rot(w, \theta) = \begin{bmatrix} w_1^2\xi_\theta + \cos\theta & w_1w_2\xi_\theta - w_3\sin\theta & w_1w_3\xi_\theta + w_2\sin\theta \\ w_1w_2\xi_\theta + w_3\sin\theta & w_2^2\xi_\theta + \cos\theta & w_2w_3\xi_\theta - w_1\sin\theta \\ w_1w_3\xi_\theta - w_2\sin\theta & w_2w_3\xi_\theta + w_1\sin\theta & w_3^2\xi_\theta + \cos\theta \end{bmatrix},$$

where $\xi_\theta = 1 - \cos\theta$. If the successor does not satisfy the safety condition, next successor of A is searched clockwise, cf. Figure 3.

Keeping close to the plane prevents the vehicle from wandering on the obstacle surface.

The decision mode is called when PG is clearly visible and a distance to the current goal is smaller than the value stored in the memory when the trace mode was originally invoked, i.e.,

$$\mathrm{dist}_F(\mathrm{PG}, OPop) \geqslant \mathrm{dist}_{T2D}, \qquad \|\,\mathrm{knot}_{\mathrm{curr}} - \mathrm{pos}\| \leqslant \mathrm{dist}_{\mathrm{PG}}, \tag{18}$$

with constant parameter dist_{T2D}. It should be noted that by following boundaries of obstacles the vehicle could increase the distance to the (sub-)goal, yet, purposefully, it avoids to be trapped in a local optimum of the criterion function.

2.2. VELOCITY AND ORIENTATION CONTROL RULES

All concepts introduced previously were geometric in nature. The control rules module (Figure 1) which takes into account vehicle's dynamic characteristics will be discussed next. The module (using control map and a current vehicle's attitude and velocity) works out the next values of the attitude and velocity. Instead of using rather complicated procedures (Elnagar and Base, 1993), it is proposed that a very simple mechanism based on the concept of negative feedback should be considered. The feedback works as follows: the vehicle compares its current velocity with a velocity considered safe in the given environment. If the difference is negative, it increases the current velocity, otherwise decreases the velocity. Below, this concept is presented more formally.

The objective of control system rules is to derive $(v(t+\Delta t), o(t+\Delta t))$ on a basis of $(v(t), o(t))$ and $Map_{CTRL}(t)$. Let us start from the simpler component: the attitude

$$o(t+\Delta t) = Rot\left(\frac{o(t) \times \mathrm{RG}(t)}{\|o(t) \times \mathrm{RG}(t)\|}, \gamma\right), \tag{19}$$

where

$$\gamma = \begin{cases} \Delta\alpha \cdot \Delta t, & \text{if } \Delta\alpha \cdot \Delta t \leqslant \|o(t), \mathrm{RG}(t)\|_{\text{ang}}. \\ \|o(t), \mathrm{RG}(t)\|_{\text{ang}}, & \text{otherwise.} \end{cases} \tag{20}$$

$\Delta\alpha$ is computed according to Equation (10).

The velocity control rule is as follows

$$\begin{aligned} \tilde{v} &= v(t) + g(t) \cdot \Delta t \cdot a_{\max}, \\ v(t+\Delta t) &= \begin{cases} 0, & \text{if } \tilde{v} \leqslant 0, \\ \tilde{v}, & \text{if } 0 < \tilde{v} < v_{\max}, \\ v_{\max}, & \text{otherwise,} \end{cases} \end{aligned} \tag{21}$$

where $g(t) = g(v(t), Map_{CTRL}(t)) \in [-1, 1]$ is a function selected by designer. This should increase the speed, if there is no obstacles, and decrease, if obstacles may be dangerous to the vehicle. In order to formalize what may constitute "the dangerous situation" to the vehicle, the measure of "safe distance" from obstacles has been introduced. This measure takes the form of the following function

$$\mathrm{dist}_{\mathrm{safe}} = \frac{v^2}{2a_{\max}} = R_{\mathrm{e}} \cdot \frac{v^2}{v_{\max}^2}. \tag{22}$$

This distance is compared with other distances to obstacles in *an active area*, AA,

$$\mathrm{dist} = \min_{x \in AA} Map_{CTRL}(x), \tag{23}$$

$$AA = \left\{x \in Trian\left(S^2\right), \left|\|x, o\|_{\text{ang}} + \|x, \mathrm{RG}\|_{\text{ang}} - \|o, \mathrm{RG}\|_{\text{ang}}\right| < \delta\right\}, \tag{24}$$

where δ is a prescribed constant and o is the current orientation of the vehicle. The difference $y = \mathrm{dist} - \mathrm{dist}_{\mathrm{safe}}$ is an argument of $g(t) = g(y)$ function depicted in Figure 4.

