BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 70(2), 2022, Article number: e140691
DOI: 10.24425/bpasts.2022.140691

CONTROL AND INFORMATICS

The obstacle vector field (OVF) method for collision-free trajectory planning of free-floating space manipulator

Tomasz RYBUS *

Centrum Badań Kosmicznych Polskiej Akademii Nauk (CBK PAN), Warsaw, Poland

Abstract. Manipulators mounted on small satellites will be used to perform on-orbit servicing, removal of space debris, and assembly of large orbital structures. During such operations, the manipulator must avoid collisions with the target object or the elements of the assembled structure. Planning of the manipulator trajectory is one of the major challenges for the proposed missions because the motion of the manipulator influences the position and orientation of the satellite. Thus, the dynamic equations of motion must be used during trajectory planning. Methods developed for fixed-base manipulators working on Earth cannot be directly applied. In this paper, we propose a new obstacle vector field (OVF) method for collision-free trajectory planning of a manipulator mounted on a free-floating satellite. The OVF method is based on a vector field that surrounds the obstacles and generates virtual forces that drive the manipulator around the obstacles. The OVF method is compared with the classical artificial potential field (APF) method and the rapidly exploring random trees (RRT) method. In the presented examples the trajectory planning problem is solved for a planar case in which the satellite is equipped with a 2 DoF manipulator. It is shown that the OVF method is more efficient than the APF method, i.e., it allows us to solve the trajectory planning problem in some of the cases, in which the APF method is unsuccessful. The time required to find the solution with the use of the OVF method is shorter than the time needed by the APF and the RRT method.

Key words: space robotics; obstacle avoidance; obstacle vector field; free-floating system; on-orbit servicing.

1. INTRODUCTION

In the near future the manipulators mounted on small satellites will be applied for several purposes: to capture space debris [1], to perform servicing of commercial satellites [2], and to perform assembly of large orbital structures [3]. These operations could be done by astronauts but using unmanned space robots will significantly reduce costs and will pose no risk for humans [4]. The satellite-manipulator system will require a high level of autonomy to accomplish the aforementioned tasks [5].

In this paper, we focus on the problem of the manipulator trajectory planning during operations performed on-orbit. In many studies, it is suggested that these operations be conducted without the active control of the servicing satellite (e.g., [6, 7]). In such a case, the satellite is using its thrusters and momentum wheels during the approach phase, but when the manipulator begins its motion, the control of the satellite is switched-off [8]. Thus, during the maneuver, the servicing satellite is in the free-floating state, and the motion of the manipulator affects the position and orientation of the satellite [9]. The satellite equipped with the manipulator is a nonholonomic system [10] (various aspects related to trajectory planning and control of nonholonomic systems were investigated in [11] and [12]). Moreover, during orbital capture maneuver and assembly operations, the manipulator is working in proximity to other objects (elements of the captured satellite or elements of the assembled structure). Before the execution of the manipulator motion, a collision-free trajectory must be planned. Methods developed for Earth-based manipulators cannot be directly applied in the orbital scenario [13]. The trajectory planning algorithm must consider the fact that the state of the satellite is influenced by the motion of the manipulator. Thus, dynamic equations of the satellite-manipulator system must be used during the trajectory planning (this is the main difference between the manipulator operating on an orbit and fixed-base manipulators working on Earth).

Various techniques are proposed for planning collision-free trajectories. In an approach based on the rapidly exploring random trees (RRT) algorithm, the search for the collision-free trajectory is performed by a random search in the state-space (a trajectory tree is constructed from the initial state of the system). Application of the RRT algorithm for a manipulator that is attached to a free-floating satellite is presented in [14] and [15]. The approach proposed in [16] is based on the bi-directional RRT algorithm and allows us to find a collision-free trajectory that results in the planned change of the satellite orientation (this change is caused only by the motion of the manipulator). The collision-free trajectory planning problem for a space robot could be solved with methods that use the A* algorithm [17] and with optimization techniques [18–20]. Another possible approach is based on the artificial potential field (APF) method. In this method, an artificial force field is constructed, and the robotic manipulator is treated as a particle that moves in this field. There are numerous examples of the application of the APF method for collision-free trajectory planning of mobile

[*] e-mail: trybus@cbk.waw.pl

Manuscript submitted 2021-06-28, revised 2021-12-02, initially accepted for publication 2021-12-26, published in April 2022.

robots (e.g., [21, 22]) and fixed-based manipulators working on Earth (e.g., [23]). This method is also widely applied in other tasks, e.g., [24]. Application of the APF method for a manipulator that works in the orbital environment is described in [25]. Mukherjee and Nakamura proposed the application of potential fields based on the Lyapunov function [26]. In [27] the potential field is based on the Laplace function that facilitates overcoming the problem of local minima (the existence of local minima is one of the major drawbacks of the APF method). Several approaches are based on a combination of different techniques (e.g., [28, 29]). The approach presented in [30] allows preventing the collision of the manipulator with the tumbling target object after unsuccessful capture.

In this paper, we propose a new method for planning a collision-free trajectory of a robotic manipulator attached to a free-floating satellite. This approach is called the obstacle vector field (OVF) method. In contrast to the classical APF method, in which obstacles generate a scalar potential field, the OVF method is based on a vector field that surrounds obstacles. This vector field determines the direction, in which the link of the manipulator should move when it is close to an obstacle. The proposed method considers the fact that the motion of the manipulator influences the position and orientation of the satellite, on which the manipulator is mounted.

The OVF method, proposed in this paper for the manipulator mounted on the free-floating satellite, is to some extent similar to several concepts known from the literature [31–34]. A harmonic potential field (HPF) approach for planning the motion of an unmanned aerial vehicle (UAV) in an environment with a drift field is proposed in [31]. The drift field is understood as an external force acting on the UAV. The HPF approach processes the geometry of the environment (obstacles) and the given drift field to produce a vector field that guides the motion of the UAV (treated as a particle). The task of simultaneous planning and control is achieved by treating the control signal as an additional fictitious state of the system. Another application of the HPF approach can be found in [32], where a method for collision-free trajectory planning of a mobile robot is proposed. The vector field that guides the robot to the desired position is produced from the harmonic potential (the field is perturbed by obstacles present in the environment).

The main difference between the proposed OVF method and methods presented in [31] and [32] is in the approach used for the construction of the vector field. In the OVF method, the direction of the vector field that surrounds the given obstacle is based on the gradient of the potential field generated by this obstacle and is calculated using simple mathematical expressions. In the approach based on the HPF, the desired trajectory is obtained by solving a gradient dynamical system (this is a necessary step to construct the vector field). Thus, the planer is much more complicated than the proposed OVF method, as it contains components based on partial differential equations.

A path planning algorithm based on parametrized vector potential functions is proposed in [33]. In this approach, the complex workspace is split into triangular regions (there are three different types of regions: regions that contain obstacles, regions that contain no obstacles, and regions that contain the desired position). The vector potential functions are calculated using the information on region vertices, the obstacles, and the desired position of the object. The velocity vector field guides the object to the desired position and the desired orientation. The approach presented in [33] is demonstrated in simulations carried out for a unicycle vehicle. The use of predefined triangular regions differentiates this approach from the proposed OVF method.

Methods proposed in [31, 32] and [33] deal with the problem of planning the trajectory of a single object (UAV or mobile robot). Application of a method based on the vector potential field for planning a trajectory of a fixed-base manipulator is presented in [34]. As in the proposed OVF method, the guiding vector field in [34] is composed of two components: the first component is responsible for driving the end-effector towards the desired position, while the second is responsible for guiding the end-effector around obstacles. One part of the second component is radial to the obstacle surface, and one is tangential to this surface. The main difference between the method presented in [34] and the proposed OVF method is in the approach used for the construction of the field: the vector field in the OVF method is constructed from the FIRAS potential function, while in [34] the field is generated by solving a scalar boundary value problem. The OVF method gives more freedom in the selection of the direction of the field near obstacles (this direction is determined by weight coefficients). Moreover, when the end-effector is close to the obstacle and this obstacle lies between the end-effector position and the target position, the motion of the manipulator in the OVF method results only from the vector field of the obstacle (the attractive potential of the target is not considered). This allows us to find the solution in more difficult scenarios.

The differences between the proposed OVF method and methods known from the literature, outlined in the above discussion, are significant. Moreover, to the best of our knowledge, this paper presents the first application of the trajectory-planning method based on the vector field for the case of the manipulator mounted on the free-floating satellite.

The paper is organized as follows. The equations of the satellite-manipulator dynamics are presented in Section 2. The OVF method is described in Section 3. Section 4 contains the results of numerical simulations used to verify the proposed approach. The conclusions are presented in Section 5.

2. DYNAMICS OF THE SATELLITE-MANIPULATOR SYSTEM

Figure 1 shows a satellite with an attached n-DoF serial manipulator. The inertial coordinate system is denoted by CS_{ine} (all vectors will be given in CS_{ine} unless indicated otherwise), the coordinate system located at the mass center of the satellite and fixed to its body is denoted by CS_S, while the coordinate system located at the i-th joint and fixed to the i-th link is denoted by CS_i. The equations are given for a general three-dimensional case, but they can be directly applied for a simplified two-dimensional (planar) case. These equations are taken from [6] and [35].

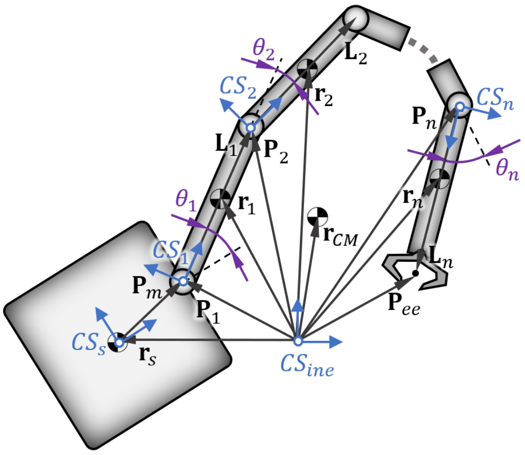

Fig. 1. A schematic view of a satellite with attached manipulator

The state of the satellite-manipulator system is defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_v \\ \mathbf{q}_p \end{bmatrix}. \tag{1}$$

The vector of generalized coordinates, $\mathbf{q}_p$, is defined as:

$$\mathbf{q}_p = \begin{bmatrix} \mathbf{r}_s \\ \boldsymbol{\Theta}_s \\ \boldsymbol{\theta} \end{bmatrix}, \tag{2}$$

where $\mathbf{r}_s$ is the position of the origin of CS_S (in CS_{ine}), $\boldsymbol{\Theta}_s$ is the orientation (attitude) of the satellite expressed using the Euler angles, $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_n \end{bmatrix}^T$, while θ_i is the angular position of the i-th joint. The vector $\mathbf{q}_v$ is defined as:

$$\mathbf{q}_v = \begin{bmatrix} \mathbf{v}_s \\ \boldsymbol{\omega}_s \\ \dot{\boldsymbol{\theta}} \end{bmatrix}, \tag{3}$$

where the linear and angular velocity of the satellite is denoted by $\mathbf{v}_s$ and $\boldsymbol{\omega}_s$, respectively, $\dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dots & \dot{\theta}_n \end{bmatrix}^T$, while $\dot{\theta}_i$ is the angular velocity of the i-th joint.

