2018 IEEE International Conference on Systems, Man, and Cybernetics

Cloud Computing Fuzzy Adaptive Predictive Control for Mobile Robots

Wen-Shyong Yu and Chien-Chih Chen

***Abstract*—The purpose of this paper is on the use of cloud computing for efficiently planning autonomous real-time prespecified trajectory tracking and obstacle avoidance control for an omnidirectional wheeled robot using fuzzy adaptive predictive control algorithm. The autonomous trajectory tracking control includes dynamic simulation, omnidirectional wheeled robot control, and the feedback signal mainly provided by the sensor object surface and depth measurement. The robot is equipped with three independent-driven omnidirectional wheels and six ultrasonic sensors. The Jacobian between Cartesian space corresponding to the joint space of the robot is setup for ellipse motion planning so that it can autonomously follow the prespecified trajectory tracking, obstacle avoidance, and other sports. An architecture is setup to split computation between the remote cloud and the robot so that a robot can interact with a computing cloud. Given this robot/cloud architecture, the stability of the closed-loop control system from the Lyapunov theorem for the fuzzy adaptive predictive control algorithm and trajectory planning is guaranteed with satisfactory performance on the cloud during a periodically updated preprocessing phase efficiently, and manipulation queries on the robots given changes in the workspace can achieve real-time trajectory tracking and obstacle avoidance with ellipse motion planning control. Finally, tradeoffs arising between path quality and computational efficiency are evaluated through simulation, and experiments are given for analyzing the control performance.**

Index Terms: Cloud computing, Omnidirectional wheeled robot, fuzzy adaptive predictive control algorithm, motion dynamics, trajectory tracking, obstacle avoidance.

I. Introduction

With the rapid progress of the technology industry, there have been many studies on researching many types of mobile robots [1]-[3] and tracking controls [4]-[6]. Among them, robots with the omnidirectional [2]-[4] are capable of moving in any direction without the need to reorient themselves, which gives them better maneuverability compared to nonholonomic robots. In development of unmanned vehicles in pedestrian avoidance, path tracking in cycles, and navigation in [1],[3], mobile robots with tracking control [4]-[5] may cause the errors between the desired path and the actual path due to slippage, disturbance, noise, vehicle-terrain interaction, and other factors. Therefore, in order to avoid the errors between the desired and actual robot paths, the is necessary. Recently, several controllers for mobile robots have been proposed [6]-[9], such as fuzzy controller [8]-[9], fuzzy PID controller [10]. In [10], it is proved that the adaptive fuzzy control rules can be automatically modified and used to improve the performance of the control system. Therefore, in this paper, the use of cloud computing for efficiently planning autonomous real-time prespecified trajectory tracking and obstacle avoidance control is proposed for an omnidirectional wheeled robot using fuzzy adaptive predictive control algorithm. The autonomous trajectory tracking control includes dynamic simulation, omnidirectional wheeled robot control, and the feedback signal mainly provided by the sensor object surface and depth measurement. The robot is equipped with three independent-driven omnidirectional wheels and six ultrasonic sensors. The Jacobian between Cartesian space corresponding to the joint space of the robot is setup for ellipse motion planning so that it can autonomously follow the prespecified trajectory tracking, obstacle avoidance, and other sports. An architecture is setup to split computation between the remote cloud and the robot so that a robot can interact with a computing cloud. Given this robot/cloud architecture, the stability of the closed-loop control system from the Lyapunov theorem for the fuzzy adaptive predictive control algorithm and trajectory planning is guaranteed with satisfactory performance on the cloud during a periodically updated preprocessing phase efficiently, and manipulation queries on the robots given changes in the workspace can achieve real-time trajectory tracking and obstacle avoidance with ellipse motion planning control. Finally, tradeoffs arising between path quality and computational efficiency are evaluated through simulation, and experiments are given for analyzing the control performance.

II. Problem formulation

The Omnidirectional wheels can move in all directions due to the manufacture design of wheels by three-axis type and its angle difference between each angle is 120 degrees. The omnidirectional mobile robot has three degrees of freedom with linearly moving in the XY-plane in any transla-

2577-1655/18/$31.00 ©2018 IEEE
DOI 10.1109/SMC.2018.00694

tional directions and Z-axis rotations. Based on the angular velocity, the other two driven wheels will generate different directions of torques, respectively, and then generate the direction of the resultant force to scrolling as shown in Fig. 1.