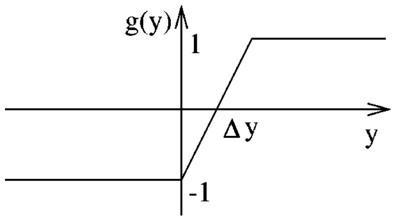

Figure 4. The $g(a)$ function.

Let us analyze how the negative feedback works in this case. If there is no obstacles, $\text{dist} > \text{dist}_{\text{safe}}$, and the vehicle is sped up until it reaches its maximal allowable velocity $v_{\max}$. If there are some obstacles, all depends on vehicle's current velocity. When it is small enough that $\text{dist} - \text{dist}_{\text{safe}} > \Delta y$, the vehicle is sped up slightly. When the difference is positive, yet smaller than Δy, the vehicle is slowed down, but not drastically. When obstacles are dangerous to the vehicle, $\text{dist} - \text{dist}_{\text{safe}} \leqslant 0$, maximal allowable deceleration is applied. The vehicle reduces its velocity (sometimes even to zero) and by reorienting itself finds sectors for which $\text{dist} - \text{dist}_{\text{safe}}$ has a value which allows to increase velocity. Consequently, the negative feedback stabilizes velocity. Now, let us consider the most demanding case when a movable obstacle periodically appears and disappears at the boundary of the perception (sonar range) area in the direction towards the goal. Because the obstacle is far from the vehicle, then the velocity might decrease, but only slightly. More serious problems may be encountered when the vehicle tries to change its orientation. Obviously, sudden changes of the orientation are forbidden, cf. Equation (19), but chattering in orientation is theoretically possible. In order to prevent this extremally rare case, the obstacle is to be added to the physical map when the obstacle enters the area with the radius $R_s - \Delta R_s$, slightly smaller than R_s centered at the current vehicle's location. The obstacle is written off from the physical map when it leaves the area of the radius R_s. In this case, no obstacles can disappear suddenly, so no chattering in orientation is possible. The value of ΔR_s can be selected in advance based on estimated motion abilities of possible obstacles.

Some vehicles, e.g., an aircraft, cannot reduce their velocities below a given velocity limit. Such constraints are to be introduced into admissible velocity (Equation (21), and attitude changes, Equations (9), (19), (20). Also they can be taken into account by the performance index given by Equation (11) to punish sudden velocity changes and/or their exhaustive reduction. With the use of modified performance index, the aircraft selects a reorienting policy rather than a significant velocity decrease.

Quite different problems one can encounter with using the proposed local trajectory planner while planning a trajectory for (small) ships. Those objects are very sensitive to wind and water currents. For ships, the performance index should reflect the fact that effective velocities depend on the mentioned above factors. Consequently, the cost of exerting velocities at different directions may not be the same. However, even in the worst case, when the vehicle's engine is too weak to keep desired course, the planner still tries to minimize an error measured by the difference between the current and desired course.

Discussed examples of vehicles show that the planner is general and flexible enough to cover different and rare cases of navigation. The planner behaves rationally even when it cannot accomplished a given task. There are three main reasons for possibility of the failure. First one is due to constraints on maximal admissible velocity and acceleration of the vehicle when a fast moving obstacle crosses the vehicle trajectory. The second one is when the vehicle is surrounded by many obstacles cannot escape the trap. The third reason is due to numerous, slow moving obstacles colliding with trajectory of the vehicle. The latter two cases of failure are characteristic to local trajectory planners as the planners, having only local information (within their sonar ranges), do not perceive dangerousness of the situation which can be clear only when a global information is involved.