The end-effector position depends not only on the angular positions of the joints but also on the satellite position and orientation. On the kinematic level, this is the main difference between the fixed-base manipulator and manipulator attached to the satellite. The end-effector position in CS_{ine} is given by:

$$\mathbf{P}_{ee} = \mathbf{r}_s + \mathbf{P}_m + \sum_{i=1}^{n} \mathbf{L}_i\,, \tag{4}$$

where $\mathbf{P}_m$ is the position of the manipulator mounting point, and $\mathbf{L}_i$ is the position of the $i+1$-th kinematic pair with respect to the i-th kinematic pair. These vectors are expressed in CS_{ine}, so they depend on $\boldsymbol{\Theta}_s$.

The velocity of the end-effector is expressed as:

$$\begin{bmatrix} \mathbf{v}_{ee} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} = \mathbf{J}_s \begin{bmatrix} \mathbf{v}_s \\ \boldsymbol{\omega}_s \end{bmatrix} + \mathbf{J}_m \dot{\boldsymbol{\theta}}, \tag{5}$$

where $\mathbf{J}_s$ is the Jacobian matrix of the satellite:

$$\mathbf{J}_s = \begin{bmatrix} \mathbf{I} & \tilde{\mathbf{P}}_{ee\_s}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{6}$$

while $\mathbf{J}_m$ is the Jacobian matrix of the manipulator:

$$\mathbf{J}_m = \begin{bmatrix} \mathbf{a}_1 \times (\mathbf{P}_{ee} - \mathbf{P}_1) & \dots & \mathbf{a}_n \times (\mathbf{P}_{ee} - \mathbf{P}_n) \\ \mathbf{a}_1 & \dots & \mathbf{a}_n \end{bmatrix}. \tag{7}$$

The ~ symbol is used to indicate a matrix that is equivalent to a vector cross-product, $\mathbf{I}$ is the identity matrix, $\mathbf{a}_i$ and $\mathbf{P}_i$ denotes the unit vector of angular velocity and position of the i-th kinematic pair, respectively, while $\mathbf{P}_{ee\_s}$ is defined as:

$$\mathbf{P}_{ee\_s} = \mathbf{P}_m + \sum_{i=1}^{n} \mathbf{L}_i\,. \tag{8}$$

The linear momentum, $\mathbf{P}$, and the angular momentum, $\mathbf{L}$, of the satellite with attached manipulator can be expressed as:

$$\begin{bmatrix} \mathbf{P} \\ \mathbf{L} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T + \tilde{\mathbf{r}}_s \mathbf{A} & \mathbf{E} + \tilde{\mathbf{r}}_s \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{v}_s \\ \boldsymbol{\omega}_s \end{bmatrix} + \begin{bmatrix} \mathbf{D} \\ \mathbf{F} + \tilde{\mathbf{r}}_s \mathbf{D} \end{bmatrix} \dot{\boldsymbol{\theta}}. \tag{9}$$

The matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{D}$, $\mathbf{E}$ and $\mathbf{F}$ are defined as:

$$\mathbf{A} = \left(m_s + \sum_{i=1}^{n} m_i \right) \mathbf{I}, \tag{10}$$

$$\mathbf{B} = \sum_{i=1}^{n} m_i \tilde{\mathbf{r}}_{i\_s}\,, \tag{11}$$

$$\mathbf{D} = \sum_{i=1}^{n} m_i \mathbf{J}_{Ti}\,, \tag{12}$$

$$\mathbf{E} = \mathbf{I}_s + \sum_{i=1}^{n} \left(\mathbf{I}_i + m_i \tilde{\mathbf{r}}_{i\_s}^T \tilde{r}_{i\_s} \right), \tag{13}$$

$$\mathbf{F} = \sum_{i=1}^{n} \left(\mathbf{I}_i \mathbf{J}_{Ri} + m_i \tilde{\mathbf{r}}_{i\_s} \mathbf{J}_{Ti} \right), \tag{14}$$

where m_s is the mass of the satellite, while $\mathbf{I}_s$ is its inertia matrix, m_i is the mass of the i-th link, while $\mathbf{I}_i$ is its inertia matrix, $\mathbf{J}_{Ti}$ and $\mathbf{J}_{Ri}$ denotes the translational and the rotational component of the i-th link Jacobian matrix. Vector $\mathbf{r}_{i\_s}$ is the position of the i-th link center of mass with respect to the satellite center of mass: $\mathbf{r}_{i\_s} = \mathbf{r}_i - \mathbf{r}_s$, where $\mathbf{r}_i$ is the position of the i-th link center of mass (in CS_{ine}). The thrusters and reaction wheels mounted on the satellite are not used during operations performed with the manipulator. All external forces acting on the satellite and the manipulator are neglected. We also assume that: $\mathbf{v}_s(t=0) = \boldsymbol{\omega}_s(t=0) = \mathbf{0}$. In this case $\mathbf{P} = \mathbf{0}$ and $\mathbf{L} = \mathbf{0}$.

The mass center of the satellite-manipulator system remains stationary in CS_{ine}.

For the system with zero momentum and angular momentum equation (9) can be transformed into the following form:

$$\begin{bmatrix} \mathbf{v}_s \\ \boldsymbol{\omega}_s \end{bmatrix} = - \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T + \tilde{\mathbf{r}}_s \mathbf{A} & \mathbf{E} + \tilde{\mathbf{r}}_s \mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{D} \\ \mathbf{F} + \tilde{\mathbf{r}}_s \mathbf{D} \end{bmatrix} \dot{\boldsymbol{\theta}}. \tag{15}$$

We define the dynamic Jacobian matrix of the manipulator attached to the free-floating satellite as:

$$\mathbf{J}_D = \mathbf{J}_m - \mathbf{J}_s \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T + \tilde{\mathbf{r}}_s \mathbf{A} & \mathbf{E} + \tilde{\mathbf{r}}_s \mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{D} \\ \mathbf{F} + \tilde{\mathbf{r}}_s \mathbf{D} \end{bmatrix}. \tag{16}$$

The Jacobian $\mathbf{J}_D$ considers the fact that the position and orientation of the satellite are influenced by the manipulator motion. By substituting equation (15) into equation (5) we obtain the following relation between the end-effector velocity and angular velocities of manipulator joints:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_D^{\#} \begin{bmatrix} \mathbf{v}_{ee} \\ \boldsymbol{\omega}_{ee} \end{bmatrix}, \tag{17}$$

where the symbol # denotes the Moore–Penrose pseudoinverse of a matrix.

To obtain the dynamic equations of the satellite-manipulator system we follow the approach described in [35]. We neglect the potential energy because the satellite is in the state of free fall. The kinetic energy is given by:

$$T = \frac{1}{2} \mathbf{q}_v^T \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{D} \\ \mathbf{B}^T & \mathbf{E} & \mathbf{F} \\ \mathbf{D}^T & \mathbf{F}^T & \mathbf{N} \end{bmatrix} \mathbf{q}_v, \tag{18}$$

where:

$$\mathbf{N} = \sum_{i=1}^{n} \left(\mathbf{J}_{Ri}^T \mathbf{I}_i \mathbf{J}_{Ri} + m_i \mathbf{J}_{Ti}^T \mathbf{J}_{Ti} \right). \tag{19}$$

By substituting equation (18) into the Lagrange equation we derive the generalized equations of motion:

$$\mathbf{Q} = \mathbf{M}\left(\mathbf{q}_p\right) \dot{\mathbf{q}}_v + \mathbf{C}\left(\mathbf{q}_p, \mathbf{q}_v\right) \mathbf{q}_v, \tag{20}$$

where $\dot{\mathbf{q}}_v$ is the first derivative of $\mathbf{q}_v$ with respect to time, $\mathbf{M}$ is the generalized mass matrix, $\mathbf{C}$ is the Coriolis matrix, and $\mathbf{Q}$ is the vector of generalized forces. $\mathbf{M}$ is given by:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{D} \\ \mathbf{B}^T & \mathbf{E} & \mathbf{F} \\ \mathbf{D}^T & \mathbf{F}^T & \mathbf{N} \end{bmatrix}. \tag{21}$$

The vector $\mathbf{Q}$ is defined as:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{F}_s \\ \mathbf{H}_s \\ \mathbf{u} \end{bmatrix}, \tag{22}$$

where $\mathbf{F}_s$ is the vector of external forces, $\mathbf{H}_s$ is the vector of external torques ($\mathbf{F}_s$ and $\mathbf{H}_s$ act on the satellite), while $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & \ldots & u_n \end{bmatrix}^T$ is the vector of control torques that are applied at the joints. We assume that: $\mathbf{F}_s = \mathbf{H}_s = \mathbf{0}$.

Finally, we use a definition of the state vector (1) and equation (20) to obtain the following equation that describes the dynamic behavior of the free-floating satellite with the attached manipulator:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{M}^{-1}\left(\mathbf{Q} - \mathbf{C}\mathbf{q}_v\right) \\ \mathbf{q}_v \end{bmatrix}. \tag{23}$$

3. THE OBSTACLE VECTOR FIELD (OVF) METHOD

3.1. The problem of collision-free trajectory planning

In order to achieve the task of grasping the target object or to perform the assembly work the manipulator must move its end-effector to the selected position, $\mathbf{P}_T$. The manipulator operates in proximity to the target object or the assembled structure and must avoid collisions with obstacles. The task of planning manipulator trajectory can be treated separately from the task of trajectory following (as explained in [6], each of these tasks can be realized by a separate module of the control system). The planning is performed before the execution of the manipulator motion. At the trajectory planning stage, perfect knowledge of the satellite-manipulator system parameters is assumed. Moreover, no uncertainties in the satellite or the end-effector position and orientation are considered. It is the role of the closed-loop controller to ensure the realization of the selected trajectory despite disturbances and uncertainties. Various methods can be used for such a purpose (e.g., predictive control [6] or adaptive control [36, 37]). The problem of tracking of a numerically defined trajectory by the manipulator mounted on the satellite is analyzed in [38], while an approach based on input-output decoupling for a free-floating satellite with manipulator under state and input disturbances is shown in [39]. The review of various methods that can be used for control of the satellite-manipulator system can be found in [40].

