

## Research Article

# Dynamic Hybrid Filter for Vision-Based Pose Estimation of a Hexa Parallel Robot

Ba-Phuc Huynh <sup>1,2</sup> and Yong-Lin Kuo <sup>1,3</sup>

<sup>1</sup>Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

<sup>2</sup>Faculty of Engineering and Technology, Kien Giang University, Kien Giang 920000, Vietnam

<sup>3</sup>Center of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

Correspondence should be addressed to Yong-Lin Kuo; [kuo@mail.ntust.edu.tw](mailto:kuo@mail.ntust.edu.tw)

Received 26 March 2021; Accepted 15 September 2021; Published 16 October 2021

Academic Editor: Jesús Lozano

Copyright © 2021 Ba-Phuc Huynh and Yong-Lin Kuo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the problems with industrial robots is their ability to accurately locate the pose of the end-effector. Over the years, many other solutions have been studied including static calibration and dynamic positioning. This paper presents a novel approach for pose estimation of a Hexa parallel robot. The vision system uses three simple color feature points fixed on the surface of the end-effector to measure the pose of the robot. The Intel RealSense Camera D435i is used as a 3D measurement of feature points, which offers a cheap solution and high accuracy in positioning. Based on the constraint of three color feature points, the pose of the end-effector, including position and orientation, is determined. A dynamic hybrid filter is designed to correct the vision-based pose measurement. The complementary filter is used to eliminate the noise of image processing due to environmental light source interference. The unscented Kalman filter is designed to smooth out the pose estimation of the vision system based on robot's kinematic parameters. The combination of two filters in the same control scheme contributes to increased stability and improved accuracy of robot's positioning. The simulation, experiment, and comparison demonstrate the effectiveness and feasibility of the proposed method.

## 1. Introduction

Robots have been used a lot in production today to replace human labor because of their ability to work quickly, continuously, and accurately, among many other outstanding advantages. Serial robots, which have an open-loop kinematic chain, are the most common in the industry. Its advantage is the large workspace, but the drawback is accumulated errors, low stiffness, relatively low load performance due to open kinematic structure, and large weight of actuators. Unlike serial robots, parallel robots, which have a close loop kinematic chain, have high stiffness, high speed and acceleration, and high accuracy and carrying capability, making them a good solution for many manufacturing applications [1–3]. In this paper, a six-degree-of-freedom parallel manipulator, Hexa parallel robot [4–7], is studied to apply the proposed control methods.

One of the problems with industrial robots is their ability to accurately locate the pose of the end-effector. Integrating absolute encoders in the joints is efficient, but expensive and difficult to install. Over the years, many other solutions have been studied including static calibration and dynamic positioning [8–10]. The authors in [11] used an inertial measurement unit (IMU) and ultrasonic triangulation sensors for self-calibration of the robot pose. Its advantage is not complex. However, when using IMU, one of the difficulties is handling cumulative errors when integrating measurement parameters due to the large noise of IMU. Li et al. [12] used the sensor frame and some reflectors fixed on the end-effector of the robot to estimate the relative pose of the robot. This method can be applied to parallel and serial robots. It requires a complex setup and calculation. In [13], the authors used a 3D visual sensor to correct the pose of the end-effector of the robot. The sensor is attached

to the end-effector and it is not easy to install for some applications. FarzanehKaloorazi et al. [14] optimized the trajectory planning of a robotic work cell by using particle swarm optimization (PSO). However, the PSO algorithm needs complex computation and can cause a delay in real-time execution.

In robot control, many various sources of error can affect the positioning of the end-effector such as uncertainties of the model, the nonlinearity of the system, backlash of the mechanical system, and unknown noises. Solving all sources of error is not feasible and effective. Therefore, the visual measurement-based solutions [15] are of interest because they directly address and correct the pose of the end effector. Liang et al. [16] used a vision system and a deep convolutional network human pose estimation algorithm to estimate the 2D and 3D poses of robots on construction sites. This method uses a vision system and deep convolutional network algorithms but does not use additional sensors and markers. The author in [17] proposed a real-time method based on deep learning to estimate the pose of the robot arm using a vision system. This method does not need calibration.

In this paper, the pose of a Hexa parallel robot is estimated dynamically by using a 3D vision system. This method uses three simple color feature points fixed on the surface of the end-effector. The Intel RealSense Camera D435i is used as a 3D measurement of the feature points, which offers a cheap solution and high accuracy in positioning. Based on the constraint of three color feature points, the pose of the end-effector, including the position and orientation, is determined.

As with other vision-based positioning methods, this study needs to address noise in image processing, noise caused by system vibrations while moving, and other unknown noises. In the previous studies, some authors used an extended Kalman filter (EKF) or adaptive Kalman filter (AKF) to stabilize the pose estimation. Wilson et al. [18] applied an EKF to the pose estimation based on visual servo control of a 5-DOF robot arm. In [19], the authors used an AKF to improve accurate pose estimates for visual servoing. In [20], a dynamic path tracking scheme for industrial robots using an AKF is proposed. González et al. [21] used an EKF for the estimation of the camera space manipulation parameters and applied it to a parallel robot. The authors in [22] proposed an adaptive robust Kalman filter to correct the vision-based pose detection of industrial robots. Compared to EKF and AKF, a new extension of the Kalman filter to nonlinear systems [23], named unscented Kalman filter (UKF), uses many sigma points to linearize a nonlinear model, so it may be more accurate. In [24], the authors proposed a two-stage Kalman filter-based algorithm for processing the data from an inertial measurement unit in order to obtain accurate position estimation over a short period of time. In [25], the authors proposed a dynamic filtered path tracking control schema for a 3RRR planar, where the unscented Kalman filter is designed to reduce the errors of the pose estimation.

Because the pose estimation error can be caused by multiple sources of noise, this paper proposes a dynamic hybrid filter to smooth out the pose measurement of the vision sys-

tem. A complementary filter is used to eliminate the noise of the image processing due to environmental light source interference. A UKF is designed to correct the pose estimation of the vision system based on robot's kinematic parameters. The combination of two filters in the same control scheme contributes to increased stability and improved accuracy of robot's positioning.

