PRZEGLĄD ELEKTROTECHNICZNY

Ukazuje się od 1919 roku **2'2022**

Organ Stowarzyszenia Elektryków Polskich • **Wydawnictwo SIGMA-NOT Sp. z o.o.**

1. Rezoug Nabil[1], 2. Zerikat Mokhtar[2], 3.Chekroun Soufyane[3]

Laboratory of Automatics and Systems Analysis (L.A.A.S.), Department of Electrical Engineering, National Polytechnic School –Maurice Audin - Oran. BP 1523 El' M'naouer, Oran, Algeria.(1)(2)
Department of Electrical Engineering, University Mustapha Stambouli of Mascara.BP.305, Route El Mamounia, 29000 Mascara Algeria.(3)
ORCID. 1.0000-0003-1613-8605, 2.0000-0002-4351-4867, 3. 0000-0002-7394-5879

doi:10.15199/48.2022.02.01

An Efficient Fuzzy PI Approach to Real-time Control of a ROS-Based Mobile Robot

***Abstract**. This paper presents the implementation and experimental validation of a fuzzy PI control framework on a Turtlebot Kobuki. Due to the robot complex kinematic model structure, we first designe a guidance law to overcome computational challenges that could become problematic under real time control, further experimentation tryals evoked the need for a robust solution which was introduced through an integral action of the PI controller cascaded into the Fuzzy controller. Control performance analysis were carried out in a real experimental setup in order to validate the effectiveness of the proposed scheme.*

Streszczenie.** Ten artykuł przedstawia implementację i eksperymentalną walidację rozmytej struktury kontroli PI na Turtlebot Kobuki. Ze względu na złożoną strukturę modelu kinematycznego robota, najpierw opracowaliśmy prawo przewodnie, aby przezwyciężyć wyzwania obliczeniowe, które mogą stać się problematyczne przy sterowaniu w czasie rzeczywistym. do kontrolera Fuzzy. Analizę działania sterowania przeprowadzono w rzeczywistym układzie doświadczalnym w celu sprawdzenia skuteczności proponowanego schematu. (**Wydajne, rozmyte podejście PI do sterowania w czasie rzeczywistym mobilnego robota opartego na ROS)

Keywords: Fuzzy PID**,** ROS, Turtlebot Kobuki, Waypoint guidance.
Słowa kluczowe: in the case of foreign Authors in this line the Editor inserts Polish translation of keywords.

Introduction

In the last decade, robotic technology witnessed a rapid grow mainly for its huge involvement in industry and service robotic, in particular Autonomous mobile robots which are often seen as a replacement for guided vehicles, they offer the advantage of being able to navigate complex environments without the need for permanent wire strips or magnetic tracks along the floor to guide their path. Moreover, since the onset of COVID-19, homebound consumers ordering goods online has spiked, and demand for rapid order fulfilment in warehousing applications has grown vital, especially at a time when fewer workers are able to occupy facilities, the adoption of mobile robots has increased considerably mostly for its ability to help meet social distancing requirements in industry and vertical ranging from healthcare to manufacturing. In this paper we address the subject of autonomous navigation in indoor environments as a step to build a control law that steers the robot through given reference locations, however developing a fully autonomous robot that can navigate in unknown environments is problematic due to difficulties that span dynamics modelling, on-board perception, trajectory generation and optimal control, not to mention the heavy computational bill that may results from such methods, Although several studies denounce limitations that are imposed by the structural obstruction of the kinematic model, little attention has been drawn to address these obstacles. As Real-time fine motion control is a fundamentally important and yet difficult issue in robotics, many studies have been done as an attempt to address the issue of how to efficiently control a mobile robot with unknown robot dynamics, and subject to significant uncertainties and unmodeled disturbances, this issue in particular has drawn lot of interest from researchers in artificial intelligence, however as reported by [1] real-time processing requires a large neural network with multiple parameters, which represent a computational burden, even for modern CPU, in [2] they used fuzzy PID controller for autonomous motion of a differential drive robot, in [3] an efficient single layer structure neural network approach is proposed for Realtime motion control of a nonholonomic mobile robot, in [4] a nonlinear model predictive controller is used to overcome the complex nonlinear terms and significant uncertainties that exhibits the vehicle dynamic, in [5] the aim was for the robot position and orientation to reach desired final values simultaneously in a user-defined time, a robust position-tracking controller for a perturbed wheeled mobile robot is designed and tested in a real robot TurtleBot 2. Previously, many approaches based on numerical analysis have sufficed in simplifying problems to allow analytical tractability for complex systems, however, now days, many processes cannot be adequately analyzed or modelled into equations without requiring multiple interacting and interconnected frameworks, especially since most of these systems contain uncertain information, hence the need to develop a process independent non-model-based method, relying on Fuzzy PID controller, we develop in this paper an algorithm that ensures the task of steering the robot between a set of reference locations, or waypoints allowing high performance and accurate path tracking.

