Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Trajectory reconfiguration for time delay reduction in the case of unexpected obstacles: application to 4-mecanum wheeled mobile robots (4-MWMR) for industrial purposes

Samia MELLAH * **Guillaume GRATON** *,**
El-Mostafa EL ADEL * **Mustapha OULADSINE** *
Alain PLANCHAIS ***

* *Aix Marseille Univ., Université de Toulon, CNRS, LIS (UMR 7020), Avenue Escadrille Normandie-Niemen, F-13397 Marseille Cedex 20, France, (e-mail: samia.mellah@lis-lab.fr)*
** *Ecole Centrale Marseille, Technopôle de Château-Gombert, 38 rue Frédéric Joliot-Curie, F-13451 Marseille Cedex 13, France*
*** *ST Microelectronics Rousset, 190 avenue Celestin Coq, ZI - Rousset-Peynier, F-13106 Rousset, France*

Abstract: Nowadays, wheeled mobile robots have a very important role in industrial applications, namely in transportation tasks thanks to their accuracy and rapidity. However, meeting obstacles while executing a mission can cause an important time delay, which is not appreciable in industry where production must be optimal. This paper proposes a new trajectory reconfiguration approach dealing with obstacle generated time delay, applied on four wheeled omnidirectional mobile robots. A strategy is proposed to compensate or minimize the time delay caused by unexpected obstacles, allowing the robot to respect as well as possible its mission planed duration. This strategy is based on updating the velocity reference profile in real time with respect to the environment changing. The aim is to provide to the industrial a support for the robot missions planing and managing, in order to optimize the production.

Keywords: Transportation systems, trajectory reconfiguration, time delay managing, intelligent autonomous robots, robotics, 4-mecanum wheeled mobile robot, dynamic model.

1. INTRODUCTION

In these last few years, wheeled mobile robots (WMRs) perform several tasks in different fields.Their capability of replacing humans to achieve repetitive and hard tasks makes their utilization growing day by day, namely in the industry for material handling and transportation.

Omnidirectional mobile robots are among the most solicited WMRs in the industry this last decade. Their stronger point lies in their wheel structure, allowing them to reach any position in their evolution world without the need to be reoriented.

The concept of omnidirectional wheels is based on a central wheel with free rollers, mounted around the periphery of the wheel. Depending on the rollers type and their inclination angle, several types of omnidirectional wheels are distinguished (Qian et al. [2017], Tătar et al. [2013]). "Mecanum wheel" is one of them with spherical rollers placed at an angle of 45° to the wheel hub circumference (see Fig. 1 and Dickerson and Lapin [1991]). According to Kanjanawanishkul [2016], robots with mecanum wheels are more appropriate for carrying heavy goods in the industrial environment. This work is focused on mecanum wheeled mobile robots, especially the one with 4 wheels: the four-mecanum wheeled mobile robot (4-MWMR), used for transportation tasks in a manufacturing industry.

Fig. 1. Mecanum wheel design.

In order to be efficient and largely smart, WMRs should be able to do self-localization, to sense their surroundings (perception), to generate a path from their initial position to their destination, and to execute it (navigation), in an efficient manner, and without any human intervention.

[⋆] Research leading to these results has received funding from the EU ECSEL Joint Undertaking under grant agreement n 737459 (project Productive4.0) and the partners national funding authorities DGE.

Navigation approaches aim to find an optimal or sub-optimal collision-free path, allowing the robot to reach a target position starting from an initial configuration, taking into account the obstacle avoidance.

Over the past two decades, many efforts have been done by researchers and scientists to respond to the autonomous navigation problematic, and a lot of navigation algorithms and approaches have been proposed. Generally, a navigation is considered to be global or local (Ni et al. [2016]). The global navigation follows an "hierarchical" paradigm and the local one a "reactive" paradigm (Murphy [2000]).