Based on the above discussion and literature review, the contributions of this study could be summarized as follows:

- (1) The pose of a Hexa parallel robot is estimated dynamically by using a 3D vision system. This method uses three simple color feature points fixed on the surface of the end-effector. The Intel RealSense Camera D435i is used as a 3D measurement of the feature points, which offers a cheap solution and high accuracy in positioning. Based on the constraint of three color feature points, the pose of the end-effector, including the position and orientation, is determined
- (2) A dynamic hybrid filter is proposed to smooth out the pose measurement of the vision system. It is the combination of two filters in the same control scheme that contributes to increased stability and improved accuracy of robot's positioning. The complementary filter is used to eliminate the noise of the image processing due to environmental light source interference. The UKF is designed to correct the pose estimation of the vision system based on robot's kinematic parameters
- (3) Although there are many applications of Kalman filters for industrial robots in the literature review, there are no specific studies on UKF's application in the positioning of parallel robots such as the Hexa robot. This study details how to define the kinetic expressions of the Hexa robot and how it is applied to the UKF, as well as presents the establishment of UKF parameters and key expressions in practical application. The simulation, experiment, and comparison are conducted to verify the effectiveness and feasibility of the proposed approaches

The remainder of this paper includes the following: Section 2 introduces the kinematics of the Hexa parallel robot. Section 3 is the system description, vision-based pose estimation, and the problem statement. Section 4 presents the dynamic hybrid filter. Section 5 shows the simulation, experiment, and comparison. Section 6 is the conclusion.

## 2. Kinematic Model of the Hexa Parallel Robot

Figure 1 presents the kinematic structure of the Hexa parallel robot. The moving plate is connected to the fixed platform by six identical pairs of arms and rods, which are divided into three groups spaced 120 degrees around the center of the fixed platform. The six arms are driven by six DC motors. In this study, the index  $i$  indicates the  $i^{\text{th}}$  pair of arm and rod, where  $i = 1, \dots, 6$  as presented

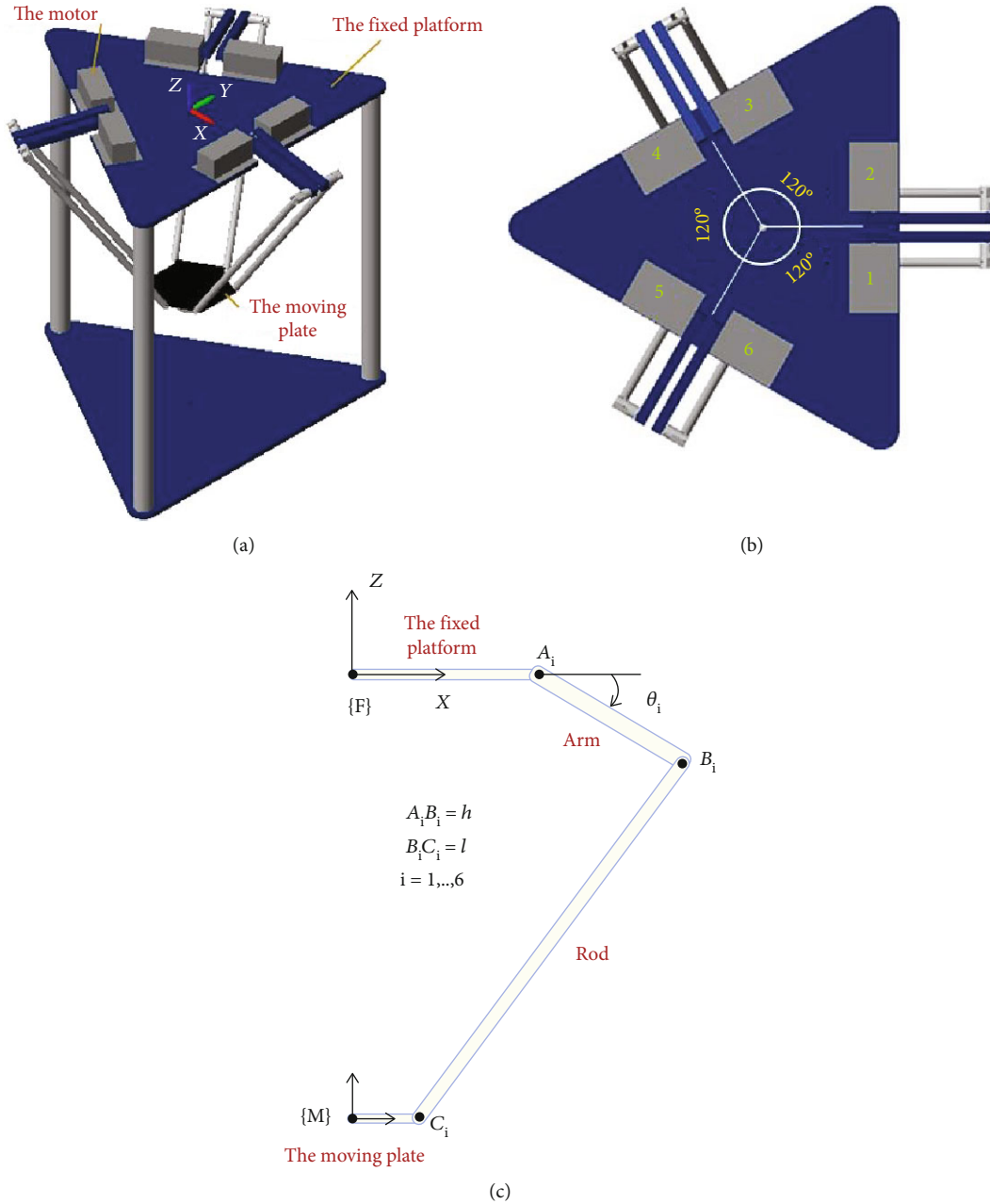


FIGURE 1: The kinematic structure of the Hexa parallel robot. (a) 3D model; (b) top view; (c) front view.

in Figure 1.  $A_i$  is the rotation center of the revolute joint of the arm.  $B_i$  is the center of the universal joint that connects the arm to the rod.  $C_i$  is the center of the spherical joint that connects the rod to the moving plate.  $R$  is the radius of the fixed platform calculated from its center to the point  $A_i$ .  $r$  is the radius of the moving plate calculated from its center to the point  $C_i$ .

