TWO-LAYERS WORKSPACE: A NEW APPROACH TO COOPERATIVE OBJECT TRANSPORTATION WITH OBSTACLE AVOIDANCE FOR MULTI-ROBOT SYSTEM

STEPHANIE KAMARRY¹, RAIMUNDO CARLOS S. FREIRE¹, ELYSON A. N. CARVALHO,², LUCAS MOLINA², PHILLIPE CARDOSO SANTOS³ and EDUARDO OLIVEIRA FREIRE²,

¹Federal University of Campina Grande, Paraíba, Brazil
²Federal University of Sergipe, Sergipe, Brazil
³Federal Institute of Sergipe, Sergipe, Brazil

Corresponding author: Stephanie Kamarry (e-mail: stephanie.sousa@ifs.edu.br).

ABSTRACT In this paper we present a new approach to cooperative overhead object transportation with obstacle avoidance for multi-robot systems. This approach considers situations in which there are obstacles that only pose a risk to the robots involved in the transport and not necessarily to the group of robots with the load. Thus, we propose to separate the workspace into two layers in a way that the obstacles which only pose risk of collision to the robots can be transposed by a local planner. This new approach to the load transportation problem allows the system to find shorter paths or even find a solution in situations which would not be possible with the classic approach. Furthermore, the developed method has a configuration space independent of the number of robots and it is robust to environmental variation, not reducing its performance in more complex environments as in narrow passages or long corridors. The proposed approach was tested by simulation and the results are compared to the classic one.

INDEX TERMS Muti-robot systems, Object Transport, Low Dimension Configuration Space, Obstacle Avoidance.

I. INTRODUCTION

In recent years, there has been increasing research interest in load transport by multi-robot systems (MRSs), making it an emerging field [1]–[4]. Cooperative transport has become a relevant problem in cooperative robotics due to the fact that several robots have potential advantages over a single one, such as fault tolerance, robustness and feasibility. A cooperative team of robots can perform a transport task even if some robots fail. They can also transport large and heavy objects in challenging environments [5].

Cooperative handling and transport of heavy payloads is required in a number of potential MRS applications [6], including construction, manufacturing, and automated warehouses [7]; disaster response and search and rescue missions [8]; assembly of ships and aircrafts [9]; and manipulation, assembly and construction tasks in inhospitable spaces and marine environments [10].

Over the past 20 years, load transport by MRS has become a classic task to study cooperation in robot groups. The strategies used are traditionally divided into three most common types: pushing, caging and grasping [1].

Carrying a load by grasping involves making multiple robots attach to the object, for example, via claws [11] or by lifting it [12]. In the vast majority of research in which MRSs use grasping strategies to transport objects collectively, the robots are pre-connected to the object and never release the connection during transport [1]. Thus, within the researches using the grasping strategy, a series of studies in which robots cooperatively transport the object over their bodies emerged [13], [14].

In most of the works which the formation needs to carry the object on top of its body from one point to another, the robots must keep the formation in balance and avoid
collisions with obstacles present in the environment [15]. Obstacle avoidance methods used in grasping strategy, generally involve the need to manipulate the load with robots that have a load handling device or when mobile manipulator robots are used. [16] In either case, the problem becomes complex due to the size of the configuration space and the necessary adjustment of forces [17].

In cases that robots carry the load over their heads, there is no possibility of handling the load. In general, the set composed of formation and load is considered as a single virtual rigid body with the objective of facilitating the planning and maintain the formation of robots [18]. In these situations, when unexpected obstacles appear on the path, the detected obstacle is incorporated into the map and the obstacle avoidance is made by replanning the path.

Nevertheless, conventional path planning methods based on virtual structures consider only geometric and topological conditions, such as the shape of obstacles and robots combined with the load. This approach limits potential applications of MRS carrying loads on top of their bodies, as there are many situations in which unknown obstacles arise during navigation, but presenting a collision risk only to the robots of the formation rather than to the entire virtual structure. Thus, adding an obstacle on the map only based on its topological format can lead to unnecessary long paths or the impossibility of finding a solution for the task.

Such situations are very common, since there are environments where its physical structure is known a priori, but during the robots’ task, unknown obstacles of different heights may arise, as well as irregular terrain, which can be avoided through a local planner. For example, in warehouses of large companies that work with systems of multiple robots transporting loads and organizing stock. In these situations, robots that are moving within the warehouse pose a collision risk only to the robots that are carrying the load and not necessarily to the complete virtual structure of the formation with the load.