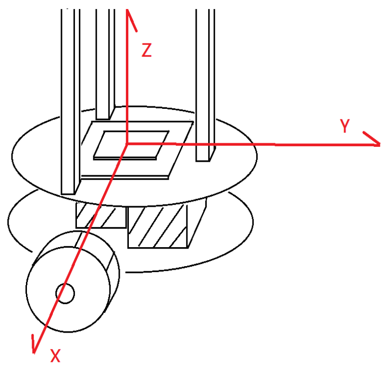

Fig. 1. Schematic diagram of omnidirectional robot

First, the human interface control program in the control computer is mainly responsible for calculating the operation interface control of the kernel algorithm with servo feedback signal from ultrasonic sensors. Through the Bluetooth wireless transmission, the Arduino Mega 2560 receives the corresponding instruction from the control computer and conducts the packet disassembly to the values of speed parameters for Pololu to trigger three DC motors and feedback their corresponding encoders signals to the Arduino. Finally, the ultrasonic sensors parameters and the encoders of the motor speeds will returns to smart phone to make the control inputs.

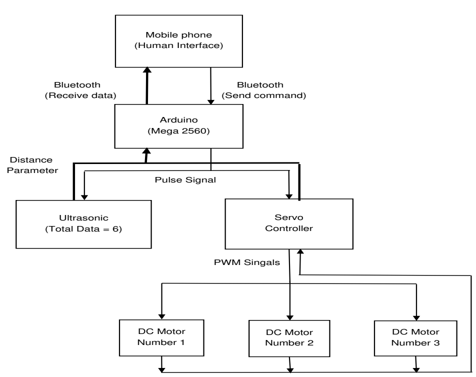

Fig. 2. Schematic diagram of system

The Omnidirectional Robot is shown in Fig. 3. The structure of the Omnidirectional robot includes mobile phone, Arduino, servo controller, motor drivers, three Omni-wheels, three DC motors, and ultrasonic sensors.

Fig. 3. Full view of the robot

The specifications of the Omnidirectional robot are as follows: Its total weight is 12 kg, Height (with tablet) is 132 cm, Height (without tablet) is 110 cm, Diameter (with wheels)is 46.8 cm, Diameter (without wheels) is 35.0 cm, DC-Motor and Encoder, DC-Motor+Encoder *3, Parallax Motor Driver *3, Pololu Maestro Motor Controller *1, Arduino Mega2560 *1, Ultrasonic Parallax Ultrasonic *6 with working range within 2cm to 300cm, Bluetooth Module Bluetooth Module *1 with working range to 10 meters, Battery pack sets 6V *2, 12V *1.

Arduino Mega 2560 is a microcontroller development board with ATmega 2560 and IDE interface for firmware writing. A complete set of packets is composed of 3 Bytes. The first Byte is the Header, the second Byte is Number, which controls the operation of the specified device, and the third Byte is the servo target for controlling the PWM of the output. The servo target controls are within the range of 0 to 254 in decimal. The numeric values are defined as: (1) The motor stop is defined as 127; (2) the servo card output from 0 to 126 is the clockwise rotation of the motor, and the motor will be accelerated by the direction of diminishing the value to 0, and (3) the servo card outputs from 128 to 254 for the counterclockwise rotation of the motor, and the motor will be accelerated by the direction of increasing the value to 254. There are six ultrasonic sensors are located at the circle of the platform with the difference in 60 degrees on the mobile platform. It is used to detect the position of the obstacle for obstacle avoidance.

