ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 6, No. 4, 348-354 (2021)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Modelling and Simulation of Fuzzy-based Coordination of Trajectory Planning and Obstacle Avoiding for RRP-Typed SCARA Robots

Ngoc-Anh Mai*

Le Quy Don Technical University, Hoang Quoc Viet str. 236, Hanoi, Vietnam

ARTICLE INFO

Article history:
Received: 14 April, 2021
Accepted: 13 August, 2021
Online: 26 August, 2021

Keywords:
Hierarchical chart
Fuzzy-based coordination
RRP-typed SCARA robot

ABSTRACT

In this article, a fuzzy-based solution of coordination between behaviors of trajectory planning and obstacle avoiding in a RRP-typed SCARA robot control is presented. The first idea of the proposed solution is to divide a robot's complex behavior into simpler parallel behaviors. The second key idea is a fuzzy-based coordination between these behaviors to make smooth robot motions without collision. The modelling and simulation on Matlab are executed to test the performance of the proposed solutions under basic circumstances.

1. Introduction

The Selective Compliance Assembly Robot Arm (SCARA robot) was firstly invented in 1978 [1]. Since then, SCARA robot has been developed for different Degree of Freedom (DoF) such as 3-DoF [2], [3], 4-DoF [4], 5-DoF [5,6], and 6-DoF [7], [8]. In particular, SCARA robot with 3-DoF has become one of the most applied industrial robots due to its simple and basic kinematic structures.

In the basic group of 3-DoF industrial robots, there are many different categories of kinematic structures including articulated manipulators [9], Spherical manipulators [10], SCARA manipulators [11], cylinder manipulators [12], cartesian manipulators [13]. Among these categories, RRP-typed SCARA robots have been strongly invested because of flexible trajectory planning solutions with rapidly exploring random algorithms.

Fuzzy logic is an intelligent control tool that simplifies the complexity of nonlinear control through IF-THEN rules. Thanks to this advantage, many fuzzy logic solutions are proposed for controlling 3-DoF SCARA robot such as trajectory tracking control [2], position control [14], path planning control [15], obstacle avoidance [16], [17]. The Matlab simulation results of [2] show that it is possible to apply fuzzy logic for the RRP-typed SCARA controller to reduce the loop trajectory errors. From the result in [14], it is proven that the designed fuzzy logic controller help the RRP SCARA robot move smoother for tracking trajectory but without obstacle avoidance. Similarly, the proposed design in [15] confirms that fuzzy logic application makes the RRP-typed SCARA robot movement faster than the conventional PD controller in path planning. Furthermore, [16] and [17] proposed fuzzy-based solutions for obstacle avoidance of the robotic manipulators. The simulation results demonstrate that fuzzy-based obstacle avoidance provides SCARA robots better solutions of local planning without any collisions. The computational complexity, however, increases due to the repeated use of the nonlinear functions of the fuzzy logic.

Different to the above-mentioned results, in [18] a fuzzy-based basic solution is proposed for coordinating obstacle avoidance and path planning behaviors of 6-DoF humanoid mobile robots. The main ideas of the solution is to divide a complex robot behavior into simpler behaviors and organize the behaviors in a hierarchical chart so that an upper class behavior consists of some behaviors in the lower class. The main ideas in [18] is reused in this research but for a 3-DoF robot to reduce the computational complexity of the robot control system. The proposed fuzzy-based coordination between parallel behaviors makes the robot movement smooth according to the changing obstacle distance. The article is presented as follows: First, the kinematic structure of RRP-typed SCARA robot is introduced. Then, a hierarchical chart of behaviors of trajectory planning and obstacle avoidance are analyzed. After that, a modular diagram of SCARA robot control system is given with the separable modules for trajectory planning and collision avoidance and fuzzy-based coordination. Finally, the Matlab simulations are analyzed to test the system performance.

*Corresponding Author: Ngoc-Anh Mai, Email: maingocanh.atc@mta.edu.vn

2. Kinematics

2.1. Kinematic structure

The structure of RRP-typed SCARA robot with 3-DoF includes 2 revolute (R) (or rotating joints) and one prismatic (P) (or translating joint) as shown on the left side in Fig.1.

