Control of Mobile Robot Using Fractional Order PI$^\lambda$D$^\mu$ Controller

Younes Boucetta[*1†], **Redouane Ayad**[**2], and **Zoubir Ahmed-Foitih**[*3], Non-members

ABSTRACT

The ideal example for studying complex systems with non-holonomic constraints is mobile-wheeled robots. In this article, we study the problem of trajectory tracking of mobile robot unicycle type. To resolve this type of problem a fractional-order control technique is proposed. The main objective of this control method is to design a robust tracking controller to eliminate disturbances. The mathematical model of the mobile robot taking explicitly into account their dynamics is used to calculate the desired linear and angular velocities. To adjust the controller optimal parameters the particle swarm optimization (PSO) algorithm is utilized. The simulation results allowed us to demonstrate the efficiency and robustness of this technique on non-holonomic systems in the form of chained chains.

Keywords: Unicycle Robot, Fractional Order, Trajectory Tracking, Control, PSO

1. INTRODUCTION

Nowadays, Wheeled Mobile Robots have found many applications in industry, transportation and other fields. Mobile robotics is becoming an important element in our everyday life; its rapid development has made it possible to use it in various fields. The big problem of mobile robotics is how to control the speed of the trajectory of movement of the robot in order to carry out a human task.

The mobile robot type unicycle is one of the robots covered in the studies of the trajectory control of the mobile robots because its kinematic and dynamic models are simple to exploit. In the last decade, many researchers propose trajectory-tracking control for a mobile robot. Authors in [1–3] propose a backstepping controller design for trajectory tracking of unicycle-type mobile based on the Lyapunov theorem. In [4] a Linear Quadratic Controller (LQR) type based on the linearized model of the robot is proposed and simulation studies were conducted to determine its performance. A fuzzy PID controller is elaborated for trajectory tracking of a mobile robot in [5–7].

Other researchers are interested in applying another type of control like sliding mode control (SMC) [8–10], or adaptive control [10–12]. To control the movement of the robot a combination of adaptive model reference control and gain programming is developed in [11]. An adaptive controller based on the dynamic model calculates the torques of the robot actuators to obtain the required velocities in the presence of the unknown dynamics in the dynamic model of the robot, as studied in [12]. In [13], the stability of an adaptive trajectory-tracking controller based on robot dynamics is proved by the Lyapunov stability theorem with experimental results. The dynamic controller is able to update the estimated parameters, which are directly related to physical parameters of the robot. In order to overcome trajectory-tracking problems, in the presence of uncertainties, an adaptive nonlinear control of a wheeled mobile robot approach is proposed in [14]. to design control algorithms for trajectory tracking of mobile robots, A methodology based on linear interpolation is used in [15]. In [16] the optimal PID controller is used to control the velocity of a unicycle type of mobile robot. Authors in [17] used an Evolutionary Algorithm to modify the parameters of the FOPID (Fractional Order PID) controller for the trajectory tracking problem of a wheeled mobile robot.

In this article, we explain how the problem of trajectory tracking of a mobile robot is resolved using a fractional order PID controller. The control of the unicycle robot is solved by considering its kinematics and dynamics model. Two controllers are presented in this document, the first controller is used to adjust the internal parameters of the robot, which are linear and angular velocities; these two parameters are obtained from a given trajectory based on the kinematic model of the mobile robot. The received parameters (the linear and angular velocities) from the first controller are input parameters for the second controller that uses the fractional order PID to improve and generate the reference linear and angular velocities to command the motor of the mobile robot by following the desired trajectory.

The optimal FOPID controller parameters, which are used to adjust the speed of the wheels of mobile robot, are calculated using the Particle Swarm Optimization algorithm (PSO).

Manuscript received on February 11, 2018 ; revised on June 27, 2019 ; accepted on August 29, 2019.

[*]The authors are with LEPESA Laboratory, University of Sciences and Technology of Oran Mohammed Boudiaf (USTO-MB), Algerie, E-mail: younes.boucetta@univ-usto.dz[1] and zfoitih@yahoo.fr[3]

[**]The author is with Hassiba Benbouali University of Chlef, Algerie, E-mail: r.ayad@univ-chlef.dz[2]

[†]Corresponding author.