3. Numerical Examples

In order to evaluate the proposed 3D local trajectory planner, several tests were performed. The objective of the first test was to verify overall performance of the 3D local planner. The assumptions for the experiment are as follows: there are only static obstacles (cuboids) whose parameters are shown in Table I. The path to follow is given also in Table I. The following initial data have been assumed: $o(0) = (0, -1, 0)$, $v(0) = 0$ [m/s], $k_1 = k_3 = 1$, $k_2 = 0$, $v_{\max} = 10$ [m/s], $a_{\max} = 10$ [m/s^2], $\alpha_{\min} = 20°$/s, $\alpha_{\max} = 100°$/s. Simulation results are presented in Figures 5–8. Let us analyze the resulting path presented in Figure 5. At the very early stages of motion, the vehicle works in the decision mode. As its orientation is not the same as the direction to the first sub-goal, it requires to reorient itself and its path approaches dashed line starting at $(0, 0)$. A crucial role in reorienting the vehicle plays the $J_1(x)$ component of the performance index $J(x)$, see Equation (11). At the point marked A, the vehicle faces obstacles on its way to the first subgoal.

Table I. Obstacle parameters (x, y, z coordinates of the lower left corner of a cuboid, Δx, Δy, Δz lengths of its edges) and knot points – a path to follow. All data expressed in [m].

Obstacles						Path			
x	y	z	Δx	Δy	Δz	x	y	z	Radius
20	30	45	20	20	5	0	0	0	0
35	30	25	5	20	20	40	70	50	10
20	45	25	20	5	20	80	70	40	10
65	65	0	5	5	100	90	35	20	5
65	80	0	5	5	100				

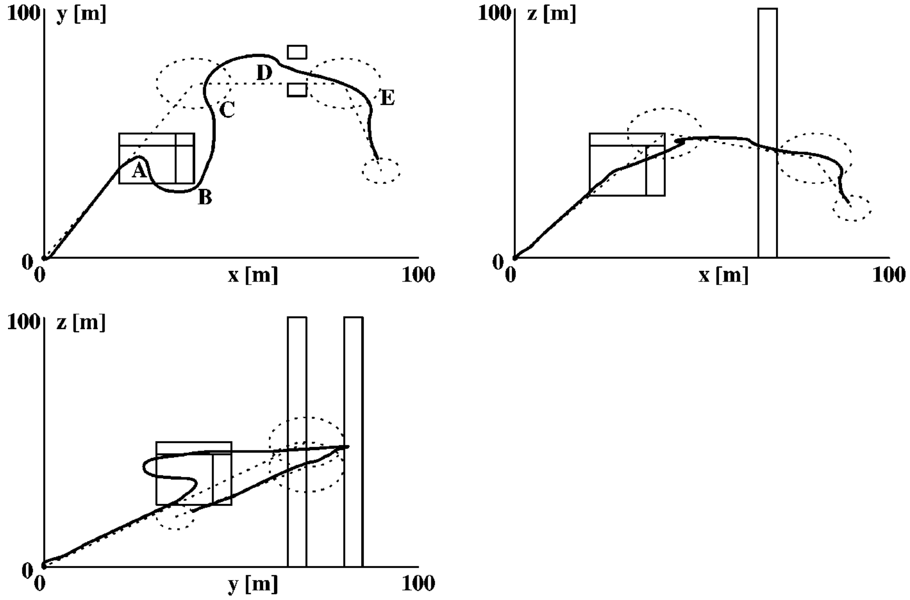

Figure 5. Motion in 3D space.

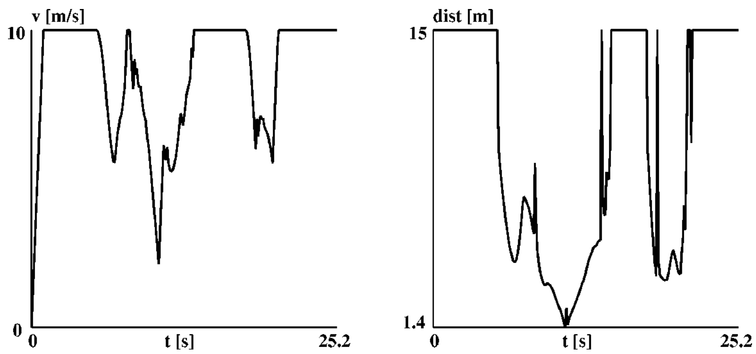

Figure 6. Velocity along the path and the distance to the nearest obstacle.