Therefore, we approach the problem proposing an alternative obstacle avoidance strategy, separating the workspace into two layers: the first consisting of the unknown obstacles that offer risk of collision only to the robots that belong to the formation (R-layer), and the second one consisting of known obstacles with a height greater than the height of the robots, that is, obstacles that pose a risk of collision to both the load and robots (L-layer).

Since the obstacles in the R-layer do not offer risks of collision to the load, these barriers can be transposed by robots in formation, keeping the load balanced over their heads. This approach allows finding topological shorter paths for the formation, and even finding paths on the environment that were not possible if the environment is considered as in the classic approach.

Furthermore, to solve the problem of transporting a load in environments with two classes of obstacles, in this work we propose an hierarchical motion planning architecture, executed in two layers: a global and a local one. The differential of the proposed solution is that the dimension of the local configuration space (Ξ) created is totally independent of the number of robots involved in the task, which makes this mapping efficacious for systems composed of multiple robots, in order to allow the execution of hybrid local planning in real-time.

Within this context, this work proposes a solution to a problem not yet addressed, presenting the problem of transporting loads with obstacle avoidance through a new perspective. This approach has as major contributions the following points:

- **Representation** - in this approach, it is possible to separate a three-dimensional environment into two-layers workspace through the classification of obstacles present in the environment into two categories;
- **Local configuration space** - one of the main advantages of this approach is the construction of a local configuration space that remains with 2 DOF, regardless of the number of robots involved in the transport;
- **Collision Avoidance** - the proposed approach, presents a scalable and efficient method for navigation of a team of robots carrying an object over their head and avoiding obstacles in 3D workspace.

The proposed solution as well as the results obtained, are presented throughout this work, which is structured as follows: in section II the main works in MRS are presented. The problem is formally described in Section III and in Section IV a description of the solution is provided. Obtained results are presented in section V. At the end of the document, conclusions and future works are discussed, followed by references.

## II. PROBLEM DEFINITION AND ASSUMPTION

In this section, the assumed considerations about the robot, the load and the environment will be addressed. In addition, the problem proposed by this paper will be detailed.

### A. MOBILE ROBOTS

In this work it is considered that there is a team of holonomic robots capable of navigating in formation, described by \( Rb = \{r_1, r_2, \ldots, r_n\} \), in which \( n \in \mathbb{N} \). For each robot \( r_i \) from the set \( Rb \), with \( i = \{1, 2, \ldots, n\} \in \mathbb{N} \), its pose in time \( t \) is denoted by \( q_i(t) = [x_i, y_i, \theta_i]^T \in \mathbb{R}^3 \).

The set \( Rb \) is composed of homogeneous robots with the same dynamic model and cylindrical shape; the robots are omnidirectional and their geometry is defined by two constants: \( R \) and \( h \), in which \( R \) represents the radius of the circular base and \( h \) the height of robots.

It is assumed that each robot can estimate its own speed and direction, as well as it is able to locate itself in relation to the global coordinate system, disregarding the dynamic effect of the motors and the inertia of the bodies. Furthermore, the robots are able to detect the presence of other robots and obstacles in the environment. Thus, the neighborhood of \( q_i \) is given by the range and the field of view of the sensor’s hardware, with \( T_i \) being the set of neighbors in this region.
The communication between the robots is direct and perfect, within a circular region defined by the radius $R_{com}$ whose center is the formation equilibrium point.

For the formation, the model $T$ [19] is considered, in which the set of robot positions and external vertices \( \{w_1, \ldots, w_m\} \) is given relative to the center of rotation (typically the centroid) of the formation, in which $m$ defines the number of external vertices of the formation. The vertex set represents the convex hull of the robot’s positions in the formation, thus reducing the complexity of formations composed of many robots.

B. OBJECT AND MANIPULATION

In this work, the load is understood as any flat structure, with uniform density, convex geometry and which can be carried by the robots.

The location of the load’s center of mass is known. The transport strategy used is grasping overhead, in which the robots are physically attached to the load and it is transported on top of the robots, that is, the entire body of the robots is located under the load.

In addition, it is considered that robots do not perform the lifting and already start the task connected to the load, so that the set of robots with the load can be considered as a single rigid body and, once the system is started, it is not possible the decoupling of robots from the load.