Using C# WPF (Windows Presentation Foundation) by Microsoft Visual Studio 2010to develop human interface program and C language protocol in the Arduino platform including Bluetooth connection, input trajectory, forward and inverse kinematics, the Arduino receives the commands via Bluetooth and sends to the respective sensors and equipments, and then feedback signals back to the human interface for control. As for the communication protocol, two set of UART serial communication transmission are used for Bluetooth module with Baud rate 9600bps. All commands issue rests with the human interface when the microprocessor receives commands, interpret the packet, and the corresponding actions. A complete packet consists of four Bytes, where the first Byte to perform an action on behalf and defined as Mode: Mode 1 is only for the reception operation, reading the values of the ultrasonic motor angular velocity, and back to the human interface. As for Mode 2, it only controls the motor speed and do not run the return operation. As for Mode 3, it not only runs the motor speed control but also reads the ultrasonic sensors, motor angular velocity value, and return to the human interface. The second Byte to the fourth Byte correspond to the motor 1 to motor 3 control parameters for a given amount of rotation of the motor to the servo control cards Servo target: Byte(1) for Mode, Byte(2) for Motor Speed1, Byte (3) for Motor Speed2, and Byte(4) for Motor Speed3.

The Arduino outputs 5 ms square wave using one prespecified pin in output mode to the ultrasonic sensors, and then this pin is converted into input mode to obtain the parameters which are converted to the actual distance of the robot with the obstacle.

III. Kinematics of the robotic system

First, we define the world coordinates as $[x_w \ y_w]^\top$ and the robot coordinates as $[x_m \ y_m]^\top$, where ϕ is the angle between the robot and the coordinates of the world. The robot and world coordinates are shown in Fig. 4. The transformation matrix between these two coordinates is given by

$$\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \end{bmatrix} = \begin{bmatrix} cos\phi & -sin\phi \\ sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} \tag{1}$$

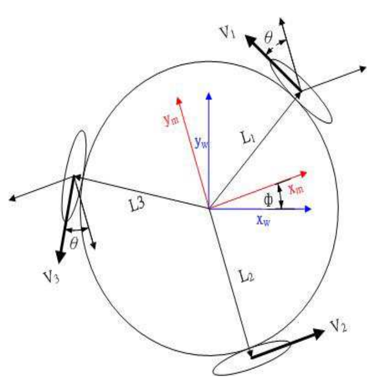

Fig. 4. Relationship Between The Robot and The World Coordinates

From Fig. 4, the omnidirectional wheel robot motion model is given by

$$V_1 = R\dot{\theta_1} = -sin(\theta+\phi)\dot{x}_w + cos(\theta+\phi)\dot{y}_w + L_1\dot{\phi} \tag{2}$$

$$V_2 = R\dot{\theta_2} = cos(\phi)\dot{x}_w + sin(\phi)\dot{y}_w + L_2\dot{\phi} \tag{3}$$

$$V_3 = R\dot{\theta_3} = -sin(\theta-\phi)\dot{x}_w - cos(\theta-\phi)\dot{y}_w + L_3\dot{\phi} \tag{4}$$

where three omnidirectional wheels M_1, M_2, M_3 are separated by locating on the Cartesian coordinate space with differences of $30°$, $270°$, $150°$, L_1, L_2, and L_3 are the center of gravity distance for each wheel, R is the omnidirectional wheel radius, θ_1, θ_2, and θ_3 are the omnidirectional wheel position, and V_1, V_2, V_3 are three wheels speeds. In order to obtain the forward kinematics, dividing by R and integrating by time on both sides of (2), (3), and (4), the relationship of the rotation joint angles and the coordinates in Cartesian system for the robot can be obtained. If the determinant of the coefficient matrix in (1) is not zero, we can obtain the inverse kinematics for $\dot{x}_w$, $\dot{y}_w$, and $\dot{z}_w$ by Cramer's rule.

IV. The Robot Motion Description

Assuming that the angle ϕ between the initial position of the omni-directional robot and the world coordinates is zero. That is, the platform is facing forward $90°$ in the direction of the Cartesian coordinates. In addition, the angle between the initial position of the omni-directional robot and the world coordinates is assumed to be zero, that is, the platform is facing forward 90 degrees in the direction of the dicardian coordinates. When the robot move to y_w coordiantes, M_1 and M_3 are with the same speed and opposite rotation directions as follows:

$$V_1 = -V_3, \ V_2 = 0 \tag{5}$$

When the robot moves to the coordinate x_w, M_1 and M_3 are with the same speeds and directions; while M_2 is in 2-fold speeds and with opposite rotation directions. The relationship for three wheel velocity is given by:

$$V_1 = V_3 = -2V_2 \tag{6}$$

When the robot moves horizontally and vertically at same times, the relationship of the three wheel velocities equation is given by:

