Trajectory tracking and time delay management of 4-mecanum wheeled mobile robots (4-MWMR)

Samia Mellah, Guillaume Graton, El-Mostafa El-Adel, Mustapha Ouladsine, Alain Planchais

Trajectory tracking and time delay management of 4-mecanum wheeled mobile robots (4-MWMR)

Samia MELLAH[1], Guillaume GRATON[1,2], El-Mostafa EL-ADEL[1], Mustapha OULADSINE[1], and Alain PLANCHAIS[3]

[1] Aix Marseille Univ, Universit de Toulon, CNRS, LIS (UMR 7020), Avenue Escadrille Normandie-Niemen, F-13397 Marseille Cedex 20, France
samia.mellah@lis-lab.fr,
[2] Ecole Centrale Marseille, Technopole de Chateau- Gombert, 38 rue Frederic Joliot-Curie, F-13451 Marseille Cedex 13, France
[3] 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 deals with the time delay management, the trajectory generation and the tracking problem applied on four wheeled omnidirectional mobile robots. A strategy is proposed to minimize or compensate the time delay caused by obstacles. The approach is done by updating the reference trajectory. This update helps to track the trajectory in real time, a new control law based on the feedback linearization control theory is synthesized to track perfectly generated or updated trajectories.

Keywords: Vision and robotics, 4-mecanum wheeled mobile robot, kinematic model, feedback linearization control, trajectory generation and reconfiguration.

1 Introduction

Mobile robots, thanks to their capability of replacing humans to achieve hard, tedious, and risky activities, become a subject of high interest in the industrial application domain as well as in the scientific research area.

Mainly, three types of mobile robots (MR) are distinguished: unicycle, car-like, and omnidirectional MR. Unicycle and car-like MR are called "nonholonomic robots", while omnidirectional ones are called "holonomic robots". The difference lies in the number of degrees of freedom (DOFs). Nonholonomic robots are limited on only two DOFs, while the holonomic ones are characterized by three DOFs in the horizontal plane [1].

Unlike nonholonomic mobile robots, each wheel on the omnidirectional mobile platform is actuated by a separate DC motor. This property allows them to reach any position on their plane of motion without the need of any angle reorientation and makes them more adaptable for transportation, especially in tight spaces, where maneuvering to change the moving direction becomes complicated.

Omnidirectionaly is the ability to move instantaneously in any direction from any configuration without having to change the orientation angle. This property is obtained thanks to the omnidirectional wheel structure.

Omnidirectional wheels are based on the concept of a central wheel with free rollers placed at an angle around the periphery of the wheel. Several different omnidirectional wheel mechanisms are distinguished depending on roller types and inclination angles [2], [3]. Mecanum wheels are one type of omnidirectional wheels among the others. They were invented by Bengt Ilon in 1973 [4] and are characterized by spherical rollers angled at 45° to their hub circumference.

Mecanum wheels are more appropriate for carrying heavy goods in the industrial environment [5]. Fig. 1 illustrates the wheel structure. More information and illustrations can be found in [6]. In this work, we are interested in mecanum wheeled mobile robot, especially the one with 4 wheels (4-MWMR).

Fig. 1. Mecanum wheel design

1.1 Problem description

This work takes place in the project PRODUCTIVE4.0, in collaboration with STMicroelectronics (Rousset, France). STMicroelectronics is an international company specialized on semiconductor production. Products in the semiconductor manufacturing fabrication facility (fab) go through about 1200 fabrication steps, where each step is allocated to one or several qualified entities or equipment.

In order to optimize the production, STMicroelectronics Rousset plans to use omnidirectional mobile robots to transport batches from one equipment to another.

Using robots instead of human operators for transportation allows to accelerate the fabrication process. Also, independently of their rapidity, mobile robots, contrary to human operators, do not get tired of doing repetitive tasks, fact that can optimize the production.

The production fab contains humans and equipment units. Unfortunately, this restricts the robot working space, and makes meeting obstacle inevitable. Meeting an obstacle during the mission execution may cause an important time delay, which is not acceptable regarding the main purpose of the manufacturing fabrication facility (fab) robotizing. The robot time delay managing is the main focus of this paper.