The article is structured as follows, we start by citing the motivation for our research along with the encountered problems, also mention different work done addressing this subject in the last few years, next, we give brief introduction to the kinematic model of nonholonomic mobile robots in general, and then tackle the approach we used for guidance, section III starts with an introduction to the strategy used to control the robot, section IV shows the experimental results of the implemented strategies, section V contains the conclusion and future work.

Robot model and path following problem formulation

When it comes to path generation in two-dimensional frame, many researchers rely on guidance laws to ensure accurate tracking, in our work we relied on waypoint guidance law to generate the desired path, a thorough explanation is given in this section

Kinematic Model

The kinematic model for nonholonomic ground robot is given by

$$\begin{aligned} \dot{x}_r &= V_a \cos(\psi), \\ \dot{y}_r &= V_a \sin(\psi), \\ \dot{\psi}_r &= \omega; \end{aligned} \tag{1}$$

where x_r and y_r denote the robot position in the fixed frame, and Ψ it's orientation. V_a is the linear velocity, and ω is the angular velocity.

Waypoint guidance strategy

Homebound consumers ordering goods online has spiked since the ongoing global health crisis, rapid fulfilling of the market demands has become more and more challenging, especially since governments derives their authority for isolation and quarantine by substituting human factor in the work place, hence the need for mobile robots in warehousing applications, where autonomous motions are needed from vertical ranging to bringing goods or tools to specified target locations, in this section we study a guidance law that steers a mobile robot through reference locations otherwise known as waypoints , simply by driving in a straight line from one waypoint to the next, which is naturally the shortest path between two points, a desired path is a consecutive association of n waypoint referred to as Pk where $k \in \{1\ldots, n\}$, technically for the robot to drive to the next waypoint it is first required to reach an acceptance range circling the current waypoint [6].

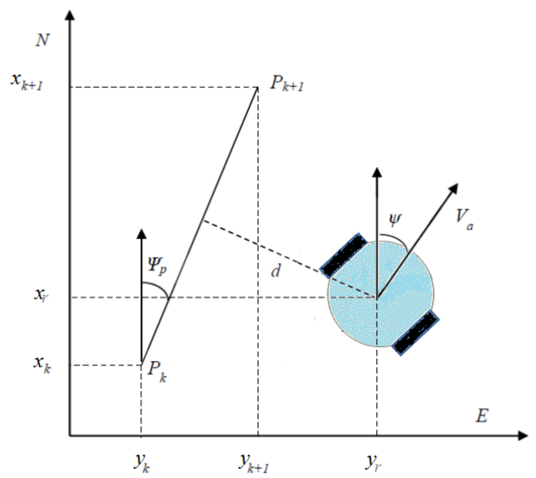

Fig. 1. Waypoint following (N: North, E: East)

d is the distance between the robot the reference path,

$$d = -(x_r - x_k)\sin(\psi_p) + (y_r - y_k)\cos(\psi_p) \tag{2}$$

where ψ_p is the angle of the path in relative to the x axe direction of the robot, defined by

$$\psi_p = \tan^{-1}\left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right) \tag{3}$$