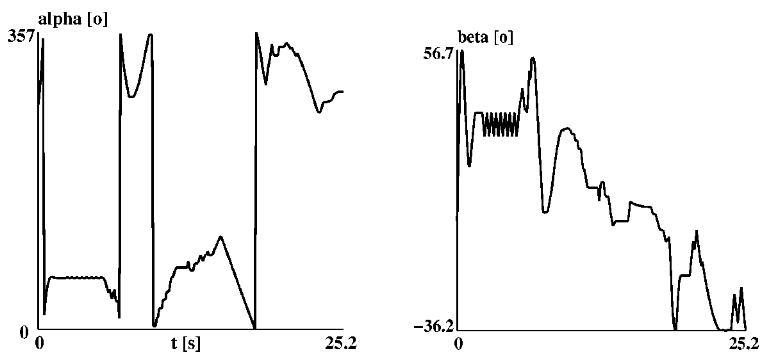

Figure 7. Vehicle orientation – a nutation (alpha) and a precession (beta) angle.

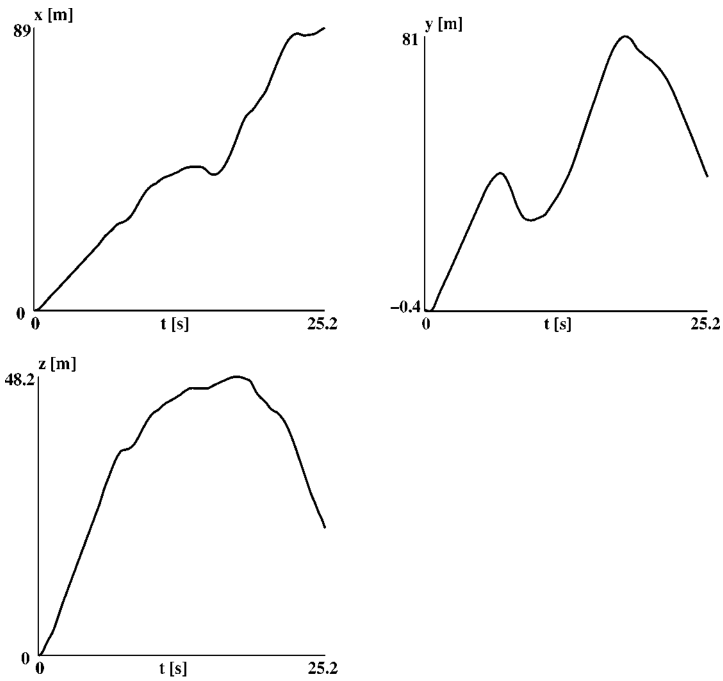

Figure 8. Coordinates of the object along the path.

The PG becomes hidden behind obstacles, velocity of the vehicle drops, and the trace mode is invoked. The vehicle selects a plane of motion and traces a boundary of obstacles at a reasonably safe distance, Figure 6. At the point marked B, the mode is changed to the decision one, as the first subgoal becomes visible. The vehicle directs to the subgoal. At its neighborhood, point C, the subgoal is changes to the third knot of the path, Table I. Finally, at point D, the vehicle carefully, with a low pace (the last decrease of speed in Figure 6), passes the gate formed by two obstacles. After reaching the obstacle-free area, the vehicle increases its speed to maximal admissible value, and therefore not immediately directs to the goal knot, see point E.

The effect of delay in reaching a prescribed path in obstacle-free regions can be reduced by the increase of reorienting abilities of the vehicle. Results of increasing $\alpha_{\max}$ to $130°/\mathrm{s}$, with other data kept the same as in the basic task, are presented in Figures 9 and 10. As can be seen in Figure 9, the resulting path is more compact than the path in Figure 5. Nevertheless, comparing the times of completing the motion in Figures 6, 10, one can notice that increasing the orientation abilities not necessarily decreases the total time.