1.2 Some literature related works

Due to the important role that occupy mobile robots in a lot of domains nowadays, especially the omnidirectional ones, many efforts have been done by researchers regarding the omnidirectional mobile robot kinematic and dynamic modeling, their control, and also the fault diagnosis in order to improve more and more their autonomy and intelligence. This section gives some relative research works that can be found in the literature.

Concerning the 4-MWMR modeling, authors in [7] propose a dynamic model based on bond graph theory, while authors in [1], [8] and references therein give an approach based on the Lagrangian theory to propose two other versions of dynamic model. In [9], the kinematic model of the same type of robot is developed. Other kinematic and dynamic modeling methods can be found in [10], [11] and references therein.

For a three wheeled omnimove mobile robot control, a feedback control law is proposed in [12], while authors in [13] propose a sliding control based on fuzzy logic theory. An other control law based on inverse Input-Output linearization is proposed for the same robot in [14].

For a 4-MWMR, robust control is proposed in [8] and an adaptive robust control is proposed in [15], while in [16], an integral sliding mode controller is synthesized.

In [17], a fault diagnosis method based on extended Kalman filters and particle filters is proposed to detect and isolate faults in a three wheeled omnidirectional mobile robot. Authors in [1] propose a control law allowing to deal with wheel faults. In [18], a 4-MWMR fault diagnosis problem is studied. A method based on extended Kalman filter is proposed to detect and isolate both sensor and actuator additive faults.

This work deals with the omnidirectional mobile robot trajectory reconfiguration and tracking problem. The main contribution focuses on proposing a way for managing the time delay caused by obstacles. For the best of our knowledge, no work has been published regarding the 4-MWMR trajectory reconfiguration for time delay management.

The remaining of this paper is organized as follows: the omnidirectional mobile robot mathematical model is described in section 2. Section 3 contains the control theory. The trajectory reconfiguration to deal with time delay caused by obstacles is explained in section 4. In section 5 some results of simulated scenarios are exposed to show the effectiveness of the proposed approach, and finally a conclusion and some perspectives are done in the last section 6.

2 4-MWMR mathematical model

Fig. 2 shows a 4-MWMR geometry. To consider the kinematic model, it is assumed 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, which coincides with its geometric center G.

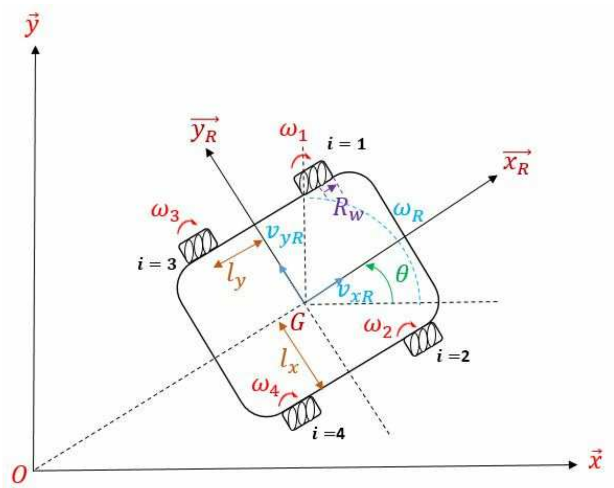

Fig. 2. Geometry of a 4-mecanum wheeled robot.

2.1 Nomenclature

The following notations presented in Table 1 will be used throughout this article. Variables are illustrated in Fig. 2.

2.2 Forward kinematic model

The robot is assumed to be equipped with a gyroscope to provide its orientation angle θ, with a positioning system (like beacon, WIFI, GPS, ...) to provide its position (x, y), and with sonars to detect obstacles. The forward kinematic model is given in the robot local frame $(G, \overrightarrow{x_R}, \overrightarrow{y_R})$ by the following equation [1]

Variable	Description & unit
x, y	Robot position along x-axes and y-axes (m).
θ	Robot orientation angle (rad).
l_x	Half distance between front or rear wheels (m).
l_y	Half distance between front and rear wheels (m).
$l = l_x + l_y$	
R_w	Wheel radius (m).
$\omega_i = \dot{\theta}_i$	Wheel angular velocities (rad/s).