These joints are operated by 3 independent actuators to control the pose (including position and direction) of the End-Effector (E) following a desired trajectory.

The basic structure of RRT-typed SCARA robot consists of 4 parts: Base B0, revolute joint R1, revolute joint R2 and prismatic joint P3 as shown on the right side in Fig. 1.

The kinematic parameters in Fig. 2 are defined as follows:

+ d_0 is the height from base B0 to joint R1 along Z_0 axis;

+ a_1 and a_2 are the lengths of the arms 1 and 2 along axes X_1 and X_2, respectively;

+ θ_1 and θ_2 are the rotation angles of joints R1 and R2 around axes Z_1 and Z_2, respectively;

+ d_3 is the distance from joint P3 to the end-effector E.

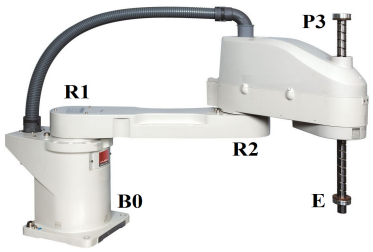

Figure 1: Basic structure of RRT-typed SCARA robot

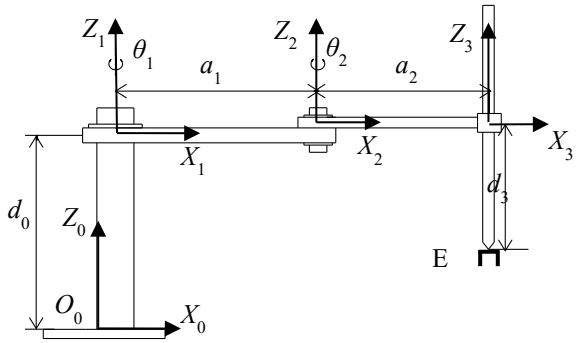

Figure 2: Kinematic structure of the SCARA robot

2.2. Homogeneous transformation

According to the Denavit-Hartenberg (D-H) of homogeneous transformation rules in [19], the D-H parameters of the SCARA robot can be setup in the table shown in Table 1. Using the D-H parameter table, the transformation matrices are as follows:

$$\mathbf{T}_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\mathbf{T}_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\mathbf{T}_3^2 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where:

- $\mathbf{T}_1^0$, $\mathbf{T}_2^1$, $\mathbf{T}_3^2$ are respectively matrices of homogeneous transformation between the coordinate systems from $OXYZ_0$ to $OXYZ_1$, from $OXYZ_1$ to $OXYZ_2$, and from $OXYZ_2$ to $OXYZ_3$.

- c_1, s_1, c_2, s_2 are respectively the symbols of $\cos(\theta_1)$, $sin(\theta_1)$, $\cos(\theta_2)$, $sin(\theta_2)$.

To calculate the homogeneous transformation matrix from the origin O_0 to the end-effector E, we perform the matrix multiplication as follows:

$$\mathbf{T}_3^0 = \mathbf{T}_1^0\mathbf{T}_2^1\mathbf{T}_3^2 = \begin{bmatrix} \mathbf{R}_3^0 & \mathbf{p}_3^0 \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (4)$$

The detail computation of equation (4) is carried out as follows:

$$\mathbf{T}_1^0\mathbf{T}_2^1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{T}_2^0 \quad (5)$$

$$\mathbf{T}_2^0 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 \cdot c_1 \\ s_{12} & c_{12} & 0 & a_1 \cdot s_1 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)\mathbf{T}_2^0\mathbf{T}_3^2 =$$

$$\begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 \cdot c_1 \\ s_{12} & c_{12} & 0 & a_1 \cdot s_1 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$\mathbf{T}_3^0 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 \cdot c_1 + a_2 \cdot c_{12} \\ s_{12} & c_{12} & 0 & a_1 \cdot s_1 + a_2 \cdot s_{12} \\ 0 & 0 & 1 & d_0 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$\mathbf{R}_3^0 = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$\mathbf{p}_3^0 = \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \begin{bmatrix} a_1 \cdot c_1 + a_2 \cdot c_{12} \\ a_1 \cdot s_1 + a_2 \cdot s_{12} \\ d_0 - d_3 \end{bmatrix} \quad (10)$$