The first paradigm is based on a prior knowledge of the robot environment, presented under a map form. The robot senses its environment using sensors such as cameras and sonars to create a map. Then, it plans the directives needed to reach the next position using this map. Finally it executes them (sense-plan-act). This paradigm was the first to be proposed and used. Among the well-known global navigation approaches, we can cite navigation functions and roadmaps (Hee et al. [2009]). These approaches help to have an optimal path. But they assume the complete knowledge of the environment. In real applications, the environment is not static, and its complete knowledge is impossible. These methods can not be used in dynamic environment (Hee et al. [2009]).

To deal with the global navigation drawbacks, researchers proposed the local navigation theory. Unlike the hierarchical paradigm, the reactive one does not need a prior knowledge of the environment. It is based on the robot embedded sensors data (laser sensors, cameras, sonars, ...). Its principle is based on sensing and acting only (sense-act), no planning step. The navigation is based on planning a list of behaviors (*e.g*: go forward, turn left, ...). Then depending on sensor data, the robot decides which behavior to execute.

Artificial potential fields, fuzzy logic, neural network, genetic algorithm, and ant colony optimization is a non exhaustive list of the local navigation approaches. More methods can be found in Patle et al. [2019] with details. These approaches are perfectly suitable to dynamic environments but they suffer from some major weaknesses like: 1) they can not guarantee an optimal path, and 2) there may be situations where the algorithm can not find the target position (see Khatoon and Ibraheem [2012]).

A new hybrid paradigm called "hybrid deliberative/reactive" paradigm has emerged in 1990's (Murphy [2000]). It aims to combine the local and global navigation advantages and eliminate some of their weaknesses. Under hybrid paradigm, the robot decompose the global task into sub-tasks (mission planning), and based on the environment map, it plans the suitable behaviors to accomplish each sub-task. Then the behaviors are executed using reactive paradigm (plan-sense-act). This hybrid paradigm allows to solve several navigation problems, and continues to be the current area of research.

1.1 Problematic

This work is a part of the European project PRODUCTIVE4.0, with an application in a semiconductor manufacturing company (STMicroelectronics, Rousset-France). Products in the semiconductor fabrication facility (fab) go through about 1200 fabrication steps, where each step is allocated to one or several qualified equipment units. In order to optimize the production, the company plans to use omnidirectional mobile robots to transport batches from one equipment to another in the fab. Contrary to human operators, robots do not get tired of doing repetitive and hard tasks. Added to that, they are accurate and rapid, fact that allows the production optimization.

The fab contains human operators and equipment units. Unfortunately, this restricts the robot working space, and means that the robot may encounter unexpected obstacles while operating.

When it meets an unexpected obstacle, the robot in the fab has two reaction possibilities: 1) Obstacle avoidance if there is enough space, 2) Stopping and waiting for the obstacle to move away. Whatever the robot reaction to bypass the problem, encountering an unexpected obstacle while operating generates a time delay with regard to the duration of the planed mission. This time delay can be important and impacts the global mission execution duration, which is not suitable according to the main purpose of the fab robotizing.

Although the importance of the time notion while executing missions in industrial applications (namely in manufacturing processes), the time delay problem has not been taken into account in the literature proposing autonomous navigation approaches.

1.2 Main contribution

This work deals with the issue of time delay generated by unexpected obstacles.

Depending on the obstacle apparition instant with respect to the remaining distance and time before reaching the final position, plus the time needed to bypass it, an analysis is done to say if the generated delay can be compensated during the global mission remaining time. The aim is to estimate the delay that can be compensated at each instant. This information is very important for the production manager, in order to decide if the mission is supposed to be realisable at time. If not, other solutions must be expected, in order not to effect the production. Then, a new strategy based on the reconfiguration of velocity trajectories in real time, with respect to the environment change is proposed to compensate, where applicable, or to minimize the time delay generated by obstacles.

The outline of this paper is as follows: in the upcoming section, the 4-MWMR mathematical model is described. Then, the trajectory generation, tracking, and obstacle avoidance principle are given to explain the robot navigation principle adopted for simulation. After that, the proposed time delay compensation approach is detailed, followed by an analysis allowing to predict if the delay will be totally compensated or not, and to estimate the non compensable delay in the end of the mission. Next, simulation results are presented for different possible scenarios to show how effective is the proposed approach. Finally, a conclusion and some perspectives are given to conclude the paper.