ψ_e is the orientation of the robot relative to the desired path

$$\psi_e = \psi_p - \psi \tag{4}$$

By differentiating (2) with respect to time and using (1), it follows that

$$\dot{d} = -V_a \sin(\psi_e) \tag{5}$$

$$\dot{\psi}_e = \omega \tag{6}$$

The considered state vector $x = [d, \psi_e]^T$

Designing Direct action Fuzzy PI Controller

In literature, various structures for fuzzy PID controllers have been studied, it can be further categorized into 3 types [7,8,9], Fuzzy Gain Scheduling it is basically a conventional PID with controllable gains tuned on line through a set of fuzzy rules, next is a hybrid type controller, a combination of a conventional PID and Direct-Action Fuzzy logic controller. In this work we used Direct-Action Fuzzy PI controller, as it is shown in Fig.2. the PI controller is cascaded to a FLC, in this case, the proportional gain act as an amplifier of the error, thus the output action will be relative to the error amplitude. It is well known that the integral action offers a robust solution to the control structure, however to avoid extra computational burden that will occur from an extra set of rules we separated the integral action from the FLC structure.

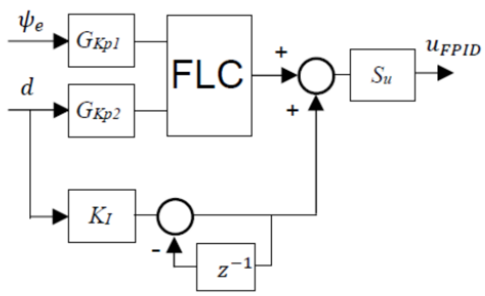

Fig. 2. One input direct action fuzzy PI control structure, S_u: scaling factors. G_{kp1} and G_{kp2}: Fuzzy gains, K_I: integral gain.

1.Selecting input variables

The input variables of the fuzzy controller are: d eq (2) and Ψ_e eq (4). The inputs are fed directly from the waypoint algorithm generator, the input error variables are transformed into a normalized region [-1 1], therefor the scale factors for error variables are defined to obtain a normalized error, in this case, Fuzzy gains for error G_{kp1} and G_{kp2} are simultaneously used as proportional gain and scale factors. The input variables set are shown below.

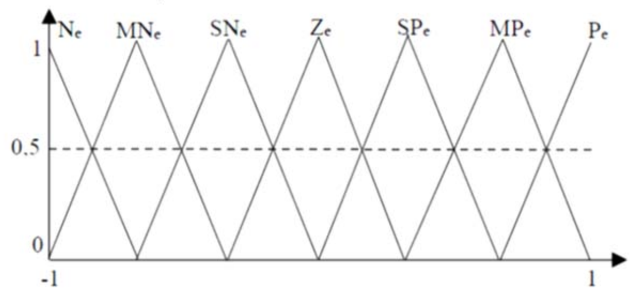

Fig. 3. Input membership function

N_e: Negative error. P_e: Positive error Z_e: No error
SN_e: Small Negative error. MN_e: Medium Negative error.
SP_e: Small Positive error. MP_e: Medium Positive error

b. Output Variables

Outputs are singletons, as shown in Fig. 4. The output varies from -1 1, singletons have the advantage to drive the control signal to its extreme values, they also provide a more intuitive way to write rules. As part of a post-processing procedure, the output has to be converted to an engineering unit by applying scaling factors S_u.

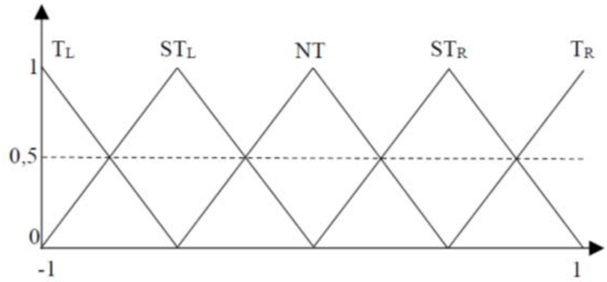

Fig. 4. Output membership function
T_L: Turn Left. ST_L: Slow Turn Left NT: No Turn
ST_R: Slow Turn Right. T_R: Turn Right.