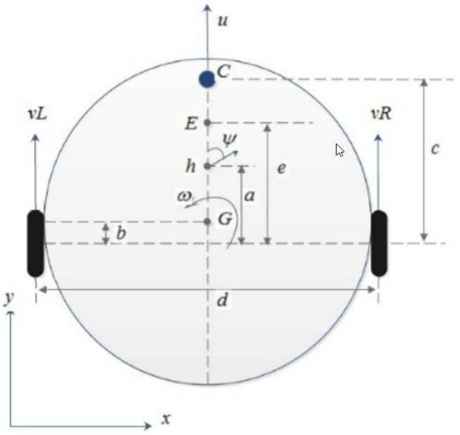


Fig.1: *Mobile Robot Unicycle.*

To evaluate the effectiveness of the proposed controller, simulations are performed using the FOMCON which is a new fractional-order modeling and control toolbox for MATLAB. It offers a set of tools in the field of fractional-order control. A fractional order transfer function is usually approximated to an integer transfer function with a similar behavior, to be easily implemented or simulated. Many researchers proposed several ways to find such approximation. In [18] and [19] a good review of the fractional order approximations can be found. Many approximations are implemented in FOMCON toolbox [20], in this work we use the Oustaloup continuous approximation [18].

2. MATHEMATICAL MODEL OF THE UNICYCLE ROBOT

A robot that moves in a 2D with an advance speed and a zero instantaneous lateral movement generally is a robot of the unicycle type. In other words, it is a nonholonomic system. The modeling of unicycle robots consists of studying their kinematics and their dynamics. Kinematic modeling makes it possible to study the trajectories followed by mobile robots when subjected to controlled speeds and dynamic modeling supplements the kinematics taking into account control forces and intrinsic friction.

Fig. 1 represents the mobile robot. G is the center of mass of the robot, C is the position of the castor wheel, E is the location of a tool onboard the robot, h is the point of interest with coordinates x and y in the plane XY, ψ is the robot orientation, and a is the distance between the point of interest and the central point of virtual axis linking the traction wheels (point B), u and ω are the linear and angular velocities of the robot.

2.1 Kinematic Modeling

The kinematic controller design is based on the kinematic model of the robot, assuming that the perturbation term of Eq. (1) is a null vector. The kinematic model of the robot is given by

$$\begin{aligned} \dot{x} &= u\cos\alpha - a\omega\sin\alpha \\ \dot{y} &= u\sin\alpha - a\omega\cos\alpha \\ \dot{\alpha} &= \omega \end{aligned} \tag{1}$$

whose output are the coordinates of the point of interest, that meaning $h = [x \quad y]^T$

$$H = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\alpha & -a\sin\alpha \\ \sin\alpha & a\cos\alpha \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = A \begin{bmatrix} u \\ \omega \end{bmatrix} \tag{2}$$

with

$$A = \begin{bmatrix} \cos\alpha & -a\sin\alpha \\ \sin\alpha & a\cos\alpha \end{bmatrix} \tag{3}$$

whose inverse is

$$A^{-1} = \begin{bmatrix} \cos\alpha & -a\sin\alpha \\ -\frac{1}{a}\sin\alpha & \frac{1}{a}\cos\alpha \end{bmatrix} \tag{4}$$

Therefore, the inverse kinematics is given by

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\frac{1}{a}\sin\alpha & \frac{1}{a}\cos\alpha \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{5}$$

2.2 Dynamic Modeling

The complete dynamics model of the mobile robot given by [21], is written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \\ \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} u\cos\alpha + a\omega\sin\alpha \\ u\sin\alpha + a\omega\cos\alpha \\ \omega \\ \frac{\alpha_3}{\alpha_1}\omega^2 - \frac{\alpha_4}{\alpha_1} \\ -\frac{\alpha_5}{\alpha_2}u\omega - \frac{\alpha_6}{\alpha_2}\omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{\alpha_1} & 0 \\ 0 & \frac{1}{\alpha_2} \end{bmatrix} \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix} \tag{6}$$

where the input signals for the system are the desired values of the linear velocity u_{ref} and angular velocity ω_{ref}.

A vector of identified parameters and a vector of parametric uncertainties of the dynamic model are

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 \end{bmatrix}^T \tag{7}$$

and

$$\delta = \begin{bmatrix} \delta_x & \delta_y & 0 & \delta_u & \delta_\omega \end{bmatrix}^T \tag{8}$$

where δ_x and δ_y are functions of the slip velocities and the robot orientation. δ_u and δ_ω are functions of physical parameters as mass, inertia. Diameters of the wheel and tyre, parameters of the motors, forces on the wheels, etc., are considered as disturbances.

The parameters of the vector α are functions of certain physical parameters of the robot, such that:

m is the mass of the robot; I_z is the moment of inertia of the robot at point G; R_a is the electrical resistance of the motors; K_b is the electromotive constant of the motors; K_a is the torque constant of the motors, B_e is the coefficient of friction; I_e is the moment of inertia of each rotor-reducing-rotor group; r is the radius of the left and right wheels; R_t is the nominal radius of the tires; and b and d are the distances.