In this section, we present the proposed trajectory planning algorithm for n-DoF manipulator mounted on the free-floating satellite. For the sake of simplicity we are considering a two-dimensional (planar) case, in which there are m rectangular obstacles in the manipulator workspace: $\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \ldots, \boldsymbol{\Gamma}_m$. Each obstacle is defined by the position of its geometrical center, $\mathbf{P}_{\Gamma_i}$, its orientation with respect to CS_{ine}, α_i, and its dimensions: width, w_i, and height, h_i. More complex shapes of obstacles can be approximated by many rectangles. The goal of the trajectory planning algorithm is to find a collision-free trajectory of the manipulator from the given initial state of the satellite-manipulator system to a state in which the manipulator end-effector is in the desired position. This task can be expressed as:

For $\mathbf{q}_v(t=0) = \mathbf{0}$ and for the given $\mathbf{q}_p(t=0)$ and $\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \ldots, \boldsymbol{\Gamma}_m$ find $\boldsymbol{\theta}(t) = \begin{bmatrix} \theta_1(t) & \theta_2(t) & \ldots & \theta_n(t) \end{bmatrix}^T$ such that:

1. $\mathbf{P}_{ee}(t = t_f) = \mathbf{P}_T$.
2. $\forall t \in (0, t_f)\, \forall\, i \in \{1, 2, \ldots, n\}\, \forall\, j \in \{1, 2, \ldots, m\}$: $\overline{\mathbf{P}_i \mathbf{P}_{i+1}} \cap \boldsymbol{\Gamma}_j = \emptyset$, where $\mathbf{P}_{n+1} \equiv \mathbf{P}_{ee}$.

The line segment between the Cartesian position (given in CS_{ine}) of the i-th kinematic pair and $i+1$-th kinematic pair is denoted by $\overline{\mathbf{P}_i\mathbf{P}_{i+1}}$. The collision between this line segment and j-th obstacle occurs if and only if there is a non-empty set of points that belong to the line segment $\overline{\mathbf{P}_i\mathbf{P}_{i+1}}$ and to the obstacle $\mathbf{\Gamma}_j$. To simplify the collision detection we do not consider the width of the manipulator links (the links are treated as line segments). This assumption is acceptable if the width of the links is much smaller than their length. The width of the links can be considered during the trajectory planning by introducing a safe zone around the obstacle (in any practical application such a zone must be introduced anyway to avoid collisions in the case of disturbances and non-perfect knowledge of parameters of the satellite-manipulator system). One possible approach to introduce a safe zone is to perform the trajectory planning with virtual obstacles that are located in the same position as real obstacles, but have their physical dimensions increased by a constant $z_{\mathbf{\Gamma}}$.

In the classical APF method trajectory, planning is based on the scalar function (called the potential) [41]. The potential field can be constructed in the physical (Euclidean) space or the configuration space of the manipulator. This field has two components: (i) the attractive potential of the target (a function that has its minimum in the desired final configuration of the manipulator), and (ii) the repulsive potential of obstacles (a function that has high values in all configurations in which the links of the manipulator collide with obstacles). The virtual forces generated by the potential field act on the manipulator and drive its motion towards the configuration, in which the end-effector is in the desired position. The general idea of the APF method applied for the free-floating satellite manipulator system is presented in Fig. 2, while the general idea of the proposed OVF method is presented in Fig. 3.

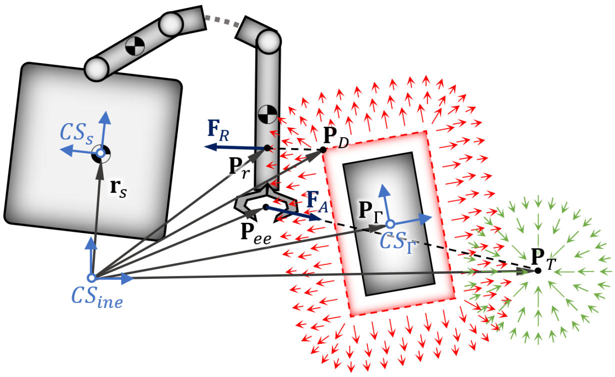

Fig. 2. A schematic view showing the general idea of one of the variants of the classical APF method. The attractive potential field is generated by the target position, while the repulsive potential field (based on the scalar FIRAS potential field function) is generated by the obstacle

The OVF method is a variation of the well-known approach based on the APF method. The main difference is that instead of a simple scalar potential function the OVF method uses a vector field. The vector field determines the direction of motion that allows us to move the links of the manipulator around the obstacles. The idea to use vector field in trajectory planning is not new [31–34]. However, to the best of our knowledge, such an approach was never used for the manipulator mounted on the free-floating satellite.

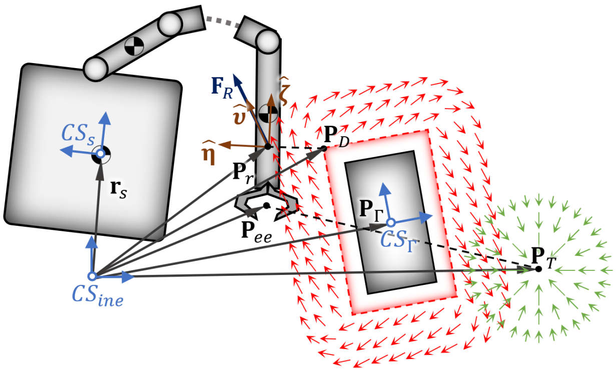

Fig. 3. A schematic view showing the general idea of the OVF method. The obstacle $\mathbf{\Gamma}$ lies between the end-effector and the target position. This obstacle generates the vector field that drives the manipulator around the obstacle

3.2. The attractive force generated by the target

The virtual attractive force, $\mathbf{F}_A$, is generated by the target position, $\mathbf{P}_T$, and acts on the end-effector. This force is directed from the current end-effector position, $\mathbf{P}_{ee}$, towards the target position and drives the motion of the end-effector. As in the classical APF method [42], the amplitude of $\mathbf{F}_A$ depends on the distance between $\mathbf{P}_{ee}$ and $\mathbf{P}_T$. However, in the proposed OVF method, when the manipulator is in proximity to the obstacle and this obstacle lies between the end-effector position and the target position, there is no need for the attractive force to act on the end-effector because the motion of the end-effector will be driven by the vector field of the obstacle. The virtual attractive force is given by:

$$\mathbf{F}_A = \begin{cases} g_{A1}\left(e^{-g_{A2}\|\mathbf{P}_T-\mathbf{P}_{ee}\|}+g_{A3}\right)\dfrac{\mathbf{P}_T-\mathbf{P}_{ee}}{\|\mathbf{P}_T-\mathbf{P}_{ee}\|}, \\ \qquad \text{if } \forall i \in \{1,2,\ldots,m\}: d_i > d_a \vee \overline{\mathbf{P}_{ee}\mathbf{P}_T} \cap \mathbf{\Gamma}_i = \emptyset \\ \mathbf{0}, \ \text{if } \exists i \in \{1,2,\ldots,m\}: d_i \le d_a \wedge \overline{\mathbf{P}_{ee}\mathbf{P}_T} \cap \mathbf{\Gamma}_i \neq \emptyset, \end{cases} \tag{24}$$

where g_{A1}, g_{A2}, and g_{A3} denote constant coefficients, while $d_i = \|(\mathbf{P}_r)_i - (\mathbf{P}_D)_i\|$ denotes the closest distance between the manipulator and the i-th obstacle, $(\mathbf{P}_r)_i$ denotes the point on the manipulator that is closest to the i-th obstacle, while $(\mathbf{P}_D)_i$ denotes the point on the i-the obstacle that is closest to the manipulator (positions of points $(\mathbf{P}_r)_i$ and $(\mathbf{P}_D)_i$ are found using a simple algorithm that calculates distances between line segments defined by the positions of manipulator kinematic pairs and sides of rectangular obstacles). A constant parameter that describes the minimal value of d_i is denoted by d_a. The $\mathbf{F}_A$ equals zero if for any obstacle $\mathbf{\Gamma}_i$ the distance d_i is smaller than some defined value d_a and if this obstacle intersects with the line segment $\overline{\mathbf{P}_{ee}\mathbf{P}_T}$ (such intersection occurs when there is a non-empty set of points that belong to the line segment $\overline{\mathbf{P}_{ee}\mathbf{P}_T}$

and to the obstacle $\boldsymbol{\Gamma}_i$). The coefficients g_{A1}, g_{A2}, g_{A3}, and the distance d_a are used to tune the OVF algorithm.

3.3. The repulsive force generated by obstacles

In the OVF method, the virtual repulsive force of obstacles results from the vector field generated by these obstacles. The vector field is constructed from the potential field that surrounds the obstacle. To describe the potential field we choose the FIRAS potential function [43]:

$$U\left(\mathbf{P}_U^{(\boldsymbol{\Gamma})}\right) = \begin{cases} \dfrac{g_P}{2}\left(\dfrac{1}{\left\|\mathbf{P}_U^{(\boldsymbol{\Gamma})} - \mathbf{P}_D^{(\boldsymbol{\Gamma})}\right\|} - \dfrac{1}{d_0}\right)^2, \\ \qquad \text{if } 0 < \left\|\mathbf{P}_U^{(\boldsymbol{\Gamma})} - \mathbf{P}_D^{(\boldsymbol{\Gamma})}\right\| < d_0 \\ 0, \qquad \text{if } \left\|\mathbf{P}_U^{(\boldsymbol{\Gamma})} - \mathbf{P}_D^{(\boldsymbol{\Gamma})}\right\| \geq d_0\,, \end{cases} \tag{25}$$

where $\mathbf{P}_U^{(\boldsymbol{\Gamma})}$ denotes a point (given in the local coordinate frame fixed to the obstacle center, CS_Γ), for which the potential is calculated, g_p denotes a constant coefficient, $\mathbf{P}_D^{(\boldsymbol{\Gamma})}$ denotes a point on the obstacle that is closest to $\mathbf{P}_U^{(\boldsymbol{\Gamma})}$, while d_0 denotes a constant parameter that describes the effective range of the obstacle potential field. Although the FIRAS potential function was introduced in the 1980s, it is still used in collision-free trajectory planning algorithms (e.g., [44]). In the classical APF method, the potential field constructed with this function can result in local minima. Therefore, various other approaches were proposed to describe the repulsive potential of obstacles (e.g., superquadrics [45] or harmonic potential functions [46]). However, in the OVF method, the potential function is only needed for the construction of the obstacle vector field. Thus, we can use the simple FIRAS potential function.

Each obstacle is treated separately. The value of the potential generated by the i-th obstacle is calculated for the point on the manipulator that is closest to this obstacle. Thus, $\mathbf{P}_U^{(\boldsymbol{\Gamma}_i)} \equiv \left(\mathbf{P}_r^{(\boldsymbol{\Gamma}_i)}\right)_i$. The direction from the i-th obstacle is calculated as a gradient of the scalar potential field:

$$\boldsymbol{\eta}_i = \nabla U_i\,, \tag{26}$$

where U_i denotes the value of the potential field from equation (25). The direction perpendicular to the gradient of the potential field is calculated with the following equation:

$$\boldsymbol{\zeta}_i = \mathbf{T}_\gamma\left(\gamma_i \frac{\pi}{2}\right)\hat{\boldsymbol{\eta}}_i\,, \tag{27}$$

where the symbol ^ denotes the unit vector $\left(\hat{\boldsymbol{\eta}}_i = \dfrac{\boldsymbol{\eta}_i}{\|\boldsymbol{\eta}_i\|}\right)$, $\mathbf{T}_\gamma$ denotes the rotation matrix:

$$\mathbf{T}_\gamma(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \tag{28}$$

while $\gamma_i \in \{-1, 1\}$ denotes the parameter that defines the direction of the vector $\boldsymbol{\zeta}_i$. For the given initial conditions and the given set of obstacles, the selection of γ_i depends on $\mathbf{P}_T$. To solve the trajectory planning problem with the OVF method, it is required to check various combinations of γ_i. For m obstacles there are 2^m possible combinations, e.g., for $m = 2$ there are four sets of γ_i coefficients: (i) $\gamma_1 = 1$, $\gamma_2 = 1$, (ii) $\gamma_1 = 1$, $\gamma_2 = -1$, (iii) $\gamma_1 = -1$, $\gamma_2 = -1$, and (iv) $\gamma_1 = -1$, $\gamma_2 = 1$. Values of γ_i coefficients in the z-th set can be computed from the following expression:

$$\gamma_i = (-1)^{\left\lfloor \frac{z+2^{i-1}-1}{2^{i-1}} \right\rfloor}, \tag{29}$$

where $\lfloor\cdot\rfloor$ denotes the floor function.