$\phi$  is the angle formed by three points  $A_1$ ,  $A_2$ , and the center of the fixed platform.  $\varphi$  is the angle formed by three points  $C_1$ ,  $C_2$ , and the center of the moving plate. The fixed coordinate frame  $\{F\}$  is located at the center of the fixed platform. The X-axis is toward the center point of the edge  $A_1 A_2$ . The Z-axis is perpendicular to the fixed platform in

the upwards direction. The Y-axis is determined by the right-hand rule. The moving coordinate frame  $\{M\}$  is centrally located in the moving plate and has axes that are oriented similar to those of the frame  $\{F\}$ . The length of the arm is  $h$ . The length of the rod is  $l$ .  $\theta_i$  is the rotation angle of the  $i^{\text{th}}$  arm. The rotation state of the arms is specified by a vector  $q = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$ . The pose (position and orientation) of the moving plate is specified by a vector  $p = [x, y, z, \gamma, \beta, \alpha]^T$  with respect to the frame  $\{F\}$ , where  $p_t = [x, y, z]^T$  is the position of the moving plate and  $p_r = [\gamma, \beta, \alpha]^T$  is the Euler angles around the X-, Y-, and Z-axes, respectively.

2.1. *Inverse Kinematics.* The coordinate of the joint  $C_i$  with respect to the frame  $\{F\}$  is calculated as follows:

$${}^F C_i = p_t + {}^F_M H \cdot {}^M C_i, \quad (1)$$

where  ${}^F_M H$  is the homogeneous transformation matrix from  $\{M\}$  to  $\{F\}$ .  ${}^M C_i$  is the fixed coordinate of the joint  $C_i$  with respect to the frame  $\{M\}$  and determined as follows:

$${}^M C_i = [r \cdot \cos \psi_i^M, r \cdot \sin \psi_i^M, 0]^T, \quad (2)$$

with

$$[\psi_1^M, \psi_2^M, \psi_3^M, \psi_4^M, \psi_5^M, \psi_6^M] = \left[ -\frac{\varphi}{2}, \frac{\varphi}{2}, \frac{2\pi}{3} - \frac{\varphi}{2}, \frac{2\pi}{3} + \frac{\varphi}{2}, \frac{4\pi}{3} - \frac{\varphi}{2}, \frac{4\pi}{3} + \frac{\varphi}{2} \right]. \quad (3)$$

The homogeneous transformation matrix  ${}^F_M H$  is as follows:

$${}^F_M H = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}, \quad (4)$$

where  $c$  and  $s$  present cosine and sine.

The coordinate of the joint  $B_i$  with respect to the frame  $\{F\}$  is calculated as follows:

$${}^F B_i = {}^F A_i + {}^A B_i, \quad (5)$$

where  ${}^F A_i$  is the fixed coordinate of the joint  $A_i$  with respect to the frame  $\{F\}$ .  ${}^A B_i$  is the coordinate of the joint  $B_i$  with respect to the joint  $A_i$ .  ${}^F A_i$  and  ${}^A B_i$  are calculated as follows:

$${}^F A_i = [R \cdot \cos \psi_i^F, R \cdot \sin \psi_i^F, 0]^T, \quad (6)$$

$${}^A B_i = \begin{bmatrix} h \cdot \cos \lambda_i \cdot \cos \theta_i \\ h \cdot \sin \lambda_i \cdot \cos \theta_i \\ -h \cdot \sin \theta_i \end{bmatrix}.$$

with

$$[\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6] = \left[ 0, 0, \frac{2\pi}{3}, \frac{2\pi}{3}, \frac{4\pi}{3}, \frac{4\pi}{3} \right],$$

$$[\psi_1^F, \psi_2^F, \psi_3^F, \psi_4^F, \psi_5^F, \psi_6^F] = \left[ -\frac{\phi}{2}, \frac{\phi}{2}, \frac{2\pi}{3} - \frac{\phi}{2}, \frac{2\pi}{3} + \frac{\phi}{2}, \frac{4\pi}{3} - \frac{\phi}{2}, \frac{4\pi}{3} + \frac{\phi}{2} \right]. \quad (7)$$

The relationship between the joint  $B_i$  and the joint  $C_i$  is expressed as follows:

$$\|{}^F C_i - {}^F B_i\| = l. \quad (8)$$

By substituting Equations (1) and (5) into Equation (8), a new equation is obtained in the following format:

$$a_i \cdot \cos \theta_i + b_i \cdot \sin \theta_i = c_i, \quad (9)$$

with

$$a_i = {}^F C_{ix} \cdot \cos \lambda_i + {}^F C_{iy} \cdot \sin \lambda_i - R \cdot \cos (\lambda_i - \psi_i^F),$$

$$b_i = {}^F C_{iz},$$

$$c_i = \frac{d_i^2 + e_i^2 + {}^F C_{iz}^2 + h^2 - l^2}{2h}, \quad (10)$$

$$d_i = {}^F C_{ix} - R \cdot \cos \psi_i^F,$$

$$e_i = {}^F C_{iy} - R \cdot \sin \psi_i^F.$$

By solving Equation (9), the rotation angles of the arms can be obtained as follows:

$$\theta_i = 2 \tan^{-1} \left( \frac{b_i \pm \sqrt{a_i^2 + b_i^2 - c_i^2}}{a_i + c_i} \right). \quad (11)$$

2.2. *Kinematic Jacobian Matrix and Singularity Analysis.* To establish the relationship between the velocity of the moving plate and the angular velocity of the arms, the Jacobian matrix needs to be determined. Equation (8) can be rewritten as follows:

$$({}^F C_i - {}^F B_i)^2 = l^2. \quad (12)$$

Substituting Equations (1) and (5) into Equation (12) and differentiating the equation will result to

$$K_i^T \cdot \dot{p}_t + K_i^T \cdot {}^F_M \dot{H} \cdot {}^M C_i - K_i^T \cdot {}^A \dot{B}_i = 0, \quad (13)$$

$$K_i = {}^F C_i - {}^F B_i,$$

where  $\dot{p}_t = [\dot{x}, \dot{y}, \dot{z}]^T$  is the velocity vector of the moving plate.