Table 1: Denavit-Hartenberg parameters of the SCARA robot

Joint	θ_i	α_i	a_i	d_i
1	θ_1^*	0	0	d_0
2	θ_2^*	0	a_1	0
3	0	0	a_2	$-d_3^*$
* means variable				

where

- $\mathbf{T}_3^0, \mathbf{R}_3^0, \mathbf{p}_3^0$are transformation matrix, orientation matrix, and position matrix of the end-effector E in comparison with base B0.

- c_{12}, s_{12} notes for $\cos(\theta_1 + \theta_2), sin(\theta_1 + \theta_2)$, respectively.

- x_E, y_E, z_E are respectively the coordinates of the end-effector E projected on the axes X, Y, Z of $OXYZ_0$.

2.3. Inverse kinematic computation

According to [20], the inverse kinematics problem consists of the determination of the joint variables corresponding to a given end-effector position and orientation. The solution to this problem is of fundamental importance in order to transform the motion specifications, assigned to the end-effector in the operational space, in to the corresponding joint space motions that allow an execution of the desired motion.

In this study, the inverse kinematics problem requires to find out the 2 joint variables concerning angles θ_1, θ_2 and the displacement d_3 based on the given pose of the end-effector E, that means $\mathbf{p}_3^0$ and $\mathbf{R}_3^0$ are known.

To solve this problem, the top-viewed projections of the SCARA robot, as shown in Figure 3, allows formula (10) to mathematically express in a different way as follows:

$$x_E^2 + y_E^2 = (a_1 \cdot c_1 + a_2 \cdot c_{12})^2 + (a_1 \cdot s_1 + a_2 \cdot s_{12})^2$$
$$= a_1^2 + a_2^2 + 2 \times a_1 \times a_2 \times (c_1 \times c_{12} + s_1 \times s_{12}) x_E^2 +$$
$$y_E^2 = a_1^2 + a_2^2 + 2 \cdot a_1 \cdot a_2 \cdot c_2 \quad (11)$$

From formula (11) we have:

$$c_2 = \frac{x_E^2 + y_E^2 - a_1^2 - a_2^2}{2 \cdot a_1 \cdot a_2} \quad (12)$$

$$\theta_2 = \operatorname{acos}\left(\frac{x_E^2 + y_E^2 - a_1^2 - a_2^2}{2 \cdot a_1 \cdot a_2}\right) \quad (13)$$

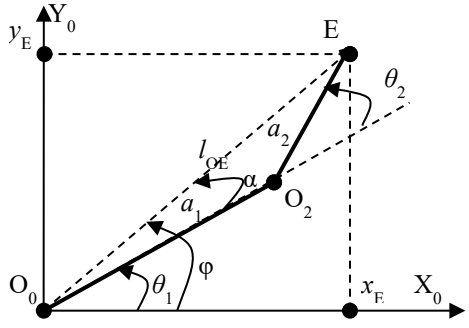

Figure 3: Top-viewed Projections on plane $OXYZ_0$

To calculate angle θ_1, the geometric calculation method can be used under the projections on the plane $OXYZ_0$ as follows:

Let consider triangle O_0O_2E and use mathematic computations:

+ The length from O_0 to E is calculated:

$$l_{OE} = \sqrt{(x_E^2 + y_E^2)} \quad (13)$$

+ The angle α between O_0O_2 and O_0E is calculated as follows:

$$\alpha = \operatorname{acos}\left(\frac{({a_1}^2 + l_{OE}^2 - {a_2}^2)}{2 \cdot a_1 \cdot l_{OE}}\right). \quad (14)$$

+ Angle θ_1 for rotating link 1 around axis Z1 is calculated:

$$\theta_1 = \varphi - \alpha = \operatorname{atan}\left(\frac{y_E}{x_E}\right) - \operatorname{acos}\left(\frac{({a_1}^2 + l^2 - {a_2}^2)}{2 \cdot a_1 \cdot l}\right). \quad (15)$$

