Dynamic Filtered Path Tracking Control for a 3RRR Robot Using Optimal Recursive Path Planning and Vision-Based Pose Estimation

BA-PHUC HUYNH\textsuperscript{1,2} AND YONG-LIN KUO\textsuperscript{2,3}
\textsuperscript{1}Faculty of Engineering and Technology, Kien Giang University, Kien Giang 920000, Vietnam
\textsuperscript{2}Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
\textsuperscript{3}Center of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
Corresponding author: Yong-Lin Kuo (kuo@mail.ntust.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-011-150.

ABSTRACT This study proposes a dynamic filtered path tracking control schema for a 3RRR planar parallel robot to reduce the positioning errors caused by various sources such as backlash in the mechanical system, system nonlinearities, uncertainties, and unknown disturbances. The optimal recursive algorithm is implemented absolutely in path planning based on the optimization problem. Compared to other methods, this algorithm not only helps the robot avoid singularities but also ensures the shortest movement distance of links thanks to the optimization in path planning. The vision system using the 3D Intel RealSense camera D435i is proposed to provide the visual measurement as feedback for the pose estimation of the 3RRR planar parallel robot. The unscented Kalman filter is designed to reduce the errors of the pose estimation due to the camera’s intrinsic parameters, the vibration of the mechanical system, and unknown noises. The establishment of important equations and parameters of the unscented Kalman filter is detailed in this study. The simulation, experiment, and comparison are conducted to verify the effectiveness and feasibility of the proposed approaches.

INDEX TERMS 3RRR planar parallel robot, dynamic filtered path tracking control, optimal recursive path planning, vision-based pose estimation, unscented Kalman filter.

I. INTRODUCTION
Robots have become very popular these days, from industrial production to daily activities. Industrial robots with fast speed, high accuracy, durable working ability, and many other outstanding advantages have gradually replaced human labor, contributed to reducing costs and increasing productivity in the manufacturing factories. Serial manipulators with the advantage of the large workspace are the most common industrial robots. However, it has many disadvantages such as low stiffness related to open kinematic structure, accumulated errors during operation, large weight of actuators, and relatively low load performance. Compared to serial manipulators, parallel manipulators have several superior advantages. With high stiffness, fast speed and acceleration, high accuracy, and high carrying capacity, these manipulators have become the solution for applications in the industry [1]–[3].

Therefore, many ideas for the application of parallel manipulators to the fields of robotics and machine tools have been developed [4]–[6]. In this study, the 3RRR robot [7], a type of three-degree-of-freedom (3-DOF) planar parallel manipulator, is selected as the research object to apply the proposed approaches.

Like many closed-loop structures, 3RRR planar parallel robots have the main drawback of limited workspace and complex singularities [8], [9]. Bonev et al. [10] used the crew theory to analyze the singularity of 3-DOF planar parallel robots. The authors in [11] analyzed the singularity to select an optimal structure of the 3-DOF planar parallel robots. However, the papers did not mention controlling robots to avoid singularities. To control the robot, the inverse kinematics problem must be solved to determine the rotation angles of the active links corresponding to a given position of the end-effector (plate). The inverse kinematics problem has eight solutions, which is called the moving state in this study. As the robot moves through a multipoint path, each point has
eight possible moving states, there will be a huge number of combination of moving states to satisfy the desired path. Selecting the optimal moving states to both minimize travel distances and avoid singularities is a problem that needs to be addressed. The optimal design problem was proposed in [12] to determine the best geometry of the 3RRR planar parallel robot to increase the workspace without the singularities for all the eight moving states. Liu et al. [13] and Zhang et al. [14] analyzed the singularity and path planning for a 3RRR planar parallel robot using the working (actuation) mode conversion. The working modes are converted in turn in different schemas. Most of the research analyzed the singularities and proposed a feasible approach for path planning of the 3RRR planar parallel robots. However, most of the works did not solve the optimization for path planning. In this study, the optimal recursive algorithm is proposed for the path planning of a 3RRR planar parallel robot. The algorithm divides the moving points of the desired path into several groups to reduce the number of combinations of moving states. After that, the absolute recursive algorithm is implemented in each group based on the optimization problem, which both minimizes travel distances and avoids singularities. The proposed algorithm ensures optimal and smooth movement of the robot without the need for mechanical structure intervention.

Over the years, many studies have been conducted to improve the ability of the robot to reach the desired pose (position and orientation) accurately. The direct solution of adding the absolute encoders to the revolute joints is highly effective but costly and sometimes difficult to implement thoroughly. Therefore, other solutions including static calibration and dynamic positioning were considered [15]–[17]. Farajtabar et al. [18] modeled the joint clearance as a virtual actuator to compensate for the errors in positioning. The authors in [19] used three sliding mode controllers and a proportional-integral-derivative (PID)-neural network controller for positioning of the 3RRR planar parallel robot in the simulation. It tries to reduce the errors between the links caused by noises and disturbances. In [20], a genetic algorithm was used to optimize a PID controller of the 3RRR planar parallel robot. It is expected to improve the precision of trajectory control. Yao et al. [21] used the vision measurement as feedback to control the position of the 3RRR planar parallel system. It combined an adaptive control algorithm, a fuzzy logic controller, and a PI controller to achieve the accuracy of positioning. [22] used particle swarm optimization (PSO) to optimize the trajectory planning for a robotic work cell.

In general, positioning errors can be caused by various sources such as backlash in the mechanical system, system nonlinearities, uncertainties, and unknown disturbances. Addressing all sources of error will be complex and less effective. Solutions based on visual measurement [23] meet positioning requirements with high accuracy for various trajectories because it addresses directly the pose of the end effector. However, the vision system often has errors due to noises during its operation. Therefore, some studies used the extended Kalman filter (EKF) [24] or adaptive Kalman filter (AKF) [25], [26] to optimize the pose estimation. In this study, the pose estimation of the 3RRR planar parallel robot is performed based on the vision system using the 3D Intel RealSense camera D435i. The unscented Kalman filter (UKF) [27] is proposed to correct the pose estimation, reduce the errors due to the camera’s intrinsic parameters, the vibration of the mechanical system, and unknown noises. Compared to EKF and AKF, UKF uses many points, called sigma points, to linearize a nonlinear model, so it can predict the system state more accurately.

Based on the literature review and the above discussion, the primary contributions of this article involve:

1. The optimal recursive path planning is proposed for the 3RRR planar parallel robot. This method divides the moving points of the desired path into several groups to reduce the number of combinations of moving states, thus speed up the performance. After that, the optimal recursive algorithm is implemented absolutely in each group based on the optimization problem. Compared to other methods, this algorithm not only helps the robot avoid singularities but also ensures the shortest movement distance of links thanks to the optimization in path planning. The proposed approach ensures optimal and smooth movement of the robot without the need for mechanical structure intervention.

2. The vision system using the 3D Intel RealSense camera D435i is proposed to provide the visual measurement as feedback for the pose estimation of the 3RRR planar parallel robot. The UKF is designed for the vision-based pose estimation to reduce the errors due to the camera’s intrinsic parameters, the vibration of the mechanical system, and unknown noises. The establishment of important equations and parameters of UKF is detailed in this study.

3. The dynamic filtered path tracking control schema is proposed for the 3RRR planar parallel robot to reduce the positioning errors caused by various sources such as backlash in the mechanical system, system nonlinearities, uncertainties, and unknown disturbances. The simulation, experiment, and comparison are conducted to verify the effectiveness and feasibility of the proposed approaches.