$$V_1 + V_2 + V_3 = 0 \tag{7}$$

When the robot is spinning around, M_1, M_2, and M_3 are with the same speeds and directions. The relationship of the three wheel velocities is given by:

$$V_1 = V_2 = V_3 \tag{8}$$

From (5)–(8), when the omnidirectional robot is moving, the rotation of the mobile platform does not change with fix $\phi = 0$. As compared the omnidirectional movements with traditional ones for the robot, the robot only moves with linear motion and need not rotate. Thus, the robot can move with less limitation and more flexible than that of the traditional mobile robots. Therefore, it can exert the characteristics of omni-bearing movements. From (8), we can known the robot does not need large radius of gyration to spinning around in a narrow space. In what follows, linear motion trajectory with obstacle avoidance is designed based on the omnidirectional wheel characteristics (platform not rotating). First, assuming that the robot moves from point A to point B, the initial position is at point A, then the platform does not rotate (ϕ is set to 0), and the resultant direction of the three omnidirectional wheel differentials moves to point B to show the advantages of omnidirectional wheel tracing and the movement of obstacle avoidance. After the original trajectory sampling, the forward speed of each motor is deduced by the forward kinematics to reach the linear non-rotating movement mode. Finally, the actual speed of each motor is obtained by the encoder, and the moving speed $\dot{x}$B$\dot{y}$ and offset angular velocity $\dot{\phi}$, and then move the speed and offset angular velocity on the time integral. The actual movement of the robot distance is can obtained. The control block diagram is shown in Fig. 5.

V. Cloud computing fuzzy adaptive predictive control design

The cloud computing for fuzzy adaptive predictive control algorithm including kinematic model will be applied to control the Omnidirectional wheel robot. Six sensor values and the errors that the reference model is compared with the desired input trajectory is fed back to the cloud for adaptive mechanism to adjust the adaptive controller.

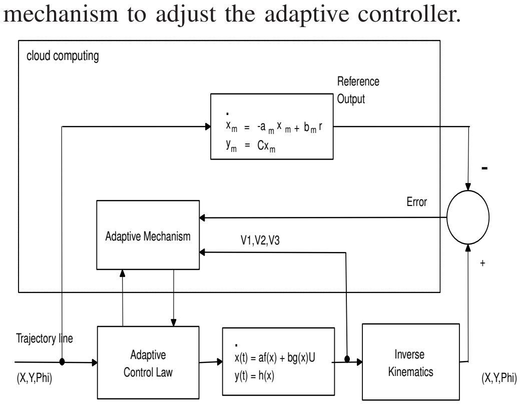

Fig. 5. The structure of the cloud computing fuzzy adaptive predictive control.

Let the T-S model for the nonlinear system be given by

Plant Rule i: IF $\theta_1(t)$ is μ_{i1} and $\theta_2(t)$ is μ_{i2} and $\cdots$ and $\theta_p(t)$ is μ_{ip}, THEN

$$\dot{\mathbf{x}}(t) = \mathbf{A}_i\mathbf{x}(t) + \mathbf{B}_i u(t),\ i = 1, \ldots, r \tag{9}$$

where $\mu_{i1},\ldots,\mu_{ip}$ are fuzzy sets, $\mathbf{A}_i$, $\mathbf{B}_i$ are constant matrices of compatible dimensions, r is the number of IF-THEN rules, and $\boldsymbol{\theta}(t) = [\theta_1(t), \theta_2(t), \ldots, \theta_p(t)]^\top$ is the premise variable vector. Given a pair of $(\mathbf{x}(t), u(t))$, the final output of the fuzzy system is inferred as

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{r} h_i(\theta(t))[\mathbf{A}_i x(t) + \mathbf{B}_i u(t)] \tag{10}$$

where

$$h_i(\theta(t)) = \omega_i(\theta(t)) / \sum_{i=1}^{r} \omega_i(\theta(t))$$

$$\omega_i(\theta(t)) = \prod_{i=1}^{p} \mu_{ij}(\theta_j(t))$$

where

$$\mathbf{A} = \sum_{i=1}^{r} h_i(\theta(t))\mathbf{A}_i \tag{11}$$

$$\mathbf{B} = \sum_{i=1}^{r} h_i(\theta(t))\mathbf{B}_i \tag{12}$$

It is desired that $\mathbf{x}$ tracks state $\mathbf{x}_m$ of the reference model is given by