The equations describing the parameters α_i were proposed by the author in [21]. Therefore the parameters α_i are

$$
\begin{aligned}
\alpha_1 &= \frac{\left[\dfrac{R_a}{K_a}(mR_t r + 2I_e) + 2rK_{DT}\right]}{2rK_{PT}} \\
\alpha_2 &= \frac{\left[\dfrac{R_a}{K_a}(I_e d^2 + 2R_t r(I_z + mb^2)) + 2rdK_{DR}\right]}{2rdK_{PR}} \\
\alpha_3 &= \frac{R_a}{K_a}\frac{mbR_t}{2K_{PT}} \\
\alpha_4 &= \frac{\dfrac{R_a}{K_a}\left(\dfrac{K_a K_b}{R_a} + B_e\right)}{(rK_{PT})} + 1 \\
\alpha_5 &= \frac{R_a}{K_a}\frac{mbR_t}{dK_{PR}} \\
\alpha_6 &= \frac{R_a}{K_a}\left(\frac{K_a K_b}{R_a} + B_e\right) + \frac{d}{(2rK_{PR})} + 1
\end{aligned}
\tag{9}
$$

3. FRACTIONAL ORDER CONTROL

3.1 Fractional Order Calculus

There are several definitions for fractional order integral and differential, which include Caputo definition, Riemann-Liouville (RL) definition, and Grünwald-Letnikov definition [22].

The basic operator of fractional differential and fractional integral is ${}_aD_t^{\alpha}$ and ${}_aD_t^{-\beta}$ $f(t)$, where $(\alpha > 0)$ and $(\beta > 0)$ represents the real order of the differential calculus and integral calculus, t is the parameter for which the differential calculus is taken, and a is the lower limit.

The RL fractional derivative of order μ of a function $f(t)$ is given by [22]:

$$
{}_0D_t^u f(t) = \frac{1}{\Gamma(\mu - n)}\frac{d^n}{dt^n}\int_0^t (t-\mu)^{n-\mu-1} f(\mu)d\mu \tag{10}
$$

where n is the nearest integer greater than μ, $(n-1 < \mu < n)$ and the RL definition for fractional integral order λ, of a function $f(t)$ is given by [12]:

$$
{}_0D_t^u f(t) = \frac{1}{\Gamma(\lambda)}\int_0^t (t-\tau)^{\lambda-1} f(\tau)d\tau,\ 0 < \lambda < 1 \tag{11}
$$

where $\Gamma(.)$ is the well-known Euler's gamma function

$$
\Gamma(z) = \int_0^{\infty} e^{-t}t^{z-1}dt, \quad Re(z) > 0 \tag{12}
$$

The Laplace transform of the RL fractional integral under zero initial condition can be derived as:

$$
L\left[{}_0D_t^{-\lambda} f(t)\right] = \frac{1}{s^{\lambda}}F(s) \tag{13}
$$

where $F(s)$ is the Laplace transform of $f(t)$ and the Laplace transform of RL fractional derivative can be derived as:

$$
L\left[{}_0D_t^{\mu} f(t)\right] = s^{\mu}F(s) \tag{14}
$$

3.2 Fractional Order PI$^{\lambda}$D$^{\mu}$ Controller

The classic PID controllers are still dominating industrial controllers, because of their simplicity of implementation and the availability of many effective and simple tuning methods based on minimum plant model knowledge.

The equation of a fractional order controller was described as [23]

$$
u(t) = -(K_p e(t) + K_d D_t^{\lambda} e(t) + K_i D_t^{-\mu} e(t) \tag{15}
$$

where $e(t)$ is the error between a measured output variable and the desired setpoint, and $u(t)$ is the control signal. The five control parameters (K_p, K_i, K_d, λ, and μ) must be set to design a precise controller. Integral order and derivation order add flexibility to the design of a FOPID controller. The continuous transfer function in s, a domain of FOPID controller, is given by [23]

$$
G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \tag{16}
$$

The orders λ and μ can be any real numbers. The equation of the classical PID controller can be obtained by setting λ=μ=1. When λ=1 (or 0) and μ=0 (or 1) a normal PI (or PD) controller can be obtained. To tune FOPID parameters many researchers used some of the conventional methods which are empirical and experimentally tuning approaches. After that, they tune the fractional order (λ and μ) in the last step of the tuning method for more flexibility. Fig. 2 shows that the fractional order PI$^{\lambda}$ D$^{\mu}$ controller generalizes and expands the classical PID controller from a point to plane.