Table 1. Nomenclature

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix} = \frac{R_w}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ \frac{-1}{l} & \frac{1}{l} & \frac{-1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{1}$$

This model can be expressed in the inertial reference frame $(O, \overrightarrow{x}, \overrightarrow{y})$ using the following transformation matrix (more details are given in [1]):

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

Finally, using (1) and (2), the forward kinematic model is done in the inertial reference frame by the following continuous state-space representation form:

$$\begin{cases} \dot{X} = B(\theta)u \\ Y = CX + e \end{cases} \tag{3}$$

where $C = I_3$ is a (3×3) identity matrix, e denotes the measurement noise or sensor error, it is assumed to be non-correlated gaussian white noise with known covariance matrix W, $X = [x, y, \theta]^T$, $u = [\omega_1, \omega_2, \omega_3, \omega_4]^T$, and

$$B(\theta) = \frac{R_w}{4} \begin{bmatrix} \cos\theta + \sin\theta & \cos\theta - \sin\theta & \cos\theta - \sin\theta & \cos\theta + \sin\theta \\ \sin\theta - \cos\theta & \sin\theta + \cos\theta & sin\theta + \cos\theta & \sin\theta - \cos\theta \\ \frac{-1}{l} & \frac{1}{l} & \frac{-1}{l} & \frac{1}{l} \end{bmatrix} \tag{4}$$

3 Feedback linearizing control

The state-space representation model given in (3) can be controlled using a feedback linearizing control [19]. In the studied case and according to (3), the first state derivative $\dot{X}$ is a linear expression of u with varying parameters given by (4). By introducing v as $v = B(Y_\theta)u$, with $B(Y_\theta)$ has the same expression as $B(\theta)$ except that θ is replaced by its measurement Y_θ, it follows:

$$\dot{X} = v$$

Then, using a pole placement method, the control law is given by: [19]

$$\begin{cases} v = a_0(Y_{ref} - Y) + a_1(\dot{Y}_{ref} - \dot{Y}) + ... + a_{n-1}(Y_{ref}^{n-1} - Y^{n-1}) + a_n Y_{ref}^n \\ u = B(Y_\theta)^\dagger v \end{cases} \quad (5)$$

where Y^n corresponds to the n^{th} derivative of Y (here $n = 1$), $(Y_{ref}, ..., Y_{ref}^n)$ the reference trajectory and its successive derivatives, $B(Y_\theta)^\dagger$ the pseudo-inverse of $B(Y_\theta)$ such that $B(Y_\theta)B(Y_\theta)^\dagger = I$, the identity matrix. Coefficients $a_0, ..., a_n$ represent the polynomial coefficients which have to be chosen such that the poles of polynomial

$$a_0 + a_1 s + ... + a_{n-1}s^{n-1} + a_n s^n$$

are with negative real parts. The system is represented by the functional scheme illustrated in Fig. 3.

When a mission is given to the robot (initial and final position, distance, and mission duration), the reference trajectory block (see Fig. 3) plans a trajectories with respect to the mission properties.

The control block then calculates in real time the control allowing the robot to track its planned trajectory. Sensors described in the beginning of subsection 2.2 provide the needed measurements to the robot in order to update its trajectory and its control law. Using the feedback linearization, the theory allows to track perfectly the desired trajectories. Results are illustrated in section 5.

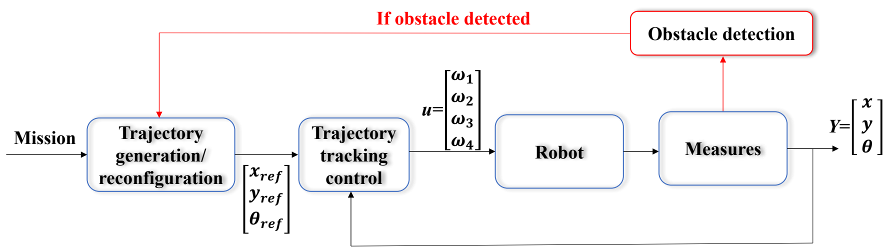

Fig. 3. System functional scheme.