C. ENVIRONMENT

The robots navigate in a global map $\mathcal{W}$ composed of two classes of static obstacles, which characterize the proposed approach:

1) $\mathcal{O} \subset \mathbb{R}^3$ - Obstacles that generate collision with the entire structure or only with the load, that is, obstacles with a height greater than $h$;
2) $\mathcal{O}_R \subset \mathbb{R}^3$ - Obstacles that generate collision only with robots, possible to be detected by their sensors, that is, obstacle with a height smaller than $h$.

There is a set $\mathcal{O}_{comp}$ that denotes the space composed of all obstacles present in the environment and is formed by

$$\mathcal{O}_{comp} = \mathcal{O} \cup \mathcal{O}_R.$$  \hfill (1)

The shape, position and orientation of obstacles present in $\mathcal{O}$ are known by the overall system. However, the obstacles of the class $\mathcal{O}_R$ are unknown, but measurable by sensors embedded in robots.

If $C_{r_i}$ denotes the configuration space of the robot $r_i$, the composite configuration space of the system $C$ is

$$C = C_{r_1} \times C_{r_2} \times \ldots \times C_{r_n}.$$  \hfill (2)

For the proposed problem there are two different configuration spaces: the $C_0$ which is the configuration space in which the obstacles generate collision with the whole structure or only with the load; and the $C_{O,R}$ which is the space for the configuration of obstacles that generate collisions only with robots, without offering any risk to the transported load. Both of the configuration spaces presented consist of a subspace of $C$.

Likewise, for the proposed problem, there are two Free Configuration Spaces: $C_{freeO}$ (3) for the first class of obstacles and $C_{freeO_R}$ (4) for the second class.

$$C_{freeO} = C \setminus C_{obsO}.$$  \hfill (3)

$$C_{freeO_R} = C \setminus C_{obsO_R}.$$  \hfill (4)

D. PROBLEM DEFINITION

The proposed task consists of transporting an object $\mathcal{M}$ in $\mathcal{W} \subset \mathbb{R}^3$ from an initial configuration $q_{init}$ to a target configuration $q_{goal}$ through the topological shortest path, without collision with the obstacles present in the set $\mathcal{O}_{comp}$.

Thus, in a problem containing both classes of obstacles defined in II-C, the proposed environment representation has several advantages, as illustrated in Figure 1, in which the formation movement planning using the classical representation of the environment (dashed path) and the movement planning using the representation proposed in this work (continuous path) are represented.

In Figure 1, the obstacle $B'$ is unknown and its height is shorter than the robots’. So, it does not offer risk of collision for the load, only for the robots. In addition, it is only visible by the formation when it enters the distance range of the sensors.

In this case, when an unknown obstacle is found, the classic methods include it on the map of the environment (obstacles from class $\mathcal{O}$) and re-plan the task. In the example of Figure 1, the formation would execute a path similar to the one illustrated by the traced line. Nevertheless, considering the environment representation proposed in this work, the obstacle $B'$ can be included in the class of obstacles $\mathcal{O}_R$ instead of in the class $\mathcal{O}$. This simple segmentation of the work space in two layers allows the system to find shorter paths for the load (continuous line in Figure 1) when compared to the re-planning traditional solution (traced line), in case the obstacle can be overcome by the robots in formation.

![FIGURE 1. Comparison between the paths obtained by the classical representation of the environment (dotted line) and the proposed representation (continuous line).](image)
The possibilities of obstacles in \( O_R \) that can be overcome by formation is totally associated with the characteristics of the formation and the coupling between robots and load. The more flexible these characteristics are, the greater are the possibilities for overcoming the obstacle. Considering the definitions presented so far in section II, a solution for the transposition of obstacles belonging to the \( O_R \) is presented below.

III. SOLUTION

The proposed approach to solve the problem of transporting a load in environments with two classes of obstacles consists of a hierarchical motion planning architecture, executed in two layers: a global and a local one.

The global planning is responsible for planning a viable way to transport the load without collision with the known obstacles of \( O \), that is, not considering the unknown obstacles that generate collisions only with robots (\( O_R \)). The local layer, on the other hand, is responsible for adjusting the orientation of the load.

The global planning is initially executed by a path planner based on the Virtual Framework approach. First, formation planning is done, then each robot locally implements its own controller.

After the global planning, the robots follow the configurations until they detect an imminent collision with an obstacle belonging to \( O_R \). At this moment, the local planner is activated to overcome the obstacle. This planner performs the task by rotating the load along the previously planned trajectory, but without affecting the overall planning already carried out.