The resultant direction of the repulsive force is defined by the unit vector $\hat{\boldsymbol{\upsilon}}_i = \dfrac{\boldsymbol{\upsilon}_i}{\|\boldsymbol{\upsilon}_i\|}$, where $\boldsymbol{\upsilon}_i$ is given by:

$$\boldsymbol{\upsilon}_i = \begin{cases} \hat{\boldsymbol{\eta}}_i + g_h\hat{\boldsymbol{\zeta}}_i, \qquad \text{if } \overline{\mathbf{P}_{ee}\mathbf{P}_T} \cap \boldsymbol{\Gamma}_i = \emptyset \\ \dfrac{-\tan^{-1}\left(g_w(d_i - d_\upsilon)\right)}{0.5\pi}\hat{\boldsymbol{\eta}}_i + g_h\hat{\boldsymbol{\zeta}}_i, \\ \qquad \text{if } \overline{\mathbf{P}_{ee}\mathbf{P}_T} \cap \boldsymbol{\Gamma}_i \neq \emptyset, \end{cases} \tag{30}$$

where g_w and g_h denote constant coefficients, while d_v denotes a constant parameter that describes the distance from the obstacle at which $\boldsymbol{\upsilon}_i$ changes its direction.

If the i-th obstacle does not lie between the current end-effector position and the target position (the set of points that belong to the line segment $\overline{\mathbf{P}_{ee}\mathbf{P}_T}$ and to the obstacle $\boldsymbol{\Gamma}_i$ is empty), then the $\boldsymbol{\upsilon}_i$ vector is calculated as a weighted sum of unit vectors $\hat{\boldsymbol{\eta}}_i$ and $\hat{\boldsymbol{\zeta}}_i$ (the former is defined by the gradient of the potential field, while the latter is parallel to the lines of the potential field). The weight coefficient for $\hat{\boldsymbol{\eta}}_i$ is equal to 1, while the weight coefficient for $\hat{\boldsymbol{\zeta}}_i$ is equal to a constant parameter g_h. For low values of g_h, the direction of $\boldsymbol{\upsilon}_i$ is close to the direction defined by the gradient of the potential field and the repulsive force is acting away from the obstacle. Higher values of g_h will result in a direction that will be moving the manipulator around the obstacle. If the end-effector is close to the obstacle and this obstacle lies between the end-effector position and the target position, then the motion of the manipulator should only result from the vector field of the obstacle. In such a case, the manipulator should be guided around the obstacle at some given distance. The first term of equation (30) under the condition that $\overline{\mathbf{P}_{ee}\mathbf{P}_T} \cap \boldsymbol{\Gamma}_i \neq \emptyset$ ensures that the manipulator will keep a constant distance from the obstacle (the arctangent function is used to obtain a smooth change of the direction from repulsive to attractive).

The virtual repulsive force generated by the i-th obstacle is given by the following expression:

$$(\mathbf{F}_R)_i = g_{R1}\left(-e^{-g_{R2}\|\mathbf{P}_T - \mathbf{P}_{ee}\|} + g_{R3}\right)U_i\hat{\boldsymbol{\upsilon}}_i\,, \tag{31}$$

where g_{R1}, g_{R2}, and g_{R3} denote constant coefficients. Potential U_i is calculated for the point $(\mathbf{P}_r)_i$, in which the force $(\mathbf{F}_R)_i$ acts on the manipulator (point $(\mathbf{P}_r)_i$ lies on one of the links of the manipulator). $\|(\mathbf{F}_R)_i\|$ depends on the distance between the target position and the current end-effector position (if $\|\mathbf{P}_T - \mathbf{P}_{ee}\|$

decreases, then the value of $(\mathbf{F}_R)_i$ also decreases). Such an approach allows us to plan a trajectory even if $\mathbf{P}_T$ lies very close to the edge of the obstacle. Without the dependence of $\|(\mathbf{F}_R)_i\|$ on $\|\mathbf{P}_T - \mathbf{P}_{ee}\|$ the attractive force $\mathbf{F}_A$ could be at some point balanced by $(\mathbf{F}_R)_i$. The values of constant coefficients g_{R1}, g_{R2}, and g_{R3} should be selected in such a way that the influence of $\|\mathbf{P}_T - \mathbf{P}_{ee}\|$ on the value of $(\mathbf{F}_R)_i$ is significant only when $\|\mathbf{P}_T - \mathbf{P}_{ee}\|$ is very small.

3.4. Trajectory planning with the OVF method

The manipulator should move under the influence of the virtual attractive force generated by the target position and the virtual repulsive force generated by m obstacles. The following equation is used to determine the desired motion of manipulator joints:

$$\dot{\boldsymbol{\theta}}_{\text{des}} = \mathbf{J}_D^{\#} g_v \mathbf{F}_A + \sum_{i=1}^{m} \mathbf{J}_{D_i}^{\#} g_v \mathbf{T}_\gamma(\varphi_i)(\mathbf{F}_R)_i \,, \tag{32}$$

where g_v denotes a constant coefficient, while $\mathbf{J}_{D_i}$ denotes the dynamic Jacobian computed not for the end-effector, but the point $(\mathbf{P}_r)_i$. Equation (16) is used to calculate $\mathbf{J}_{D_i}$, but only these joints are considered that are between the manipulator mounting point and the link, on which the point $(\mathbf{P}_r)_i$ is located. If the point $(\mathbf{P}_r)_i$ lies on the j-th link, then $\mathbf{J}_{D_i}$ is computed for j-DoF manipulator, in which the position of the end of the last link with respect to the last kinematic pair is given by: $\mathbf{L}_j = (\mathbf{P}_r)_i - \mathbf{P}_j$. Matrix $\mathbf{T}_\gamma$ in equation (32) is used to transform $(\mathbf{F}_R)_i$ from CS_{ine} to the local coordinate frame fixed at the beginning of the link, on which the point $(\mathbf{P}_r)_i$ is located. Orientation of this link is denoted by φ_i.

The control torques that result in the desired motion of the manipulator joints are calculated from the following equation:

$$\mathbf{u} = g_u \left(\dot{\boldsymbol{\theta}}_{\text{des}} - \dot{\boldsymbol{\theta}}\right), \tag{33}$$

where g_u denotes a constant coefficient, while the vector of control torques $\mathbf{u}$ is the only non-zero component of the vector of generalized forces $\mathbf{Q}$ defined in equation (22). The influence of the manipulator motion on the state of the servicing satellite must be considered during the trajectory planning. The 4th order Runge-Kutta (RK IV) method is used to solve equation (23) in order to determine the state of the satellite-manipulator system after the application of the control torque given by equation (33). The period between the previous state and the new state is denoted by t_{RK} (it is assumed that the control torque is constant during this period). Steps of the OVF method are repeated for the new state obtained from the RK IV method. This process is continued until the solution is found (the end-effector reaches the desired target position: $|\mathbf{P}_{ee} - \mathbf{P}_T| < \varepsilon$, where ε denotes the allowable error) or the maximal number of repetitions is reached (this number is denoted by k_{step}). When the solution is found, the algorithm obtains the reference manipulator trajectory in the joint space from all the states that lie on the trajectory. At this stage the time of motion, t_f, can be arbitrarily selected and the trajectory can be adjusted. The pseudocode of the OVF trajectory planning method is presented as Algorithm 1.

Algorithm 1. Pseudocode of the OVF trajectory planning method

```
Input:  Parameters of the system, the initial state ($\mathbf{x}(t=0)$),
        obstacles ($\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \ldots, \boldsymbol{\Gamma}_m$), desired target position ($\mathbf{P}_T$)
Output: The trajectory of the manipulator defined in the joint
        space: $\boldsymbol{\theta}(t)$

$\mathbf{x}_1 \leftarrow$ Assign the initial state as the first state on the trajectory;
for $z \leftarrow 1$ to $2^m$ do
    $\gamma_1(z), \gamma_2(z), \ldots, \gamma_m(z) \leftarrow$ Calculate coefficients using equation (29);
    for $j \leftarrow 1$ to $k_{step}$ do
        $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{ee} \leftarrow$ Solve the forward kinematics for $\mathbf{x}_j$ to obtain positions of joints and position of the end-effector;
        if $|\mathbf{P}_{ee} - \mathbf{P}_T| < \varepsilon$ then
            $\boldsymbol{\theta}(t) \leftarrow$ Obtain the reference trajectory in the joint space from $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_j$;
            Solution found, break the loop;
        end if
        $\mathbf{F}_A \leftarrow$ Calculate the virtual attractive force using equation (24);
        for $i \leftarrow 1$ to $m$ do
            $(\mathbf{P}_r)_i, (\mathbf{P}_D)_i \leftarrow$ Find point on the manipulator that is closest to the $i$-th obstacle, and point on the $i$-the obstacle that is closest to the manipulator;
            $U_i \leftarrow$ Calculate the value of the FIRAS function in $(\mathbf{P}_r)_i$ using equation (25);
            $\hat{\boldsymbol{\eta}}_i$, $\hat{\boldsymbol{\zeta}}_i$, $\hat{\boldsymbol{\upsilon}}_i \leftarrow$ Build vector field in the point $(\mathbf{P}_r)_i$;
            $(\mathbf{F}_R)_i \leftarrow$ Calculate the virtual repulsive force using equation (31);
        end for
        $\dot{\boldsymbol{\theta}}_{\text{des}} \leftarrow$ Calculate the desired motion of manipulator joints using equation (32);
        $\mathbf{u} \leftarrow$ Calculate control torques using equation (33);
        $\mathbf{x}_{j+1} \leftarrow$ Use RK IV method to solve equation (23) in order to determine the state of the satellite-manipulator system after the application of $\mathbf{u}$;
    end for
end for
```

4. RESULTS OF NUMERICAL SIMULATIONS

4.1. Parameters of the satellite-manipulator system

Simulations presented in this section were performed for a planar satellite equipped with a non-redundant 2 DoF manipulator. Parameters of the manipulator are shown in Table 1. Parameters of the satellite are presented in Table 2.