The rest of this article includes: Section II is the system description and problem statement. Section III presents the optimal recursive path planning. Section IV presents the vision-based pose estimation using the UKF. Section V describes the dynamic filtered path tracking control. Section VI shows the simulation, experiment, and comparison results. Section VII presents the conclusions. Section VIII discusses future improvement.

**II. SYSTEM DESCRIPTION AND PROBLEM STATEMENT**

Fig. 1 is the model of the 3RRR robot used in this study. The robot has three equal kinematics chains, which is described as a serial RRR chain, where R stands for the revolute joint. The fixed base frame {F} is defined at the center of the joint A₁.
The active frame \( \{E\} \) is defined at the center of the plate with the \( X \) axis parallel to \( C_1C_2 \) and the \( Y \) axis toward \( C_3 \). The \( Z \) axis is directed from the base up. The camera frame \( \{C\} \) is defined at the center of the 3D Intel RealSense camera D435i with the \( Z \) axis pointing up. For the \( i^{th} \) kinematic chain, \( i = 1, 2, 3, \theta_i \) is the rotation angle of the active link \( A_iB_i, \phi_i \) is the rotation angle of the passive link \( B_iC_i \). \( l_i \) is the length of the active links. \( l_2 \) is the length of the passive links. \( R \) is the radius of the circle of the joints \( C_i \). The vector \( q = [\theta_1 \theta_2 \theta_3]^T \) defines the joint space vector as the input variable of the robot. The vector \( p = [x \ y \ \alpha]^T \) defines the task space vector as the pose of the plate, where \([x \ y]^T\) is the position of the plate with respect to the frame \( \{F\} \), \( \alpha \) is the rotation angle of the plate. The 3D Intel RealSense camera points to the surface of the plate, captures the image of the colored feature points. It detects and provides the 3D coordinate of the feature points for the pose estimation of the plate. To control motion, the inverse kinematics problem needs to be solved to determine the input variable of the robot based on a given pose of the plate.

### A. INVERSE KINEMATICS PROBLEMS

The coordinate of the motor joints with respect to the frame \( \{F\} \) are:

\[
A = \begin{bmatrix} A_1 & A_2 & A_3 \end{bmatrix} = \begin{bmatrix} x_{A_1} & x_{A_2} & x_{A_3} \\ y_{A_1} & y_{A_2} & y_{A_3} \end{bmatrix} = \begin{bmatrix} 0 & \sqrt{3}R & 0 \\ \sqrt{3}R/2 & \frac{3}{2}R & 0 \end{bmatrix}
\]  

The coordinate of the joints \( C_i \) with respect to the frame \( \{E\} \) are:

\[
\begin{bmatrix} C_1^E \\ C_2^E \\ C_3^E \end{bmatrix} = \begin{bmatrix} -\sqrt{3}r & \sqrt{3}r & 0 \\ \frac{3}{2r} & \frac{1}{2r} & 0 \end{bmatrix}
\]  

The coordinate of the joints \( C_i \) with respect to the frame \( \{F\} \) is expressed as follow:

\[
\begin{bmatrix} C_i^F \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{E}{F}H \end{bmatrix} \begin{bmatrix} C_i^E \\ 1 \end{bmatrix}
\]  

where \( \frac{E}{F}H \) is the homogeneous transformation matrix from the frame \( \{E\} \) to the frame \( \{F\} \):

\[
\frac{E}{F}H = \begin{bmatrix} \cos \alpha & -\sin \alpha & x \\ \sin \alpha & \cos \alpha & y \\ 0 & 0 & 1 \end{bmatrix}
\]

From Equations (2), (3), and (4), the coordinate of the joints \( C_i \) with respect to the frame \( \{F\} \) can be expressed as follow:

\[
\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} x_{C_1} \\ x_{C_2} \\ x_{C_3} \\ y_{C_1} \\ y_{C_2} \\ y_{C_3} \end{bmatrix} = \begin{bmatrix} x - r \cos(\frac{\pi}{6} + \alpha) \ x + r \cos(\frac{\pi}{6} - \alpha) \ x - r \sin \alpha \\ y - r \sin(\frac{\pi}{6} + \alpha) \ y - r \sin(\frac{\pi}{6} - \alpha) \ y + r \cos \alpha \end{bmatrix}
\]

The constraint of the closed-loop vector in each kinematic chain is:

\[
\overrightarrow{FC_i} = \overrightarrow{FA_i} + \overrightarrow{A_iB_i} + \overrightarrow{B_iC_i}
\]
Equation (6) can be rewritten as follow:
\[
\begin{align*}
 x_{Ci} &= x_{Ai} + l_1 \cos \theta_i + l_2 \cos \varphi_i \\
 y_{Ci} &= y_{Ai} + l_1 \sin \theta_i + l_2 \sin \varphi_i
\end{align*}
\] (7)

After solving Equation (7), the input variable of the robot can be obtained as:
\[
\theta_i = 2 \arctan \left( \frac{-k_1 \pm \sqrt{k_{ws}^2 - k_2^2}}{k_3 - k_2} \right) \tag{8}
\]

where:
\[
\begin{align*}
 k_{ws} &= k_1^2 + k_2^2 - k_3^2 \\
 k_1 &= 2l_1(y_{Ai} - y_{Ci}) \\
 k_2 &= 2l_1(x_{Ai} - x_{Ci}) \\
 k_3 &= x_{Ci}^2 + y_{Ci}^2 + l_2^2 - 2x_{Ci}x_{Ai} - 2y_{Ci}y_{Ai} + l_1^2
\end{align*}
\] (9)

**B. WORKSPACE AND SINGULARITY ANALYSIS**

Using the time derivative of Equation (8) yields:
\[
J_p \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = -J_q \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}
\] (10)

where $J_p$ is the forward kinematics Jacobian matrix, $J_q$ is the inverse kinematics Jacobian matrix.

\[
J_p = \begin{bmatrix} BC_{x1} & BC_{y1} & EC_{x1} \cdot BC_{y1} - EC_{y1} \cdot BC_{x1} \\ BC_{x2} & BC_{y2} & EC_{x2} \cdot BC_{y2} - EC_{y2} \cdot BC_{x2} \\ BC_{x3} & BC_{y3} & EC_{x3} \cdot BC_{y3} - EC_{y3} \cdot BC_{x3} \end{bmatrix}
\]

$J_q = \begin{bmatrix} u & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & w \end{bmatrix}$

\[
u = AB_{x1} \cdot BC_{x1} - AB_{y1} \cdot BC_{y1}
\] (12)

\[
v = AB_{x2} \cdot BC_{x2} - AB_{y2} \cdot BC_{y2}
\] (13)

\[
v = AB_{x3} \cdot BC_{x3} - AB_{y3} \cdot BC_{y3}
\] (14)

\[
\begin{bmatrix} AB_{x1} & AB_{x2} & AB_{x3} \\ AB_{y1} & AB_{y2} & AB_{y3} \end{bmatrix}
\] (15)

\[
\begin{bmatrix} l_1 \cos \theta_1 & l_1 \cos \theta_2 & l_1 \cos \theta_3 \\ l_1 \sin \theta_1 & l_1 \sin \theta_2 & l_1 \sin \theta_3 \end{bmatrix}
\] (16)

\[
\begin{bmatrix} BC_{x1} & BC_{x2} & BC_{x3} \\ BC_{y1} & BC_{y2} & BC_{y3} \end{bmatrix}
\] (17)

\[
\begin{bmatrix} r \cos(\frac{\pi}{6} + \alpha) & r \cos(\frac{\pi}{6} - \alpha) & r \sin \alpha \\ r \sin(\frac{\pi}{6} + \alpha) & r \sin(\frac{\pi}{6} - \alpha) & r \cos \alpha \end{bmatrix}
\] (18)

The forward singularity happens when the determinant of $J_q$ is zero. However, when the pose of the plate is outside the workspace, $k_{ws}$ in Equation (9) will have a negative value, Equation (8) cannot be solved, thereby the determinant of $J_q$ is impossible to be calculated. Therefore, in practice, the workspace and inverse singularity is checked by the value of $k_{ws}$ in Equation (9).