Using the bottom expression in formula (10) we get:

$$d_3 = d_0 - z_E. \quad (16)$$

Let draw from equations (15), (13) and (16) we have:

$$\begin{bmatrix}\theta_1 \\ \theta_2 \\ d_3\end{bmatrix} = \begin{bmatrix}\operatorname{atan}\left(\frac{y_E}{x_E}\right) - \operatorname{acos}\left(\frac{({a_1}^2 + l^2 - {a_2}^2)}{2 \cdot a_1 \cdot l}\right) \\ \operatorname{acos}\left(\frac{x_E^2 + y_E^2 - a_1^2 - a_2^2}{2 \cdot a_1 \cdot a_2}\right) \\ d_0 - z_E\end{bmatrix} \quad (17)$$

In summary, based on formula (17) it is possible to simulate RRT-typed SCARA robot movement by giving the coordinates of waypoints (or path points) to form a desired trajectory.

In the next section, the calculation based on these waypoints for planning trajectory is presented. After that, the calculation for avoiding obstacle is analyzed. Based on these calculations, both kinds of behaviors concerning planning trajectory and avoiding obstacle are organized in a hierarchical chart before applying fuzzy logic for coordinating them.

3. Planning trajectory and avoiding obstacle

3.1. Planning trajectory

According to [19], to reduce the complexity of motion control, a complex path can be replaced with a sequence of *n* waypoints to ensure motion along the desired trajectory$[p_1..p_i, p_j..p_n]$.

To travel over all *n* waypoints at certain instants of time, it should be better to compute the actuating commands $\mathbf{v}_T = (v_T, \omega_T)$ for tracking each segment of the desired trajectory between two adjacent waypoints (p_i, p_j) by assigning the initial and final velocities.

Let briefly depict the computation by interpolating polynomials in a segment of trajectory between two adjacent waypoints (p_i, p_j) at two instants of time t_i and t_j as follows:

It is noticed that, if the order of interpolating polynomials increase, the nature of the desired trajectories is reduced and the numerical accuracy for computation polynomial coefficients decreases. For this reason, cubic polynomials are chosen in the following computation.

The generic equation of interpolating polynomials with velocity constraints at the two waypoints are:

$$s(t) = k_0 + k_1 \cdot t + k_2 \cdot t^2 + k_3 \cdot t^3. \quad (18)$$

$$v(t) = \dot{s}(t) = k_1 + 2k_2 \cdot t + 3k_3 \cdot t^2. \quad (19)$$

where

+ $s(t)$ is a cubic polynomial of the path;

+ $v(t)$ is a cubic polynomial velocity respected to $s(t)$;

+ k_i, i = 0...3, are polynomial coefficients depended on the arbitrary motion parameters at the two considering waypoints.

To solve the polynomials, the user has to specify the desired velocities at each points as follows:

+ at the initial time t_i at p_i: the position value is given $s(t_i) = \theta_i$; the velocity value is given $\dot{s}(t_i) = \dot{\theta}_i$;

+ at the final time t_j at p_j: the position value is given $s(t_j) = \theta_j$; the velocity value is given $\dot{s}(t_j) = \dot{\theta}_j$.

Equations (18) and (19) are defined as follows:

$$\theta_i = k_0 + k_1 \cdot t_i + k_2 \cdot {t_i}^2 + k_3 \cdot {t_i}^3. \quad (20)$$

$$\dot{\theta}_i = k_1 + 2 \cdot k_2 \cdot t_i + 3 \cdot k_3 \cdot {t_i}^2. \quad (21)$$

$$\theta_j = k_0 + k_1 \cdot t_j + k_2 \cdot t_j^2 + k_3 \cdot t_j^3. \quad (22)$$

$$\dot{\theta}_j = k_1 + 2 \cdot k_2 \cdot t_j + 3 \cdot k_3 \cdot t_j^2. \quad (23)$$

Expressing equations from (20) to (23) in a matrix form, we have the following matrix:

$$\begin{bmatrix} \theta_i \\ \dot{\theta}_i \\ \theta_j \\ \dot{\theta}_j \end{bmatrix} = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 \\ 0 & 1 & 2 \cdot t_i & 3 \cdot t_i^2 \\ 1 & t_j & t_j^2 & t_j^3 \\ 0 & 1 & 2 \cdot t_j & 3 \cdot t_j^2 \end{bmatrix} \cdot \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix}. \quad (24)$$