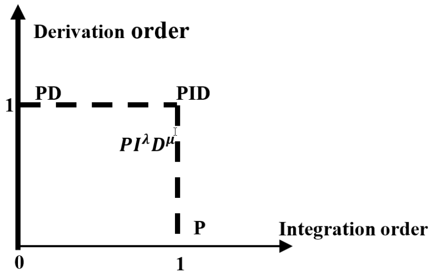


Fig.2: *Controller FOPID.*

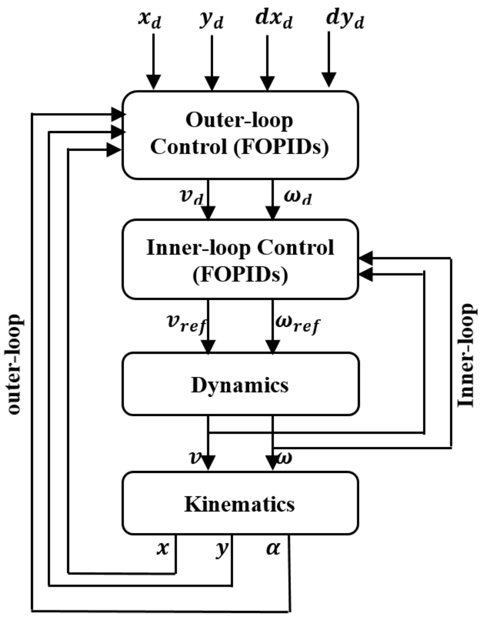


Fig.3: *Control System.*

4. CONTROL STRATEGY

To control our mobile robot we use two subsystems, four FOPID controllers are used. Two FOPID are used in the inner loop to control linear and angular velocities and others FOPID are used in the outer loop for the control of x, y position (Fig. 3). The tuning of all FOPIDs parameters is set in the same way using PSO described in Section 5.

To obtain the desired position of the mobile robot with a precise linear velocity and angular velocity we opted for a well-defined control strategy that is divided into two stages.

The first one allows us to control the mobile robot in an internal loop to have the linear velocity and angular velocity desired using a FOPID controller, and the dynamic model of the robot. Fig. 4 shows this control in this step.

In the second step we controlled the robot in an external loop to adjust its position; this step is divided into 3 blocks: the first block for the FOPID controller, the second contains the blocks of the first step and in the third block we put the kinematic model of the robot as shown in Fig. 5.

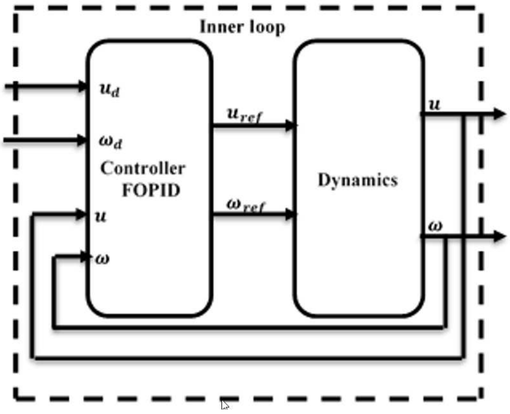


Fig.4: *Inner Loop.*

The values of the FOPID controller parameters used in this system are designed from the PSO algorithm.

5. TUNING OF FOPID WITH PSO

The PSO is viewed as the best calculation compared to different techniques. Its favorable position is due to execution time, cost and better results. Another benefit for which PSO is appealing is that there are few parameters to change. The system is initialized with a population of arbitrary solutions and looks for optima by refreshing the generations. To find Pbest and Gbest in each iteration each particle updates the five components K_p, K_i, K_d, λ and μ. Finally, the algorithm runs to locate the ideal solution.

Particle swarm optimization has been used for approaches that can be connected over an extensive variety of applications, as well as for specific applications focused on a specific requirement. Fig. 6 describes the structure of PSO algorithm.

5.1 Fitness Function

In the present study, an integral absolute error is utilized to assess the performance of the controller as it is widely adopted to evaluate the dynamic performance of the control system, based on the calculation by [24].

The fitness function can be calculated

$$F_i = \int_0^\infty (w|e_i(t)|)dt \qquad (17)$$

where w is the inertia weight factor between $w_{\max} = 0.9$ and $w_{\min} = 0.4$. and $i \in \omega, u, x, y$.