**FIGURE 2.** The two types of forward singularity.

The forward singularity happens when the determinant of $J_q$ is zero. Fig. 2 shows two types of forward singularity. In type (a), the centerlines of the passive links pass through the center of the plate. In type (b), three passive links are parallel. In both types of forward singularity, the motors cannot provide enough torque to move the plate. Therefore, the forward singularities must be detected and avoided when planning the path for the robot.

**C. PROBLEM STATEMENT**

In operation, the robot is desired to move point to point following a certain task path with high accuracy. In this study, the desired task path $\mathcal{S}$ of the plate with respect to the frame (F) is defined as follow:

\[
[p_x, p_y, p_\alpha] = [x[n], y[n], \alpha[n]]
\]

where $n = 0, \ldots, N$, $N$ is the total number of points in the path. $n = 0$ indicates the start point, which is also the current point.

The control objective is to control the plate following the desired path $\mathcal{S}$ and track the pose $(p_x, p_y, p_\alpha)$ to minimize the error. To achieve the control objective, this study addresses path planning and pose estimation, then proposes a dynamic filtered path tracking control schema to improve the movement accuracy in real time.

**III. OPTIMAL RECURSIVE PATH PLANNING**

For a given pose $p = [p_x, p_y, p_\alpha]^T$ of the plate, by solving the inverse kinematics problem, the rotation vector $q = [\theta_1 \theta_2 \theta_3]^T$ of the active links are obtained. Each active link has two solutions as expressed in Equation (8). Therefore, there are a total of eight moving states of the active links that can correspond to a specific pose of the plate geometrically. Consider Equation (8), the solution corresponding to the positive sign is solution 1, the solution corresponding to the negative sign is solution 2. The eight moving states are defined as in Table 1.

To ensure a stable movement, the path $\mathcal{S}$ is divided into $N$ equal segments. $\mathcal{S}[0]$ indicates the start point of the path.
TABLE 1. The eight moving states in the path planning.

| Moving state | The solution number |
|--------------|---------------------|
| 1            | 1 1 1               |
| 2            | 1 1 2               |
| 3            | 1 2 1               |
| 4            | 1 2 2               |
| 5            | 2 1 1               |
| 6            | 2 1 2               |
| 7            | 2 2 1               |
| 8            | 2 2 2               |

The path planning will drive the plate across points from $\mathcal{S}[1]$ to $\mathcal{S}[N]$. The points are spaced $\Delta p = (\Delta p_x, \Delta p_y, \Delta p_y)$, where $\Delta p_x$ and $\Delta p_y$ are the distances between two adjacent points in the $X$ and $Y$ axes, respectively, $\Delta p_y$ is the change in the rotation angle of the plate between two adjacent points. In practice, the distance between two adjacent points $\Delta p_t = \sqrt{\Delta p_x^2 + \Delta p_y^2}$ is defined and used. In this study, the Bresenham algorithm [28], [29] is used and modified to divide the path between the start point and the endpoint into $N$ segments contained in the path $\mathcal{S}$ with $(\Delta p_t, \Delta p_y)$ given. By experiment, $\Delta p_t = 10$ mm is selected in this study ($\Delta p_y$ is determined by equally dividing the desired rotation angle of the plate by $N$). Each point in the path $\mathcal{S}$, from $\mathcal{S}[1]$ to $\mathcal{S}[N]$, will be solved by the inverse kinematics problem and singularity analysis to obtain the two-dimensional array $A$. The first dimension of $A$, denoted as $row$, is the index of points, which has the value of zero to $N$. The second dimension, denoted as $col$, is the index of possible combinations of moving states of the active links for each point, which has a value of 1 to 8. Each element of $A$, expressed as $A[\text{row, col}]$, is a record of $q$ and $|\det(J_p)|$.

Each point in the path $\mathcal{S}$ has a total of eight moving states of the active links, so the path $\mathcal{S}$ has a total of $8^N$ possible combinations of moving states of the active links. Each combination needs to be evaluated to select the best states for the movement. The evaluation includes avoiding forward singularity and minimizing the distance traveled by active links. If all $8^N$ combinations are evaluated, the solution is optimal and the evaluation is called the absolute evaluation. However, it is impossible in practice due to the computation time. For example, suppose $N = 12$, there are more than sixty-eight billion possible combinations. It will take a very long time for evaluation and be impossible to apply in real-time control. Therefore, this study proposes a novel approach, which still considers all possible states but reduces the number of evaluated combinations to speed up the calculation.

In this approach, $N$ points contained in the path $\mathcal{S}$ are divided into $G$ groups with $\Delta N$ points in each group. The absolute evaluation will be implemented in each group. It considers all the moving states to select the best states for the movement. Transitions between groups are negligible because the points are a small distance apart. Finally, the movement of the path $\mathcal{S}$ is very smooth and optimal. The total of possible combinations of moving states is $G \times 8^{\Delta N}$. Returning to the above example with $N = 12$, suppose $G = 3$ and $\Delta N = 4$, the total combinations is 12288, which is greatly reduced. The evaluation will be executed very fast and easily in real time.

The evaluation process can be expressed as the following optimization problem:

$$\min \sum_{n=1}^{\Delta N} \sum_{k=1}^{3} |\theta^n_k - \theta^{n-1}_k|$$

subject to: $|\det(J_p)| \geq \zeta, \quad n = 1, \ldots, \Delta N$ (20)

where $\Delta N$ is the number of points in each group, $n$ is the index of the point in the group, $k$ is the index of the kinematic chain. $\zeta$ is the limit value to ensure that the robot does not fall into the forward singularity. $|\theta^n_k - \theta^{n-1}_k|$ presents the distance traveled by the $k^{th}$ active link corresponding to the plate movement between two adjacent points. Note that in each group, $n - 1 = 0$ indicates the start point of the group. For the first group, the start point is the start point $\mathcal{S}[0]$ of the path $\mathcal{S}$. For the other groups, the start point is the endpoint of the previous group. $J_p$ is the forward kinematics Jacobian matrix. $\det(\cdot)$ is the determinant of the matrix.

The optimal recursive path planning is presented in Algorithm 1. The returned parameter is the $N$-element array BestState in which each element contains the index of the best of the eight moving states of the active links at the corresponding point. $\Delta N$ is an optional parameter, which is selected to ensure a balance in computation time and smoothness of the movement. In this study, $\Delta N = 4$ produces the desired results.