C. rule base
this process is built in the "IF-Else-THEN" format, since we have 2 inputs with 7 membership function each.

Table 1. Rule base for steering the robot

d / ψ_e	N_e	MN_e	SN_e	Z_e	SP_e	MP_e	P_e
N_e	/	/	/	STL	STL	TL	TL
MN_e	/	/	/	STL	STL	TL	TL
SN_e	/	/	/	STL	STL	STL	TL
Z_e	STR	STR	STR	NT	STL	STL	STL
SP_e	TR	STR	STR	STR	/	/	/
MP_e	TR	TR	STR	STR	/	/	/
P_e	TR	TR	STR	STR	/	/	/

Real Time Implementation
Experimental testbench

Fig.5. shows the mobile robot we have in our school Laboratory of Automatic & Analyses of Systems (LAAS), it was used in this work to assess the framework reliability, it is a ROS-enabled nonholonomic mobile robot, Turtlebot Kobuki 2 is from a robot series, powered by Willow Garage and ClearPath Robotics, more details are provided in [10].

Fig.5. Experimental platform Turtlebot Kobuki

The Turtlebot is equipped with a Kinect sensor, a Kobuki base, for monitoring and control we used an i5-6300HQ 2.30 GHz laptop with 8 Go of RAM. For movement the robot has two motorized rear wheels equipped with an encoder, the robot also contains proximity sensors and gyro meter, harware specification of the Turtlebot are as follows [10]

- Diameter: *351,5 mm*; Height:*124,8 mm*; Weight: *2,35 kg.*
- Maximum translational velocity: *70 cm/s.*
- Maximum rotational velocity: *180 deg/s* (>*110 deg/s* gyro performance will degrade).
- Odometry: *52 ticks/enc* rev, *2578.33 ticks/wheel* rev, *11.7 ticks/mm.*
- Gyro: factory calibrated, 1 axis (*110 deg/s*).

As mentionned above, the Turtlebot is equiped whith a 3-Axis digital gyroscope and magnetic type encoder for each wheel, it is clear from equation (7) and (8) that the data provided from these sensors can be used to estimate the change in position and heading over time, thus having a thorough tracking of the robot evolution in space relative to it's starting position

(7) $$v = \frac{r(\omega_R + \omega_L)}{2}$$

(8) $$\omega = \frac{r(\omega_R - \omega_L)}{l}$$

The angular speed of two wheels denoted by ω_L and ω_R for left and right wheels, respectively, r is the radius of the wheels and l is the distance between the wheels. As shown in Fig.6. real time odometry data is streamed on a ROS topic "*odom*", however a conversion from quaternion to Euler angles is necessary in order to give a physical sense to the robot's heading angle.

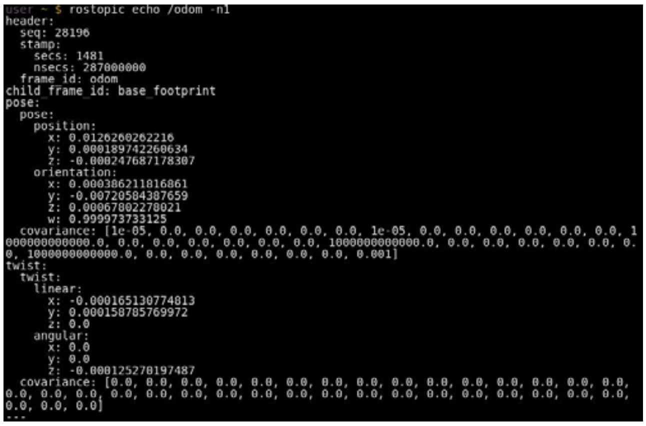

Fig. 6. ROS topic *odom*

In [10], a profound study has been done to estimate the accuracy of the Turtlebot odometry and heading angle. As it is denounced in various studies, odometry is a source of a certain degree of innacuracies, since it is a cumulative measurement, any error from the encoders will increase as time passes as it is shown in the study in [10]. We will discuss below the effect of such inacuracies on the tracking performance of the Turtlebot.