The aim is to minimize this fitness function in order to improve the system response in terms of steady-state error. The stop criteria used was the one that defines the maximum number of generations to be produced. When the PSO algorithm runs, the new populations generating process is finished, and the best solution to complete the generation number

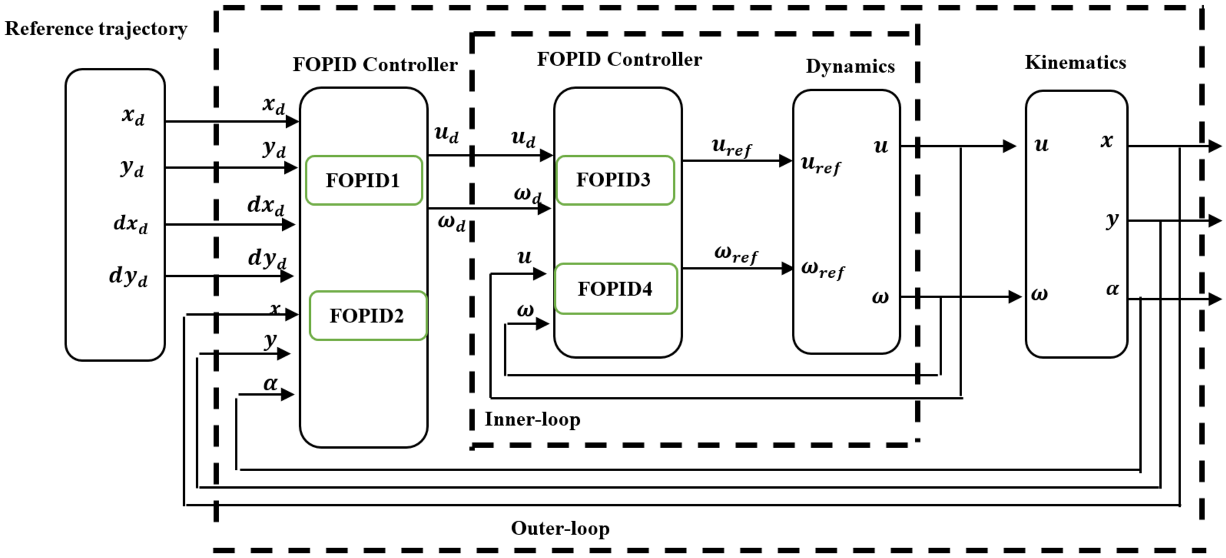


Fig.5: *Model Control System.*

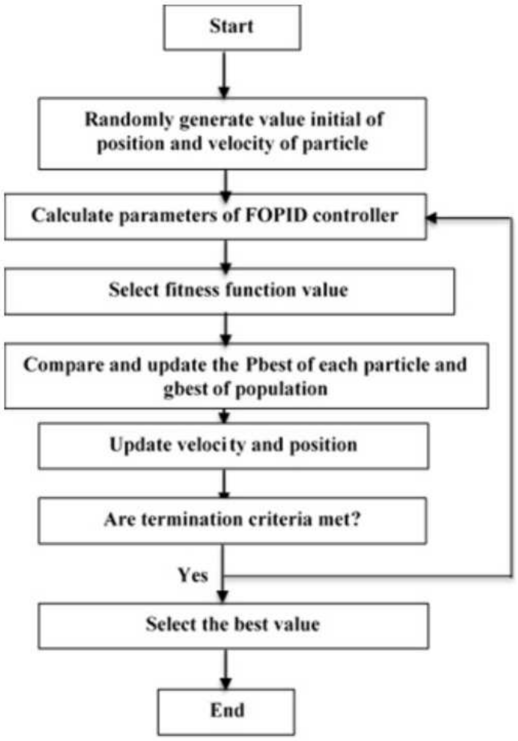

Start
Randomly generate value initial of position and velocity of particle
Calculate parameters of FOPID controller
Select fitness function value
Compare and update the Pbest of each particle and gbest of population
Update velocity and position
Are termination criteria met?
Yes
Select the best value
End

Fig.6: *Structure of PSO Algorithm.*

is the one among the individuals better adapted to the evaluation function.

5.2 Calculation of Parameters of Fractional Order

From Eq. (2) five parameters K_p, K_i, K_d and λ, μ must be calculated, depending on the control objectives. For the typical PID controller design, all poles of the closed loop transfer function must be limited to the left half of the s plane. Based on the stability condition of the fractional order system, a controller that stabilizes the integer order system stabilizes the fractional order system. The FOPID controller parameter can adopt the parameter of integer order controller and add fractional orders of integral and derivative [24].

In this paper, four FOPID controllers are used to control the robot. Two FOPID controllers are used in the inner loop to control linear and angular velocities and other FOPID controllers are used in the outer loop for the control of x, y position. The tuning of all FOPID controller parameters is set in the same way.

For each FOPID controller, five dimensional vectors represent the initial positions of the ith particles of the swarm, then the initial values are generated randomly as a function of the extreme values.

To optimize parameters of each FOPID controller, the PSO creates a population of 200 swarms that contain the parameters necessary to minimize the objective function.