IV. VISION-BASED POSE ESTIMATION USING THE UNSCENTED KALMAN FILTER

A. VISION-BASED POSE ESTIMATION

In the pose estimation, the position and rotation angle of the plate is determined with respect to the frame $\{C\}$ by the vision system using the 3D Intel RealSense Camera D435i. After that, the pose with respect to the frame $\{F\}$ will be calculated by using the homogeneous transformation matrix $C_F$ from $\{C\}$ to $\{F\}$. Because the 3RRR robot is a planar robot, the depth $Z$ respect to the frame $\{C\}$ is ignored. The camera is placed at the center of the triangle $A_1A_2A_3$. The homogeneous transformation matrix $C_F$ is as follow:

$$C_F = \begin{bmatrix} 0 & 0 & -\sqrt{3}R \\ 0 & 1/2 & -R \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 3 shows the pose estimation of the plate. The Intel RealSense Camera D435i provides continuous depth image acquisition in real time. The 2D coordinate of feature points
Algorithm 1 Optimal Recursive Path Planning

**Inputs:** $\Delta N$, $\zeta$, $P_0$, $P_N$

1. $[3, N, G] \leftarrow \text{Bresenham\_Algorithm}(P_0, P_N, \Delta N)$
2. $A \leftarrow \text{Inverse\_Kinematics\_Analysis}(3)$
3. $P[0] \leftarrow P_0$
4. $j \leftarrow 1$
5. Initialize ($BestState$, $LocalState$)
6. while ($j \leq N$) do
7. \hspace{1em} $BestCost \leftarrow 1000000$
8. \hspace{1em} $ORA(j, j, \min(j + \Delta N - 1, N))$
9. \hspace{1em} $j \leftarrow j + \Delta N$
10. end while
11. procedure ORA(pos, index, des)
12. \hspace{1em} if ($index > des$)
13. \hspace{2em} $Cost \leftarrow 0$
14. \hspace{2em} $ppos \leftarrow pos - 1$
15. \hspace{2em} if ($ppos > 0$)
16. \hspace{3em} $p_start \leftarrow A[ppos, BestState[ppos]].q$
17. \hspace{2em} else
18. \hspace{3em} $p_start \leftarrow q_0$ corresponding to $P_0$
19. \hspace{2em} end if
20. \hspace{1em} $Cost \leftarrow 0$
21. \hspace{1em} for $m \leftarrow pos$ to $des$
22. \hspace{2em} $n \leftarrow m - pos$
23. \hspace{2em} for $i \leftarrow 1$ to $3$
24. \hspace{3em} $dis[i] \leftarrow | A[m, LocalState[n]].q[i] - p_start[i]|$
25. \hspace{3em} $q_start[i] \leftarrow A[m, LocalState[n]].q[i]$
26. \hspace{2em} end for
27. \hspace{1em} $Cost \leftarrow Cost + dis[1] + dis[2] + dis[3]$
28. end for
29. end if
30. $BestCost \leftarrow Cost$
31. $n \leftarrow 0$
32. for $m \leftarrow pos$ to $des$
33. $BestState[m] \leftarrow LocalState[n]$
34. $n \leftarrow n + 1$
35. end for
36. end if
37. return
38. end if
39. $n \leftarrow index - pos$
40. for $state \leftarrow 1$ to $8$
41. \hspace{1em} if ($\det(J_p) \geq \zeta$)
42. \hspace{2em} $LocalState[n] \leftarrow state$
43. \hspace{2em} $ORA(pos, index + 1, des)$
44. \hspace{1em} end if
45. end for
46. end

Output: $BestState$

The 3D coordinate of feature points can be calculated by using the SDK of the Intel RealSense Camera D435i. Fig. 4 presents the feature points detected in the control software.

The pose of the plate with respect to the frame \{C\} is calculated as follows:

$$
\begin{bmatrix}
  x \\
  y \\
  \alpha
\end{bmatrix}^C = \begin{bmatrix}
  \frac{F_1 x + F_2 y + F_3 z}{3} \\
  \frac{F_1 y + F_2 y + F_3 y}{3}
\end{bmatrix}
$$

where $(F_{ix}, F_{iy})$ is the coordinate of the $i$th feature point with respect to the frame \{C\}.

Finally, the pose of the plate with respect to the frame \{F\} is determined as follows:

$$
\begin{bmatrix}
  x \\
  y \\
  \alpha
\end{bmatrix}^F = \begin{bmatrix}
  \frac{C H}{F} \begin{bmatrix}
    x \\
    y
  \end{bmatrix}^C \\
  \tan(2 \left( F_{2y} - F_{1y}, F_{2x} - F_{1x} \right)
\end{bmatrix}
$$

B. UNSCENTED KALMAN FILTER FOR THE POSE ESTIMATION

The vision-based pose estimation has an error due to the calibration of the camera, the noise in image processing, the vibration of the mechanical system, and unknown disturbances. In this study, the UKF is proposed and configured to reduce the error in the pose estimation before pass it to the dynamic filtered path tracking control. Fig. 5 shows the process of the UKF for the pose estimation.

1) DESIGN THE SYSTEM MODEL

Equation (10) is rewritten as follow:

$$
\begin{bmatrix}
  \dot{x} \\
  \dot{y} \\
  \dot{\alpha}
\end{bmatrix} = J_c \begin{bmatrix}
  \dot{\theta}_1 \\
  \dot{\theta}_2 \\
  \dot{\theta}_3
\end{bmatrix} = J_c \begin{bmatrix}
  \omega_1 \\
  \omega_2 \\
  \omega_3
\end{bmatrix}
$$

$$
J_c = -J_p^{-1} J_q
$$
where \( J_p \) is the forward kinematics Jacobian matrix, \( J_q \) is the inverse kinematics Jacobian matrix. \( J_p \) and \( J_q \) are calculated by Equations (11) to (18). \( \omega_i \) is the angular velocity of the \( i \)th active link of the robot.

The vision system tracks the pose of the plate following the path \( \mathcal{S} \). The state variable is expressed as \( p = [x \ y \ \alpha]^T \). Equation (24) can be rewritten as follow:
\[
\dot{p} = J_c \omega
\]
(25)
where \( \omega = [\omega_1 \ \omega_2 \ \omega_3]^T \) is the angular velocity vector of the active links.

Equation (25) in discrete form is described as follow:
\[
p^{k+1} = p^k + J_c^{k+1} \dot{\omega}^{k+1} \Delta t
\]
(26)
where \( k \) and \( k+1 \) are the current and next instant, respectively. \( \Delta t \) is the discrete interval.

2) DESIGN THE SIGMA POINTS
The dimensionality of the state variable is \( d = 3 \), so there are \( 2d + 1 = 7 \) sigma points required for the filter. The sigma points are computed as follows:
\[
X_0 = \hat{p}
\]
\[
X_{col} = \hat{p} + \sqrt{(d + \lambda)} C_p \text{col}, \quad \text{for col} = 1, \ldots, d
\]
\[
X_{col} = \hat{p} - \sqrt{(d + \lambda)} C_p \text{col}^{d}, \quad \text{for col} = d + 1, \ldots, 2d
\]
(27)
where \( X \) is the sigma points matrix of size \( 3 \times 7 \). \( \hat{p} \) is the mean vector of the state variable. \( \lambda \) is the scaling factor. \( C_p \) is the covariance matrix of size \( d \times d = 3 \times 3 \). The squared root of the matrix \( \sqrt{(d + \lambda)} C_p \) can be obtained by using the Cholesky factorization. The subscript \( \text{col} \) indicates the column of the matrix. For initialization, the covariance matrix \( C_p \) is set by the conjectured standard deviation of elements of the state variable, the mean vector \( \hat{p} \) is set equal to the desired pose of the plate. For other time instants, \( \hat{p} \) and \( C_p \) are the estimation from the end of the previous time instant.