4.2. Comparison with other methods

The proposed method is compared with two other trajectory planning methods: the APF method and the RRT method. We use a variant of the APF method presented in [25]. In this approach, the potential field is based on superquadrics that describe the obstacles. The direction of the repulsive force generated by the i-th obstacle is not based on the local gradient

Table 1
Parameters of the 2 DoF manipulator

Parameter	Symbol	Value	
		Link 1	Link 2
Position of CS_{i+1} with respect to CS_i	$\mathbf{L}_i^{(CS_i)}$	$[0.6\text{ m}\ \ 0]^T$	$[0.6\text{ m}\ \ 0]^T$
Position of the center of mass	$\mathbf{r}_i^{(CS_i)}$	$[0.3\text{ m}\ \ 0]^T$	$[0.3\text{ m}\ \ 0]^T$
Mass	m_i	4.5 kg	1.5 kg
Mass moment of inertia	$\mathbf{I}_i$	0.135 kg·m^2	0.045 kg·m^2
Range of allowed positions of joint	θ_i	$(-150°, +150°)$	$(-170°, +170°)$

Table 2
Parameters of the satellite

Parameter	Symbol	Value
Mass	m_s	60 kg
Mass moment of inertia	$\mathbf{I}_s$	1.875 kg·m^2
Manipulator mounting point	$\mathbf{P}_m^{(CS_S)}$	$[0.4\text{ m }0]^T$

of the potential field but is set halfway between the direction determined by the points $(\mathbf{P}_G)_i$ and $(\mathbf{P}_D)_i$, and the direction determined by the points $(\mathbf{P}_r)_i$ and $(\mathbf{P}_D)_i$ (point $(\mathbf{P}_G)_i$ lies in the geometric center of the i-th link). As demonstrated in [25], such an approach to the selection of the direction of the repulsive force in the APF method reduces the risk of stopping the algorithm after reaching the local minimum of the field.

In our simulations, we use a one-directional variant of the RRT algorithm described in [14] (with some modifications introduced in [16]). In the approach presented in [14], the RRT algorithm is used in a different way than it is usually used for planning trajectories of mobile robots and fixed-base manipulators (usually the construction of the trajectory tree continues until the desired configuration is reached, e.g., [47]). We follow the approach, in which a single tree is constructed until a predefined number of vertices is reached (this tree begins from the initial state of the system and is constructed in the configuration space). The number of tree vertices is set to 100 000. The position of the end-effector is calculated for each vertex of the tree (vertices are defined in the configuration space). The vertex that corresponds to the end-effector position that is sufficiently close to the target position is selected. The trajectory in the configuration space is constructed backwards from this vertex to the initial vertex of the trajectory tree.

To the best of our knowledge in the case of the free-floating satellite-manipulator system, no known approach allows us to determine if for the given conditions the solution of the trajectory planning problem exists. Thus, to compare the OVF method with other methods we select the following criterion. We arbitrarily chose part of the manipulator workspace and denote this part as A_T. The trajectory planning is performed for a set of points uniformly distributed in A_T. This set is denoted as $\Pi_T = \{(\mathbf{P}_T)_1, (\mathbf{P}_T)_2, \ldots, (\mathbf{P}_T)_k\}$, where $(\mathbf{P}_T)_i \in A_T$, while k is the number of points. Some of the points that belong to Π_T may lie inside obstacles. Such points belong to the set $\Pi_\mathbf{\Gamma} \subseteq \Pi_T$. The set of points for which the tested method can find the solution to the trajectory planning problem is denoted as $\Pi_S \subseteq \Pi_T$. The effectiveness of the trajectory planning method is measured by δ:

$$\delta = \frac{|\Pi_S|}{|\Pi_T| - |\Pi_\mathbf{\Gamma}|}, \tag{34}$$

where $|\Pi|$ denotes the number of elements in the set Π. The solution to the trajectory planning problem may not exist for some points that belong to Π_T. Thus, for the given conditions the maximal possible effectiveness may be lower than 1.

The OVF, the APF, and the RRT method were implemented in Matlab (R2014a). Simulations were performed on a PC with Intel Core i7-3820 CPU (3.60 GHz) with 32 GB RAM.

4.3. Results of numerical simulations

In the considered example there are three rectangular obstacles. The center of the first obstacle is in $\mathbf{P}_{\mathbf{\Gamma}_1} = [0.9\text{ m}\ \ -0.215\text{ m}]^\text{T}$. Edges of this obstacle are parallel to the axes of CS_{ine} and have the following lengths: $w_1 = 0.06$ m and $h_1 = 0.21$ m. The center of the second obstacle is in $\mathbf{P}_{\mathbf{\Gamma}_2} = [1.105\text{ m }0]^T$. This obstacle is oriented at an angle of $\alpha_2 = 10°$ with respect to the axes of CS_{ine} and its edges have the following lengths: $w_2 = h_2 = 0.13$ m. The center of the third obstacle is in $\mathbf{P}_{\mathbf{\Gamma}_3} = [1.3\text{ m}\ \ 0.3\text{ m}]^T$. This obstacle is oriented at an angle of $\alpha_3 = -45°$ and its edges have the following lengths: $w_3 = 0.21$ m and $h_3 = 0.11$ m. The width of the manipulator links is considered during the trajectory planning by introducing a safe zone around each obstacle. To create this safe zone the dimensions of each obstacle are increased by $z_\mathbf{\Gamma} = 0.04$ m. The initial state is defined by $\mathbf{q}_p(t{=}0)$ $= \left[(r_s)_x\ (r_s)_y\ \psi_s\ \theta_1\ \theta_2\right]^T = \left[0\ 0\ 0\ 1.14\text{ rad}\ \ -2.28\text{ rad}\right]^T$ and $\mathbf{q}_v(t=0) = \mathbf{0}$, where $(r_s)_x$ and $(r_s)_y$ denotes the X and Y components of $\mathbf{r}_s$, respectively, while ψ_s is the orientation of the satellite around the Z-axis. For the given $\mathbf{q}_p(t=0)$ the initial position of the end-effector is: $\mathbf{P}_{ee}(t=0) = [0.7\text{ m}\ \ -0.1\text{ m}]^T$. The goal of the trajectory planning algorithm is to find the collision-free trajectory from this initial state to a state in which the end-effector is in the desired target position: $\mathbf{P}_{ee}(t=t_f) = \mathbf{P}_T$.

The parameters used by the OVF method were selected by a trial-and-error method. Selected values allow successful trajectory planning in a wide range of conditions: $g_{A1} = 10$, $g_{A2} = 70$, $g_{A3} = 1$, $d_a = 0.1$ m, $g_p = 0.005$, $d_0 = 10$ m, $g_w = 10^4$, $g_h = 2$, $d_v = 0.02$ m, $g_{R1} = 0.5$, $g_{R2} = 50$, $g_{R3} = 1$, $g_v = 1$, $g_u = 50$, $t_{RK} = 0.001$ s, $k_{step} = 2000$, $\varepsilon = 0.002$ m, $t_f = 20$ s. The following desired target position was chosen for the initial simulations: $\mathbf{P}_T = [1\text{ m}\ \ -0.25\text{ m}]^T$. The OVF and the RRT method were able to solve the trajectory planning problem. The APF method was not able to find the trajectory that would bring the system to a state, in which the end-effector is in the desired position.

The visualization of the potential field generated by obstacles using the FIRAS function is shown in Fig. 4. The solution of the trajectory planning problem was found with the OVF method for the following set of γ coefficients: $\gamma_1 = -1$, $\gamma_2 = 1$, and

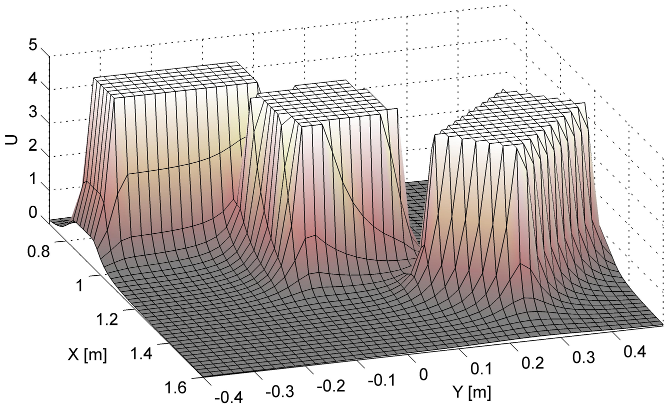

Fig. 4. Visualization of the artificial potential field generated by obstacles using the FIRAS potential field function (values above 5 are truncated)

$\gamma_3 = 1$. The vector field generated by obstacles for this set is presented in Fig. 5. The end-effector trajectory obtained with the OVF, the APF, and the RRT method is shown on the XY plane in Fig. 6. The X and Y components of this trajectory are presented in Figs. 7 and 8, respectively. The end-effector trajectory obtained with the OVF method and the RRT method leads to the desired target position, while the trajectory obtained with the APF method ends at the safe zone that surrounds the obstacle $\mathbf{\Gamma}_1$. Figures 9 and 10 show the trajectory in the joint space. The satellite-manipulator system during realization of the trajectory planned with the OVF method is depicted in Fig. 11, which shows the influence of the manipulator motion on the satellite. The mass center of the system remains stationary, while the mass center of the satellite is moving during realization of the trajectory.

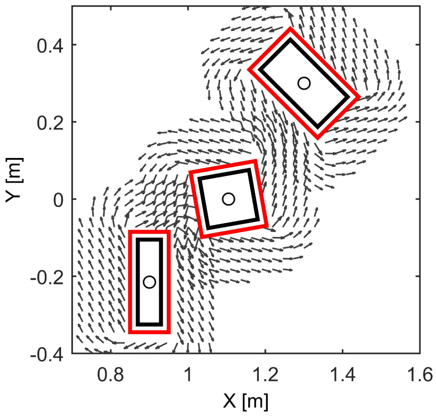

Fig. 5. The vector field generated by obstacles for: $\gamma_1 = -1$, $\gamma_2 = 1$, and $\gamma_3 = 1$. Each obstacle is surrounded by a safe zone