The second experiment shows the need for a two level map. In the case of passive movable obstacles they are just filtered out from the physical map and they are not seen on the control map. If we do not perform filtering (only the physical map), the vehicle may react to the moving obstacles which do not interrupt its

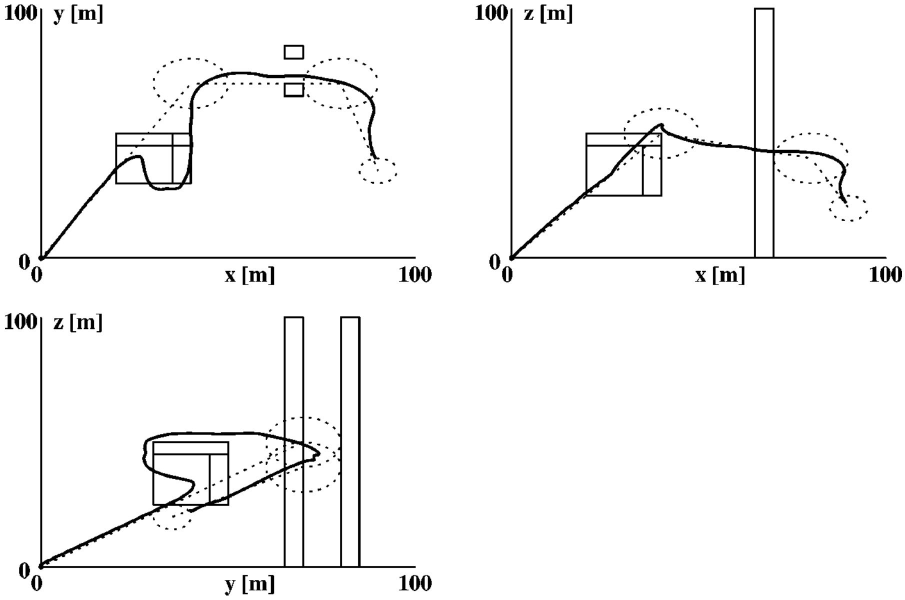

Figure 9. Motion in 3D space.

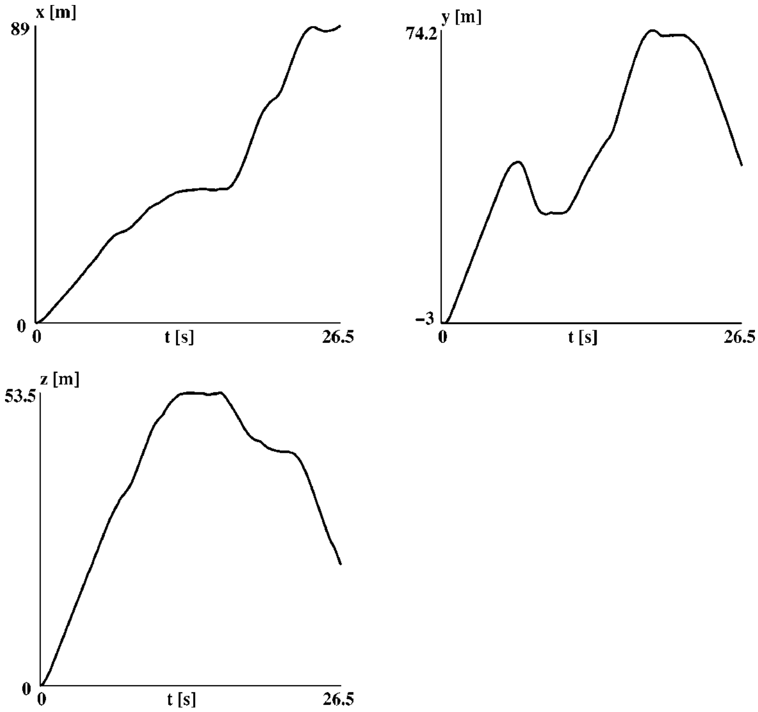

Figure 10. Coordinates of the object along the path.