The time derivative of the homogeneous transformation matrix can be calculated by the skew-symmetry operator [26]:

$${}^F_M \dot{H} = \begin{bmatrix} 0 & -\dot{\alpha} & \dot{\beta} \\ \dot{\alpha} & 0 & -\dot{\gamma} \\ -\dot{\beta} & \dot{\gamma} & 0 \end{bmatrix} \cdot {}^F_M H. \quad (14)$$

The time derivative of  ${}^A B_i$  is as follows:

$${}^A \dot{B}_i = h \cdot \begin{bmatrix} -\cos \lambda_i \cdot \sin \theta_i \\ -\sin \lambda_i \cdot \sin \theta_i \\ -\cos \theta_i \end{bmatrix} \cdot \dot{\theta}_i. \quad (15)$$

Equation (13) can be rewritten as follows:

$$K_i^T \cdot \dot{p}_t + \left( [{}^F_M H \cdot {}^M C_i]_X \cdot K_i \right)^T \dot{p}_r = Q_i \cdot \dot{\theta}_i, \quad (16)$$

$$Q_i = K_i^T \cdot h \cdot \begin{bmatrix} -\cos \lambda_i \cdot \sin \theta_i \\ -\sin \lambda_i \cdot \sin \theta_i \\ -\cos \theta_i \end{bmatrix},$$

where  $\dot{p}_r = [\dot{\gamma}, \dot{\beta}, \dot{\alpha}]^T$  is the angular velocity vector of the moving plate.  $[\cdot]_X$  is the skew-symmetry operator of a vector  $3 \times 1$ . Suppose  ${}^F_M H \cdot {}^M C_i = [w_1, w_2, w_3]^T$ , the skew-symmetry operator will result to

$$[{}^F_M H \cdot {}^M C_i]_X = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \quad (17)$$

Equation (16) can be rewritten as follows:

$$J_p \cdot \dot{p} = J_q \cdot \dot{q}, \quad (18)$$

where

$$\dot{p} = \begin{bmatrix} \dot{p}_t^T \\ \dot{p}_r^T \end{bmatrix}^T = \begin{bmatrix} \dot{x}, \dot{y}, \dot{z}, \dot{\gamma}, \dot{\beta}, \dot{\alpha} \end{bmatrix}^T, \quad (19)$$

$$\dot{q} = \begin{bmatrix} \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6 \end{bmatrix}^T,$$

$$J_q = \text{diag} [Q],$$

$$J_p = [e_1 \ e_2 \ e_3 \ e_4 \ e_5 \ e_6]^T,$$

$$e_i = \left[ K_i^T \quad \left( [{}^F_M H \cdot {}^M C_i]_X \cdot K_i \right)^T \right].$$

$J_p$  and  $J_q$  are the forward and inverse Jacobian matrices, respectively. Equation (18) is used to find the system equation for the UKF in Section 4.2.1.

### 3. System Description, Vision-Based Pose Estimation, and Problem Statement

**3.1. System Description.** Figure 2 is the block diagram of the hybrid filter for the pose estimation of the Hexa parallel robot. Block A is the motion control, which includes the path planning and motion control of the robot. In this paper, motion control is not covered in detail because it is not the scope of the study. It is assumed that the desired trajectory is ready to be set and that the motion controller controls the motors through all points in the trajectory. Block B is the image filter, which includes image processing, features detection, and a complementary filter to stabilize the feature points. Details will be presented in Sections 3.2 and 4.1. Block C is the pose estimation, which includes the 3D reconstruction to calculate the 3D coordinate of the feature points and through which the pose of the moving plate will be calculated. Details will be presented in Section 3.2. Block D is

the dynamic filter, which includes a UKF to smooth out the pose estimation. Details will be presented in Section 4.

$p_d$  is the desired pose of the moving plate at one point in the trajectory.  $q_d$  is the rotation vector of the arms corresponding to  $p_d$ . The relationship between  $q_d$  and  $p_d$  is described as follows:

$$q_d = T_{ikp}(p_d), \quad (20)$$

where  $T_{ikp}$  is the apparent function of the inverse kinematics problem presented in Section 2.1.

$q_r$  is the real rotation vector of the arms.  $\omega_d$  is the desired angular velocity of the arms.  $\omega_r$  is the real angular velocity of the arms.  $q_r$  and  $\omega_r$  are obtained by tracking the encoders of the motors.  $p_r$  is the real pose of the moving plate after the motion control corresponding to  $p_d$ . There is always an error between  $p_r$  and  $p_d$  due to many reasons such as uncertainties, system nonlinearities, the backlash of the motor gearbox, and unknown disturbances. In the experiment,  $p_r$  is measured by a highly accurate instrument after each moving point in the trajectory and used as ground truth to evaluate the results.

$(u_j, v_j)$  is the coordinate of the  $j^{\text{th}}$  feature point in the image plane.  $p_e$  is the pose of the moving plate after the vision-based pose estimation.  $p_f$  is the pose of the moving plate after using the UKF to smooth out the pose estimation. The filtered value after the dynamic filter is used to calculate the angles of the arms using the inverse kinematics problem and compared with the desired value to compensate for the errors.

**3.2. Vision-Based Pose Estimation.** Figure 3 presents the vision system of the Hexa parallel robot using in this study. The Intel RealSense Camera D435i is located below the fixed platform, pointing towards the surface of the moving plate to capture the image of the feature points.  $\{C\}$  is the camera coordinate system. The feature points are detected by color, shape, and area using OpenCV 4.3.0, which can be searched on the internet. After the image processing, the coordinate  $(u_j, v_j)$  of the  $j^{\text{th}}$  feature point in the image plane is obtained.

The 3D coordinate of the feature point with respect to the frame  $\{C\}$  is calculated as follows:

$${}^C [x_j, y_j, z_j]^T = T_R(u_j, v_j), \quad (21)$$

$$j = 1, 2, 3,$$

where  $T_R$  is the apparent function of the 3D reconstruction using Application Programming Interfaces (APIs) for the Intel RealSense Camera, which is supplied with the device or can be found on manufacturer's website.

The 3D coordinate of the feature point with respect to the fixed frame  $\{F\}$  is calculated as follows:

$${}^F [x_j, y_j, z_j]^T = {}^F_C R \cdot {}^C [x_j, y_j, z_j]^T, \quad (22)$$

$$j = 1, 2, 3,$$

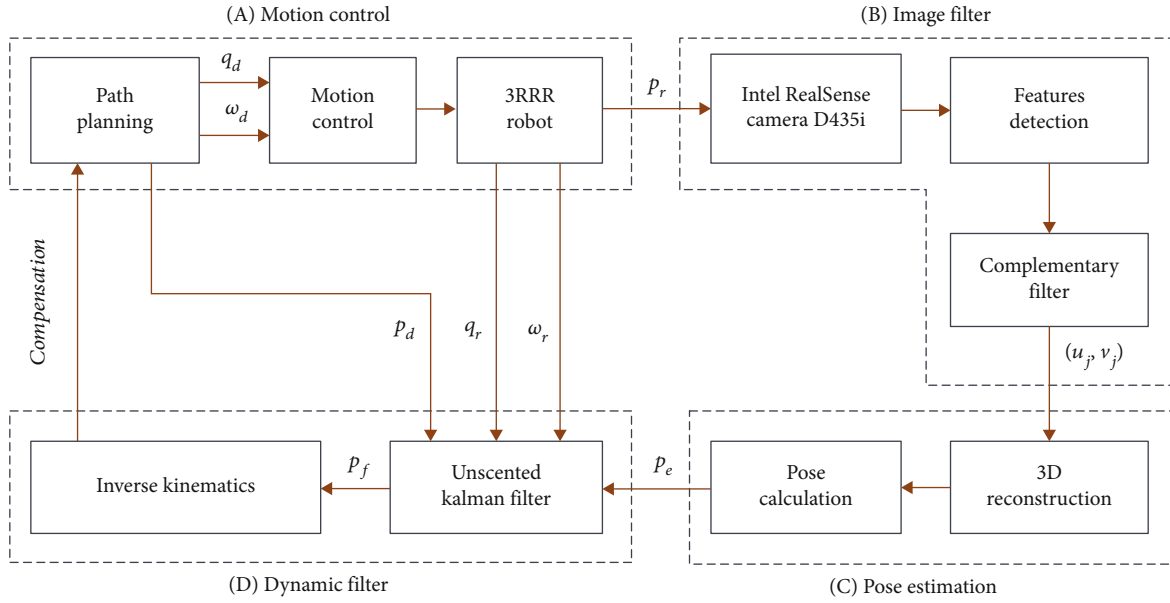


FIGURE 2: Block diagram of hybrid filter for the pose estimation of the Hexa parallel robot.



FIGURE 3: The vision system of the real robot (a) and the vision system on the control software (b).

where  ${}^F_C R$  is the homogeneous transformation matrix from  $\{C\}$  to  $\{F\}$ .

The position of the moving plate with respect to the fixed frame  $\{F\}$  is calculated as follows:

$$p_t = [x, y, z]^T = \left[ \frac{x_1 + x_2 + x_3}{3}, \frac{y_1 + y_2 + y_3}{3}, \frac{z_1 + z_2 + z_3}{3} \right]^T. \quad (23)$$

To calculate the orientation of the moving plate, the plane  $\Pi$  passing through the three feature points is defined as follows:

$$\Pi : ax + by + cz + 1 = 0, \quad (24)$$

with

$$a = \frac{-1}{\Delta} \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix}, \quad b = \frac{-1}{\Delta} \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}, \quad (25)$$

$$c = \frac{-1}{\Delta} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}, \quad \Delta = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}.$$

The orientation  $p_r = [\gamma, \beta, \alpha]^T$  of the moving plate is calculated as follows:

- (i) The Euler angle  $\beta$  is the angle between the plane  $\Pi$  and the  $OYZ$ -plane, where the  $OYZ$ -plane has an equation of  $x = 0$

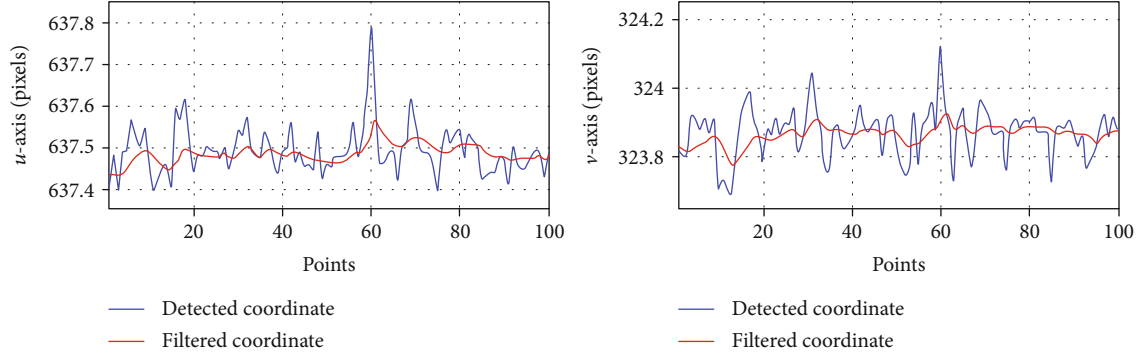


FIGURE 4: The coordinate of detected feature point before and after applying the complementary filter (100 continuous instants at the ground truth of  $u = 637.50$  and  $v = 323.85$ . The ground truth is obtained by calculating the inverse problem in 3D reconstruction for a known physical location with exact coordinates).

- (ii) The Euler angle  $\gamma$  is the angle between the plane  $II$  and the  $OXZ$ -plane, where the  $OXZ$ -plane has an equation of  $y = 0$
- (iii) The Euler angle  $\alpha$  is the angle between the  $X$ -axis and the line passing through the feature point 1 and the center of the plate

Finally, the pose of the moving plate is determined as follows:

$$p = \begin{bmatrix} p_t \\ p_r \end{bmatrix}. \quad (26)$$

3.3. *Problem Statement.* Two factors may influence the outcome of the pose estimate in Figure 2:

- (i) Feature detection in block B is noisy and has a certain margin of error in image processing mainly due to the influence of ambient light sources as demonstrated in Figure 4. The noise affects the results of the depth measurement and the calculation of the 3D coordinates of the feature points by Equation (21).
- (ii) The pose calculation in block C has errors due to many reasons such as uncertainties, system nonlinearities, the backlash of the motor gearbox, and unknown disturbances. The trajectory and speed of the robot also affect the error

The objective of this study is to design a dynamic hybrid filter, which is the combination of the complementary filter in block B and the UKF in block D to smooth out and correct the pose estimation of the moving plate. The complementary filter reduces the noise and stabilizes the measurement of image processing. The UKF smooths out the pose calculation based on robot's kinetics. The corrected value will be used as the feedback of the motion control to increase the accuracy.

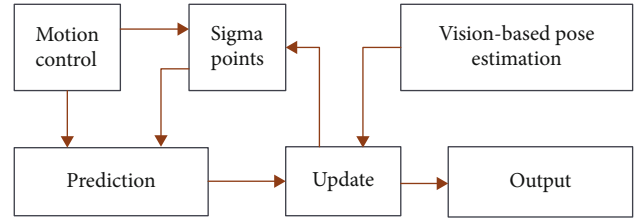


FIGURE 5: The process of the unscented Kalman filter.