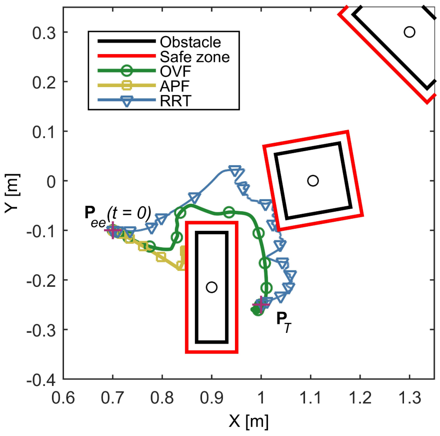

Fig. 6. The end-effector trajectory on the XY plane

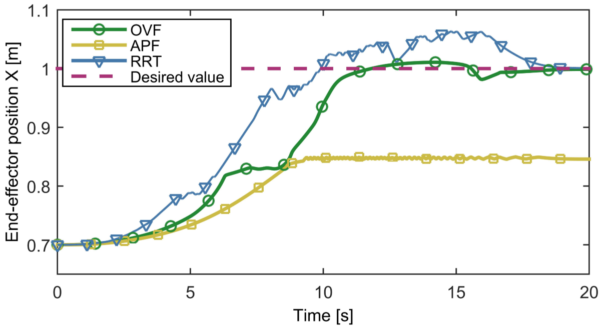

Fig. 7. The X-component of the end-effector position

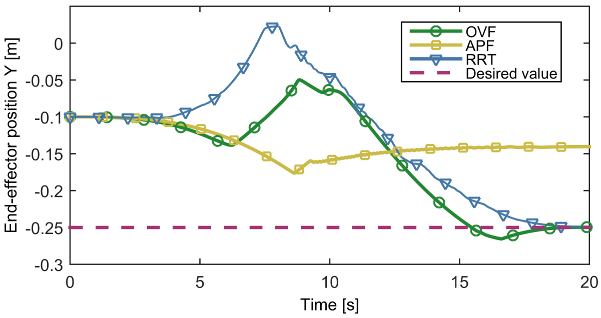

Fig. 8. The Y-component of the end-effector position

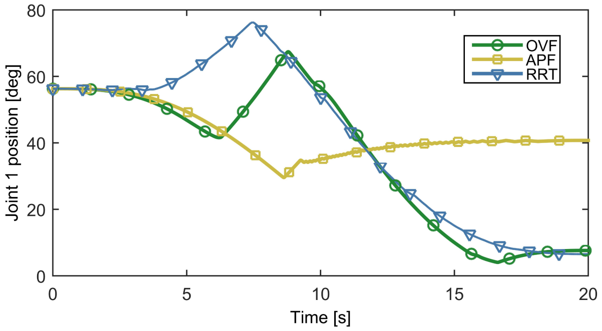

Fig. 9. The angular position of the first joint of the manipulator

The OVF method and the RRT method were able to solve the trajectory planning problem for the specific desired target position $\mathbf{P}_T = [1\ \mathrm{m}\ \ -0.25\ \mathrm{m}]^T$. During the trajectory planning performed with the APF method, the manipulator has reached a position in which the control torques that arise from the virtual attractive force generated by the target are compensated by torques that arise from the virtual repulsive force generated by the obstacle $\mathbf{\Gamma}_1$. As a result, the APF method is not

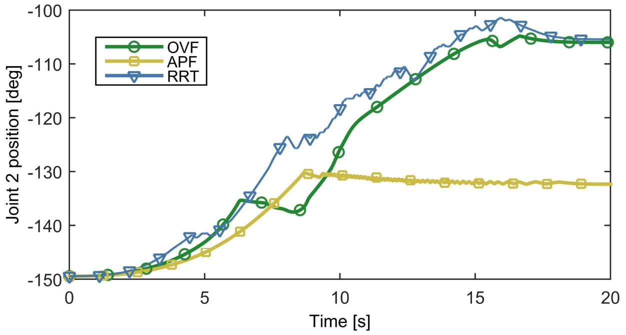

Fig. 10. The angular position of the second joint of the manipulator

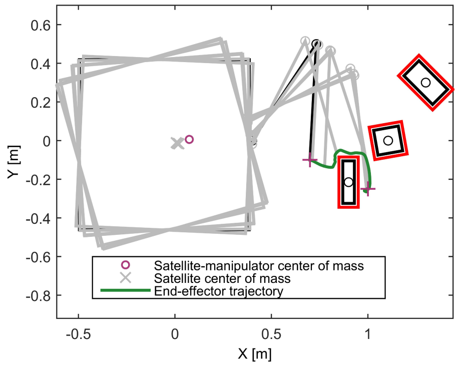

Fig. 11. The satellite-manipulator system during the realization of the trajectory planned with the OVF method

able to solve the trajectory planning problem. The final end-effector position obtained with the APF method is: $\mathbf{P}_{ee}(t = t_f) = [0.846 \text{ m} \ -0.141 \text{ m}]^T$. A comparison between the solutions obtained with the OVF method and the RRT method shows that the trajectory obtained with the OVF method has higher smoothness. The final end-effector position obtained after the realization of the trajectory is the same for the OVF and the RRT method (the difference between the obtained and the desired position is in both cases below 0.0016 m. However, the final positions of joints are different: $\theta_1(t = t_f) = 0.133$ rad and $\theta_2(t = t_f) = -1.85$ rad for the trajectory obtained with the OVF method, $\theta_1(t = t_f) = 0.114$ rad and $\theta_2(t = t_f) = -1.84$ rad for the trajectory obtained with the RRT method. The final orientation of the satellite is also different: $\psi_s(t = t_f) = 0.268$ rad for the OVF method, $\psi_s(t = t_f) = 0.281$ rad for the RRT method. These differences arise from the fact that the satellite equipped with the manipulator is a nonholonomic system and its final configuration depends on the trajectory. The change of the orientation and position of the satellite is caused by the reaction forces and reaction torques induced by the motion of the manipulator. Due to the free-floating nature of the considered system the equations that describe the dynamics of this system (equation (23)) are used during the trajectory planning.

4.4. Effectiveness of the trajectory planning methods

In order to compare the OVF, APF, and RRT methods using the criterion defined in equation (34), trajectory planning is performed for a set of points $\Pi_T = \{(\mathbf{P}_T)_1, (\mathbf{P}_T)_2, \ldots, (\mathbf{P}_T)_k\}$. We select $k = 4096$ points that are uniformly distributed on a grid and cover rectangular area defined by vertices: $(\mathbf{P}_V)_1 = [0.7 \text{ m} \ -0.4 \text{ m}]^T$, $(\mathbf{P}_V)_2 = [1.6 \text{ m} \ -0.4 \text{ m}]^T$, $(\mathbf{P}_V)_3 = [1.6 \text{ m} \ 0.5 \text{ m}]^T$, and $(\mathbf{P}_V)_4 = [0.7 \text{ m} \ 0.5 \text{ m}]^T$. In the case of the OVF and APF methods, the trajectory planning algorithm is used to find the solution to the trajectory planning problem for every point in the set Π_T (each point is passed to the algorithm as a target position). In the case of the RRT method, one trajectory tree is constructed for the given scenario. It is checked which points that belong to Π_T are sufficiently close to the end-effector positions obtained for tree vertices.

The OVF, APF, and RRT methods are compared in three different scenarios. The first scenario, considered in Section 4.3, is named 'case 1'. We introduce two other cases. In the second case, there are two rectangular obstacles in the manipulator workspace. The center of the first obstacle is in $\mathbf{P}_{\Gamma_1} = [0.95 \text{ m} \ 0.15 \text{ m}]^T$. This obstacle is oriented at an angle of $\alpha_1 = -45°$ with respect to the axes of CS_{ine} and its edges have the following lengths: $w_1 = 0.16$ m and $h_1 = 0.06$ m. The center of the second obstacle is in $\mathbf{P}_{\Gamma_2} = [0.95 \text{ m} \ -0.15 \text{ m}]^T$. This obstacle is oriented at an angle of $\alpha_2 = 45°$ with respect to the axes of CS_{ine} and its edges have the following lengths: $w_2 = 0.16$ m and $h_2 = 0.06$ m. As in 'case 1', to create a safe zone around obstacles the dimensions of each obstacle are increased by $z_\Gamma = 0.04$ m. The initial state of the satellite-manipulator system is defined by $\mathbf{q}_p(t = 0) = [0 \ 0 \ 0 \ 0.982 \text{ rad} \ -2.608 \text{ rad}]^T$ and $\mathbf{q}_v(t = 0) = \mathbf{0}$. For the given $\mathbf{q}_p(t = 0)$ the initial position of the end-effector is: $\mathbf{P}_{ee}(t = 0) = [0.9 \text{ m} \ 0]^T$. Although in this case, the number of obstacles is smaller than in 'case 1', their arrangement with respect to the initial configuration of the manipulator makes the trajectory planning problem very difficult. It is not possible to use the free space between the two obstacles to access the area of the workspace that is located to the right of the obstacles. The end-effector must move around the obstacles in order to access this area.

In the third case, there are six rectangular obstacles. The centers of the obstacles are located in the following positions: $\mathbf{P}_{\Gamma_1} = [0.8 \text{ m} \ -0.215 \text{ m}]^T$, $\mathbf{P}_{\Gamma_2} = [1 \text{ m} \ -0.215 \text{ m}]^T$, $\mathbf{P}_{\Gamma_3} = [1.2 \text{ m} \ -0.215 \text{ m}]^T$, $\mathbf{P}_{\Gamma_4} = [1.35 \text{ m} \ 0.4 \text{ m}]^T$, $\mathbf{P}_{\Gamma_5} = [1.35 \text{ m} \ 0.2 \text{ m}]^T$, and $\mathbf{P}_{\Gamma_6} = [1.35 \text{ m} \ 0]^T$. These obstacles are oriented at the following angles: $\alpha_1 = -30°$, $\alpha_2 = 30°$, $\alpha_3 = -30°$, $\alpha_4 = \alpha_5 = \alpha_6 = 90°$. All obstacles have the same size: $w_1 = w_2 = w_3 = w_4 = w_5 = w_6 = 0.06$ m and $h_1 = h_2 = h_3 = h_4 = h_5 = h_6 = 0.21$ m. As in 'case 1', to create a safe zone around obstacles the dimensions of each obstacle are increased by $z_\Gamma = 0.04$ m. We select exactly the same initial state as in 'case 1': $\mathbf{q}_p(t = 0) = [0 \ 0 \ 0 \ 1.14 \text{ rad} \ -2.28 \text{ rad}]^T$ and $\mathbf{q}_v(t = 0) = \mathbf{0}$. The initial position of the end-effector is: $\mathbf{P}_{ee}(t = 0) = [0.7 \text{ m} \ -0.1 \text{ m}]^T$. Although in 'case 3' there are twice as many obstacles as in 'case 1', the accessible area of the workspace hidden behind obstacles is small. Thus, from the perspective of the collision-free trajectory planning algorithm, this last case is an easy one. The main difficulty, especially for the OVF method, arises from the number of obstacles, not from their configuration with respect to the initial configuration of the manipulator.