To determine the polynomial coefficients k_i, equation (24) can be changed into the following matrix:

$$\begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 \\ 0 & 1 & 2 \cdot t_i & 3 \cdot t_i^2 \\ 1 & t_j & t_j^2 & t_j^3 \\ 0 & 1 & 2 \cdot t_j & 3 \cdot t_j^2 \end{bmatrix}^{-1} \begin{bmatrix} \theta_i \\ \dot{\theta}_i \\ \theta_j \\ \dot{\theta}_j \end{bmatrix} =\cdot \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix}. \quad (25)$$

Equation (25) enables the control system planning the desired trajectory independently in pairs of waypoints. The computed velocity command at the waypoints has the following form:

$$\mathbf{v}_T = (v_T, \omega_T). \quad (26)$$

where

- $v_T$is linear velocity for tracking the planned trajectory.

- ω_T is angular velocity for tracking the planned trajectory.

The detail calculation of the velocities at the given waypoints can be seen in [20].

3.2. Avoiding obstacle

In the scope of educational goal, let consider an obstacle with a cylinder shape shown in Figure 4. The obstacle stays in segment of path assuming (p_i, p_j). In this situation, the obstacle have to be avoided by following an additional path to make a suitable curvature in segment (p_i, p_j).

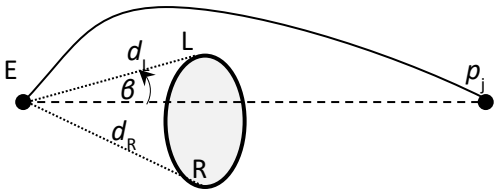

Figure 4: Additional path for avoiding obstacle

Let note the distance to the left obstacle edge L be d_L, the distance to the right edge R be d_R. and the angle deviation β from the direction to p_j to the nearest obstacle edge.

The actuating command for avoiding obstacle are follows:

$$\mathbf{v}_A = (v_A, \omega_A). \quad (27)$$

$$v_A = f_{vA}(min(d_L, d_R) . cos(\beta)). \quad (28)$$

$$\omega_A = f_{\omega A}\left(\frac{sin(\beta)}{min(d_L, d_R)}\right).\text{sign}(d_L - d_R). \quad (29)$$

where

- v_A and ω_A are linear velocity and angular velocity for avoiding obstacle, respectively.

- f_{vA} and $f_{\omega A}$ are user-defined non-linear functions related to d_L, d_R and β for calculating the linear velocity and angular velocity, respectively.

4. Modelling the system with fuzzy-based coordination

Let call the behaviors be planning trajectory PT and avoiding obstacle AO. Then, the AO behavior is divided into 2 simpler behaviors Turning-Left (TL) and Turning-Right (TR) to control the end-effector avoiding a collision by moving to the left or the right edge, respectively. These behaviors are organized in a hierarchical chart shown in Figure 5 and the robot control system can be designed in modules as the block diagram shown in Figure6.

Module PT defines a behavior for planning trajectory and provides a motion command $\mathbf{v}_T$ to module FC.

Module AO selects a suitable behavior between behavior TL and TR based on the results of obstacle detection and measurement in module DM. Then, an actuating command $\mathbf{v}_A$ for avoiding obstacle is computed and sent to module FC.