The scaling factor \( \lambda \) is defined as follow:
\[
\lambda = \alpha^2 (d + \kappa) - d
\]
(28)
where \( \alpha \) is the spread of the sigma points around the mean value. \( \kappa = 3 - d \) is the second scaling factor. In this study, \( \alpha = 10 \) and \( \kappa = 0 \).

The weights are computed as follows:
\[
w_m^m = \frac{\lambda}{d + \lambda}
\]
\[
w_m^c = \frac{\lambda}{d + \lambda} + 1 - \alpha^2 + \beta
\]
\[
w_{col}^m = w_{col}^c = \frac{1}{2(d + \lambda)}, \quad \text{for col} = 1, \ldots, 2d
\]
(29)
where \( w_{col}^m \) is the weights for the mean corresponding to the \( \text{col}^{th} \) sigma point. \( w_{col}^c \) is the weights for the covariance corresponding to the \( \text{col}^{th} \) sigma point. The factor \( \beta = 2 \) is the optimal value for Gaussian problems.

3) THE PREDICTION STEP
The predicted mean and covariance are calculated as follows:
\[
\hat{p}' = \sum_{\text{col}=0}^{2d} w_{col}^m \cdot P_{col}
\]
\[
P_{p} = \sum_{\text{col}=0}^{2d} w_{col}^c \cdot \left( (P_{col} - \hat{p}') (P_{col} - \hat{p}')^T \right) + N_p
\]
(30)
where \( \text{col} \) indicates the column of the matrix. \( N_p \) is the process noise covariance matrix of size \( d \times d = 3 \times 3 \). \( P \) is the mapping sigma points matrix. \( g (\cdot) \) is the nonlinear function, which is calculated based on Equation (26).

4) THE UPDATE STEP
The vision system measures the pose of the plate and results in a pose vector \( \mathbf{M} = [x \ y \ \alpha]^T \). This measurement has an error due to the calibration of the camera, the noise in image processing, the vibration of the mechanical system, and unknown disturbances. The difference \( \Delta \mathbf{M} \) between the measurement \( \mathbf{M} \) and the predicted mean \( \hat{p}' \) must be calculated as follow:
\[
\Delta \mathbf{M} = \mathbf{M} - \hat{p}'
\]
(31)
After that, the sigma points need to be transformed into the measurement space based on the difference \( \Delta \mathbf{M} \) by using a function \( h (\cdot) \) as follow:
\[
\mathbf{Z} = h (\mathbf{P})
\]
(32)
where \( \mathbf{Z} \) is the transformed sigma points matrix of a size \( 3 \times 7 \) corresponding to \( \mathbf{P} \). Each column of \( \mathbf{Z} \) can be calculated as follow:
\[
\mathbf{Z}_{col} = P_{col} + \Delta \mathbf{M}, \quad \text{for col} = 1, \ldots, 2d
\]
(33)
The measured mean and covariance are calculated as follows:
\[
\hat{\mathbf{Z}} = \sum_{\text{col}=0}^{2d} w_{col}^m \cdot \mathbf{Z}_{col}
\]
by the UKF as presented in Section IV.B. The corrected pose of the plate is used as the feedback to compensate for the pose error. This process is performed in real time when the robot is running.

In Fig. 6, $p_d$ is the desired pose of the plate corresponding to each point in the desired path. $q_d$ is the desired rotation vector of the active links. $p_d$ and $q_d$ are presented in Section III. The relation between $p_d$ and $q_d$ is as follow:

$$ q_d = T(p_d) $$

where $T$ is the transformation of the inverse kinematics problem presented in Section II.A.

$q_r$ is the real rotation vector of the active links determined by the motion controller. $\omega_d$ is the desired angular velocity of the active links. $\omega_r$ is the real angular velocity determined by the motion controller. $p_e$ is the real pose of the plate after the position control. $p_e$ is the pose estimation of the vision system after corrected by UKF. The relation between $p_r$ and $p_e$ is as follow:

$$ p_e = F(p_r) $$

where $F$ is the apparent function for the vision-based pose estimation and UKF presented in Section IV.

$p_e$ is then compared with $p_d$ to calculate the compensation for the pose error. This is a closed-loop control that is performed repeatedly so that the pose error is controlled within the desired limit.

The verification of the proposed control approach will be presented in the next section.

VI. SIMULATION, EXPERIMENT, AND COMPARISON

A. HARDWARE

Fig. 1 shows the 3RRR robot used in this study. The length of links are $l_1 = 180\text{mm}$, $l_2 = 180\text{mm}$. The other dimensions include $A_1A_2 = A_2A_3 = A_1A_3 = 466\text{mm}$. $C_1C_2 = C_2C_3 = C_1C_3 = 100\text{mm}$. The vision system is the 3D Intel RealSense Camera D435i. The camera has the resolution of $1280 \times 720$, the frame rate of $30\text{fps}$, the depth accuracy of less
than 0.5 percent of the distance from the object. Because the
process interval of the control software is $\Delta t = 50ms$, the real
frame rate of the camera in the program is 20fps. The distance
between the camera and the object is 780mm, so the expected
accuracy of the camera is between 1.8mm to 3.9mm.

The simulation is implemented in MATLAB. The control
software for the experiment is developed in Visual C++ by
our lab. The experiment was conducted on a laptop with
medium hardware included Intel Core i5 9th Gen, 8GB RAM,
SDD 120GB, Graphics card Nvidia GeForce GTX 1050 3GB.

**B. WORKSPACE AND SINGULARITY ANALYSIS**

Fig. 7 to Fig. 9 are the workspace scanning data for three
different rotation angles $\alpha$ of the plate. The red lines are the
connections between the motors. The blue area is where the
plate can reach. As presented in Section II.B, the workspace
can be determined by checking the value of $k_{ws}$ in Equa-
tion (9). When $k_{ws} < 0$ it is out of the workspace. The
workspace is depended on the rotation angle of the plate.

**FIGURE 7.** The workspace of the 3RRR robot ($\alpha = 0^\circ$).

**FIGURE 8.** The workspace of the 3RRR robot ($\alpha = 60^\circ$).

**FIGURE 9.** The workspace of the 3RRR robot ($\alpha = -60^\circ$).

**FIGURE 10.** The singularity scanning for moving state 1 ($\alpha = 0^\circ$).

**FIGURE 11.** The singularity scanning for moving state 2 ($\alpha = 0^\circ$).

Fig. 10 to Fig. 17 are the forward singularity scanning
data of eight moving states corresponding to the rotation
angle $\alpha = 0^\circ$. The black area is the forward singularity area. As presented in Section II.B, the forward singularity occurs when $\det(J_p) = 0$. In practice, the control program will detect the forward singularity when the value of $\det(J_p)$
close to zero. In the figures, the dark black area is very near to the singularity point. The black fades as it gets further away from the singularity point. It can be seen that, at every position in the workspace, there is always at least one state that does not fall into the forward singularity. This is still true when changing the rotation angle $\alpha$ of the plate. This fact makes sure that the proposed optimal recursive path planning algorithm always can find an optimal path to avoid the forward singularities.