2. 4-MWMR MATHEMATICAL MODEL

Let's assume that the robot is placed on a plane surface where $((O,\overrightarrow{x},\overrightarrow{y})$ is the inertial reference frame and $(G,\overrightarrow{x_R},\overrightarrow{y_R})$ is a local coordinate frame fixed on the robot at its center of mass and geometric center G, (see Fig. 2).

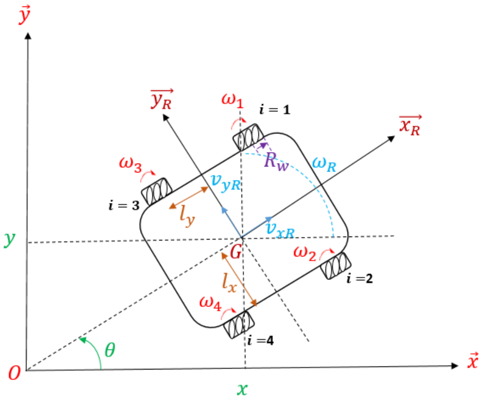

Fig. 2. Geometry of a 4-mecanum wheeled robot.

The following assumptions are made to consider the robot dynamic model:

- Disturbances are neglected due to the robot evolving environment (low velocity, no slippage, no slops, ...).
- Three measurements are available: x and y positions provided by a positioning system, and the rotation angle θ returned by a gyroscope.
- Obstacles are detected using laser distance sensors, allowing to know the distance between the robot and the obstacle from each side.
- Measurement noises are modeled by taking into account the sensors accuracy.

The following notations in Table 1 are used throughout this article (see Fig. 2).

Table 1.

Variable	Description & unit
x, y	Robot position along x and y $axis$ $[m]$
θ	Robot orientation angle $[rad]$
l_x	Half distance between front wheels $[m]$
l_y	Half distance between front and rear wheels $[m]$
R_w	Wheel radius $[m]$
$(\dot{x},\dot{y})/(\dot{x}_R,\dot{y}_R)$	Linear velocities $[m/s]$ in $\left((O,\overrightarrow{x},\overrightarrow{y}\right)$ and $(G,\overrightarrow{x_R},\overrightarrow{y_R})$ respectively
$(\ddot{x},\ddot{y})/(\ddot{x}_R,\ddot{y}_R)$	Linear accelerations $[m/s^2]$ $\left((O,\overrightarrow{x},\overrightarrow{y}\right)$ and $(G,\overrightarrow{x_R},\overrightarrow{y_R})$ respectively
$\dot{\theta}$, $\ddot{\theta}$	Rotational velocity rad/s/acceleration $[rad/s^2]$
I_z	Moment of inertia of the platform $[kg.m^2]$
m	Robot overall mass $[m]$
τ_i	Applied torque to wheel i $[N.m]$

Dynamic model Neglecting the model uncertainties and frictions and denoting $l = l_x + l_y$, the dynamic model is given in $(G,\overrightarrow{x_R},\overrightarrow{y_R})$ by the following equations: (see Sahoo et al. [2018] for more details)

$$\begin{cases} \ddot{x}_R = \dfrac{1}{2mR_w}(\tau_1+\tau_2+\tau_3+\tau_4) \\ \ddot{y}_R = \dfrac{1}{2mR_w}(\tau_1-\tau_2+\tau_3-\tau_4) \\ \ddot{\theta} = \dfrac{l}{2I_zR_w}(\tau_1-\tau_2-\tau_3+\tau_4) \end{cases} \tag{1}$$

This model can be expressed in the inertial reference frame $(O,\overrightarrow{x},\overrightarrow{y})$ using the following transformation matrix: (Vlantis et al. [2016])