## 4. The Dynamic Hybrid Filter

4.1. *The Image Filter Using a Complementary Filter.* In image processing, the coordinate of feature points in the image plane may have an error due to noises, mainly light source noise and vibration when moving the robot. A 1-pixel error can also lead to a position error of several millimeters after 3D reconstruction as shown in Equation (21). To increase accuracy, the pixel positions in this study are calculated by real numbers instead of integers. In this study, a complementary filter is used to remove the noise and stabilize the coordinate of the feature points in the image plane. Figure 4 shows the role of the complementary filter in the proposed hybrid filter. It can reduce noise significantly.

The formula of the filter is as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix}_j^k = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}_j^{k-1} + \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix}_j^k, \quad (27)$$

$$\beta = 1 - \alpha,$$

with initial condition :  $\begin{bmatrix} u \\ v \end{bmatrix}_j^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$

$$j = 1, 2, 3,$$

where  $0 < \alpha < 1$  is the filter gain, which is chosen experimentally to balance smoothness and lag.  $k - 1$  and  $k$  are the previous and current instant, respectively.  $j$  is the index of the feature point.  $(\bar{u}_j, \bar{v}_j)$  is the coordinate of the feature point on the image plane calculated by image processing at the

current instant.  $(u_j, v_j)$  is the coordinate of the feature point on the image plane after filtering.

**4.2. The Dynamic Filter Using an Unscented Kalman Filter.** The UKF [23] is used to smooth out the pose calculation based on robot's kinetics. The corrected value will be used as the feedback of the motion control to increase the accuracy. This process is demonstrated in Figure 5.

**4.2.1. The System Equation.** Equation (18) can be rewritten as follows:

$$\begin{aligned} \dot{p} &= \mathbf{J}_c \cdot \dot{q} = \mathbf{J}_c \cdot \omega, \\ \mathbf{J}_c &= \mathbf{J}_p^{-1} \cdot \mathbf{J}_q, \\ \omega &= [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6]^T, \end{aligned} \quad (28)$$

where  $\mathbf{J}_p$  is the forward kinematics Jacobian matrix of size  $6 \times 6$  and  $\mathbf{J}_q$  is the inverse kinematics Jacobian matrix of size  $6 \times 6$ .  $\mathbf{J}_p$  and  $\mathbf{J}_q$  are calculated by Equation (19).  $\omega$  is the angular velocity vector of the arms of the robot obtained by tracking the encoders of the motors.

Equation (28) is expressed in discrete form as follows:

$$p^k = p^{k-1} + \mathbf{J}_c^{k-1} \cdot \omega^{k-1} \cdot \Delta t, \quad (29)$$

where  $k-1$  and  $k$  are the previous and current instant, respectively.  $\Delta t$  is the discrete interval.  $p$  is the pose vector of the moving plate.

**4.2.2. Step 1: Design Sigma Points.** UKF uses many sigma points to linearize a nonlinear model. The state variable (the pose vector  $p$  of the moving plate) has a dimensionality of  $d = 6$ , so the number of sigma points will be  $2d + 1 = 13$ . To start a new instant, the sigma points are calculated as follows:

$$\begin{aligned} \mathbf{X}_0 &= \hat{p}, \\ \mathbf{X}_{col} &= \hat{p} + \left[ \sqrt{(d+\lambda)\mathbf{C}_p} \right]_{col}, \quad \text{for } col = 1, \dots, d, \\ \mathbf{X}_{col} &= \hat{p} - \left[ \sqrt{(d+\lambda)\mathbf{C}_p} \right]_{col-d}, \quad \text{for } col = d+1, \dots, 2d, \\ \lambda &= \alpha^2(d+\kappa) - d, \end{aligned} \quad (30)$$

where the sigma points matrix  $\mathbf{X}$  has the size of  $6 \times 13$ .  $\hat{p}$  is the mean value of  $p$ . The covariance matrix  $\mathbf{C}_p$  has the size of  $d \times d = 6 \times 6$ . The squared root  $\sqrt{(d+\lambda)\mathbf{C}_p}$  can be calculated by using the Cholesky decomposition, which is available on the Internet (such as the website of Wikipedia). The index  $col$  denotes the column of the matrix. The initial value of  $\mathbf{C}_p$  is set by the estimated standard deviation of elements of  $p$ . The initial value of  $\hat{p}$  is set equal to the desired pose of the moving plate at the current moving point. For other instants,  $\hat{p}$  and  $\mathbf{C}_p$  are the estimation in the update step of the previous instant.  $\alpha$

is the difference between the sigma points around  $\hat{p}$ .  $\kappa$  is the scaling coefficient. In this paper,  $\alpha = 10$  and  $\kappa = 0$ .

The weights are calculated as follows:

$$\begin{aligned} w_0^m &= \frac{\lambda}{d+\lambda}, \\ w_0^c &= \frac{\lambda}{d+\lambda} + 1 - \alpha^2 + \beta, \\ w_{col}^m &= w_{col}^c = \frac{1}{2(d+\lambda)}, \quad \text{for } col = 1, \dots, 2d, \end{aligned} \quad (31)$$

where  $w_{col}^m$  is the weight to calculate the mean associated with the  $col^{\text{th}}$  sigma point.  $w_{col}^c$  is the weight to calculate the covariance associated with the  $col^{\text{th}}$  sigma point.  $w^m$  and  $w^c$  have the size of  $6 \times 13$ .  $\beta = 2$  is the optimal coefficient.

**4.2.3. Step 2: Prediction.** The predicted mean and covariance of UKF are determined as follows:

$$\begin{aligned} \hat{p}' &= \sum_{col=0}^{2d} w_{col}^m \cdot \mathbf{P}_{col}, \\ \mathbf{C}_p' &= \sum_{col=0}^{2d} w_{col}^c \cdot \left[ (\mathbf{P}_{col} - \hat{p}') (\mathbf{P}_{col} - \hat{p}')^T \right] + \mathbf{N}_p, \\ \mathbf{P}_{col} &= g(\mathbf{X}_{col}), \quad \text{for } col = 1, \dots, 2d, \end{aligned} \quad (32)$$

where the index  $col$  denotes the column of the matrix.  $\mathbf{N}_p$  is the process noise covariance matrix, which has the size of  $d \times d = 6 \times 6$ .  $\mathbf{P}_{col}$  is mapped associated with the  $col^{\text{th}}$  sigma points.  $g(\cdot)$  is the nonlinear function, which is estimated based on the system kinetics as shown in Equation (29).