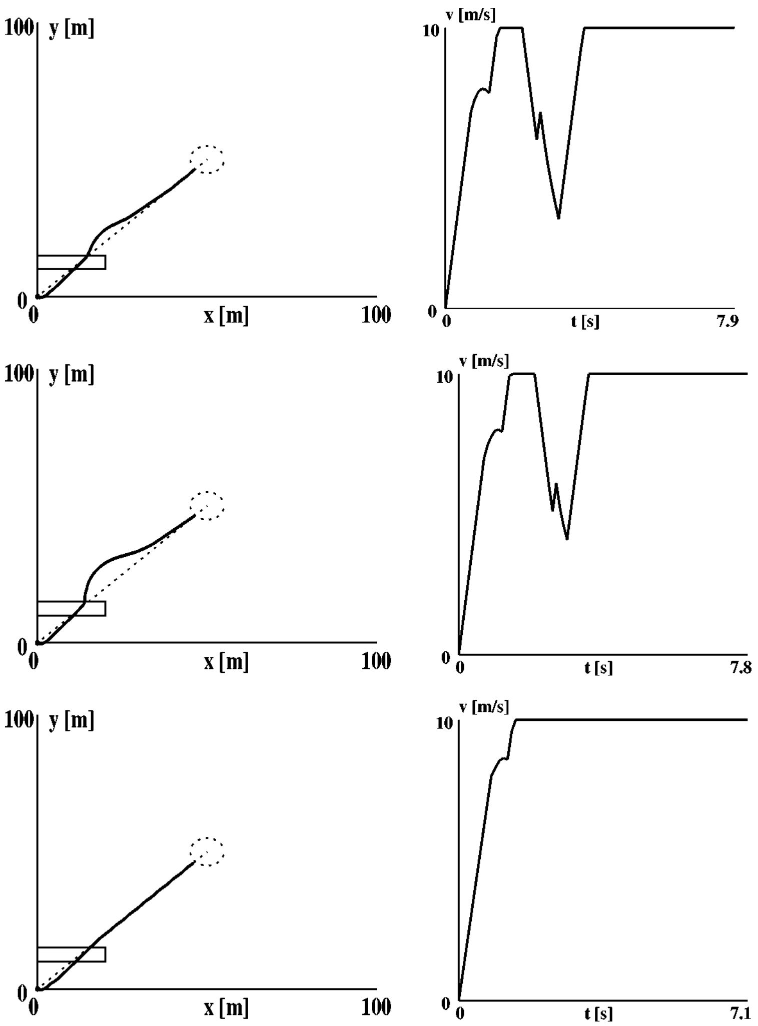

Figure 11. Path followed by vehicle in (x, y)-plane and vehicle's velocity profile for obstacle's velocity equal to 3, 5, 7 [m/s] (the 1st, 2nd and 3rd row, respectively).

motion. Assumptions for this experiment are as follows: there is only one obstacle $(x = 0,\ y = 10,\ z = 0)$, $(\Delta x = 20,\ \Delta y = 5,\ \Delta z = 20)$ moving along x-axis with different speed. The vehicle should reach the point $(50, 50, 0)$. Other data are the same as in the basic task. Results are presented in Figure 11. It could be seen from plots in Figure 11 (the left column) that the vehicle may not choose the best strategy of avoiding movable obstacles (the best strategy would be to pass the obstacle by moving to the left from the goal direction not to the right as in the plots), because local information can be sometimes too poor to work out a right decision. Moreover, when the obstacle moves relatively fast, its presence only slightly slows down the speed of the vehicle, cf. the third row in Figure 11, as the obstacle is not considered dangerous to the vehicle mission and is filtered out from the control map. When the moving obstacle collides with the path of the vehicle, the vehicle has to slow down significantly and reorient itself. The effect of reorientation is visible when the obstacle is passed by the vehicle. When the vehicle reduced its speed significantly, as for the obstacle velocity equal to 3 [m/s], it can return reasonably fast to the prescribed path (dashed line). When the vehicle speed is high after passing the obstacle, as for the obstacle velocity 5 [m/s], the vehicle needs some time to drive back to the prescribed path.