If it is not possible to overcome the obstacle, a new global planning will be realized including this object in the class of known obstacles that cannot be overcome (\( O \)).

Each layer will be explained in more detail in the next sections.

A. R-LAYER

1) Global Planning

Load movement planning can be seen as a simple problem: given the map \( W \) and the configurations for the start (\( q_{init} \)) and destination (\( q_{goal} \)), it must find a collision-free path that connects the start configuration to the destination configuration. As the movement of the load in the environment is not the main theme of this work, the initial and final orientations of the robots were set free. Thus, the chosen orientation criterion is that the load must always be oriented in the direction of the movement, and this criterion is added at the end of the planning.

The technique used for the global planning was the Classic RRT, as it is probabilistically complete and of rapid execution [20]. It is noteworthy that given the problem presented, other motion planning techniques can be used.

The choice for RRT is due to its easy implementation and its proof of completeness. Although it does not produce an optimal solution to the problem, this algorithm is computationally very fast. An RRT is probabilistically complete under general conditions, which means that the chance of the algorithm to return a solution tends to 1 when the execution time of the algorithm tends to infinity.

The RRT algorithm basically tries to build a graph covering the free space \( C_{free} \) with the largest number of possible paths. The graph begins in \( q_{init} \) and expand the tree for the environment by incorporating random nodes created in space.

Starting from an initial configuration state, the algorithm draws a new sample (\( q_{rand} \)) from \( C_{free} \). It is checked in the tree the distance of each node to \( q_{rand} \), the node with the shortest distance (\( q_{near} \)), is chosen as an attempt to grow the tree. It traces a vector from \( q_{near} \) toward \( q_{rand} \), however, only part of the vector is used, because there is a constant (\( d \)) limiting the maximum size an edge can grow. This constant value defines the vector segment size to be evaluated. The segment goes from \( q_{near} \) to \( q_{new} \).

If the segment is contained entirely in \( C_{free} \), the node defined by \( q_{new} \) is then incorporated into the tree, otherwise, the tree does not change and a new sample is drawn.

This process repeats until the destination configuration is incorporated into the tree. Then, a valid path is found from the start configuration to the destination (Figure 2).

In summary, Classic RRT is constructed as shown in Algorithm 1.

In most cases, the path found by the RRT has sudden variations in direction between two configurations, which makes it difficult for the robot to follow the trajectory. To solve this problem, a first-order low-pass filter was used, with a cutoff frequency determined empirically (Figure 3).

The result of the global planner is a viable path, free from collision with obstacles belonging to \( O \), composed of a sequence of configurations \( q_{init}, q_2, \ldots, q_{goal} \). This sequence of settings correspond to the center of the virtual structure and its control is presented below.
Algorithm 1 Algorithm which generates RRT.

```plaintext
function GENERATE RRT(qinit, qgoal, d, W)
    RRT.init(qinit)
    qnew = qinit
    while dist(qgoal, qnew) ≥ d do
        qrand ← State_Random()
        qnear ← Nearest(qrand, RRT)
        qnew ← New_State(qnear, Δt)
        if Check_Collision(qnew, qnear) = Free then
            RRT.add_node(qnew)
        end if
    end while
    RRT.add_node(qgoal)
    return RRT
end function
```

2) Virtual Structure Control

To illustrate the proposed solution, a formation with three robots, \( Rb = \{r_1, r_2, r_3\} \) is considered and organized in a formation of model \( T \), such as explained in the II-A section. Each configuration \( q_i \) of the tree represents the center \( c \) of the formation \( T \) of the robots. So, to find the corresponding configuration of each robot (vertices of the formation), a simple translation and rotation is performed, given that \( l \) is the predefined distance between the robots and \( h \) the height of the equilateral triangle of the formation, the calculation of the position of each robot from the central configuration \( c \),

\[
c = \begin{bmatrix} x_c \\ y_c \end{bmatrix},
\]

given:

\[
r_1 = \begin{bmatrix} x_c \\ y_c + 2h/3 \end{bmatrix},
\]

\[
r_2 = \begin{bmatrix} x_c - l/2 \\ y_c - h/3 \end{bmatrix},
\]

\[
r_3 = \begin{bmatrix} x_c + l/2 \\ y_c + h/3 \end{bmatrix}.
\]

The orientation of the formation is defined so that one of the robots is always ahead of the direction of the movement. This is performed by calculating the angle formed between the current configuration and the next configuration in the tree:

\[
\theta_i = \arctan(q_{i+1}, q_i).
\]

Figure 4 shows the map with the path found through the RRT with the sequence of configurations of the center of the formation (in yellow).