4 Trajectory generation and reconfiguration

The robot is controlled to accomplish a number of small missions between several production areas. The summation of these elementary missions gives the global mission for the robot. Before generating trajectories with respect to $x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}$, missions have to respect the robot physical capacities limitation regarding to accelerations and velocities.

For that, and according to mission i, the minimal necessitated time $t_{min}(i)$ to execute the mission i by the robot is calculated according to the distance, the maximum velocity and the maximum acceleration and deceleration. Each mission duration must be greater than $t_{min}(i)$ to be sure that the mission does not induce any time delay.

According to the time given for the mission, the maximum acceleration and deceleration and the distance to go, the mission velocity is updated for the elementary mission.

The trajectory planning for each elementary mission is done as follows: given the initial and the final desired positions Z_0 and Z_f, a velocity profile is planned respecting the mission duration, the maximum acceleration and deceleration, and the maximum velocity as explained here above. Then, from the velocity profile, position profile is deduced by time integration. Finally, these two velocity and position profiles are projected on the x-axes and y-axes respecting the orientation angle θ. This generates the velocity and position trajectories with respect to x-axes and y-axes as follows: $v_x = v \cos\theta$, and $v_y = v \sin\theta$. Fig. 4 shows an example of velocity and position generated profiles to move from point $(0, 0)$ for a distance of 40 *meters* in a duration of 35 *seconds*.

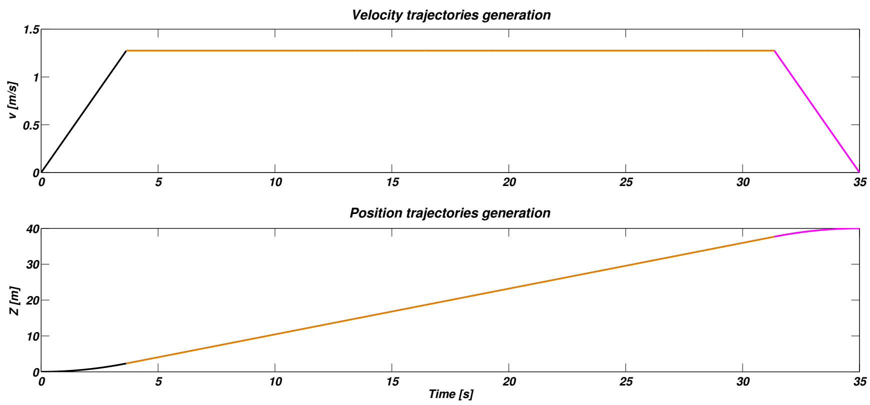

Fig. 4. Velocity and position trajectories example

When a temporary and unexpected obstacle (like human or another robot) is detected on the robot trajectory by the sonars, the robot stops and waits for the obstacle to move away. This causes a time delay with respect to the planned elementary mission.

To prevent the time delay to have a significant impact on the robot global mission, trajectory references are reconfigured. In the recalculated trajectories, robot is allowed to reach its maximum allowed velocity.

Once the time delay is compensated, 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 mission, the delay is transmitted to the next mission, which goal is to reduce or compensate the delay by acting on the velocity.

5 Simulation results

To explain and to show the effectiveness of the proposed method, a scenario is proposed without and with obstacles and obtained simulation results are exposed in this section.

The scenario consists in doing three consecutive and elementary missions (see Fig. 5):

$Mission_1$ moving from point $(0, 0)$ with respect to x-axes for a distance of 40 meters in 35 seconds

$Mission_2$ from point $(40, 0)$, moving with respect to y-axes for a distance of 20 meters in 37 seconds

$Mission_3$ form point $(40, 20)$ moving with respect to both of x-axes and y-axes for 40 meters in 52 seconds.