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = R(\theta)\begin{bmatrix}\dot{x}_R\\ \dot{y}_R\\ \dot{\theta}\end{bmatrix}, R(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0\\ sin\theta & cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{2}$$

Continuous-state space representation (CSSR) Using (1) and (2), the robot CSSR model is given as follows:

$$\begin{cases} \dot{X} = AX + B_\theta u \\ Y = CX + w \end{cases} \tag{3}$$

where $X = [x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}]^T$, $u = [\tau_1, \tau_2, \tau_3, \tau_4]^T$, w denotes the sensor noises, assumed to be uncorrelated Gaussian white noises with known variances linked to the sensors accuracy. Now, considering $a = 2mR_w$, $b = 2I_zR_w$, $c = cos\theta$, $d = sin\theta$, it follows:

$$A = \begin{bmatrix} 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0 \end{bmatrix}, C = \begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0 \end{bmatrix} \tag{4}$$

$$B_\theta = \begin{bmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ \frac{c-d}{a}&\frac{c+d}{a}&\frac{c-d}{a}&\frac{c+d}{a}\\ \frac{c+d}{a}&\frac{d-c}{a}&\frac{c+d}{a}&\frac{d-c}{a}\\ \frac{l}{b}&\frac{-l}{b}&\frac{-l}{b}&\frac{l}{b} \end{bmatrix} = \begin{bmatrix}0\\ \beta_\theta\end{bmatrix} \tag{5}$$

3. TRAJECTORY GENERATION, TRACKING, & OBSTACLE AVOIDANCE

3.1 Trajectory generation

The aim of this work is to meet the company expectations regarding the robot planed use cases in the fab. A simple hybrid navigation methodology respecting the fab conditions is used here.

The robot in the fab has to accomplish transportation missions from one equipment situated in an initial position denoted P_i to another equipment situated in a target position denoted P_t. To do that, and according to the fab conditions (lack of space, ...), the robot has to pass through intermediate positions (sub-tasks) following straight lines planned based on the environment map (predefined paths), before reaching the final destination. Let's assume that starting from P_i to reach P_t is the global mission, and passing from the intermediate positions are intermediate missions.

To respond to the fabrication process expectations, the robot has to do its global mission by respecting as well as possible the time planned for each destination. So generating trajectories taking into account the time notion and the predefined paths is important. Before generating trajectories, each sub-task feasibility regarding its fixed duration must be verified. For that, for each mission i, the planed duration must be equal or greater than the minimum necessitated time $t_{min}(i)$ to execute it.

This $t_{min}(i)$ is calculated taking into account the distance between the starting and the final positions, the robot recommended velocity v_{rec}, acceleration a_{rec} and deceleration d_{rec} values. These values are fixed to facilitate the human/robots collaboration in the fab. Each intermediate mission defined duration must be equal or greater than $t_{min}(i)$ to be sure that the mission does not induce any time delay in the absence of unexpected obstacles.

The trajectory generation for each intermediate mission is done as follows: given the starting and the final desired positions P_0 and P_f, a velocity trajectory $v(t)$ is planned respecting the mission duration, v_{rec}, a_{rec}, and d_{rec}. Then, from the velocity trajectory, position profile or trajectory $p(t)$ is deduced by time integration. Finally, these two velocity and position trajectories are projected on the x and y axes respecting the orientation angle θ. This generates the velocity and position trajectories with respect to x-axis and y-axis as follows: $v_x(t) = v(t)\cos\theta$, $v_y(t) = v(t)\sin\theta$, $x(t) = p(t)\cos\theta$, and $y(t) = p(t)\sin\theta$