4. Conclusions

In this paper a local trajectory planner has been proposed, working in 3D environment populated with static and passive mobile obstacle. The task for this planner falls somewhere in between a local path planner and a global planner. The local path planner is able to produce fast collision-free trajectories with a weighted safety and goal approaching criterion. It can be adjusted in higher level system by changing its weights. Moreover, the planner takes kinematic and dynamic constraints into account. Those constraints are given in the form of maximal allowable velocity, acceleration and attitude changes. The planner has been presented as a complete algorithm of autonomous navigation ready to be used in real applications. The algorithm is general and flexible, and applicable to many classes of autonomous guided vehicles. The novelty of the proposed algorithm is in splitting map of the vehicle's world into two components: the physical and the control map. This allows to represent the world effectively for control applications and avoids a serious drawback related to planners based on potential fields. In the latter case, a vehicle could be trapped in local optimum of the criterion function. The second advantage of the system is that a control velocity loop of the vehicle has been proposed based on a negative feedback from the environment. The feedback stabilizes velocity around a value which is considered safe in a given environment. Computational requirements needed to apply the algorithm in real situations are not very heavy and therefore, it may be applied in real-time mode to the 3D case for free-flying robots as well as to 2D case for mobile robots. Computational load can be varied by changing the number of sectors considered at a given vehicle's location. In obstacle-free areas, this number can be reduced, while in navigation in obstacle-cluttered areas the number should be increased. Computations for each sector are virtually independent, so parallel data processing could also be applied.

References

Borenstein, J. and Koren, Y.: 1991a, The vector field histogram – fast obstacle avoidance for mobile robots, *IEEE Trans. Robotics Automat.* **7**(3), 278–288.

Borenstein, J. and Koren, Y.: 1991b, Histogramic in-motion mapping for mobile robot obstacle avoidance, *IEEE Trans. Robotics Automat.* **7**(4), 535–539.

Brooks, R.: 1986, A robust layered control system for a mobile robot, *IEEE J. Robotics Automat.* **2**(1), 14–23.

Burlina, P., DeMenthon, D., and Davis, L. S.: 1992, Navigation with uncertainty: Reaching a goal in a high collision risk region, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, Nice, pp. 2440–2445.

Clark, R. J., Arkin, R. C., and Ram, A.: 1992, Learning momentum: On-line performance enhancement for reactive systems, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, Nice, pp. 111–116.

Elnagar, A. and Base, A.: 1993, Heuristics for local path planning, *IEEE Systems Man Cybernet.* **23**(2), 624–634.

Hwang, Y. K. and Ahuja, N.: 1992, A potential field approach to path planning, *IEEE Trans. Robotics Automat.* **8**(1), 23–32.

Ilari, J. and Torras, C.: 1990, 2D path planning: A configuration space heuristic approach, *Internat. J. Robotics Res.* **9**(1), 75–91.

Kyriakopoulos, K. J. and Saridis, G. N.: 1992, An integrated collision prediction and avoidance scheme for mobile robots in non-stationary environments, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, Nice, pp. 194–199.

Ratering, S. and Gini, M.: 1993, Robot navigation in a known environment with unknown moving obstacles, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, Atlanta, pp. 25–30.

Sasiadek, J. and Dulęba, I.: 1995, Local trajectory planner, in: *Proc. AIAA Guidance, Navigation and Control Conf.*, Baltimore, pp. 1474–1483.

Sasiadek, J. and Dulęba, I.: 1997, 3D local trajectory planner, in: *Proc. AIAA Guidance, Navigation and Control Conf.*, New Orleans, pp. 517–528.

Slack, M. G.: 1993, Navigation templates: Mediating qualitative guidance and qualitative control in mobile robots, *IEEE Systems Man Cybernet.* **23**(2), 452–466.

Spong, M. and Vidyasagar, M.: 1989, *Introduction to Robotics*, Addison-Wesley, Reading, MA.

Takahashi, O. and Schilling, R. J.: 1989, Motion planning in a plane using generalized Voronoi diagrams, *IEEE Trans. Robotics Automat.* **5**(2), 143–150.

Zhu, Q.: 1991, Hidden Markov model for dynamic obstacles avoidance of mobile robot navigation, *IEEE Trans. Robotics Automat.* **7**(3), 390–397.