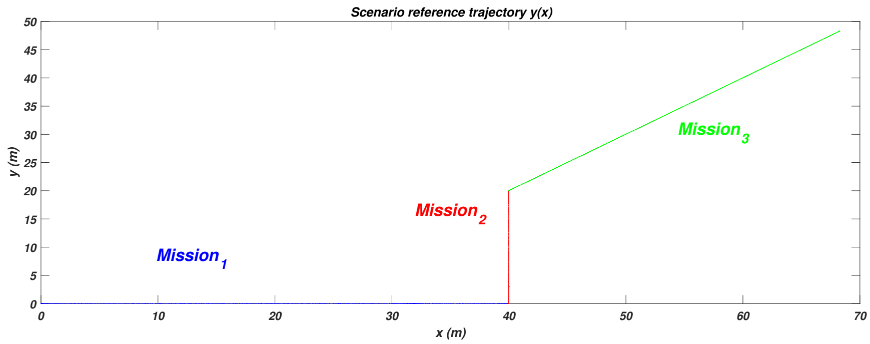

Fig. 5. Global mission scenario.

Respecting the robot physical limitations, and for security reasons, the robot maximal acceleration a_{max}, its maximal deceleration d_{max} and its maximal velocity v_{max} in the fab are limited to: $a_{max} = d_{max} = 0.5\ m/s^2$, $v_{max} = 1.8\ m/s$. According to the x-axes and the y-axes, the acceleration or deceleration and the velocity defined before are projected using the orientation angle θ.

To complete the simulation background, measurements are observed with noises which are non-correlated gaussian white noises with the following covariance:

$$W = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.001 \end{pmatrix} \quad (6)$$

For the first simulation, no obstacle is on the way of the robot. Fig. 6 illustrates the reference temporal velocity trajectory for each mission, while Fig.

7 shows in blue the reference temporal trajectories with respect to x-axes, and y-axes for each mission, and the robot trajectories in red.

By introducing the error $e_i(k)$, for each time instant k, as the difference between the reference and the measurement,

$$e_i(k) = y_{i,ref}(k) - y_i(k),$$

where $i = 1$ for the x-axes, $i = 2$ for the y-axes and $i = 3$ for the angle θ. The absolute error mean m_i and the error standard deviation s_i are defined by:

$$m_i = \frac{1}{N} \sum_{k=1}^{N} |e_i(k)|$$

$$s_i = \frac{1}{N-1} \sum_{k=1}^{N} (|e_i(k)| - m_i)^2$$

As it can be clearly seen in Fig. 7, the feedback linearization control allows to track perfectly reference trajectories with respect to x-axes, y-axes. In fact, the absolute error mean and the error standard deviation given in Table 2 are small, less than 10^{-2}.

i	Variable	m_i	s_i
1	e_x	10^{-2}	$7.6\ 10^{-3}$
2	e_y	$9\ 10^{-3}$	$7.1\ 10^{-3}$
3	e_θ	$8\ 10^{-3}$	$6.1\ 10^{-3}$

Table 2. Absolute error mean and the error standard deviation

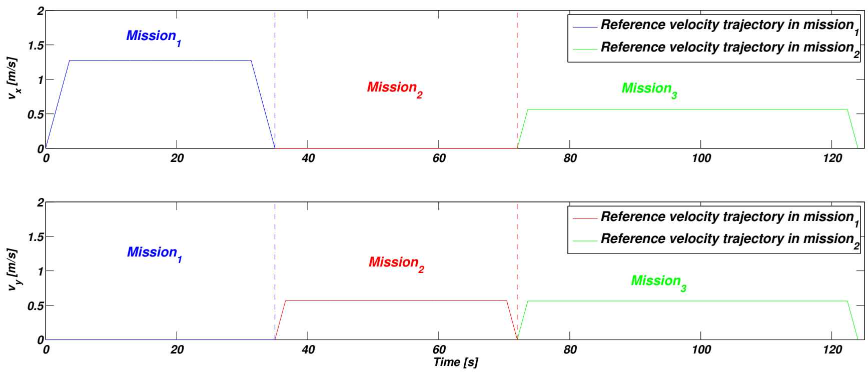

Fig. 6. Temporal velocity reference without time delay.