3) Robot’s control

In order for the robot to follow the path generated by the tree, a final position controller was implemented in which the node connected to the point where the robot is located is considered, temporarily, as the desired position, until the distance from the robot and this node becomes smaller than a predefined threshold, when the next node in path is defined as the desired position. This process is repeated so that the robot navigates the entire way and reaches the destination point.

The controller used was based on the one presented in [21] and contains the following equations for the linear \( (V_{pf}) \) and angular \( (W_{pf}) \) speeds of the robot, respectively:

\[
V_{pf} = kv_{pf} \tanh(\rho) \cos(\alpha),
\]

\[
W_{pf} = kw_{pf} \alpha + kv_{pf} \frac{\tanh(\rho)}{\rho} \tanh(\rho) \cos(\alpha).
\]

in which \( kv_{pf} \) and \( kw_{pf} \) are positive constants, \( \rho \) is the distance between the robot coordinate system and the destination coordinate system, and \( \alpha \) the error of robot orientation.
in relation to the line connecting the robot to the next node which is considered as temporary destination point. This controller was chosen because of its asymptotic stability is proved [21].

B. R-LAYER - LOCAL PLANNING

Initially, the robots in formation follow the global planning made through the RRT. When the sensors detect an obstacle that generates a collision with one or more robots in the formation, the movement of the formation is stopped and the local planning is carried out to overcome the obstacle.

The proposed method of obstacle avoidance through load rotation consists of mapping the local workspace of the formation, defined by the portion of the globally planned path that is covered by the radius of the sensors, to a 2-dimensional configuration space, independent of the number of robots involved in the task. The definition of this space, the construction process and the local planning is presented below.

1) Configuration Space \( \mathcal{L} \)

The proposal to build a new configuration space model, to solve the local hybrid navigation task, aims to reduce the dimensionality of the solution to map the local motion planning problem of an \( n \) robots team, with three dimensions \((x, y, \theta)\) for each robot, to a punctual robot local motion planning problem, in the two-dimensional configuration space.

The configuration space \( \mathcal{L} \) is always composed of 2 dimensions, being the first defined by the load displacement along the globally planned path \((\text{disp})\), and the second dimension defined by the rotation angle of the load around the center of the formation \((\theta_{\text{rot}})\).

This configuration space is not known beforehand, being built iteratively as the robot moves and its observation of the unknown obstacle is more complete. Through this mapping of the robots’ local workspace to the 2 dimensional \( \mathcal{L} \) configuration space, it is possible to implement any classical algorithm of probabilistic motion planning, and to carry out the obstacle avoidance locally in a fast way.

For a better visualization of the proposed configuration space, Figure 5 shows the representation of a set of robots carrying a load in the workspace and the same representation in the configuration space \( \mathcal{L} \) respectively.

\[ \text{The illustrated experiment highlights the simplification provided by the use of } \mathcal{L} \text{ space, which maps the local motion planning problem of a team of } \text{3 robots, into a local motion planning problem of a point robot, in the two-dimensional configurations space.} \]

\[ \text{The first dimension (disp) provides the displacement of the formation along the trajectory previously planned by the global system, the starting point (disp = 0) is determined from the moment the sensors encounter an obstacle of the class } \mathcal{O}_R, \text{ and grows continuously as the formation moves away from this point.} \]

\[ \text{In the global planner, the orientation of the formation is defined in a way that one of the robots is always ahead of the direction of movement. To enable obstacle avoidance, in the local planner, the choice of formation orientation is set free through the second dimension of the configuration space } \mathcal{L} \text{ (}\theta_{\text{rot}}\text{). In this way, the criterion of fixing the load orientation in the direction of movement is switched off and the load is set free to rotate, allowing the formation to transpose the obstacle without generating a collision.} \]

\[ \text{This new representation is clearly a simpler problem to solve. Furthermore, the resulting path defines the sequence of angles to be assigned to the virtual structure along the globally planned path, so that the robot-load set can overcome the obstacle.} \]

\[ \text{It is noteworthy that the dimension of the space created is totally independent of the number of robots involved in the task, which makes this mapping very useful for systems with multiple robots, in order to allow real-time execution of reactive local planning.} \]

\[ \text{Within the new proposed configuration space it is possible to apply any probabilistic motion planning approach, since the space is not known a priori. In this work, the use of RRT in the local planner was also chosen for its demonstration of probabilistic completeness and fast execution.} \]