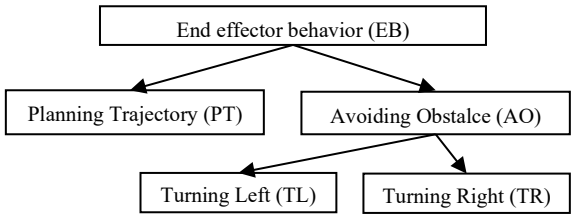

Figure 5: Hierarchical chart for the end-effector behavior

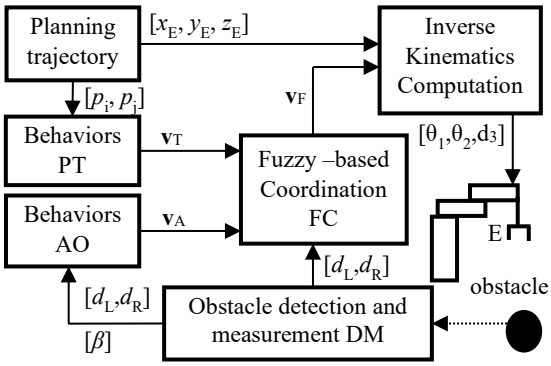

Figure 6: Block diagram of the control system with fuzzy-based coordination

Module FC carries out a fuzzy-based coordination between two behaviors $\mathbf{v}_T$ and $\mathbf{v}_A$ to give a suitable actuating command so that the end-effector can avoid the obstacle and reach to path point p_j.

The fuzzy-based behavioral coordination mechanism is computed as follows:

$$\mathbf{v}_E = (v_E, \omega_E). \tag{30}$$

$$\begin{cases} v_E = k_T \cdot v_T + k_c \cdot v_A \\ \omega_E = k_T \cdot \omega_T + k_c \cdot y_A \end{cases}. \tag{31}$$

where k_T and k_C are weights of tracking trajectory and avoiding collision, which are determined by the fuzzy rules as in Table 2.

Table 2: Fuzzy-based estimation of k_T and k_C

Fuzzy-based values of k_T and k_C		β		
		small	*medium*	*big*
min (d_L,d_R)	*far*	k_T big, k_C big	k_T big, k_C medium	k_T big, k_C small
	middle	k_T medium, k_C big	k_T medium, k_C medium	k_T medium, k_C small
	near	k_T small, k_C big	k_T small, k_C medium	k_T small, k_C small

In the next section, the simulations on Matlab are executed under the guide in [21] and [22] to test the control system with the fuzzy-based coordination.

5. Simulation

5.1. GUI interface of simulation

The robot system is tested by a simulation on Matlab. The robot's motions concerning trajectory and obstacle are tracked and displayed on the GUI interface shown in Figure7.

The left area on the GUI interface displays the coordinates of the three joints concerning the motion and obstacle parameters relative to position and dimensions of height h, long diameter r_1 and short diameter r_2. The middle area of the GUI interface is a 3D illustration of a truth-based trajectory. The right area is a 2D view of the robot's workspace and control buttons.

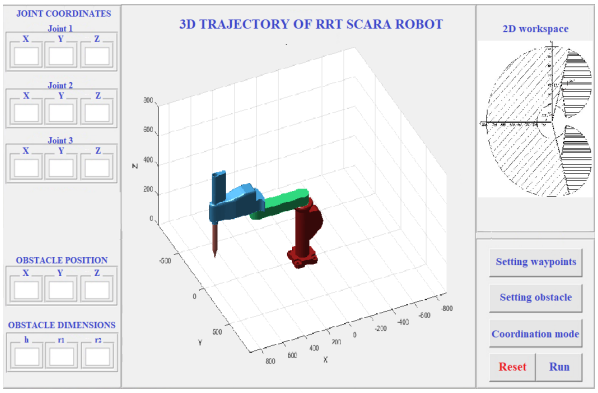

Figure 7: GUI interface of the simulation on Matlab

The five control functions are:

+ "Setting waypoints" button is used for setting the coordinates of each waypoint.
+ "Setting obstacle" button is used for setting the position and dimensions of the obstacle.
+ "Coordination mode" button is used for selecting a coordination mode including conventional coordination without fuzzy logic rules and fuzzy-based coordination.
+ "Reset" button is used for resetting all parameters.
+ "Run" button is used for starting the program.

5.2. Simulation goals and results

Two simulation goals are to test the conventional coordination without fuzzy rules to avoid obstacles and to test the capability of obstacle avoidance under the fuzzy-based coordination.