3.2 Trajectory tracking

Using a feedback linearizing control (Isidori and Persis [1996]) to track a predefined robot trajectories, (3) can be controlled by introducing $\mathcal{Y} = \mathcal{X} = [x, y, \theta]^T$. The second derivative $\ddot{\mathcal{X}}$ is a linear expression of u with varying parameters given by (5). By introducing v as $v = \beta_\theta u$, it follows:

$$\ddot{\mathcal{Y}} = \ddot{\mathcal{X}} = \beta_\theta u = v$$

Then, using a pole placement method (Isidori and Persis [1996]), the control law is given by:

$$\begin{cases} v = \sum_{i=0}^{n-1} a_i(Y^i_{ref} - \mathcal{Y}^i) + a_n Y^n_{ref} \\ u = \beta^\dagger_\theta v \end{cases} \tag{6}$$

where $\mathcal{Y}^i$ corresponds to the i^{th} derivative of $\mathcal{Y}$, Y^i_{ref} the reference trajectory and its successive derivatives, n the derivatives order (here $n = 2$), $\beta^\dagger_\theta$ the pseudo-inverse of β_θ such that $\beta_\theta\beta^\dagger_\theta = I$ with I an identity matrix with appropriate dimension. The polynomial coefficients $a_i \in \mathbb{R}$ are chosen such that the poles of polynomial P defined as $P(s) = \sum_{i=0}^{n} a_i s^i$ are with non-positive real parts.

3.3 Obstacle avoidance

Obstacle avoidance is a part of the navigation problem, and it is not the main focus of this work. To deal with the company exigences regarding obstacle overcoming, we propose an approach, simple to implement, aiming to bypass obstacles and to respect the robot evolving environment conditions, before continuing the predefined path tracking. As assumed previously, obstacles are detected using laser distance sensors mounted at each angle of the robot. These sensors provide the distance between the robot and the obstacle from each side. Based on the environment map, the robot estimates how far is the mobile obstacle from the fixed ones (schematized in the map) on each side. The robot is asked to bypass the obstacle by its left side in preference. Based on its estimations, if there is no enough space by the left, it is asked to do it from the right side. If it is not possible no more, it stops and waits to the obstacle moves away from its path. Note that the fab human operators are formed to collaborate with robots. If they notice that an obstacle blocks the robot to continue its task, they move it away.

The obstacle avoidance principle is resumed as follows: detecting an obstacle at a certain distance, the robot reduces its velocity until approaching a minimum distance noted protection distance. If it estimates that it is possible to bypass the obstacle, it moves to the left (right) at a distance allowing him to move sufficiently far away from the obstacle. Then it moves forward based always on its map and the laser sensor measures. When it overcomes the obstacle, it joins its predefined path respecting the updated trajectories, by moving again to the right (left), and all this by following straight lines (see Fig. 4).

It is important to note that many other navigation approaches for obstacle avoidance with better performances can be found in the literature, like it is mentioned in the introduction. The proposed obstacle avoidance here may cause a greater delay comparing with if a better approach is used. Note that a smaller time delay will be more easy to compensate.

4. TIME DELAY MANAGEMENT APPROACH

As mentioned previously, in the trajectory generation step, and for security reasons in the fab, it is preferable not to exceed some recommended values regarding velocity v_{rec}, acceleration a_{rec} and deceleration d_{rec}. In the case of obstacle meeting on the robot path, and to prevent the generated time delay to have a significant impact on the robot global mission, the proposed approach consists of updating and reconfigurating the trajectory references, starting from the trajectory recovery point after the obstacle. In the recalculated trajectories, the robot is permitted to reach its maximum allowed velocity v_{max}, acceleration a_{max} and deceleration d_{max} values. These values are greater than the recommended ones, and they are fixed by the constructor. They respect the robot physical capacities limitation and the fab operator's security.

Once the time delay is compensated, the robot can reduce its velocity and track the first planned trajectories for the rest of its missions. If the delay cannot be completely compensated in the current intermediate mission, it is transmitted to the next mission, which goal is to reduce or compensate it by acting on the velocity trajectory in the current mission with the same manner.

Fig. 3 bellow illustrates the functional schema bloc of the proposed principle.