2) Local Planning

The local planning is carried out in the configuration space \( \mathcal{L} \) and it consists of randomly choosing a combination of a rotation angle value around the center of mass of the load \((\theta_{\text{rot}})\) and the displacement of the virtual structure along the global path \((\text{desl})\), that makes possible to overcome the obstacle encountered.

\[ \text{This process is performed through the construction of an RRT: given the configuration space } \mathcal{L}, \text{ the starting point (}q_k\text{) set at the moment that an obstacle is detected, and destination } (q_{k+R}), \text{ in which } R \text{ is defined by the range of distance sensors, a collision-free path must be found to connect the configuration } q_k \text{ to } q_{k+R} \text{ by varying the rotation of the load along the globally planned path.} \]

\[ \text{This algorithm is performed iteratively until no obstacles are detected within the range of the sensors in the direction of the formation movement. From this moment on, the closest point on the global path is set as the destination point and the formation returns to navigate based on the global planning.} \]
The proposed algorithm basically tries to cover the free space of $L$ with the maximum possible number of paths. The construction of the tree starts at the moment of obstacle detection, and it grows in the environment, incorporating nodes randomly selected until the distance from one of the nodes added to the tree to the root ($q_k$) is close to the radius of range of distance sensors ($R_{sens}$). When this happens the tree’s growth is stopped and navigation is started.

The logic of the RRT algorithm follows the sequence explained in III-A1, the only difference is that the generation of new random configurations is performed in the space $L$ and that each configuration is composed of the angle of rotation of the load around the center of the formation ($\theta_{rot}$) and the position of the load along the path ($desl$).

Figure 6 shows the path obtained in the local planner using the configuration space $L$ from the moment the robots enter the local planner until the return to the globally planned path.

The proposed solution was tested by simulation and the results can be seen in the next section.

**IV. RESULTS**

In order to demonstrate the proposed solution for the presented problem, we defined three types of experiments. For all of them, the proposed approach is compared to the classical method, which is performed by replanning the path for the entire virtual structure when it finds an unknown obstacle.

In the first two experiments (Sections IV-A and B), we compared the transport of a load in the same environment, using formations with 3 and 4 homogeneous and omnidirectional robots. In the third experiment (Section IV-C), we present a different map to demonstrate how the local planner behaves in a more challenging environment.

For all experiments, simulation was performed in Matlab R2019a software, and robots navigate a $1000 \times 1000$ u.m. map, with start setting at $q_{init} = (150, 150)$ and destination setting at $q_{dest} = (850, 150)$. For the first and second experiments, the map shown in Fig. 7.

For the third experiment it was used the map shown in Fig. 8, with the same dimensions of the previous map.

In the two maps used, the obstacles that offer a risk of collision only to the robots that belong to the formation are presented in green.

In these experiments, each configuration is given by the geometric center of the formation.

The experiments were carried out in order to compare the classical approach to the solution proposed in this work.

**A. EXPERIMENT 1**

For this first analysis, 3 robots were used to transport the load. As explained in the section III-A1, first the global motion planning of the virtual structure is performed using the Classical RRT algorithm.

For a visual analysis, the comparative result of the path obtained after global planning and filtering is shown in Figure 9: in (a) is presented the result using the classical approach and in (b) with the proposed approach. A longer path is clearly perceived when the problem is approached in the classic way, approximately $1.78$ times longer, as it is the only possible path since the possibility of crossing the green obstacle is not considered.

In Figure 10 the result for the navigation of the formation using the proposed approach is presented. In this map, the formation follows the path planned by the RRT (in blue). However, when an obstacle that can be transposed is detected, the system enters the local planner (in red), rotating the load and transposing the obstacle. The formation then resumes the planning carried out globally up to the destination point.
Figure 9. Path result obtained by the RRT for: (a) Classic Approach and (b) Proposed Approach.

(a)  
(b)  

FIGURE 9. Path result obtained by the RRT for: (a) Classic Approach and (b) Proposed Approach.

Figure 10. Formation navigation using the Proposed Approach: in blue when performing global planning and in red when performing local planning.

FIGURE 10. Formation navigation using the Proposed Approach: in blue when performing global planning and in red when performing local planning.