The initial range of parameters are selected, these are $K_p \in [0, 50]$, $K_i \in [0, 10]$, $K_d \in [0, 1]$, $\lambda \in [0, 1]$ and $\mu \in [0, 1]$.

After 50 runs, the five parameters K_p, K_i, K_d, λ and μ are in the predefined search range and the fitness value becomes smaller.

Table 1 lists the best performance of FOPID controllers

After calculating the parameters of a FOPID controller using a PSO algorithm, for our simulation we applied these values in our mobile robot by taking into consideration the complete dynamic model including its speed and acceleration limitations.

In general, numerical coefficients, swarm size, and topology must be predefined. The overall optimiza-

Table 1: *Performance of FOPID Controller.*

	K_p	K_i	K_d	λ	μ
Linear velocity	48.80	7.5401	0.2001	0.9	1.0
Angular velocity	3.84	7.5388	0.2001	0.9	1.0
X Position	48	7.5005	0.204	0.9	0.95
Y Position	3.8	7.5010	0.203	0.9	0.95

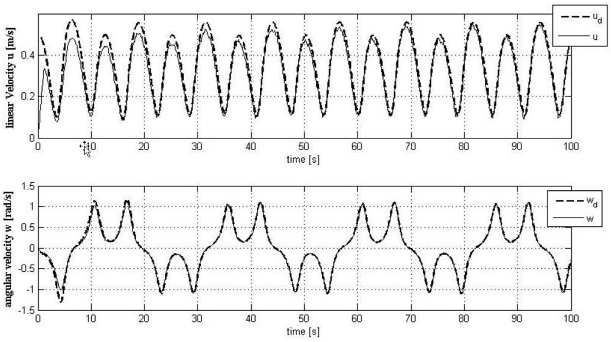


Fig.7: *Linear (top) and Angular (bottom) Velocities.*

tion capability relies heavily on the setting of these parameters.

6. SIMULATIONS AND RESULTS

In this section, we present our simulation results using MATLAB/Simulink to validate our controller FOPID; after the controller parameters are calculated with the PSO algorithm we implement the FOPID controller in our system.

The following values are used in our simulation setup for fractional order parameters optimization:

1) The dimension of research space order is 6.

2) Initial population size = 50.

3) The maximum number of iterations = 200.

4) Vmin is set equal to Xmin and Vmax equal to Xmax.

5) The swarm coefficients c1=c2=2.5

6) The inertia weight factor w is between $w_{max} = 0.9$ and $w_{min} = 0.4$.

7) The simulation time $t = 100$ s.

8) The process is repeated 15 times.

The simulations are made using MATLAB/ Simulink following a sinusoidal trajectory in which the desired position and orientation of the unicycle are specified. This trajectory is described as:

$$\begin{aligned} x_d(t) &= \sin(2\omega t) \\ y_d(t) &= \cos(\omega t) \end{aligned} \tag{18}$$

Fig. 7 shows that before the parameters update there is a difference between the linear and angular velocities, but after updating the parameters, both velocities follow the reference signal well.

In Fig. 8 it can be seen that the robot's current position (positions x and y) perfectly follows the desired position after a certain time. The latter is the update time that sets the PSO algorithm to calculate these positions. In our case, the position of the robot follows the desired position after a time of 40 s.

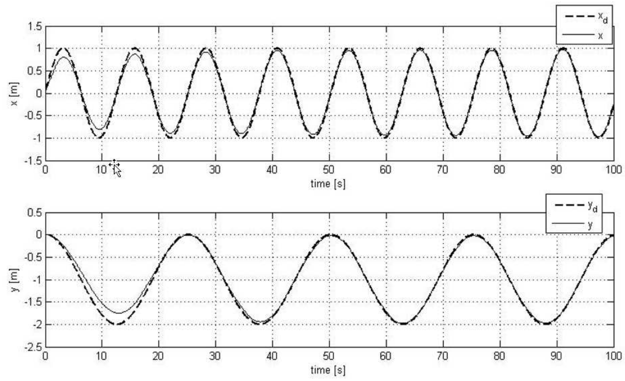
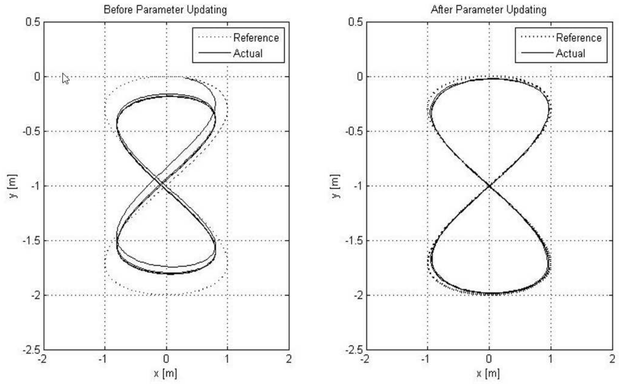


Fig.8: *Desired and Actual Position On X (top) and Y Axes (bottom).*



(a) (b)

Fig.9: *Robot Trajectory.*

Our objective to use the FOPID controller is to control the path of the mobile robot by minimizing the error. For that, we do two tests of trajectory, the first test without use of the algorithm PSO and the second with using the algorithm PSO. The reference path used is an 8 shape. Referring to Fig. 9(a), we note that despite the moving robot following the trajectory shape, the error between the current and desired trajectory is considerable.