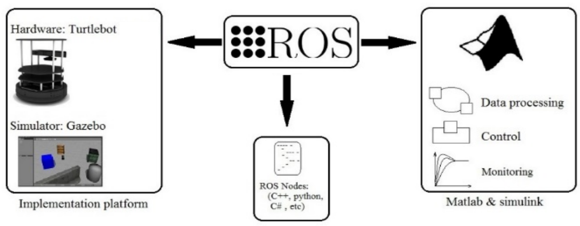

Fig. 7. ROS communication diagram

In the presence of a running ROS master under Linux, we initialize a Matlab ROS slave with the IP address and port number of the ROS master, Matlab-Simulink was essential for monitoring and data processing. The algorithm generating the system state vector $x = [d, \psi_e]^T$ was written with a MATLAB function, to ensure accurate tracking, the algorithm require real time values of the robot position and angle which are transferred through a C++ code under ROS [11].

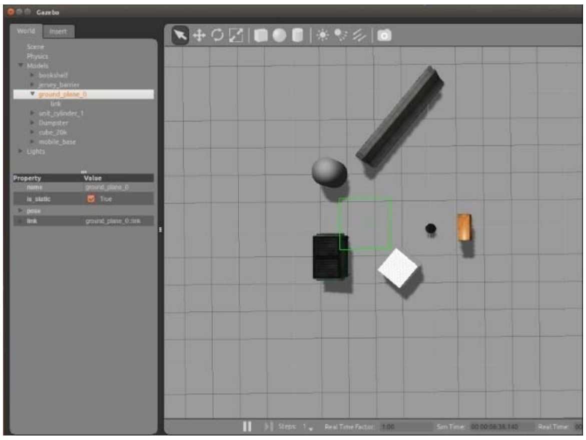

Fig. 8. Gazebo World Environment

It is mandatory to run the control strategy on a simulator before implementing it on a real robot, to this end we used Gazebo World Fig.8. as a mean to check the robot behavior seeking to track different reference trajectories, as a matter of fact, by combining Gazebo world with ROS, we mimic the real time experiment to a certain extent allowing realistic simulation of the robot behavior to different trajectories and with different controllers. Below, we discuss real time indoor tracking of the Turtlebot seeking to track a circular trajectory under Fuzzy Pi controller, the linear speed was kept at a constant value of 0.2 m/s.

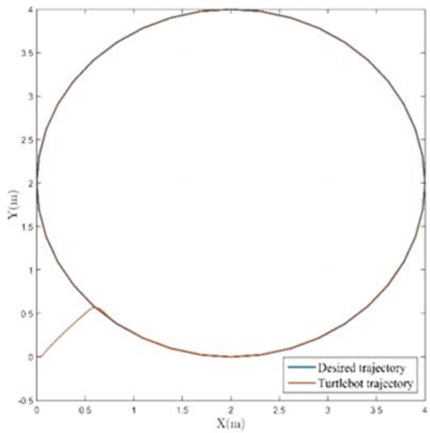

Fig.9. Circular trajectory diagram of using Fuzzy-Pi controller.

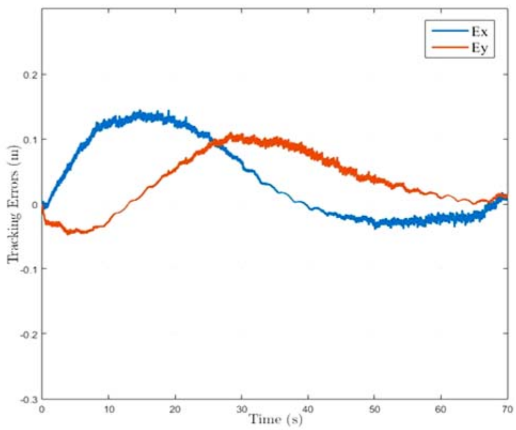

Fig.10. Tracking error diagram along the X and Y axis of the circular trajectory.