Figure 11 highlights the moment when the formation rotates in order to avoid the obstacle in the local planner.

FIGURE 11. Emphasis on the sequence of positions when training performs the obstacle avoidance.

Figure 12 shows a representation of the local planning using RRT in the configuration space $L$ with the obstacles highlighted in black, the constructed tree and the final path obtained in yellow.

FIGURE 12. Representation of the local configuration space with obstacles highlighted in black and the path obtained in yellow.

As the purpose of this work is to keep the robot on the globally planned path, only varying the rotation when it encounters an obstacle that can be transposed, in Figure 13 we presented the planned path for the virtual structure in blue and the path run by formation in red, highlighting that there was no change in the global planned path.

FIGURE 13. Planned path for the virtual structure in blue and the path run by formation in red.

Figure 14 presents the difference between the orientation performed in the experiment ($\theta_{\text{perf}}$) and the planned one ($\theta_{\text{planned}}$). This graphic demonstrates the great variation of orientation when the formation makes the transposition of the obstacle in the local planner, as expected.

FIGURE 14. Graph of orientation error over time.
B. EXPERIMENT 2

For this second analysis, 4 robots were used to transport the same load as the previous experiment. For this reason, we chose to use the same path planned previously in the section IV-A. The comparative result of the path obtained after global planning and filtering is shown in Figure 9.

The result for the navigation of the formation using the proposed approach is presented in Figure 15. In this map, the formation follows the path planned by the RRT (in blue). However, when an obstacle that can be transposed is detected, the system enters the local planner (in red), rotating the load and transposing the obstacle. The formation then resumes the planning carried out globally up to the destination point. Figure 16 highlights the moment when the formation rotates to avoid the obstacle using the local planner.

In Figure 17 is presented a representation of the local planning using RRT in the configuration space $L$ with the obstacles highlighted in black, the constructed tree and the final path obtained in green.

![Image 15](image15.png)

**FIGURE 15.** Formation navigation using the proposed approach executing global planning (in blue) and local planning (in red).

![Image 16](image16.png)

**FIGURE 16.** Steps carried out to avoid the obstacle in the local planner.

C. EXPERIMENT 3

The third experiment was conducted in order to demonstrate the efficiency of the method in explore environments with narrow passages with obstacles obstructing the path. This situation cannot be solved by the classical approach.

In this experiment, the robot avoids three obstacles at the same time. The solution found by the proposed algorithm is shown in Figure 19 and the sequence of movements performed in the local planner is highlighted in the Figure 20.
V. CONCLUSIONS

In this paper, a new way of approaching the problem of cargo transport by MRS with obstacle avoidance was presented, considering that there are two different classes of obstacles: those that can collide with the load and that are known a priori, and those that can only collide with one or more robots, unknown a priori, but which are visible by the sensors embedded in the robots.

The proposed problem and its formulation are common situations in robot navigation in dynamic and partially unknown environments. Thus, the proposed formulation enables a new look at the navigation of MRSs in homes, warehouses and uncontrolled environments.

In addition to formulating the problem, this work presents an initial solution that provides a collision-free path to transport the object, through the use of a global and local planner. The differential of the proposed work lies in the local planner that rotates the load, keeping the structure on the globally planned path in order to overcome the obstacles of the second class.

This approach allows for the mapping of the workspace in a locally built configuration space that has the advantage of having only two dimensions, regardless of the number of robots involved in the task, as demonstrated in the experiments carried out. This mapping reduces the dimensionality of the problem and makes it possible to use several classical motion planning approaches, to solve the local trajectory in real-time.

The presented results demonstrate possible ways to effectively solve the proposed problem and highlight the advantages of splitting the environment in two layers, since it allows the MRSs to find shorter paths for the load, and even find paths on maps that were not possible before. This is possible due to the fact that obstacles can now be bypassed, rather than avoided as in the classic approach.