In contrast to Fig. 9(a), in Fig. 9(b) we notice the effect of using the PSO algorithm to designate the exact parameters of the FOPID controller. After a certain time of calculation, the mobile robot perfectly follows the trajectory desired with a very small or almost zero error.

In Fig. 10 and Fig. 11, we can see the difference between a PID controller and a FOPID controller. The FOPID controller allows the robot to perfectly follow the desired signal; by contrast, with the PID controller we find that there is an error between the

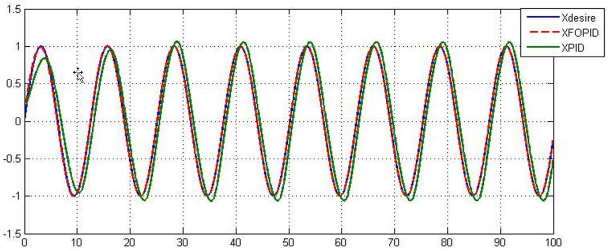


Fig.10: *Difference Between Controller FOPID and PID Position X.*

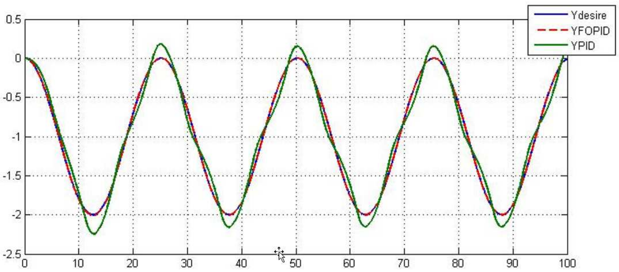


Fig.11: *Difference Between Controller FOPID and PID Position Y.*

desired signal and the current signal.

7. CONCLUSION

In this work, we presented a control technique for mobile robots. Firstly a study of the kinematics and dynamics of a mobile robot was given Then for the implementation of the controller we chose the PSO algorithm to determine the parameters of the FOPID controller. The control law was used by giving the mobile robot linear and angular reference speeds obtained by comparing the values of the calculated and desired speeds. The efficiency of the PSO algorithm allowed us to have the best parameters for our controller. The analysis of the results shows the good performance and the efficiency of the controller for the follow-up of trajectory in terms of precision, stability and convergence. In future work, we will try remote control of the mobile robot with avoidance of obstacles with or without use of a camera. We can also investigate collaborations of several mobile robots in a real environment.

References

[1] I. Benaoumeur, L. Benchikh,"Backstepping Approach for Autonomous Mobile Robot Trajectory Tracking," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2, no. 3, pp. 478-485, 2016.

[2] R. Fierro, F. L. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," in *Proceedings of the 34th IEEE Conference on Decision and Control*, vol. 4, pp. 3805-3810, 1995.

[3] G. Yanfeng and Z. Hua. "Back-stepping and neural network control of a mobile robot for curved weld seam tracking," *Procedia Engineering*, vol. 15, pp.38-44, 2011.

[4] M. Z. Ab Rashid, S. N. Sidek. "Controller Design of Unicycle Mobile Robot," *Iium Engineering Journal*, vol.3, no. 2, pp. 173-192, 2012.

[5] Q. Xu, J. Kan. "Fuzzy PID based trajectory tracking control of mobile robot and its simulation in Simulink," *International journal of control and automation*, vol.17, no. 8, pp.233-244, 2014.

[6] T. Das, I. N. Kar, "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 14 no. 3, pp. 501-510, 2006.

[7] Y. Cai, Q. Zhan, "Path tracking control of a spherical mobile robot," *Mechanism and Machine Theory*, vol. 51, pp. 58-73, 2012.

[8] G. Guoqin, R. Yi, "Smooth sliding mode control for trajectory tracking of greenhouse spraying mobile robot," *telkomnika Indonesian Journal of Electrical Engineering*, vol. 11 ,no. 2, pp. 642-652, 2013.

