Abstract—Efficient path planning considering a sequence of goals in a manufacturing scenario is still challenging for nonholonomic robots. In this paper, we present an approach to improve the efficiency of path planning considering a sequence of goals for four-wheeled Drive Robots. Besides using a near-optimal sampling-based algorithm applied in a map with real inflating obstacles, virtual obstacles are added to the map to help the generation of smoother paths that lead the robot from the start pose to its goals poses faster, with fewer mechanical efforts, and less power consumption.

I. INTRODUCTION

Autonomous mobile robots will be protagonists in several manufacturing tasks in the Fourth Industrial Revolution (4IR). The successful application of mobile robots requires not only safe but also an efficient operation where the robot has to achieve a predefined sequence of goals.

The usual planning approach used in the navigation software (Navigation Stack [1]) of the ROS framework [2] utilizes two separate algorithms: the global planner and local planner. The goal of the global planner is only to find a feasible path between two points in the free space of the map. On the other hand, the local planner is a more sophisticated algorithm responsible to handle the obstacle avoidance, trajectory optimization, and the closed-loop controller of the robot. When using this approach for differential nonholonomic robots, the robot is just driven to the goal position and then a pure rotation movement is made to achieve the goal orientation if it is necessary. A simplified navigation architecture using global and local planners is shown in the Fig. I.

The contribution of this work is improving the path planning for a four-wheeled differential drive robots. [3] that need to accomplish a sequence of goals. The paths generate with this approach will demand from the robot less:

1) torque, and consequently less power consumption;
2) mechanical wear;
3) and time to accomplish the goals.

Four-wheeled differential drive robots do not perform well pure rotations as two-wheeled differential drive robots and depending on how rough is the grounds, pure rotations demand to much torque and raise the mechanical wear. Therefore, it is necessary to generate trajectories that lead directly to the goal’s orientation, guaranteeing lower power consumption and lower mechanical wear [4].

This work was supported by the Federal University of Bahia and the FASTEN Project.

A. Problem Statement

In autonomous manufacturing, mobile robots have to accomplish different tasks in different places in the plant, forming a sequence of goals. For nonholonomic robots, only considering individual goals usually raise unnecessary maneuvers and mechanical wear, increasing time and cost of the manufacturing process.

The Fig. 3 shows the difference between the two approaches with a differential robot. On the left, the robot performs a trajectory that ignores the goal orientation and the existence of future goals and it results in an unnecessary pure rotation to achieve the orientation of the goal or the orientation to start the next path. On the right, on the other hand, the robot performs a trajectory to Goal 1 that makes it easier to accomplish future goals 2 and 3.
B. Proposed Solution

We present a path planning for a predefined sequence of goals to be applied for a four-wheeled differential robot into a small manufacturing plant. With this approach, the robot generates a set of more efficient paths that allow the robot to reach each goal position easier, where at each point a robotic arm (attached on robot’s base) has to execute some fabrication task. Examples of these tasks are manipulation, grasping, and moving of 3D printer parts.

A near-optimal sampling-based algorithm [7] is used to find a path between two points in the free space. The widely used RRT* [8] algorithm was chosen for this task due to its simplicity and effectiveness, but any other algorithms [9] could be chosen. And besides using occupancy grid mapping with real inflating obstacles, we propose the addition of virtual obstacles to to help the generation of smoother paths that lead the robot from the start pose to its goals poses faster and with fewer mechanical efforts.

C. Related Work

Multi-goals path planning problem is a vast field of study [10] that involves other areas than robotics. Commonly, mobile robots have to execute a work sequence in a manufacturing setting. This means that the task of the robot may contain a sequence of multiple future goals, where the order of the goals usually matters [11]. Besides generation of short paths, other metrics are important to evaluate the performance of a navigation system, and in our case, the generation of smooth paths is fundamental to decrease the power consumption and the mechanical wear [4] of the robot.

II. PATH PLANNING

A. Global Planner Using RRT*

RRT* is presented as an extension to RRT (Rapidly-Exploring Random Trees [12]). Its major advantage over other algorithms is that it finds an initial path very quickly and then later keeps on optimizing it as the number of samples increases. Pseudocode describing RRT* is shown in Algorithm 1. An example of a resultant path between two points is shown in Fig. 4. As the orientations of the start point and the final point do not match with the path generated the robot shall perform pure rotation to start to follow the path and to achieve the final orientation of the goal.

Algorithm 1: $T = (V, E) \leftarrow \text{RRT*}$

\[
T \leftarrow \text{InitializeTree}(); \\
T \leftarrow \text{InsertNode}(\beta, z_{\text{init}}, T); \\
\text{for } i = 0 \text{ to } i = N \text{ do } \\
\quad z_{\text{rand}} \leftarrow \text{Sample}(); \\
\quad z_{\text{nearest}} \leftarrow \text{Nearest}(T, z_{\text{rand}}); \\
\quad (x_{\text{new}}, y_{\text{new}}, z_{\text{new}}) \leftarrow \text{Steer}(z_{\text{nearest}}, z_{\text{rand}}); \\
\quad \text{if } \text{ObstacleFree}(z_{\text{new}}) \text{ then } \\
\quad \quad z_{\text{near}} \leftarrow \text{Near}(T, z_{\text{new}}, [V]); \\
\quad \quad z_{\text{min}} \leftarrow \text{ChooseParent}(z_{\text{near}}, z_{\text{nearest}}, z_{\text{new}}, z_{\text{new}}); \\
\quad \quad T \leftarrow \text{InsertNode}(z_{\text{min}}, z_{\text{new}}, T); \\
\quad T \leftarrow \text{Rewire}(T, z_{\text{near}}, z_{\text{min}}, z_{\text{new}}); \\
\text{return } T
\]