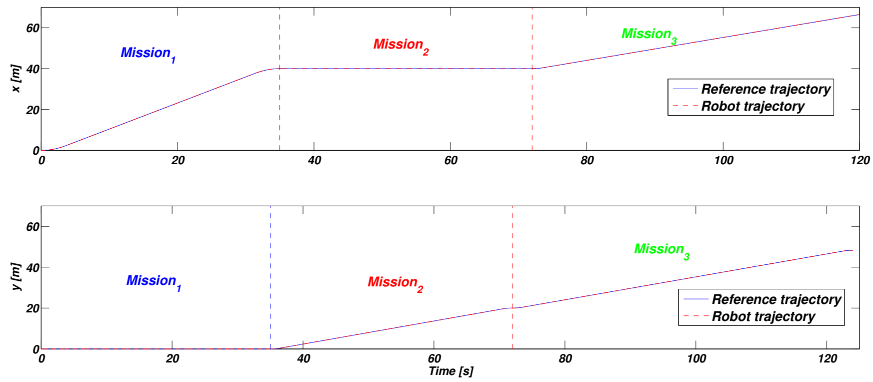

Fig. 7. Reference position trajectories tracking

For the second simulation, the robot has to execute the same scenario (see Fig. 5), but it meets an obstacle while it is executing the first mission at time 28 s, and is going to have a time delay of 5 s. Then, an other obstacle at time 66 s while it is executing the second mission, that create a time delay of 10 s.

The trajectory reconfiguration bloc (see Fig. 3) updates the reference position and velocity trajectories to compensate the time delay caused by these obstacles. Results are shown in figures 8 to 11 below.

Fig. 8 shows planned position trajectories before obstacles and the updated ones after meeting obstacles for the first and the second missions.

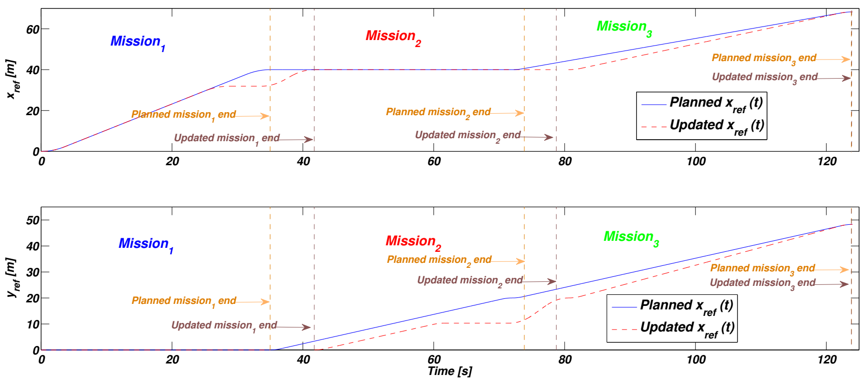

Fig. 8. Planned and updated reference position trajectories

Fig. 9 illustrates the updated velocity trajectories after detecting obstacles.

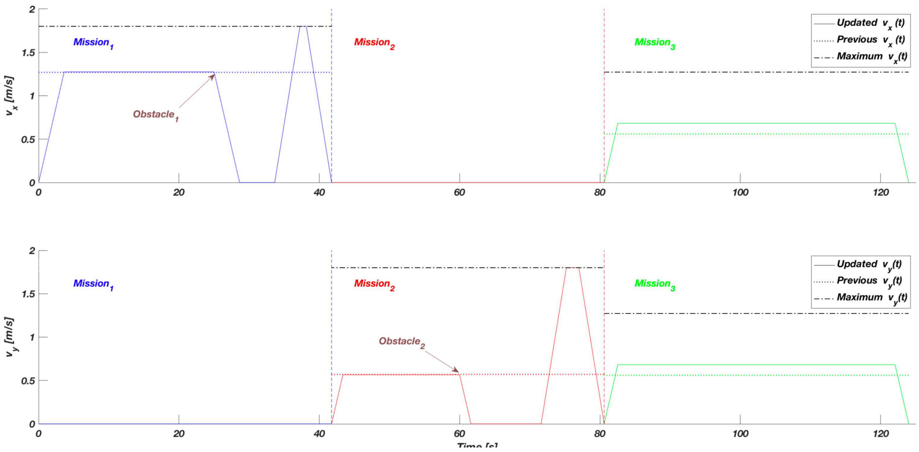

Fig. 9. Updated reference velocity trajectories

In order to illustrate the effectiveness of the proposed control, the updated reference position trajectories are plotted in the same figure with the robot trajectories. Tracking results are illustrated in Fig. 10 below.