The area covered by a set of points, for which the trajectory planning method can find the solution of the trajectory planning problem, is presented in Fig. 12 for each of the considered methods. Table 3 presents the summary of the results obtained for all three cases. The RRT method has the highest effectiveness, while the APF method has the lowest effectiveness. In 'case 1' the effectiveness of the proposed OVF method is higher than the effectiveness of the APF method by 18.1%, while the effectiveness of the RRT method is higher than the effectiveness of the OVF method by 8.2%. In 'case 2' the effectiveness of the proposed OVF method is higher than the effectiveness of the APF method by 38.5%, while the effectiveness of the RRT method is higher than the effectiveness of the OVF method by 8.1%. In these two cases, there exists a subset of desired positions, for which the OVF and RRT methods can find the solution to the trajectory planning problem, while the APF method is not able to find the solution. In 'case 3' all methods have similar effectiveness. The effectiveness of the proposed OVF method is lower than the effectiveness of the APF method by 0.9%, while the effectiveness of the RRT method is higher than the effectiveness of the OVF method by 3.3%. This is the only case, in which the advantage of the OVF method over the APF method is not visible. This is because the trajectory planning problem is less challenging for the configuration of the obstacles defined in 'case 3' than in 'case 1' and 'case 2'. Thus, the higher effectiveness of the OVF method over the APF method becomes apparent in difficult scenarios. In many cases, the APF method is not able to find the solution because the algorithm is stopped in local minima. This problem is limited in the OVF method because in the proposed approach the poten-

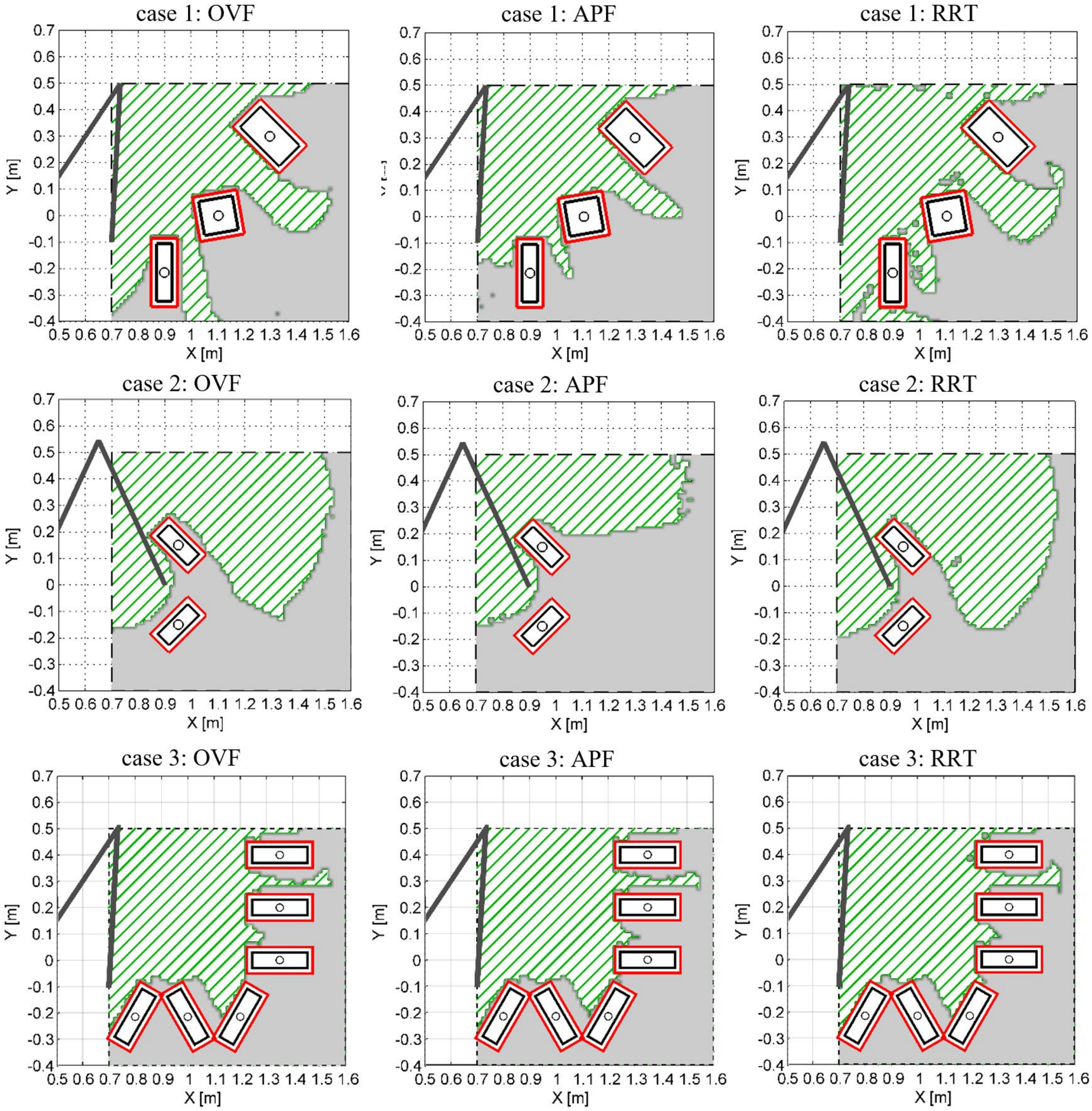

Fig. 12. Map showing the final end-effector positions that can be reached with the OVF method (left column), the APF method (middle column) and the RRT method (right column) in 'case 1' (the first row), in 'case 2' (the second row), and in 'case 3' (the third row)

tial function is only used for construction of the vector field. As a result, there are no points where the algorithm would stop due to the equilibrium of attractive and repulsive potential (the vector field drives the manipulator around the obstacles). However, there are some configurations of obstacles for which the resulting vector field does not allow the algorithm to find a solution, even though this solution exists (and is found with the RRT method). The effectiveness of the RRT method is highest in all three cases, but the differences between the effectiveness of the OVF and RRT methods are relatively small. If the solution to a given problem exists, it is not guaranteed that the OVF method will be able to find this solution. This is a disadvantage of the proposed method, but to the best of our knowledge none of the methods proposed for the collision-free trajectory planning of manipulator mounted on the free-floating satellite (including the RRT method) ensures that the solution of the trajectory planning problem will be found if it exists [13].

Table 3
Effectiveness of the trajectory planning methods

Planning method	Case 1	Case 2	Case 3
Obstacle vector field (OVF)	0.5403	0.5260	0.5283
Artificial potential field (APF)	0.4575	0.3797	0.5331
Rapidly exploring random trees (RRT)	0.5846	0.5686	0.5459

The computational time required to find the solution with the proposed OVF trajectory planning method depends on the number of obstacles. As explained in Section 3.3, each obstacle generates its own vector field, and the direction of this field is determined by γ_i. For m obstacles, there are 2^m sets of possible combinations of γ_i coefficients. However, the OVF algorithm works sequentially, trying different combinations until the solution is found or until all combinations are checked. If there are no obstacles between the initial end-effector position and the target position, then the trajectory planning problem can be solved for any combination of γ_i coefficients. Detailed analysis of the results obtained for all three cases shows that for many target positions from the set Π_T the solution with the OVF method is found for the first combination of γ_i coefficients.

The average computational time required to find the solution of the trajectory planning problem using the OVF, the APF, and the RRT method is presented in Table 4 (cases in which the trajectory planning was not successful are not considered in this calculation). Comparison between the OVF and APF methods shows that in every case the average computational time required to find the solution of the trajectory planning problem is lower for the OVF method. In the OVF method, the increase in the number of obstacles does not significantly affect the required time, but it should be noted that the scenario considered in ‘case 3’ is less demanding for the trajectory planning algorithm. The increase in the number of obstacles also slowed down the APF method. The time required by the RRT method is three orders of magnitude higher than the time required by the OVF method and APF methods (the construction of the trajectory tree for the free-floating satellite-manipulator system is time-consuming). However, in a given scenario one tree could be used to solve the trajectory planning problem for all points in the set Π_T. In the classical approach based on the RRT algorithm, a solution could be found very quickly for some desired positions (without the need to construct the tree that has 100 000 vertices). The values presented in Table 4 should be treated with caution because the Matlab implementations of the considered algorithms were not optimized.

Table 4
Average computational time required to solve the trajectory planning problem

Planning method	Case 1	Case 2	Case 3
Obstacle vector field (OVF)	16.31s	14.22s	18.57 s
Artificial potential field (APF)	32.47s	26.66s	53.71 s
Rapidly exploring random trees (RRT)*	~43000 s	~41000 s	~44700 s

* Values presented for the RRT method include the time required for the construction of the trajectory tree (100 000 vertices). One tree can be used for all points in the set Π_T.

5. CONCLUSIONS

In this paper, we presented a new approach to the collision-free trajectory planning of the manipulator mounted on the free-floating satellite. The proposed OVF method is based on a vector field that surrounds the obstacles and generates virtual forces that drive the manipulator around the obstacles. This method considers the fact that the motion of the manipulator influences the position and orientation of the satellite (the equations that describe the dynamics of the satellite-manipulator system are used during the trajectory planning). The OVF method was compared with two other methods: the APF and RRT. The comparison was performed for the planar satellite equipped with the 2 DoF manipulator. Three different scenarios were analyzed. It was shown that the OVF method is more efficient than the APF method in two cases (in ‘case 1’ the effectiveness of the OVF method is higher by 18.1%, while in ‘case 2’ the effectiveness of the OVF method is higher by 38.5%). In ‘case 3’ the effectiveness of the OVF method is very similar to the effectiveness of the APF method (the efficiency of the OVF method is lower by 0.9%). The OVF and APF methods are less efficient than the RRT method. The average computational time required to solve the trajectory planning problem with the OVF method is lower than the time required by the APF and RRT methods. The practical applicability of the RRT method is limited due to the time-consuming process of trajectory tree construction. Moreover, the RRT method is based on a random search and every run of the trajectory planning algorithm will result in a different solution obtained for the same scenario. The OVF method is fully deterministic and will always produce the same result. Thus, the proposed OVF method may be an interesting alternative for the RRT method.

In the proposed approach perfect knowledge of the satellite-manipulator parameters is assumed in the trajectory planning stage and no uncertainties are considered. A closed-loop controller is required to ensure the realization of the trajectory planned with the OVF method. The OVF method could be applied for collision-free trajectory planning in the proposed on-orbit servicing missions and during the assembly of orbital structures.

ACKNOWLEDGEMENTS

This paper was partially supported by the Polish National Centre for Research and Development project no. LIDER/19/0117/L-10/18/NCBR/2019 and by the Polish National Science Center project no. 2015/17/B/ST7/03995.

REFERENCES

[1] S. Estable *et al.*, "Capturing and deorbiting Envisat with an Airbus Spacetug. Results from the ESA e. Deorbit consolidation phase study," *J. Space Saf. Eng.*, vol. 7, no. 1, pp. 52–66, 2020, doi: 10.1016/j.jsse.2020.01.003.

[2] J.S. Hudson and D. Kolosa, "Versatile On-Orbit Servicing Mission Design in Geosynchronous Earth Orbit," *J. Spacecr. Rockets*, vol. 57, no. 4, pp. 844–850, 2020, doi: 10.2514/1.A34701.

[3] Z. Xue, J. Liu, C. Wu, and Y. Tong, "Review of in-space assembly technologies," *Chin. J. Aeronaut.*, vol. 34, no. 11, pp. 21–47, 2021, doi: 10.1016/j.cja.2020.09.043.

[4] S. Mohan and D.W. Miller, "Dynamic control model calculation: a model generation architecture for autonomous on-orbit assembly," *J. Spacecr. Rockets.*, vol. 51, no. 5, pp. 1430–1453, 2014, doi: 10.2514/1.A32581.

[5] I. Rekleitis, E. Martin, G. Rouleau, R. L'Archevêque, K. Parsa, and E. Dupuis, "Autonomous capture of a tumbling satellite," *J. Field Robot.*, vol. 24, no. 4, pp. 275–296, 2007, doi: 10.1002/rob.20194.