The robot start converging with an initial error that cannot be avoided due to the nonholonomic constraint character of this kind of robot. Generating a circular trajectory comes down to dividing the circle into an association of consecutive reference locations or waypoints, basically to have a more accurate path we can simply add more waypoints, however with this approach, we are limited to a certain degree of accuracy, mainly because of the localization error delivered from wheels encoder. With the circle centered at (2,2) and a 2 meters radius, the generated circle allows smooth control over the steering angle, since the heading error equation 4 varies in a small range from one waypoint to the next, the tracking error along the X and Y axis in Fig.10. varies in the range of +-0.15 margin which is considered to be negligeable since the acceptance reaching range is set to 0.1.

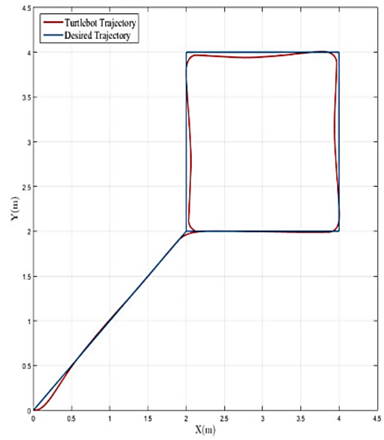

Fig.11. Square trajectory diagram of Turtlebot using Fuzzy-Pi controller

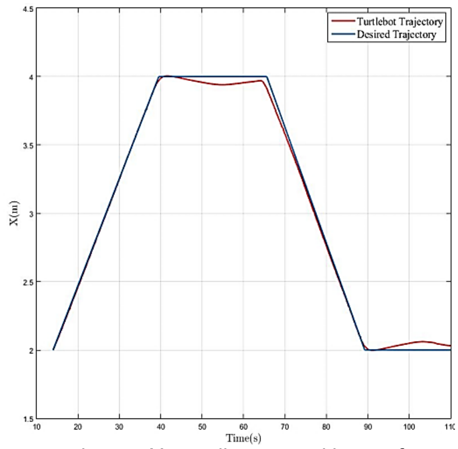

Fig.12. Square trajectory X coordinates tracking performance using Fuzzy-Pi controller

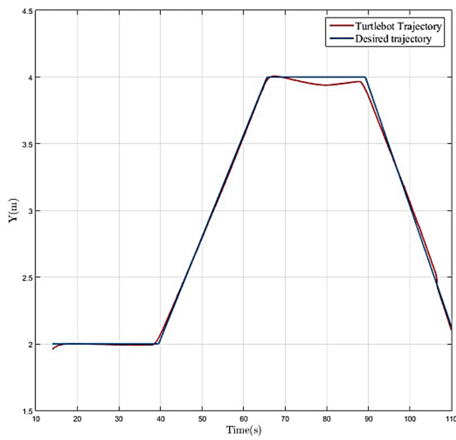

Fig.13. Square trajectory Y coordinates tracking performance using Fuzzy-Pi controller

The robot originally at the 0 coordinate (0,0) start to converge with an inevitable initial error that is due to nonholonomic constraints character of the Turtlebot, the consecutive association of the given waypoints give track to a square trajectory located at (2,2); (4,2); (4,4); (2,4).

Since a 90° turn is a challenging maneuver to execute especially with a safety limitation on the steering angle, the sudden change in direction around the square trajectory edges causes a tracking error, mainly because the heading error ψ_e equation 4 varies abruptly from 0° to 90°. in comparison to other studies results [12], Direct action Fuzzy-PI controller shows better tracking performance. However improvement is required to overcome flawed tracking performance especially around the edges of the square trajectory as shown in Fig.11.12.13.