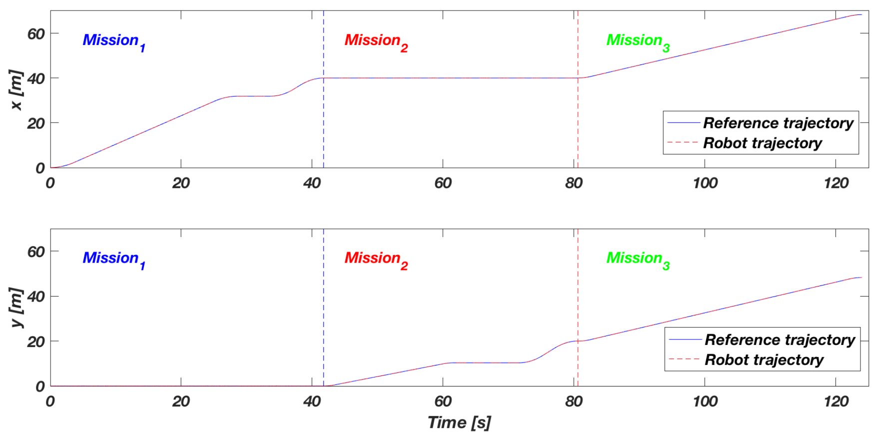

Fig. 10. Updated position trajectories tracking

The temporal delay evolution is represented in the last Fig. 11 to show how the proposed method compensates the time delay. As can be seen in this figure, the delay caused by $Obstacle_1$ could not be compensated during $Mission_1$, and the robot could not reach the position $(40, 0)$ in 35 seconds. This delay impacted the beginning of $Mission_2$, but it has rapidly been compensated by the robot in $Mission_2$. An other obstacle $Obstacle_2$ appears before the complete execution of $Mission_2$, and causes an other delay impacting the beginning of $Mission_3$. Finally, this second delay is compensated during $Mission_3$ and the global mission is executed on time.

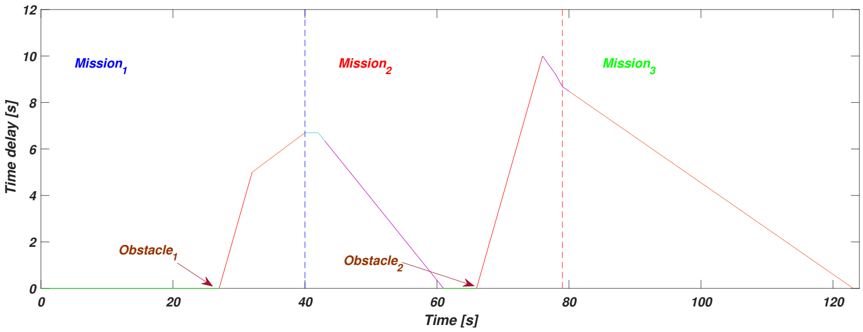

Fig. 11. Delay temporal evolution

6 Conclusion

In this paper, a four-mecanum wheeled mobile robot time delay managing, trajectory generation/reconfiguration, and trajectory tracking problems are treated. Because of obstacles met in the way of robots, especially in tight places where humans and robots collaborate, robot can have a considerable time delay preventing it to do its charged mission in time. However, mission duration must be respected especially in industry where time delay can cause important consequences for production.

A good strategy to compensate or eliminate obstacle generated time delay is proposed in this work. This strategy consists on updating planned trajectories with respect to the delay while respecting robot physical limitations, then using feedback linearized theory, a control law is synthesized to track perfectly the concerned trajectories.

In this work, only one robot is considered. The obtained results are satisfying and convincing. In future studies, we plane to use the same strategy for a fleet of robots. The fleet will be charged to do a global transportation mission composed with many elementary missions destined to many robots. The goal is to respect the global mission. Each robot must communicate its time delay to the others if it can not compensate it by its own. Planned trajectories for the rest of robots must be updated in real time so that to try to compensate this time delay.