### C. OPTIMAL RECURSIVE PATH PLANNING

To verify the proposed optimal recursive path planning, the path $\hat{\mathcal{I}}$ is divided into $231$ paths. The inverse kinematics problem is solved for all points, then the optimal recursive path planning is conducted to determine the best working state. Table $2$ is the resulted data of points from $166$ to $173$. Before point $170$, the moving state $5$ is chosen because it satisfies the least moving distance of angles $\theta_1$ and the links do not fall into the forward singularity. From point $170$, the moving state is changed to $1$ because the moving state $5$ is too close to the forward singularity. The angle $\theta_1$ changes quite large. However, it remains the state with the least distance traveled.

**TABLE 2.** The data of points in the path created by the optimal recursive path planning (The translation unit is millimeter, the rotation unit is degree).

| #  | $X$   | $Y$   | $\alpha$ | State | $\theta_1$ | $\theta_2$ | $\theta_3$ |
|----|-------|-------|----------|-------|-------------|-------------|-------------|
| 166| 258   | 185   | 0        | 5     | -6.9        | 187.2       | 342.0       |
| 167| 259   | 186   | 0        | 5     | -6.5        | 186.9       | 342.5       |
| 168| 259   | 187   | 0        | 5     | -6.2        | 186.6       | 342.7       |
| 169| 260   | 188   | 0        | 5     | -5.8        | 186.2       | 343.3       |
| 170| 261   | 189   | 0        | 1     | 79.8        | 185.8       | 343.9       |
| 171| 261   | 190   | 0        | 1     | 79.9        | 185.5       | 344.1       |
| 172| 262   | 191   | 0        | 1     | 79.6        | 185.1       | 344.7       |
| 173| 263   | 192   | 0        | 1     | 79.3        | 184.8       | 345.2       |

**D. DYNAMIC FILTERED PATH TRACKING CONTROL**

Section V has presented the dynamic filtered path tracking control schema, which applies the optimal recursive path planning and vision-based pose estimation using UKF. This section will verify the proposed control in both simulation and experiment.

1) **INITIALIZATION OF UNSCENTED KALMAN FILTER**

In Equations (28) and (29), the factors are set as $\alpha = 10$, $\beta = 2$, $\kappa = 3 - d = 0$. In Equation (30), $\mathbf{N}_p = \text{diag}(\sigma_p \odot \sigma_p)$, where $\odot$ is the Hadamard product, $\sigma_p$ is the standard deviation of the process, $\sigma_p = (0.5, 0.5, 0.00873)$. In Equation (34), $\mathbf{N}_s = \text{diag}(\sigma_s \odot \sigma_s)$, where $\sigma_s$ is the standard deviation of the measurement, $\sigma_s = (3, 3, 0.0262)$. In Equation (27), the initial covariance matrix is set temporarily as $\mathbf{C}_p = \mathbf{N}_p$, the initial mean vector $\hat{\mathbf{p}}$ is set equal to the desired pose of the plate at the start point of the path.

2) **SIMULATION**

The desired path $\mathcal{I}$ is defined from the start point $(x = 100\text{mm}, y = 100\text{mm}, \alpha = 0^\circ)$ to the endpoint $(x = 150\text{mm}, y = 150\text{mm}, \alpha = 0^\circ)$. This is a short trajectory to be easily observed data on the figure. The desired angular velocity is $\omega_d = \pi \text{ rad/s}$. As presented in Fig. 6, the feedback angular velocity $\omega_r$, the feedback rotation $q_r$, and the measurement of the vision system are required for the UKF.

In the simulation, the feedback angular velocity is set as $\omega_r = \omega_d + \text{random noise}_\omega$. Consider that the motor has a maximum error of 0.5 degrees after a rotation, $\text{random noise}_\omega$ is generated randomly in the range $[-0.00139\omega_d, 0.00139\omega_d]$ rad/s, where the unit of $\omega_d$ is rad/s. The feedback rotation is set as $q_r = q_d + \text{random noise}_q$, where $\text{random noise}_q$ is generated randomly in the range $[-0.54, 0.54]$ degrees. The value of 0.54 degrees is corresponding to three pulses of the motor encoder. $q_r$, $\omega_r$ are used in Equation (26) to calculate the nonlinear function $g(\cdot)$ in Equation (30).

The measurement of the vision system is simulated by adding noise into the desired pose at the current point. Based on the accuracy of the 3D Intel RealSense Camera D435i, the noise is generated randomly in the range $[-5, 5]$ mm for the coordinates $X$ and $Y$, and in the range $[-0.06, 0.06]$ rad for the rotation angle $\alpha$.

**FIGURE 18.** The path tracking data on $X$-axis in the first simulation.

**FIGURE 19.** The path tracking data on $Y$-axis in the first simulation.

The simulation is repeated three times. Fig. 18 to Fig. 20 show the path tracking data in the first simulation.
FIGURE 20. The path tracking data on the rotation angle $\alpha$ in the first simulation.

TABLE 3. The summary of errors of the path tracking data in the simulations.

| #  | MSE Vision | MSE UKF | MAE Vision | MAE UKF |
|----|------------|---------|------------|---------|
| 1  | $x$ (mm)   | 7.774   | 0.021      | 2.442   | 0.133   |
|    | $y$ (mm)   | 8.087   | 0.022      | 2.448   | 0.133   |
|    | $\alpha$ (degrees) | 4.218 | 0.017      | 1.812   | 0.110   |
| 2  | $x$ (mm)   | 10.512  | 0.023      | 2.881   | 0.151   |
|    | $y$ (mm)   | 7.512   | 0.024      | 2.384   | 0.139   |
|    | $\alpha$ (degrees) | 4.232 | 0.017      | 1.703   | 0.112   |
| 3  | $x$ (mm)   | 8.752   | 0.019      | 2.586   | 0.116   |
|    | $y$ (mm)   | 6.177   | 0.020      | 2.100   | 0.123   |
|    | $\alpha$ (degrees) | 4.121 | 0.020      | 1.709   | 0.126   |

The dot green line is the measurement of the vision system, which is simulated noise very much like the real system. The dashed red line is the real value. The solid blue line is the estimation of the UKF. It can be seen that the UKF can estimate the pose of the plate very close to the real pose. Table 3 shows the summary of mean squared error (MSE) and mean absolute error (MAE) of the path tracking data in three simulations. The data changes in each simulation due to the random value of noise. However, in all three simulations, the pose estimation is corrected and the errors are greatly reduced by using UKF.

3) EXPERIMENT
In the experiment, the robot will move the plate following the spiral trajectory as shown in Fig. 21. The trajectory includes sixty points. The feedback angular velocity $\omega_r$ and rotation $q_r$ are provided by the motor controller board. $\omega_r$ and $q_r$ are used in Equation (26) to calculate the nonlinear function $g(\cdot)$ in Equation (30). The measurement of the vision system is provided based on the real 3D Intel RealSense Camera D435i. When the plate moves from one point to the next point, the UKF vision-based measurement is used as the feedback of the position control to compensate for the pose error. At each point, the program will be temporarily stopped to measure the real pose manually by instrument. The manually measured real pose is considered as the ground truth in the experiment.

Fig. 22 to Fig. 24 show the path tracking data in the experiment. The dashed red line is the manual measurement. The dash-dot yellow line is the desired value created by the control program. The dot green line is the measurement of the vision system. The solid blue line is the measurement of the vision system after smoothing by the UKF. By applying the UKF, the pose measurement is corrected very close to the real pose. It will help the position control to compensate for the pose error continuously. That is why when the program stops to measure the pose manually, the real value is very close to the desired value as shown in the figures. Table 4 is a summary of the errors in the experiment. This is the comparison to the real values calculated during the movement. The MAE of the vision system is larger than the inherent measurement error of the 3D Intel RealSense Camera D435i and the measurement has a lot of noise when the robot is moving. However, The UKF can smooth and correct the measurement. The MAE after using UKF is small.