Depending on obstacle apparition instant, the time needed to bypass it, the remaining distance before reaching the desired final position, and the mission remaining execution

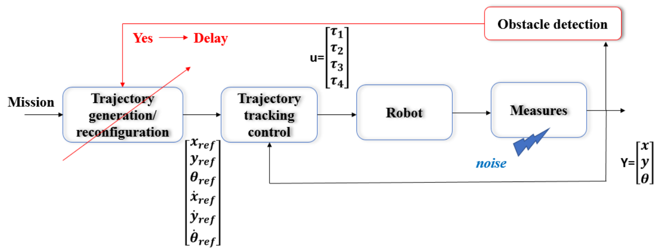

Fig. 3. Functional schema bloc

time, the delay can be compensated or not. Estimating at each instant the delay that can be compensated if an unexpected obstacle is encountered allows to the production manager to well supervise transportation missions. Hence, for each instant t if an obstacle appears, the maximum generated delay that can be compensated is estimated.

5. SIMULATION RESULTS

The robot physical parameters are given by the following values: $m = 390$ kg, $R_w = 0.125$ m, $l_x = 0.1825$ m, $l_y = 0.28$ m, and $I_z = 50$ kg.m^2. The recommended values regarding velocity, acceleration, and deceleration, are given by: $v_{rec} = 1.36\ m/s$, $a_{rec} = d_{rec} = 0.35\ m/s^2$, whereas the maximum allowed values are: $v_{max} = 1.8\ m/s$, $a_{max} = d_{max} = 0.5\ m/s^2$. Note that while avoiding obstacles, the maximum allowed velocity and acceleration/deceleration values are set to the recommended ones.

To illustrate the accuracy of the proposed method, four scenarios are studied. The first one without unexpected obstacles and the three others with the presence of obstacles.

Scenario #1: Suppose that the robot is charged to execute a global mission starting from the initial position $P_i = (0, 0, 0)$ to reach the target position $P_t = (150, 1, 0)$ in 180 seconds, according the following scenario:

- Starting from P_i and reaching position $P_1 = (50, 50, 0)$.
- Starting from P_1 and reaching position $P_2 = (90, 0, 0)$.
- Starting from P_2 and reaching P_t.

Each intermediate mission is planed to be accomplished in 60 s by engineers. The first step is to test the feasibility of this missions in the case with no obstacles. Thereby, for each mission i, $t_{min}(i)$ is calculated, taken into account the recommended values. Indeed, $t_{min}(1) = 58.2$ s, $t_{min}(2) = 53$ s, and $t_{min}(3) = 49.8$ s. The duration of each mission respects its corresponding $t_{min}(i)$. Hence, the global mission is feasible.

Scenario #2: The second scenario consists of executing the first one, but the robot is assumed to meet three obstacles while executing the global mission as follows:

- An obstacle at instant 30 s, bypassed by the left side.
- An obstacle at instant 85 s, cannot be bypassed, so the robot waits for 10 s until the obstacle moves.
- An obstacle at instant 140 s, bypassed by the right side.

Scenario #3: The third scenario is similar to the second one but the waiting time for the second obstacle is 25 s instead of 10 s.

Scenario #4: The last scenario has the first obstacle like in scenario #2, the second obstacle lasts 50 s, and the third obstacle does not exist.

Fig. 4 below illustrates the robot trajectory $y(x)$ with and without obstacles, with their tracking using the control given previously. As can be seen, the robot tracks perfectly the reference trajectory using the proposed control.

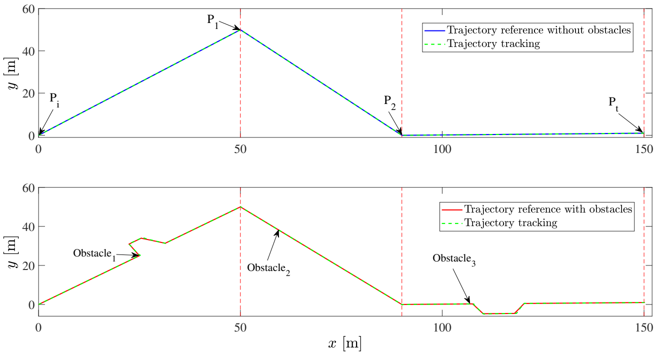

Fig. 4. Global mission scenario with and without unexpected obstacles

Fig. 5 illustrates the updated velocity trajectory references $v(t)$ based on the proposed approach, for the three last scenarios, compared to the one generated for the first scenario, besides on the deduced position profiles $p(t)$.