REFERENCES

[1] E. Tuci, M. H. Alkilabi, and O. Akanyeti, “Cooperative object transport in multi-robot systems: A review of the state-of-the-art,” Frontiers in Robotics and AI, vol. 5, p. 59, 2018.
[2] Z. Feng, G. Hu, Y. Sun, and J. Soon, “An overview of collaborative robotic manipulation in multi-robot systems,” Annual Reviews in Control, vol. 49, pp. 113–127, 2020.
[3] S. Wilson, A. Buffin, S. C. Pratt, and S. Berman, “Multi-robot replication of ant collective towing behaviours,” Royal Society open science, vol. 5, no. 10, p. 180409, 2018.
[4] R. Herguedas, G. López-Nicolás, R. Aragüés, and C. Sagüés, “Survey on multi-robot manipulation of deformable objects,” in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2019, pp. 977–984.
[5] G. Habibi, W. Xie, M. Jellins, and J. McLurkin, “Distributed path planning for collective transport using homogeneous multi-robot systems,” in Distributed Autonomous Robotic Systems. Springer, 2016, pp. 151–164.
[6] R. N. Darmanin and M. K. Bugeja, “A review on multi-robot systems categorised by application domain,” in 2017 25th mediterranean conference on control and automation (MED). IEEE, 2017, pp. 701–706.
[7] F. Xue, H. Tang, Q. Su, and T. Li, “Task allocation of intelligent warehouse picking system based on multi-robot coalition,” KSII Transactions on Internet and Information Systems (TIIS), vol. 13, no. 7, pp. 3566–3582, 2019.
[8] A. Q. Li, R. Cipolleschi, M. Giusto, and F. Amigoni, “A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings,” Autonomous Robots, vol. 40, no. 4, pp. 581–597, 2016.

[9] V. Terechik, J. Stewart, N. Bykov, S. Pedigo, S. Devasia, and A. G. Banerjee, “An efficient scheduling algorithm for multi-robot task allocation in assembling aircraft structures,” IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3844–3851, 2019.

[10] V. Kuzin, A. Kurkin, D. Y. Tyugin, A. Zaitsev, D. Zezyulin, P. Beresnev, and V. Belyakov, “Multi-robot system for monitoring ice and wave climate in coastal zone,” in Proceedings of The 33rd International Symposium on Okhotsk Sea & Polar Oceans 2018, 2018, pp. 147–150.

[11] J. Alonso-Mora, S. Baker, and D. Rus, “Multi-robot formation control and object transport in dynamic environments via constrained optimization,” The International Journal of Robotics Research, vol. 36, no. 9, pp. 1000–1021, 2017.

[12] L. Zhang, H. Xiong, O. Ma, and Z. Wang, “Multi-robot cooperative object transportation using decentralized deep reinforcement learning,” arXiv preprint arXiv:2007.09243, 2020.

[13] H. K. Kheebuz and P. Jensfelt, “Object detection approach for robot grasp detection,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 4953–4959.

[14] S. Joshi, S. Kumra, and F. Sahin, “Robotic grasping using deep reinforcement learning,” in 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE). IEEE, 2020, pp. 1461–1466.

[15] R. W. Beard, J. Lawton, and F. Y. Hadaegh, “A coordination architecture for spacecraft formation control,” IEEE Transactions on Control Systems Technology, vol. 9, no. 6, pp. 777–790, 2001.

[16] W. Li and R. Xiong, “Dynamical obstacle avoidance of task-constrained mobile manipulation using model predictive control,” IEEE Access, vol. 7, pp. 88 301–88 311, 2019.

[17] X. Ding, Y. Liu, J. Hou, and Q. Ma, “Online dynamic tip-over avoidance for a wheeled mobile manipulator with an improved tip-over moment stability criterion,” IEEE Access, vol. 7, pp. 67 632–67 645, 2019.

[18] K.-H. Tan and M. A. Lewis, “Virtual structures for high-precision cooperative mobile robotic control,” in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS’96, vol. 1. IEEE, 1996, pp. 132–139.

[19] Y. Baljaggi, O. Sölyns, and E. Sahin, “A review: Pattern formation and adaptation in multi-robot systems,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-43, 2003.

[20] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” Tech. Rep., 1998.

[21] R. Carelli, H. Secchi, and V. Mut, “Algorithms for stable control of mobile robots with obstacle avoidance,” Latin American Applied Research, vol. 29, no. 3/4, pp. 191–196, 1999.

[22] M. A. Lewis, “Virtual structures for high-precision cooperative mobile robotic control,” in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS’96, vol. 1. IEEE, 1996, pp. 132–139.

[23] E. Bahceci, O. Soysal, and E. Sahin, “A review: Pattern formation and adaptation in multi-robot systems,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-43, 2003.

[24] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” Tech. Rep., 1998.

[25] R. Carelli, H. Secchi, and V. Mut, “Algorithms for stable control of mobile robots with obstacle avoidance,” Latin American Applied Research, vol. 29, no. 3/4, pp. 191–196, 1999.