[9] L. Gracia, F. Garelli. "Integrated sliding-mode algorithms in robot tracking applications," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp.53-62, 2013.

[10] B. Aicha, A. Benalia and F. Boudjema, "Trajectory Tracking of a Unicycle-Type Mobile Robot Using Second-Order Sliding Mode Control," in *International Conference on Electrical Engineering and Automatic Control*, Algeria, 2013

[11] M. Ashoorirad, R. Barzamini, "Model reference adaptive path following for wheeled mobile robots," in *IEEE International Conference on Information and Automation (ICIA)*, pp. 289-294, 2006.

[12] E. Canigur, M. Ozkan, "Model reference adaptive control of a nonholonomic wheeled mobile robot for trajectory tracking," in *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1-5, 2012.

[13] F. N. Martins, W. C. Celeste, "An adaptive dynamic controller for autonomous mobile robot trajectory tracking," *Control Engineering Practice*, vol. 16, no. 11, pp.1354-1363, 2008.

[14] J. Taheri-Kalani and M. Khosrowjerdi, "Adaptive trajectory tracking control of wheeled mobile robots with disturbance observer," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 1, pp. 14-27, 2014.

[15] G. Scaglia, A. Rosales, "A linear-interpolation-based controller design for trajectory tracking

of mobile robots," *Control Engineering Practice*, vol. 18, no. 3, pp. 318-329, 2010.

[16] Z. Mohamed, A. Basri, "Velocity control of a unicycle type of mobile robot using optimal pid controller," *Journal teknologie (sciences and Engineering)*, vol. 78, pp. 7-14, 2016.

[17] M. A. Tawfik,E. N. Abdulwahb, S. M. Swadi "Trajectory Tracking Control for a Wheeled Mobile Robot Using Fractional Order Controller," *Al-Khwarizmi Engineering Journal*, Vol. 10, No 3, pp. 39- 52, 2014.

[18] A. Oustaloup, "La dérivation non entière," Hermes, 1995.

[19] W. Hui-fang, H. Qiu-sheng, Z. Zhi-cheng and Z. Jing-gang, "A design method of fractional order $PI^{\lambda}D^{\mu}$ controller for higher order systems," *Proceeding of the 34th Chinese Control Conference*, pp. 272-277, 2015.

[20] A. Tepljakov, E. Petlenkov and J. Belikov, "FOMCON: Fractional-order modeling and control toolbox for MATLAB," *Proceeding 18th Int Mixed Design of Integrated Circuits and Systems (MIXDES) Conference*, pp. 684-689, 2011.

[21] C. De La Cruz and R. Carelli, "Dynamic modeling and centralized formation control of mobile robots," in *IEEE Industrial Electronics*, IECON, 2006, pp. 3880-3885

[22] F. Padula and A. Visioli, "Set-point weight tuning rules for fractionalorder PID controllers," *Asian Journal of Control*, vol. 15, no. 3, pp. 678-690, 2013.

[23] D. Xue, Y. Chen and D. P. Atherton, "Linear Feedback Control Analysis and Design wih MATLAB," *USA: The soceity for Industrial and Applied Mathematics*, 2007.

[24] J. Y. Cao and B. G. Cao, "Design of Fractional Order Controller Based on Particle Swarm Optimization," *International Journal of Control, Automation, and Systems*, vol. 4, no. 6, pp. 775-781, 2006.

Younes Boucetta received the State Engineer degree in Electronics in 2005, and the Magister degree in automatics and robotics in 2009, from the University of Sciences and Technology of Oran Mohammed Boudiaf (USTO-MB), Algeria. Since October 2012, he has been with the Department of Electronics, Hassiba Benbouali University of Chlef as an assistant professor. He is currently working towards PhD degree in Automatics at the University of Sciences and Technology of Oran Mohammed Boudiaf (USTO-MB), Algeria. His current research interests include linear and nonlinear control, robotics, and automatics.

Redouane Ayad received the engineering degree in automatics in 2006, magister degree in automatics in 2010, and PhD in automatics, 2019 from the University of Sciences and Technology of Oran Mohammed Boudiaf (USTO-MB), Algeria. Interested in autonomous systems, UAVs. He is currently an assistant professor at the Hassiba Benbouali University of Chlef (UHBC), Algeria.

Zoubir Ahmed-Foitih received the engineering degree in electronics in 1980, the magister in automatic in 1988, and the PhD degree in 2004 from the University of Sciences and Technology of Oran Mohammed Boudiaf (USTO-MB), Algeria. Since 2012, he is a full professor at the (USTO-MB) and head of augmented reality team, which is implied in several French and european projects. His current research interests include robotics, computer automation, control of industrial processes and supervision, identification of process, soft computing.