B. Point to Point Global Planner with Virtual Obstacles

Given a map of the environment, the common path planners only search for a path between an initial point and a final point in the free space. In other words, there is no worry about the orientation of the waypoints. The Fig. 5 is shown the sketch of the virtual obstacles (in blue) that add to the RRT* algorithm the ability to generate a near-optimal path with smoother orientation variation. As the initial pose of the robot is known in the beginning, the free space around the initial node is very limited. On the other hand, the goal node can be located around the goal tolerance, in this case, with radius $0.2$ m.

Differential robots can handle paths with abrupt orientation variation using pure rotations, and this is useful when the robot has to navigate through narrow spaces. Therefore, the last option of path planning for differential robots should be the usual point to point global planner. The priority is searching a path with virtual obstacles that force the RRT* algorithm to generate a path with less or without pure rotations in the initial part and in the final part of the path. The path planning of a single goal is shown in Fig. 6 where a robot with initial pose red has to achieve...
the pose in black. Using these virtual obstacles he sampling algorithm generates a point to point global planner with no abrupt orientation variation.

C. Radius of Curvature

A 2D kinematic modeling of a four-wheeled differential robot is presented in [13] and the Husky robot is used in [3]. The paper [14] analysis the power consumption through the torque of the wheels for different radius of Curvature and is shown that small values of radius demand much power of the motors. Fig. 7 shows the relationship between Power and Curvature Radius (turning radius).

In order to simplify the kinematic calculation, we can treat a four-wheeled differential robot as two-wheeled. This is made modeling with two virtual wheels localized between the real four wheels as is shown in Fig. 9, being the figure description:

- $R_c$ - robot geometric center
- $(x_c, y_c)$ - robot geometric center position
- $\alpha$ - robot angular position
- $W_{FL}$ - front left wheel
- $W_{FR}$ - front right wheel
- $W_{BL}$ - rear left wheel
- $W_{BR}$ - rear right wheel
- $W_L$ - virtual left wheel
- $W_R$ - virtual right wheel
- $l_1$ - distance between robot center and front/rear wheels
- $l_2$ - distance between robot left and right wheels

The paper presents [4] a method to compute the minimum curvature radius for a two-wheeled differential robot to control of smoothing. The minimum radius of curvature that describes the robot to avoid rotational inversion in the wheels is $R > l/2$ because it ensures that the ICC (instantaneous center of curvature) will always be outside of the robot body. If $R = l/2$, the ICC will be in the center of one of the wheels, that is, during rotation only one of the wheels will be in motion and the other in zero speed, this condition is undesirable because the kinetic energy will be completely...
D. Sequence of Goals

The usual path planning approach only considers the processing of one trajectory between two points each time. This leads to a non-optimal solution when using nonholonomic robots where the orientation of the goal is fundamental. After each intermediary goal, the robot shall be already correctly oriented to follow the path of the next goal, avoiding unnecessary maneuvers.

E. Proposed Method

For all points of the sequence, the paths are computed using RRT* algorithm only with the real inflating obstacles of the map. The sequence of goals planning is shown in Fig. 10. With these paths, the Direction of Arrival (DOA) and the Direction of Departure (DOD) of each goal can be calculated. These angles are represented, respectively, in green and black arrows.

Each goal consists of a cartesian position and two possible orientations shifted by pi radians. The goal orientation with the smaller angular distance between the DOD angle of the goal point is chosen. With the definition of the goal pose $(x, y, orientation)$, the virtual obstacles are added and a new path planning is computed.

III. RESULTS

Two sequences of goals were tested using the proposed method. The first sequence has 3 goals and the second sequence has 4 goals. The second includes one goal at a narrow space (goal G2) where the robot can not perform a smooth maneuver, therefore the traditional method was used at this point.

The Fig. 12 shows the paths generated using the proposed method. The sequence consists of three goals (G1, G3, and G4) that are accomplished by a smoother set of paths. The Fig. 13 shows the second sequence of 4 goals, including the goal G2 located at narrow space. In both cases was applied an B-Spline interpolation to generate a smooth curve to be used in the robot trajectory tracking controller.

Although the method proposed generates longer paths, the robot does not need execute pure rotations after reach each goal pose. Therefore, the overall trajectory duration can be even shorter, depending on the angular velocity of the final rotation movement of the usual method.

IV. CONCLUSIONS

In this paper, a novel path planning method to handle a sequence of goals to be performed by a differential 4-wheeled robot was presented. The set of paths generated with this method allows the robot to accomplish the manufacturing tasks faster, with less power consumption, and fewer mechanical efforts.

REFERENCES

[2] Stanford Artificial Intelligence Laboratory et al., “Robotic operating system ROS.”
Fig. 12. Mission with goals sequence G1, G3, and G4.

Fig. 13. Mission with goals sequence G1, G3, G2, and G4.