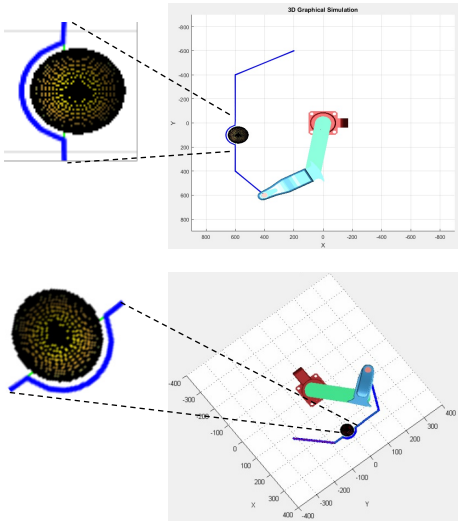

Figure 8: Avoiding obstacle without fuzzy-based coordination

The first simulation results are shown in Figure 8. During following the given waypoints, the robot avoids the obstacle without fuzzy-based coordination. The second simulation results are shown in Figure 9. During following the given waypoints, the robot avoids the obstacle without fuzzy-based coordination.

It is noticed that, the recoded trajectories in Figure 8 are not smooth at the points for changing from a behavior of trajectory planning to a behavior of obstacle avoiding. Otherwise, the recoded ones in Figure 9 become smooth at the points for changing

from a behavior of trajectory planning to a behavior of obstacle avoiding due to the fuzzy-based coordination.

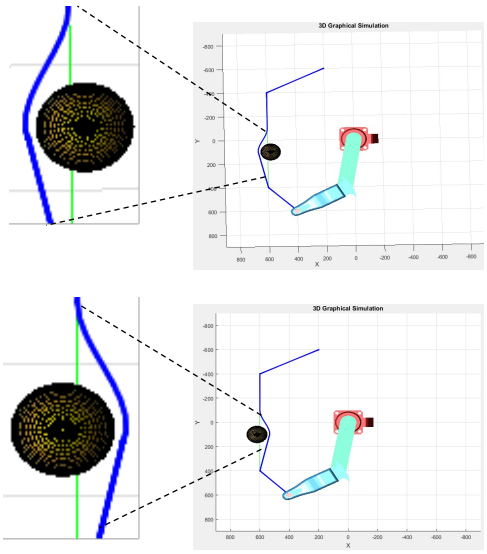

Figure 9: Avoiding obstacle with fuzzy-based coordination

The simulation results in Figure 9 prove that the fuzzy-based coordination make the robot motions non-stop before avoiding obstacle. That means the singularity problem of robotic manipulators can be avoided.

6. Conclusion

The proposed fuzzy-based coordination help the robot safely move and avoid the singularity problem of robotic manipulators.

Matlab simulations are performed to test the performance of the proposed system. Simulation results demonstrate that fuzzy-based coordination helps the robot move more smoothly to eliminate singularity that may appear at dead-points.

In further research, the proposed solution will be tested on a real robot to evaluate the accuracy of trajectory planning after applying the fuzzy-based coordination.

Conflict of Interest

The author declares that there are no conflicts of interest.

References

[1] K. Yamafuji, "Development of SCARA robots," Journal of Robotics and Mechatronics, **31**(1), 10–15, 2019, doi:10.20965/jrm.2019.p0010.

[2] S.M. Raafat, S.M. Mahdi, "Improved Trajectory Tracking Control for a Three Axis SCARA Robot Using Fuzzy Logic," In IJCCCE, **16**(January), 11–19, 2016.

[3] I.S. Karem, T.A.J. A.Wahabt, M.J. Yahyh, "Design and Implementation for 3-DoF SCARA Robot based PLC," Al-Khwarizmi Engineering Journal, **13**(2), 40–50, 2017, doi:10.22153/kej.2017.01.002.

[4] U.T. Nasional, C. Engineering, "Design and development of 4-DoF SCARA robot for educational purposes," Teknologi, **54**(1), 193–215, 2011, doi:10.11113/jt.v54.810.

[5] F. Cao, J. Liu, C. Zhou, Y. Zhao, Z. Fu, W. Yan, "A Novel 5-DOF Welding Robot Based on SCARA," in in 10th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2016–2019, 2016, doi:10.1109/ICIEA.2015.7334444.