References

1. Vlantis, P., Bechlioulis, C. P. & G. Karras and G. Fourlas and, K. J. K. *Fault tolerant control for omni-directional mobile platforms with 4 mecanum wheels* in *IEEE ICRA* (May 2016).
2. Tătar, Olimpiu, M., Cirebea, C. & Mândru, D. *Structures of the Omnidirectional Robots with Swedish Wheels* in *Mechatronic Systems and Materials IV* **198** (May 2013), 132–137.
3. Qian, J., Zi, B., Wang, D., Ma, Y. & Zhang, D. The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System. *in Sensors (Basel)* (Sept. 2017).
4. Han, K., Kim, H. & Lee, J. S. *The sources of position errors of omnidirectional mobile robot with Mecanum wheel* in *2010 IEEE International Conference on Systems, Man and Cybernetics* (Oct. 2010).
5. Kanjanawanishkul, K. Omnidirectional wheeled mobile robots: wheel types and practical applications. **6** (Feb. 2016).
6. Dickerson, S. L. & Lapin, B. D. *Control of an omni-directional robotic vehicle with Mecanum wheels* in *NTC '91 - National Telesystems Conference Proceedings* (Mar. 1991), 323–328.
7. Sahoo, S. R., Chiddarwar, S. S. & Alakshendra, V. Intuitive dynamic modeling and flatness-based nonlinear control of a mobile robot. *in SIMULATION* (2018).
8. Alakshendra, V., Chiddarwar, S. S., Nikhade, G., Jha, A. & Sahoo, S. R. *Robust Trajectory Tracking Control for an Omnidirectional Mobile Robot* in *Proceedings of the Advances in Robotics* (ACM, New York, NY, USA, 2017).
9. Taheri, H., Qiao, B. & Ghaeminezhad, N. Kinematic Model of a Four Mecanum Wheeled Mobile Robot. *International Journal of Computer Applications,* **113,** 6–9 (Mar. 2015).
10. Muir, P. & Neuman, C. *Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot* in *Proceedings. 1987 IEEE International Conference on Robotics and Automation* **4** (Mar. 1987), 1772–1778.
11. Conceicao, A. S., Moreira, A. P. & Costa, P. J. Practical Approach of Modeling and Parameters Estimation for Omnidirectional Mobile Robots. *IEEE/ASME Transactions on Mechatronics* **14,** 377–381 (June 2009).
12. Watanabe, K., Shiraishi, Y., Tzafestas, S. G., Tang, J. & Fukuda, T. Feedback Control of an Omnidirectional Autonomous Platform for Mobile Service Robots. *Journal of Intelligent and Robotic Systems* **22,** 315–330. ISSN: 1573-0409 (July 1998).
13. Tsai, C.-C. & Wu, H.-L. Nonsingular terminal sliding control using fuzzy wavelet networks for Mecanum wheeled omni-directional vehicles. *International Conference on Fuzzy Systems,* 1–6 (2010).
14. Li, X. & Zell, A. in *Motion Control of an Omnidirectional Mobile Robot, In Informatics in Control, Automation and Robotics* 181–193 (Springer Berlin Heidelberg, 2009).

15. Alakshendra, V. & Chiddarwar, S. S. Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties. *Nonlinear Dynamics* **87,** 2147–2169 (2017).
16. Hung, N., Viet, T. D., Im, J. S., Kim, H. K. & Kim, S. B. Motion control of an omnidirectional mobile platform for trajectory tracking using an integral sliding mode controller. *International Journal of Control, Automation and Systems* **8,** 1221–1231 (Dec. 2010).
17. Valdivieso, C. & Cipriano, A. *Fault detection and isolation system design for omnidirectional soccer-playing robots* in *IEEE Conference on Computer Aided Control System Design, IEEE International Conference on Control Applications,IEEE International Symposium on Intelligent Control* (2006).
18. Mellah, S., Graton, G., El-Adel, E., Ouladsine, M. & Planchais, A. Detection isolation of sensor and actuator additive faults in a 4-mecanum wheeled mobile robot (4-MWMR), In International Conference on Control, Automation and Diagnosis (2019).
19. Isidori, A. & Persis, C. D. Feedback linearization of nonlinear systems. *Control Systems, Robotics ans Automation - Vol. XII.*