[6] T. Rybus, K. Seweryn, and J.Z. Sąsiadek, "Control system for free-floating space manipulator based on Nonlinear Model Predictive Control (NMPC)," *J. Intell. Robot. Syst.*, vol. 85, no. 3, pp. 491–509, 2017, doi: 10.1007/s10846-016-0396-2.

[7] P. Rank, Q. Mühlbauer, W. Naumann, and K. Landzettel, "The DEOS automation and robotics payload," in *Proc. 11th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*, ESTEC, Noordwijk, The Netherlands, 2011.

[8] J. Virgili-Llop, J.V. Drew, R. Zappulla II, and M. Romano, "Laboratory experiments of resident space object capture by a spacecraft–manipulator system," *Aerosp. Sci. Technol.*, vol. 71, pp. 530–545, 2017, doi: 10.1016/j.ast.2017.09.043.

[9] E. Papadopoulos and S. Dubowsky, "On the nature of control algorithms for free-floating space manipulators," *IEEE Trans. Robot. Autom.*, vol. 7, no. 6, pp. 750–758, 1991, doi: 10.1109/70.105384.

[10] J. Ratajczak and K. Tchoń, "Normal forms and singularities of non-holonomic robotic systems: A study of free-floating space robots," *Syst. Control. Lett.*, vol. 138, 104661, 2020, doi: 10.1016/j.sysconle.2020.104661.

[11] A. Ratajczak, "Trajectory reproduction and trajectory tracking problem for the nonholonomic systems," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 64, no. 1, pp. 63–70, 2016, doi: 10.1515/bpasts-2016-0008.

[12] I. Duleba, "Impact of control representations on efficiency of local nonholonomic motion planning," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 59, no. 2, pp. 213–218, 2011, doi: 10.2478/v10175-011-0026-x.

[13] T. Rybus, "Obstacle avoidance in space robotics: review of major challenges and proposed solutions," *Prog. Aerosp. Sci.*, vol. 101, pp. 31–48, 2018, doi: 10.1016/j.paerosci.2018.07.001.

[14] T. Rybus and K. Seweryn, "Application of rapidly-exploring random trees (RRT) algorithm for trajectory planning of free-floating space manipulator," in *Proc. 10th International Workshop on Robot Motion and Control (RoMoCo)*, Poznań, Poland, 2015, doi: 10.1109/RoMoCo.2015.7219719.

[15] J.R. Benevides and V. Grassi, "Autonomous path planning of free-floating manipulators using RRT-based algorithms," in *Proc. 12th Latin American Robotics Symposium and 3rd Brazilian Symposium on Robotics (LARS-SBR)*, Uberlandia, Minas Gerais, Brazil, 2015, doi: 10.1109/LARS-SBR.2015.47.

[16] T. Rybus, "Point-to-point motion planning of a free-floating space manipulator using the Rapidly-Exploring Random Trees (RRT) method," *Robotica*, vol. 38, no. 6, pp. 957–982, 2020, doi: 10.1017/S0263574719001176.

[17] X. Gao, Q. Jia, H. Sun, and G. Chen, "Research on path planning for 7-DOF space manipulator to avoid obstacle based on A* algorithm," *Sens. Lett.*, vol. 9, no. 4, pp. 1515–1519, 2011, doi: 10.1166/sl.2011.1665.

[18] C. Toglia, M. Sabatini, P. Gasbarri, and G.B. Palmerini, "Optimal target grasping of a flexible space manipulator for a class of objectives," *Acta Astronaut.*, vol. 68, no. 7-8, pp. 1031–1041, 2011, doi: 10.1016/j.actaastro.2010.09.013.

[19] G. Misra and X. Bai, "Optimal path planning for free-flying space manipulators via sequential convex programming," *J. Guid. Control Dyn.*, vol. 40, no. 11, pp. 3019–3026, 2017, doi: 10.2514/1.G002487.

[20] T. Rybus, M. Wojtunik, and F.L. Basmadji, "Optimal collision-free path planning of a free-floating space robot using spline-based trajectories," *Acta Astronaut.*, vol. 190, pp. 395–408, 2022, doi: 10.1016/j.actaastro.2021.10.012.

[21] Z. Hendzel, "Collision free path planning and control of wheeled mobile robot using Kohonen self-organising map," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 53, no. 1, pp. 39–47, 2005.

[22] W. Kowalczyk, M. Michałek, and K. Kozłowski, "Trajectory tracking control with obstacle avoidance capability for unicycle-like mobile robot," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 60, no. 3, pp. 537–546, 2012, doi: 10.2478/v10175-012-0066-x.

[23] C.C. Lin and J.H. Chuang, "Potential-based path planning for robot manipulators in 3-D workspace", in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003, doi: 10.1109/ROBOT.2003.1242108.

[24] A. Wojciechowski, "Camera navigation support in a virtual environment," *Bull. Polish Acad. Sci. Tech.*, vol. 61, no. 4, pp. 871–884, 2013, doi: 10.2478/bpasts-2013-0094.

[25] T. Rybus and K. Seweryn, "Zastosowanie metody sztucznych pól potencjału do planowania trajektorii manipulatora satelitarnego (Application of the artificial potential field method for trajectory planning of space manipulator)," in *Postępy Robotyki, Prace Naukowe Politechniki Warszawskiej: Elektronika*, vol. 196, K. Tchoń and C. Zieliński, Eds. Oficyna Wydawnicza Politechniki Warszawskiej, 2018, pp. 61–74 [in Polish].

[26] R. Mukherjee and Y. Nakamura, "Nonholonomic redundancy of space robots and its utilization via hierarchical liapunov functions," in *Proc. American Control Conference (ACC)*, Boston, USA, 1991, doi: 10.23919/ACC.1991.4791630.

[27] Y. Yanoshita and S. Tsuda, "Space Robot Path Planning for Collision Avoidance," in *Proc. International MultiConference of Engineers and Computer Scientists (IMECS)*, Hong Kong, 2009, vol. 2.

[28] J.B. Balaram and H. W. Stone, "Automated Assembly in the JPL Telerobot Testbed," in *Intelligent Robotic Systems for Space Exploration*, A.A. Desrochers, Ed. Springer, New York, 1992, pp. 297–342, doi: 10.1007/978-1-4615-3634-5_8.

[29] R.K. Mathur, R. Münger, and A.C. Sanderson, "Hierarchical Planning for Space-Truss Assembly," in *Intelligent Robotic Systems for Space Exploration*, A.A. Desrochers, Ed. Springer, New York, 1992, pp. 141–184, doi: 10.1007/978-1-4615-3634-5_4.

[30] A. Ratajczak and J. Ratajczak, "Trajectory Reproduction Algorithm in Application to an On-Orbit Docking Maneuver with Tumbling Target," in *Proc. 12th International Workshop on Robot Motion and Control (RoMoCo)*, Poznań, Poland, 2019, pp. 172–177, doi: 10.1109/RoMoCo.2019.8787367.

[31] A.A. Masoud, "A harmonic potential field approach for planning motion of a UAV in a cluttered environment with a drift field," in *Proc. 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, USA, 2011, doi: 10.1109/CDC.2011.6160192.

[32] A.A. Masoud and A. Al-Shaikhi, "Time-sensitive, sensor-based, joint planning and control of mobile robots in cluttered spaces: A harmonic potential approach," in *Proc. 54th IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, 2015, doi: 10.1109/CDC.2015.7402634.

[33] A.K. Pamosoaji and K.S. Hong, "A path-planning algorithm using vector potential functions in triangular regions," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 43, no. 4, pp. 832–842, 2013, doi: 10.1109/TSMCA.2012.2221457.

[34] A.A. Masoud and M.M. Bayoumi, "Robot navigation using the vector potential approach," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, 1993, doi: 10.1109/ROBOT.1993.292076.

[35] T. Rybus *et al.*, "Application of a planar air-bearing microgravity simulator for demonstration of operations required for an orbital capture with a manipulator," *Acta Astronaut.*, vol. 155, pp. 211–229, 2019, doi: 10.1016/j.actaastro.2018.12.004.

[36] S. Ulrich, J.Z. Sąsiadek, and I. Barkana, "Nonlinear adaptive output feedback control of flexible-joint space manipulators with joint stiffness uncertainties," *J. Guid. Control Dyn.*, vol. 37, no. 6, pp. 1961–1975, 2014, doi: 10.2514/1.G000197.

[37] M.S. Sakha, H. Kharrati, and F. Mehdifar, "Adaptive control of free-floating manipulator in presence of stochastic input disturbances and unknown dynamical parameters," *J. Vib. Control*, 2021, doi: 10.1177/10775463211010525.

[38] W. Domski and A. Mazur, "Tracking of numerically defined trajectory by free-floating 3D satellite," in *Proc. 12th International Workshop on Robot Motion and Control (RoMoCo)*, Poznań, Poland, 2019, pp. 178–183, doi: 10.1109/RoMoCo.2019.8787342.

[39] W. Domski and A. Mazur, "Input-output decoupling for a 3D free-floating satellite with a 3R manipulator with state and input disturbances," *Bull. Polish Acad. Sci. Tech.*, vol. 67, no. 6, pp. 1031–1039, 2019, doi: 10.24425/bpasts.2019.130885.

[40] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Prog. Aerosp. Sci.*, vol. 68, pp. 1–26, 2014, doi: 10.1016/j.paerosci.014.03.002.

[41] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," in *Autonomous Robot Vehicles*, I.J. Cox and G.T. Wilfong, Eds. Springer, New York, 1986, pp. 396–404, doi: 10.1007/978-1-4613-8997-2_29.

[42] R. Volpe and P. Khosla, "Artificial potentials with elliptical isopotential contours for obstacle avoidance," in *Proc. 26th IEEE Conference on Decision and Control (CDC)*, vol. 26, Los Angeles, USA, 1987, pp. 180–185, doi: 10.1109/CDC.1987.272738.

[43] O. Khatib, "The potential field approach and operational space formulation in robot control," in *Adaptive and Learning Systems*, K.S. Narendra, Ed. Springer, Boston, 1986, pp. 367–377, doi: 10.1007/978-1-4757-1895-9_26.

[44] X. Xu, L. Zhang, S. Sotiriadis, E. Asimakopoulou, M. Li, and N. Bessis, "CLOTHO: A large-scale Internet of Things-based crowd evacuation planning system for disaster management," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3559–3568, 2018, doi: 10.1109/JIOT.2018.2818885.

[45] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: Theory and experiments," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 6, pp. 1423–1436, 1990, doi: 10.1109/21.61211.

[46] J.O. Kim and P.K. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *EEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 338–349, 1992, doi: 10.1109/70.143352.

[47] H. Zhang, Y. Wang, J. Zheng, and J. Yu, "Path Planning of Industrial Robot Based on Improved RRT Algorithm in Complex Environments," *IEEE Access*, vol. 6, pp. 53296–53306, 2018, doi: 10.1109/ACCESS.2018.2871222.