$$\dot{\mathbf{x}}_m = \mathbf{A}_m\mathbf{x}_m + \mathbf{B}_m r \tag{13}$$

where $\mathbf{x}_m$ is reference model states. For state-feedback sampled data stabilization, only discrete measurements can be used for control purpose, that is, we only have measurements $\mathbf{x}(t_k)$ at the sampling instant t_k with

$$0 = t_0 < t_1 < \cdots < t_k < \cdots < \lim_{k\to\infty} t_k = \infty \tag{14}$$

For the fuzzy model represented by (10), the fuzzy PDC controller shares the same IF parts with the following structure.

Controller Form:

Rule i : IF $\theta_1(t)$ is μ_{i1} and $\theta_2(t)$ is μ_{i2} and $\ldots$ and $\theta_p(t)$ is μ_{ip}, THEN

$$u(t) = K_i x(t_k), t_k \leq t < t_{k+1}, i = 1, \ldots, r. \tag{15}$$

Closed Loop System ($t_k \leq< t_{k+1}$:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(\boldsymbol{\theta}(t))h_j(\theta(t))[\mathbf{A}_i\mathbf{x}(t) + \mathbf{B}_i\mathbf{K}_j\mathbf{x}(t_k)] \tag{16}$$

Assume that the state variable $\mathbf{x}(t)$ is measured at the sampling instant t_k, where t_k satisfies $c = 15ms$, that is, the sampling distances are allowed to change with time but do not exceed 15 ms.

Theorem 1 Consider the nonlinear system (10). There exists a positive definite symmetric matrix $\mathbf{P}$, which satisfies the following Lyapunov equation

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{Q} \tag{17}$$

with $\mathbf{Q}$ being arbitrary $n \times n$ positive definite matrix, and the parameter adaptive laws are given as follows:

$$\begin{cases} \dot{\tilde{\mathbf{a}}}_i^\top = \dot{\hat{\mathbf{a}}}_i^\top = \gamma_{1i} h_i(\mathbf{x}) \mathbf{p}_1^\top \mathbf{e} \mathbf{x}^\top(t), \\ \dot{\tilde{\mathbf{b}}}_i = \dot{\hat{\mathbf{b}}}_i = \gamma_{2i} h_i(\mathbf{x}) \mathbf{p}_1^\top \mathbf{e}(u(t) + \psi(u(t))) \end{cases} \tag{18}$$

if $|\hat{\mathbf{b}}_i| > \mathbf{b}_{i0}$ or if $|\hat{\mathbf{b}}_i| = \mathbf{b}_{i0}$ and $\mathbf{p}_1^\top \mathbf{e} usgn(\mathbf{b}_i) \geq 0$, where $\hat{\mathbf{a}}_i$, $\hat{\mathbf{b}}_i$, and $\mathbf{p}_1$ are the elements of $\hat{\mathbf{A}}_i$, $\hat{\mathbf{B}}_i$, and $\mathbf{P}$, respectively. Furthermore, let the control law be given by

$$u(t) = \sum_{i=1}^{\ell} h_i(\mathbf{x}) \frac{(\mathbf{a}_d^\top - \hat{\mathbf{a}}_i^\top)\mathbf{x}}{m\hat{b}_i} - \psi(u(t)) \tag{19}$$

Then, $\mathbf{e}$ and $\hat{\mathbf{x}}$ as well as the parameter estimation errors $\hat{\mathbf{a}}_i$ and $\hat{b}_i$ are guaranteed to be uniformly ultimately bounded (UUB) for systems with dead zone control inputs. $\square$

VI. Experimental results

The hardware for experiments includes (1) Bluetooth connection: mobile phone and microcontroller build in Arduino can communicate each other, (2) Drawing interface: Provides a block that can be freely drawn to control the omnidirectional wheel, (3) Ultrasonic: An ultrasonic test interface is provided to observe the detected values and the travel status of the mobile robot. The desired trajectory input be mainly divided into two steps: First, when the finger on the screen, read the position X, Y pixel values on the interface from the initial starting coordinates. Second, when the finger continues to slide, program will obtain a second sampling coordinate point. The first and the second position to each other is used to determine the direction to give the robot action command. At the third step, when the mouse button is released, the coordinates of the trajectory is sampled for each time interval and stored in a global variable. The global variable can be used to calculate the velocities and radians for the subsequent program of the algorithm.