E. COMPARISON
1) THE COMPARISON OF THE PATH TRACKING CONTROL
To see the difference when applying the dynamic filtered path tracking control, the spiral trajectory moving experiment is repeated using the PID control without the UKF vision-based feedback. Fig. 25 shows the comparison of the manual measurement (real position) in two cases. Table 5 is the summary of errors in two cases compared to the desired values. If only the PID control is used, the pose error is quite big due to the vibration of the mechanical system and the noise such as the backlash of the motor gearbox. By using the UKF vision-based feedback, the pose error is reduced significantly.

2) THE EXTENDED COMPARISON OF THE OPTIMAL RECURSIVE PATH PLANNING
A popular path planning method of the 3RRR planar parallel robot to improve the kinematics and avoid the singularities is the working (actuation) mode conversion, which is proposed in [13] and [14]. In this method, the selected working mode
out of the eight possible working modes is changed after each step. By experiments, the best schema of the change is determined. It is expected to avoid the singularity, but it did not mention about the optimum travel distance.

Let’s review the experiment of the optimal recursive path planning shown in Section VI.C. As presented in Table 2, the moving state is changed at point 170. However, at many points earlier, the moving state was unchanged. Consider the rotation angles of the active links corresponding to the selected moving states, the working mode (as defined in [13]) is also not changed. The proposed optimal recursive path planning does not need to change the working mode each step, but it still looks to avoid all cases of singularity. Besides, it ensures the movement distance of the active links is minimal. Minimize changing the working state after each step also ensures smooth movement of the links. The smoothness and effectiveness of the optimal recursive path planning proposed in this study can be verified by the video in [30].
TABLE 4. The summary of errors of the path tracking data in the experiment.

|            | MSE     | MAE     |
|------------|---------|---------|
|            | Vision  | UKF     | Vision  | UKF     |
| $X$ (mm)   | 16.800  | 0.029   | 3.590   | 0.147   |
| $Y$ (mm)   | 16.080  | 0.032   | 3.516   | 0.142   |
| $\alpha$ (degrees) | 5.565  | 0.051   | 2.095   | 0.197   |

TABLE 5. The comparison of errors between the PID control and the dynamic filtered path tracking control in the experiment.

|            | PID     | UKF filtered | PID     | UKF filtered |
|------------|---------|--------------|---------|--------------|
|            | $X$ (mm) | 4.979        | 1.926   | 0.193        |
|            | $Y$ (mm) | 5.583        | 2.082   | 0.210        |
|            | $\alpha$ (degrees) | 6.262 | 2.116 | 0.360 |

FIGURE 25. The comparison of the manual measurement between the PID control and the dynamic filtered path tracking control in the experiment.

VII. CONCLUSION

In this study, the dynamic filtered path tracking control is proposed for the 3RRR planar parallel robot. The optimal recursive algorithm is designed in detail for path planning. By grouping the moving points in the path, the algorithm reduces computation but still ensures all possible moving states are considered. Compared to other methods, this algorithm not only helps the robot avoid singularities but also ensures the shortest movement distance of links thanks to the optimization in path planning. The 3D vision system is used to measure the pose of the robot. To reduce the errors due to the camera’s intrinsic parameters, the vibration of the mechanical system, and unknown noises, the unscented Kalman filter is designed to correct the pose measurement and smooth out the movement of the robot. Compared with PID control, the proposed method has the remarkable advantage of reducing the positioning error to about 0.193 to 0.21 mm for translation and 0.36 degrees for rotation. This result is not too good compared to systems using high-tech positioning equipment. However, it is an encouraging result for low-cost robot systems. The simulation, experiment, and comparison are performed to verify the effectiveness and feasibility of the proposed methods.

VIII. FUTURE IMPROVEMENT

The path planning should be conducted by using the well-known algorithms, then compare with the proposed optimal recursive algorithm in this study.

Some other methodologies of the path tracking control can be studied and applied to the 3RRR robot such as a finite-time tracker for nonholonomic systems [31], fast terminal sliding mode tracking control [32], neural networks-based distributed adaptive control [33], or type-2 fuzzy-based distributed supervisory control [34].

For further research, the dynamic filtered path tracking control could be applied to the robots in 3D such as the Hexa parallel robot or the robot arms.

REFERENCES

[1] L. W. Tsai, Robot Analysis: The Mechanics of Serial and Parallel Manipulators, 1st ed. Hoboken, NJ, USA: Wiley, 1999.
[2] S. Küçük, Serial and Parallel Robot Manipulators: Kinematics, Dynamics, Control, and Optimization. London, U.K.: IntechOpen, 2012, doi: 10.5772/2301.
[3] Z. Pandilov and V. Dukovski, “Comparison of the characteristics between serial and parallel robots,” Acta Technica Corvinensis-Bull. Eng., vol. 7, no. 1, pp. 143–160, Jan. 2014.
[4] Y. Patel and P. George, “Parallel manipulators applications—A survey,” Mod. Mech. Eng., vol. 2, no. 3, 2012, pp. 57–64, doi: 10.4236/mme.2012.23008.
[5] A. S. Shirodkar, G. J. Bombe, P. D. Mohite, and R. R. Pandey, “Recent trends in applications of parallel manipulators,” Int. J. Innov. Res. Sci., Eng. Technol., vol. 6, no. 10, pp. 19776–19781, Oct. 2017, doi: 10.15680/IRJSET.2017.0610105.
[6] A. Hernández, M. Urízar, E. Macho, and V. Petuya, “Parallel manipulators: Practical applications and kinematic design criteria. towards the modular reconfigurable robots,” in Mechanisms, Transmissions and Applications, vol. 52, Cham, Switzerland: Springer, Jun. 2017, pp. 131–140, doi: 10.1007/978-3-319-60702-3_14.
[7] J. Wu, J. Wang, L. Wang, and Z. You, “Performance comparison of three planar 3-DOF parallel manipulators with 4-RRR, 3-RRR and 2-2-2-2 RRR structures,” Mechatronics, vol. 20, no. 4, pp. 510–517, Jun. 2010, doi: 10.1016/j.mechatronics.2010.04.012.
[8] C. Gosselin and J. Angeles, “Singularity analysis of closed-loop kinematic chains,” IEEE Trans. Robot. Autom., vol. 6, no. 3, pp. 281–290, Jan. 1990, doi: 10.1109/70.56660.
[9] G. Yang, W. Chen, and J.-M. Chen, “A geometrical method for the singularity analysis of 3-RRR planar parallel robots with different actuation schemes,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Lausanne, Switzerland, Sep./Oct. 2002, pp. 2055–2060, doi: 10.1109/IROS.2002.1041568.
[10] I. A. Bonev, D. Zlatanov, and C. M. Gosselin, “Singularity analysis of 3-DOF planar parallel mechanisms via screw theory,” J. Mech. Des., vol. 125, no. 3, pp. 573–581, Sep. 2003, doi: 10.1115/1.1582878.
[11] I. Heerah, B. Benhabib, B. Kang, and J. K. Mills, “Architecture Selection and Singularity Analysis of a Three-Degree-of-Freedom Planar Parallel Manipulator,” J. Intell. Robotic Syst., vol. 37, pp. 355–374, Aug. 2003, doi: 10.1023/A:1026163319039.
[12] K. Ramji, S. R. Babu, and V. R. Raju, “Singularity loci of actuation schemes for 3RRR planar parallel manipulator,” Int. J. Mech. Machtron. Eng., vol. 9, no. 12, pp. 2189–2199, 2015, doi: 10.5281/zenodo.1125025.
[13] S. Liu, Z.-C. Qu, and X.-M. Zhang, “Singularity and path-planning with the working mode conversion of a 3-DOF 3-RRR planar parallel manipulator,” Mechanism Mach. Theory, vol. 107, pp. 166–182, Jan. 2017, doi: 10.1016/j.mechmachtheory.2016.09.004.
[14] Z. Zhang, L. Wang, and Z. Shao, “Improving the kinematic performance of a planar 3-RRR parallel manipulator through actuation mode conversion,” Mechanism Mach. Theory, vol. 130, pp. 86–108, Dec. 2018, doi: 10.1016/j.mechmachtheory.2018.08.011.