Concluding remarks

in order to achieve computational efficiency, we first worked toward overcoming limitations that are imposed by the structural obstruction of the kinematic model, by designing a guidance law that simplifies the structure of the control system. Moreover, unlike a pure fuzzy logic controller, the cascade-form (Fuzzy-PI) controller gives a robust solution to trajectory tracking control problem, also, using triangular input membership functions, makes their structure even simpler, the experimental results show an accurate tracking from various reference trajectories tested in real time and 3D simulation under Gazebo world. However Fuzzy Pi control showed flawed tracking performance and even with good understanding of the process we had to undergo multiple real time trials, to overcome these difficulties, for future work we will aim to use a Kinect to determine the robot's position and prevent excessive odometry error buildup, also develop a learning method that enables self-tuning of Fuzzy Logic Controller.

Authors. *Nabil REZOUG Laboratory of Automatics and Systems Analysis (L.A.A.S.), Department of Electrical Engineering, National Polytechnic School –Maurice Audin - Oran. BP 1523 El' M'naouer, Oran, Algeria, E-mail: rezoug.nabil@gmail.com ; Pr.Mokhtar ZERIKAT Laboratory of Automatics and Systems Analysis (L.A.A.S.), Department of Electrical Engineering, National Polytechnic School –Maurice Audin - Oran. BP 1523 El' M'naouer,Oran, Algeria, E-mail:mokhtar.zerikat@enp-oran.dz; Soufyane CHEKROUN, Department of Electrical Engineering, University Mustapha Stambouli of Mascara.BP.305, Route El Mamounia,29000 Mascara Algeria. E-mail: s.chekroun@univ-mascara.dz*

REFERENCES

[1] Vincent Vanhoucke and Andrew Senior and Mark Z. Mao "Improving the speed of neural networks on CPUs", Deep Learning and Unsupervised Feature Learning, NIPS 2011.
[2] J.Heikkinen, T. Minav and A. D. Stotckaia, "Self-tuning parameter fuzzy PID controller for autonomous differential drive mobile robot," *2017 XX IEEE International Conference on Soft Computing and Measurements (SCM)*, St. Petersburg, 2017, pp. 382-385, doi: 10.1109/SCM.2017.7970592.
[3] Yang, Simon & Hu, Tiemin. (2002). An Efficient Neural Network Approach to Real-time Control of a Mobile Robot with Unknown Dynamics. Differential Equations and Dynamical Systems. 10.
[4] Nascimento, Tiago & Dorea, Carlos & Gonçalves, Luiz. (2018). Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots: A modified approach. International Journal of Advanced Robotic Systems.
[5] Héctor M Becerra, J Armando Colunga, Jose Guadalupe Romero.(2018). Simultaneous convergence of position and orientation of wheeled mobile robots using trajectory planning and robust controllers.International Journal of Advanced Robotic Systems
[6] Souanef, Toufik. (2019). Adaptive Guidance and Control of Small Unmanned Aerial Vehicles. 10.13140/RG.2.2.21153.56166.
[7] Yesil, Engin & Guzelkaya, M. & Eksin, Ibrahim. (2003). Fuzzy PID controllers: An overview.
[8] Xu, J. X., Hang, C. C, Liu, C., 2000. Parallel structure and tuning of a fuzzy PID controller. Automatica, 36, 673-684.
[9] Hu, B., Mann, G. K. I., Gosine, R. G., 1999. New methodology for analytical and optimal design of fuzzy PID controllers. IEEE Trans. Fuzzy Systems 7(5), 521-539.
[10] "Kobuki Documentation", relase 2.0, Dabit industries.
[11] Corke, P. "Integrating ROS and MATLAB [ROS Topics]." IEEE Robotics & Automation Magazine 22 (2015): 18-20.
[12] Besseghieur, Khadir Lakhdar AU, Trębiński, RadosławAU, Kaczmarek, Wojciech AU, Panasiuk, Jarosław PY, "Trajectory tracking control for a nonholonomic mobile robot under ROS", Journal of Physics: Conference Series, Volume 1016, 6th International Conference on Mechatronics and Control Engineering (ICMCE 2017) 2–4 December 2017, St. Petersburg, Russian Federation.