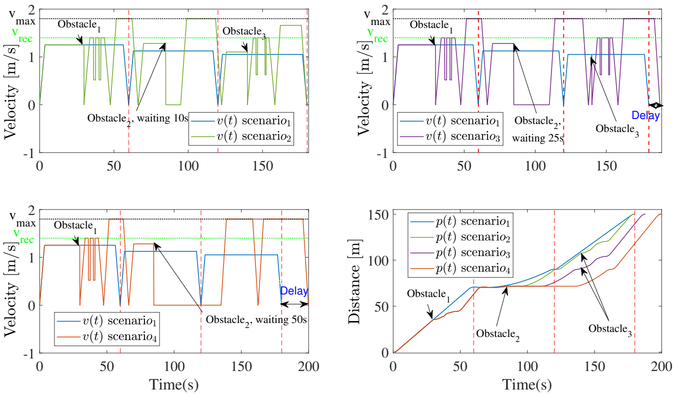

Fig. 5. Velocity reference trajectories and position profiles for all the studied scenarios.

For each scenario, $x(t)$ and $y(t)$ trajectories are obtained from the projection of the trajectory with respect to the x and y axes, as illustrated in Fig. 6.

As it can be noticed in these results, the delay compensation depends on the time taken to bypass the obstacle. In scenario #2, the second obstacle takes only 10 s before moving away. Using the proposed approach, the robot is able to compensate the delay generated by the three obstacles. In the third scenario, the fact of waiting 25 s for the second obstacle to move away added to the two obstacles impacts the global mission duration. The desired final position is reached with a time delay of 7.8 s. Finally in the fourth scenario, the second obstacle takes 50 s to move away. This time delay can not be compensated, and the mission is done with a great delay of 19.8 s even if the third obstacle does not exist.

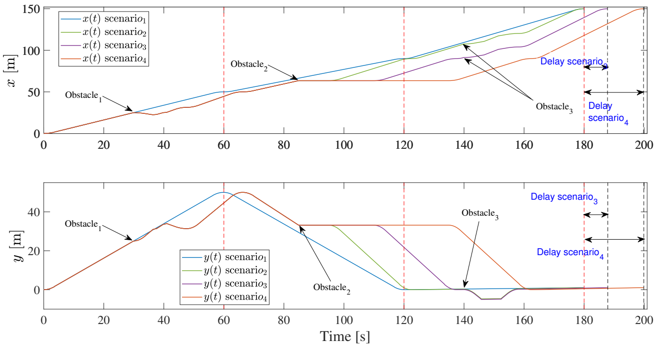

Fig. 6. $x(t)$ and $y(t)$ trajectories of all the studied scenarios.

In order to estimate at each instant the maximum delay that can be compensated, an analysis is done taking into account the obstacle apparition instant, the remaining time and the distance before reaching the desired final destination. This analysis is presented in Fig. 7. A zero delay means that, from this point (the current position), the robot can not compensate any delay if it encounters an obstacle again. A negative delay means that the robot will arrive at destination with a delay, even it will not meet a new obstacle on its path. This analysis is very important for an optimal supervision of the robot.

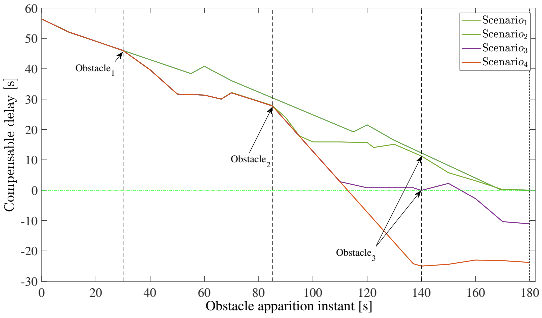

Fig. 7. Delay compensation quantification for the four scenarios.