[15] M. Gaudreault, A. Joubair, and I. Bonev, “Self-calibration of an industrial robot using a novel affordable 3D measuring device,” Sensors, vol. 18, no. 10, pp. 3380–3398, Oct. 2018, doi: 10.3390/s18103380.

[16] S. Gharaaty, T. Shu, A. Joubair, W. F. Xie, and I. A. Bonev, “Online pose correction of an industrial robot using an optical coordinate measure system,” Int. J. Adv. Robotic Syst., vol. 15, no. 4, pp. 1–6, 2018, doi: 10.1177/17298814187915.

[17] K. Kamali and I. A. Bonev, “Optimal experiment design for elasto-geometrical calibration of industrial robots,” IEEE/ASME Trans. Mechatronics, vol. 24, no. 6, pp. 2733–2744, Dec. 2019, doi: 10.1109/TMECH.2019.2944428.

[18] M. Farajtabar, H. M. Daniali, and S. M. Vareedi, “Pick and place trajectory planning of planar 3-RRR parallel manipulator in the presence of joint clearance,” Roboticica, vol. 35, no. 2, pp. 241–253, Feb. 2017, doi: 10.1017/S0263574714002768.

[19] K. D. Nguyen, C. Ha, T. Q. Dinh, and J. Marco, “Synchronization controller for a 3-RRR parallel manipulator,” Int. J. Precis. Eng. Manuf., vol. 19, no. 3, pp. 339–347, Mar. 2018, doi: 10.1007/s12541-018-0041-z.

[20] L. Sheng and W. Li, “Optimization design by genetic algorithm controller for trajectory control of a 3-RRR parallel robot,” Algorithms, vol. 11, no. 1, p. 7, 2018, doi: 10.3390/a11010007.

[21] S. Yao, H. Li, L. Zeng, and X. Zhang, “Vision-based adaptive control of a 3-RRR parallel positioning system,” Sci. China Technol. Sci., vol. 61, no. 8, pp. 1253–1264, Aug. 2018, doi: 10.1007/s11431-017-9181-9.

[22] M. FarzanehKaloorazi, I. A. Bonev, and L. Birglen, “Simultaneous path placement and trajectory planning optimization for a redundant coordinated robotic workcell,” Mechanism Mach. Theory, vol. 130, pp. 346–362, Dec. 2018, doi: 10.1016/j.mechmachtheory.2018.08.022.

[23] F. Janabi-Sharifi, “Visual servoing: Theory and applications,” in Optomechatronic Systems Handbook. Cleveland, OH, USA: CRC Press, 2003.

[24] W. J. Wilson, C. C. Williams Hulls, and G. S. Bell, “Relative end-effector control using Cartesian position based visual servoing,” IEEE Trans. Robot. Autom., vol. 12, no. 5, pp. 684–696, Oct. 1996, doi: 10.1109/70.538974.

[25] M. Ficocelli and F. Janabi-Sharifi, “Adaptive filtering for pose estimation in visual servoing,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Societal Role Robot, Next Millennium, Oct./Nov. 2001, pp. 19–24, doi: 10.1109/IROS.2001.973330.

[26] T. Shu, S. Gharaaty, W. Xie, A. Joubair, and I. A. Bonev, “Dynamic path tracking of industrial robots with high accuracy using photogrammetry sensor,” IEEE/ASME Trans. Mechatronics, vol. 23, no. 3, pp. 1159–1170, Jun. 2018, doi: 10.1109/TMECH.2018.2821600.

[27] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” Proc. SPIE, vol. 3068, Jul. 1997, Art. no. 280797, doi: 10.1117/12.280797.

[28] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” IBM Syst. J., vol. 4, no. 1, pp. 25–30, 1965, doi: 10.1147/sj.41.0025.

[29] Wikipedia. Bresenham’s Line Algorithm. Accessed: Jun. 6, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

[30] B. P. Huynh. (May 5, 2020). Painting Engraving Machine Using a 3-RRR Planar Robot V-REP Model. Accessed: Aug. 10, 2020. [Online]. Available: https://www.youtube.com/watch?v=10DVszXWY

[31] S. Mobayen, M. J. Yazdanpanah, and V. J. Majd, “A finite-time tracker for nonholonomic systems using recursive singularity-free FTSM,” in Proc. Amer. Control Conf., San Francisco, CA, USA, Jun. 2011, pp. 1720–1725, doi: 10.1109/ACC.2011.5991040.

[32] S. Amirkhani, S. Mobayen, N. Iliaee, O. Boubaker, and S. H. Hosseini, “Fast terminal sliding mode tracking control of nonlinear uncertain mass-spring system with experimental verifications,” Int. J. Adv. Robotic Syst., vol. 16, no. 1, pp. 1–10, Feb. 2019, doi: 10.1177/1729881419821767.

[33] Q. Shen, P. Shi, J. Zhu, S. Wang, and Y. Shi, “Neural networks-based distributed adaptive control of nonlinear multiagent systems,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 3, pp. 1010–1021, Mar. 2020, doi: 10.1109/TNNLS.2019.2915376.

[34] Q. Shen, Y. Shi, R. Jia, and P. Shi, “Design on type-2 fuzzy-based distributed supervisory control with backlash-like hysteresis,” IEEE Trans. Fuzzy Syst., vol. 6, 2020, doi: 10.1109/TFUZZ.2020.2992864.

BA-PHUC HUYNH received the B.E. degree in mechanical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 2003, the M.S. degree in precision mechatronics engineering from the Minghsin University of Science and Technology, Taiwan, in 2010, and the Ph.D. degree in automation and control from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2020. He is currently a Lecturer with the Faculty of Engineering and Technology, Kien Giang University, Vietnam. He is also researching at the Graduate Institute of Automation and Control, National Taiwan University of Science and Technology. His research interests include industrial robots, intelligent robots, machine vision, control theories, dynamics of mechatronics systems, evolutionary algorithms, and mobile robots.

YONG-LIN KUO received the Ph.D. degree in mechanical engineering from the University of Toronto, Canada, in 2005. He is currently a Professor with the National Taiwan University of Science and Technology, Taiwan. His research interests include system dynamics and control, robot manipulators, spacecraft dynamics and control, computational solid mechanics, nonlinear vibrations, kineto-elasto-dynamics, and enhanced finite element method.