After inputting the trajectory in the drawing mode, the omnidirectional wheels will track traces, and at the same time to detect whether there are obstacles around. If omnidirectional wheels detect obstacles, the APP will temporarily stop the user's drawing function and give priority to the function of obstacle avoidance. Obstacle avoidance function will give the elliptical trajectory path to robot to avoid obstacles. After avoiding obstacles, the robot will return to the original track and restart the drawing function.

Each time the robot has completed a small section of the road, getting the angle between the two coordinate and through the inverse kinematics calculation of the robot to determine the direction of movement. There are six ultrasonic sensors on the mobile platform which are located in $0°$, $60°$, $120°$, $180°$, $240°$, $300°$ position.

The detection angle of each sensor is only $11°$. In order to compensate for the invalid area between the sensors, the omnidirectional robot detect each time by rotating 15 degrees to achieve a complete detection as shown in Fig. 6

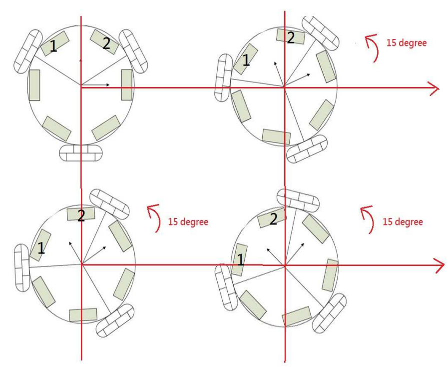

Fig. 6. Omnidirectional Rotating Detection Diagram

When the height of the object is lower than the ultrasound, the ultrasound is still detected. The ultrasound in the vertical surface also has a valid range, this range angle is $43°$ between the range and ground. The height of the object has a relationship with the actual distance. Assuming an object height of 12cm. If the error must be less than fine percentage, the distance between the object and the ultrasound is 76cm or more. If the ultrasonic sensors detect the distance to the obstacle smaller than the the defined safety value, the robot will continue to close the obstacle as well as construct a new elliptical trajectory to achieve obstacle avoidance. The distance from the obstacles feedback by ultrasonic sensors also gives the subroutine to determine if the mobile robot is away from the obstacle. Sensors 1 and 2 are used as the distance parameter for comparisons. The smaller one represents the relative position is closer. Sensors 3 and 6 will be read by the subroutine for comparisons by the same method. The schematic diagram is shown in Fig. 7. Given a straight trajectory and placed two obstacles on the road to Perform the obstacle avoidance experiments. Fig. 8 shows the obstacle avoidance experiments with two obstacles on the front trajectory.