6. CONCLUSION

Thanks to their accuracy and rapidity, mobile robots are highly solicited in the industrial application domain, namely for achieving transportation tasks. This work aims to respond to a semiconductor manufacturing company, which plans to use a four mecanum mobile robot (4-MWMR) to transport products between equipment tools in the fabrication process. Robots have to execute missions with specified and fixed time duration in order not to generate a delay for the fabrication process. Robots have to work in collaboration with human operators in the semiconductor fabrication facility (fab). Unfortunately, this restricts its working space. Added to that, meeting unexpected obstacles while operating is inevitable in such conditions. Meeting an obstacle may generate a great time delay, delaying the fabrication process, and this is not acceptable.

This paper deals with the time delay generated by mobile and unexpected obstacles, by proposing an approach based on updating and reconfigurating trajectories to track. Without obstacle, the generated trajectories respect the recommended velocity, acceleration, and deceleration values, fixed to facilitate the human/robots collaboration in the fabrication facility (fab). Whereas, when the robot meets unexpected obstacles on its predefined path preventing it to respect its mission fixed duration, the maximum velocity, acceleration and deceleration values are allowed to be reached by the robot, taking into account the fact that human operators are formed to collaborate with robots. This aims to reduce or completely compensate the time delay to best meet the company expectations behind the fab robotizing. Added to that, an analysis allowing to estimate at each instant the maximum delay that can be compensated is proposed. This information is very useful for the production manager aiming to well supervise transportation missions. The obtained simulation results are very promising. This method is planned to be validated experimentally sooner.

REFERENCES

Dickerson, S.L. and Lapin, B.D. (1991). Control of an omni-directional robotic vehicle with mecanum wheels. In *NTC '91 - National Telesystems Conference Proceedings*, 323–328.

Hee, L.G., Ang, J., and Marcelo, H. (2009). *Mobile Robots Navigation, Mapping, and Localization Part I*, 1072–1079. doi:doi:10.4018/978-1-59904-849-9.ch158.

Isidori, A. and Persis, C.D. (1996). Feedback linearization of nonlinear systems. *Control Systems, Robotics ans Automation,Encyclopedia of Life Support Systems*, Vol. XII.

Kanjanawanishkul, K. (2016). Omnidirectional wheeled mobile robots: wheel types and practical applications. In *International Journal of Advanced Mechatronic Systems*, volume 6.

Khatoon, S. and Ibraheem (2012). Autonomous mobile robot navigation by combining local and global techniques. *International Journal of Computer Applications*, 37(3), 0975–8887.

Murphy, R.R. (2000). *Introduction to AI Robotics*. MIT Press, Cambridge, MA, USA, 1st edition.

Ni, J., Wu, L., Fan, X., and Yang, S.X. (2016). Bioinspired intelligent algorithm and its applications for mobile robot control: a survey. *Computational intelligence and neuroscience*.

Patle, B., Pandey, A., Parhi, D., Jagadeesh, A., et al. (2019). A review: On path planning strategies for navigation of mobile robot. *Defence Technology*.

Qian, J., Zi, B., Wang, D., Ma, Y., and Zhang, D. (2017). The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system. *Sensors (Basel)*.

Sahoo, S., Chiddarwar, S., and Alakshendra, V. (2018). Intuitive dynamic modeling and flatness-based nonlinear control of a mobile robot. *SIMULATION*.

Tătar, Olimpiu, M., Cirebea, C., and Mândru, D. (2013). Structures of the omnidirectional robots with swedish wheels. In *Mechatronic Systems and Materials IV*, volume 198, 132–137.

Vlantis, P., Bechlioulis, C.P., Karras, G., Fourlas, G., and Kyriakopoulos, K.J. (2016). Fault tolerant control for omni-directional mobile platforms with 4 mecanum wheels. In *IEEE ICRA*.