[6] M. Abu Qassem, I. Abuhadrous, H. Elaydi, "Simulation and Interfacing of 5 DOF Educational Robot Arm," in Proceedings - 2nd IEEE International Conference on Advanced Computer Control, ICACC 2010, 569–574, 2010, doi:10.1109/ICACC.2010.5487136.

[7] N.A. Mai, X.B. Duong, "Algorithm for improving feeding rates of industrial welding robot TA 1400 in combination with a turntable frame," Computer Science and Cybernetics, **3**(3), 285–294, 2020, doi:10.15625/1813-9663/36/3/14968.

[8] N.A. Mai, X.B. Duong, "Voice Recognition and Inverse Kinematics Control for a Redundant Manipulator Based on a Multilayer Artificial," Hindawi Robotics, **2021**, 1–10, 2021, doi:10.1155/2021/5805232.

[9] E.C. Agbaraji, H.C. Inyiama, C.C. Okezie, "Dynamic Modeling of a 3-DOF Articulated Robotic Manipulator Based on Independent Joint Scheme," Physical Science International Journal, **15**(1), 1–10, 2017, doi:10.9734/PSIJ/2017/33578.

[10] T. Taunyazov, M. Rubagotti, A. Shintemirov, "Constrained Orientation Control of a Spherical Parallel Manipulator via Online Convex Optimization," IEEE/ASME Transactions on Mechatronics, **23**(1), 252–261, 2018.

[11] K. Wei, B. Ren, "A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm," MDPI Special Issue Smart Sensors for Mechatronic and Robotic Systems, **18**(2), 571–585, 2018, doi:10.3390/s18020571.

[12] A. Prasad, B. Sharma, J. Vanualailai, "Motion Control of a Pair of Cylindrical Manipulators in a Constrained 3-dimensional Workspace," in 4th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), 75–81, 2017, doi:10.1109/APWConCSE.2017.00022.

[13] A. Suarez, M. Perez, G. Heredia, "Cartesian Aerial Manipulator with Compliant Arm," MDPI Special Issue Aerial Robotics for Inspection and Maintenance, **11**(3), 1001–1020, 2021.

[14] M.Z.A.- Faiz, A.M. Abbass, "Design and Implementation of Fuzzy Logic Controller for IVAX SCARA Robot Using Real Time Window Target," JCSET, **3**(10), 373–381, 2013.

[15] D. Prabu;, S. Kumar;, R. Prasad, "Dynamic Control of Three-Link SCARA Manipulator using Adaptive Neuro Fuzzy Inference System," in 2008 IEEE International Conference on Networking, Sensing and Control, 1609–1614, 2008.

[16] P.G.. Zavlangas, S.G. Tzafestas, "Industrial Robot Navigation and Obstacle Avoidance Employing Fuzzy Logic," Intelligent & Robotic Systems, **27**(1), 85–97, 2000, doi:10.1023/A:1008150113712.

[17] Y. Chen, Y. Wang, "Obstacle avoidance path planning strategy for robot arm based on fuzzy logic," in 12th International Conference on Control Automation Robotics & Vision (ICARCV), 5–7, 2012, doi:10.1109/ICARCV.2012.6485438.

[18] H.C. Nguyen, H.X. Nguyen, N.A. Mai, L.B. Dang, H.M. Pham, "A Modular Design Process for Developing Humanoid Mobile Robot Viebot," Advances in Science, Technology and Engineering Systems Journal (ASTES), **3**(4), 230–235, 2018.

[19] R.M. Murray, A Mathematical Introduction to Robotic Manipulation, CRC press, 1994.

[20] L. Sciavicco, B. Siciliano, Modelling and Control of Robot Manipulators, Springer, 2000.

[21] S. Shrivastava, "Matlab guide for forward kinematic calculation of 3 to 6 DoF SCARA robots," IJRET, **6**(8), 46–52, 2017, doi:10.15623/ijret.2017.0609009.

[22] M.F. Aly, A.T. Abbas, "Simulation of obstacles ' effect on industrial robots ' working space using genetic algorithm," King Saud University – Engineering Sciences, **26**, 132–143, 2014, doi:10.1016/j.jksues.2012.12.005.