**4.2.4. Step 3: Update.** The Intel RealSense camera measures the pose of the moving plate and returns a measured pose vector  $p_M = [x, y, z, \gamma, \beta, \alpha]^T$ . This measurement has a lot of noise and is not fixed between instants even at the same moving point. The difference vector  $\Delta p$  between  $p_M$  and  $\hat{p}'$  is calculated as follows:

$$\Delta p = p_M - \hat{p}'. \quad (33)$$

After that, the sigma points are mapped into the measurement space based on the difference  $\Delta p$  by using a function  $h(\cdot)$  as follows:

$$\mathbf{Z} = h(\mathbf{P}) + v, \quad (34)$$

where  $\mathbf{Z}$  is the mapped sigma points matrix, which has the size of  $6 \times 13$  associated with  $\mathbf{P}$ .  $v$  is the measurement noise signal, which can be assumed to be a zero-mean Gaussian white sequences noise.

In practice, each column of  $h(\mathbf{P})$  is calculated as follows:

$$h(\mathbf{P})_{col} = \mathbf{P}_{col} + \Delta p, \quad \text{for } col = 1, \dots, 2d. \quad (35)$$



TABLE 1: The parameters of the vision system.

Resolution	Frame rate	Depth measurement accuracy	Maximum measuring distance	Expected accuracy
1080 × 720	30 fps	≤0.5% of measuring distance	500 mm	1.25 to 2.5 mm

The measured mean and covariance of UKF are determined as follows:

$$\begin{aligned}\hat{\mathbf{z}} &= \sum_{col=0}^{2d} w_{col}^m \cdot \mathbf{Z}_{col}, \\ \mathbf{C}_z &= \sum_{col=0}^{2d} w_{col}^c \cdot \left[ (\mathbf{Z}_{col} - \hat{\mathbf{z}})(\mathbf{Z}_{col} - \hat{\mathbf{z}})^T \right] + \mathbf{N}_s,\end{aligned}\quad (36)$$

where the index  $col$  denotes the column of the matrix.  $\mathbf{N}_s$  is the sequences noise covariance matrix, which has the size of  $d \times d = 6 \times 6$ .

The cross variance is determined as follows:

$$\mathbf{C}_v = \sum_{col=0}^{2d} w_{col}^c \left( \mathbf{P}_{col} - \hat{\mathbf{p}}' \right) (\mathbf{Z}_{col} - \hat{\mathbf{z}})^T. \quad (37)$$

The Kalman gain is calculated as follows:

$$\mathbf{K} = \frac{\mathbf{C}_v}{\mathbf{C}_z}. \quad (38)$$

Finally, the new state variable and covariance matrix are updated as follows:

$$\begin{aligned}\hat{\mathbf{p}} &= \hat{\mathbf{p}}' + \mathbf{K}(\mathbf{p}_M - \hat{\mathbf{z}}), \\ \mathbf{C}_p &= \mathbf{C}_p' - \mathbf{K}\mathbf{C}_z\mathbf{K}^T.\end{aligned}\quad (39)$$

The new state variable and covariance matrix are used for the next instant from step 1 in Section 4.2.2.

**4.3. The Stability of the Unscented Kalman Filter.** The stability of the UKFs is complicated especially since the measurement of the update step is provided from a vision system. The study in [27] proved that by choosing some initial conditions of the filter, the estimation error will be bounded, and the filter will be stable. In this paper, the stability analysis is not performed again but achieved experimentally by tuning the initial filter parameters.

## 5. Simulation, Experiment, and Comparison

**5.1. Hardware.** Figure 3 is the real Hexa parallel robot using in this study. The Intel RealSense camera D435i is used as a 3D vision system. The simulation is conducted in MATLAB. The experimental robot control software is written in Visual C++ by the authors. Table 1 shows the parameters of the 3D vision system. The real frame rate of the vision system is 20 fps because the process interval of the program is  $\Delta t = 50$  ms. Figure 6 is the block diagram of the simulation in MATLAB. To verify the performance of the hybrid filter,

in both simulation and experiment, the Hexa robot is moved by the helix path as shown in Figure 7. The angular velocity of the arms is set  $\omega_d = 0.15\pi$  rad/s.

**5.2. Initial Parameters of UKF.** In Equations (30) and (31), the coefficients are tuned as  $\alpha = 10$ ,  $\beta = 2$ , and  $\kappa = 0$ . In Equation (32),  $\mathbf{N}_p = \text{diag}(\sigma_p \odot \sigma_p)$ , where  $\odot$  is the Hadamard product,  $\sigma_p$  is the process standard deviation, and  $\sigma_p = (1, 1, 1, 0.0349, 0.0349, 0.0349)$ . In Equation (36),  $\mathbf{N}_s = \text{diag}(\sigma_s \odot \sigma_s)$ , where  $\sigma_s$  is the measurement standard deviation and  $\sigma_s = (5, 5, 5, 0.0611, 0.0611, 0.0611)$ . In Equation (30), the initial covariance matrix is set temporarily  $\mathbf{C}_p = \mathbf{N}_p$ , the initial mean vector  $\hat{\mathbf{p}}$  is set equal to the desired pose of the moving plate at the start point of the UKF process. These values are tuned in the simulation for best performance, then applied to the experiment.

**5.3. Simulation.** The simulation creates the feedback angular velocity  $\omega_r$  by adding a random noise into the desired angular velocity  $\omega_d$ . The random noise is controlled in the range. Consider that the real motor has a maximum error of 2 degrees per revolution; the random noise of angular velocity is in the range  $[-0.00417\omega_d, 0.00417\omega_d]$  rad/s. The feedback rotation  $q_r$  is simulated by adding a random noise into the desired rotation  $q_d$ . Consider that the real motor encoder has a maximum error of 3 pulses; the random noise of rotation is in the range  $[-0.54, 0.54]$  degrees. The simulation creates the measurement for the UKF by adding the random noise into the desired pose  $p_d$  at the corresponding instant. Based on the accuracy of the vision system as shown in Table 1, the random noise of the measurement is set in the range  $[-2.5, 2.5]$  mm for the translation and  $[-0.06, 0.06]$  rad for the Euler angles. The tracking data is recorded at each moving point.