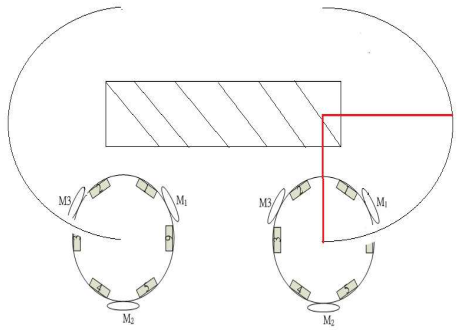

Fig. 7. Relative Position Schematic Diagram

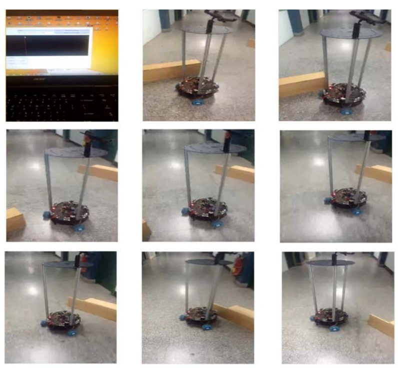

Fig. 8. Obstacle avoidance experiments with fuzzy adaptive controller

VII. Conclusion

In the paper, we have presented the cloud computing for efficiently planning autonomous real-time prespecified trajectory tracking and obstacle avoidance control for an omnidirectional wheeled robot using fuzzy adaptive predictive control algorithm. The autonomous trajectory tracking control includes dynamic simulation, omnidirectional wheeled robot control, and the feedback signal mainly provided the direct/inverse kinematics, the sensor object surface and depth measurement. The robot is equipped with three independent-driven omnidirectional wheels and six ultrasonic sensors. The Jacobian between Cartesian space corresponding to the joint space of the robot is setup for ellipse motion planning so that it can autonomously follow the prespecified trajectory tracking, obstacle avoidance, and other sports. An architecture is setup to split computation between the remote cloud and the robot so that a robot can interact with a computing cloud. Given this robot/cloud architecture, the stability of the closed-loop control system from the Lyapunov theorem for the fuzzy adaptive predictive control algorithm and trajectory planning is guaranteed with satisfactory performance on the cloud during a periodically updated preprocessing phase efficiently, and manipulation queries on the robots given changes in the workspace can achieve real-time trajectory tracking and obstacle avoidance with ellipse motion planning control. Finally, trade-offs arise between path quality and computational efficiency, which are evaluated through simulation, and experiments are given for analyzing the control performance.

Acknowledgment

Financial support of this research by Ministry of Science and Technology, Taiwan, under the grant MOST 106-2221-E-036-001- is gratefully acknowledged.

References

[1] A. Cherubini, F. Spindlerand F. Chaumette, "Autonomous Visual Navigation and Laser-Based Moving Obstacle Avoidance, "IEEE Trans. Intell. Transp. Syst., vol. 15, no. 5, pp. 2101-2110, Oct. 2014.

[2] J. C. L. Barreto, A. G. S. Conceicao, C. E. T. Dorea, L. Martinez, E. R. de Pieri, "Design and Implementation of Model-Predictive Control With Friction Compensation on an Omnidirectional Mobile Robot," IEEE/ASME Trans. Mechatronics, vol. 19, no. 2, pp. 467-465, Feb. 2014.

[3] P. J. Costa, N. Moreira, D. Campos, J. Gonalves, J. Lima, and P. L. Costa, "Localization and Navigation of an Omnidirectional Mobile Robot: The Robot@Factory Case Study," IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 11, no. 1, pp. 1-9, Jan. 2016.

[4] H. Kim, B. K. Kim, "Online Minimum-Energy Trajectory Planning and Control on a Straight-Line Path for Three-Wheeled Omnidirectional Mobile Robots," IEEE Trans. Ind. Electron, vol. 61, no. 9, pp. 4771-4779, Sep. 2014.

[5] H. Vorobieva, S. Glaser, N. M. Enache, S. Mammar, "Automatic Parallel Parking in Tiny Spots: Path Planning and Control," IEEE Trans. Intelligent Transp. Systems, vol. 16, no. 1, pp. 396-410, Aug. 2015.

[6] P. Sun, S. Wang, "Robust fault-tolerant compensation tracking control for omni-directional rehabilitative training walker," 2013 IEEE Int. Conf. Mechatronics and Automation, pp.575-580, Aug. 2013.

[7] J. T. Huang, T. V. Hung, M. L. Tseng, "Smooth Switching Robust Adaptive Control for Omnidirectional Mobile Robots," IEEE Transactions on Control Systems Technology, vol. 23, no. 5, pp. 1986-1993, Jan. 2015.

[8] H. C. Huang, T. F. Wu, C. H. Yu, H. S. Hsu, "Intelligent fuzzy motion control of three-wheeled omnidirectional mobile robots for trajectory tracking and stabilization," 2012 International Conf. Fuzzy Theory and its Applications (iFUZZY2012), pp. 107-112, Nov. 2012.

[9] M. Pena, J. A. Gomez, R. O. Comparan, I. L. Juarez, V. Lomas, H. Gomez, G. Lefranc, "Fuzzy Logic for Omni directional Mobile Platform Control Displacement using FPGA and Bluetooth," IEEE Latin America Transactions, vol. 13, no. 6, pp. 1907-1914, Jul. 2015.

[10] H. T. Abatari, A. D. Tafti, "Using a fuzzy PID controller for the path following of a car-like mobile robot," Robotics and Mechatronics (ICRoM), 2013 First Int. Conf., pp. 189-193, Feb. 2013.