Figures 8–13 show the tracking data of the first 50 points of the helix path in the simulation. The dot blue line indicates the pose estimation of the vision system. The vision system is simulated noise close to the real system. The dashed green line is the real pose of the plate. The solid red line is the smoothed estimation after using the UKF. For easy comparison of values, mean squared error (MSE) and mean absolute error (MAE) of the simulation are summarized in Table 2. After using the UKF, the vision-based pose estimation is smoothed out, and the errors are reduced significantly.

**5.4. The Experiment.** Similar in simulation, the Hexa robot is controlled to follow the helix trajectory through 101 points as shown in Figure 7. The motion control of the motor controller board will measure the angular velocity  $\omega_r$  and rotation  $q_r$  to use in the UKF process to calculate the nonlinear function  $g(\cdot)$  in Equation (32). The 3D Intel RealSense Camera D435i will estimate the real pose of the plate and use it in the update step of the UKF. The pose

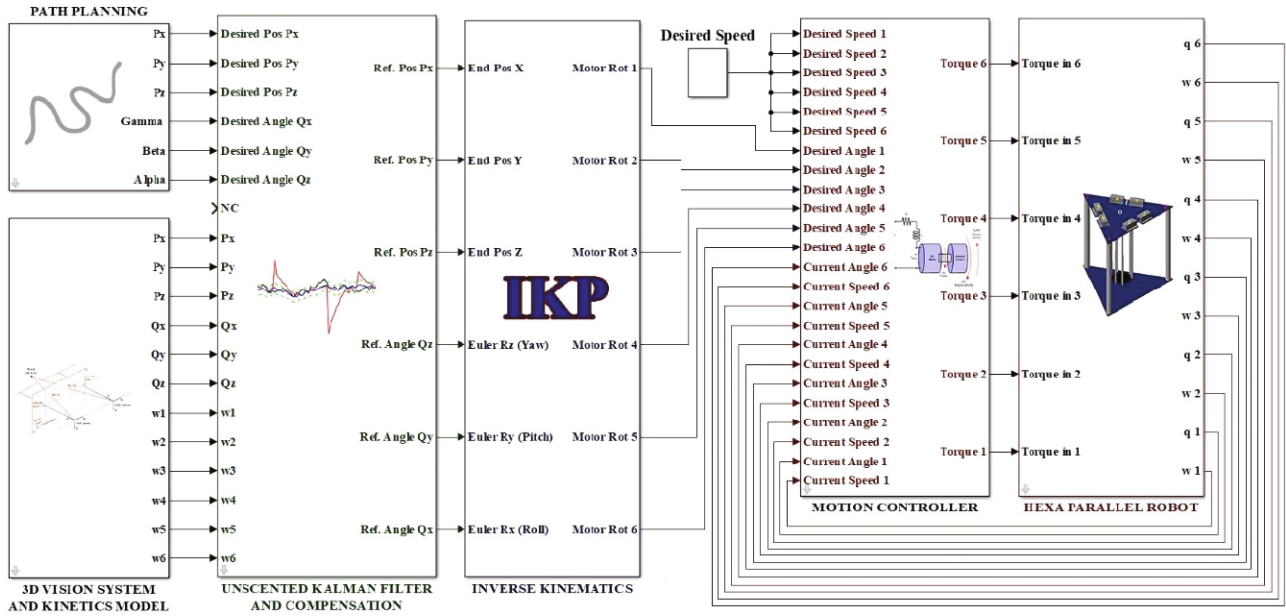


FIGURE 6: The simulation in MATLAB SimMechanics.

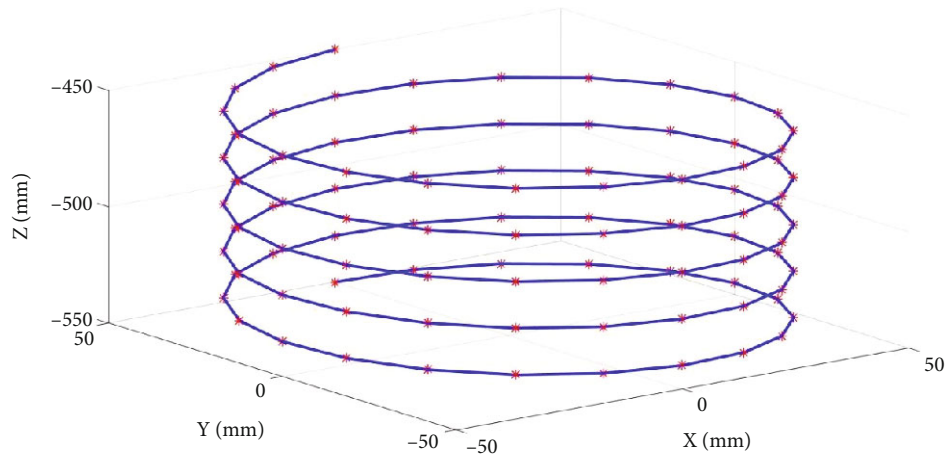


FIGURE 7: The helix path with 101 moving points.

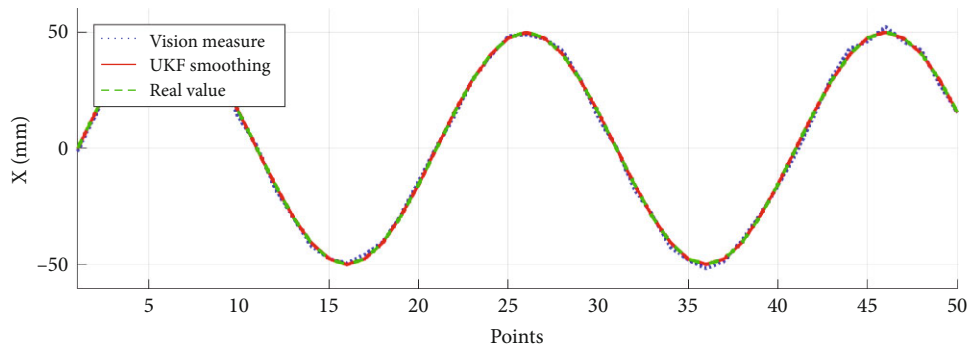


FIGURE 8: The tracking data on the X-axis of the first 50 points in the simulation.

estimation after smoothing by UKF is used as the feedback of the motion control to correct the pose of the plate. When the plate is moved point to point, the UKF is calcu-

lated continuously with a cycle of  $\Delta t = 50$  ms as presented in Section 5.1. After each movement, the robot is stopped for a moment to measure the real pose manually by

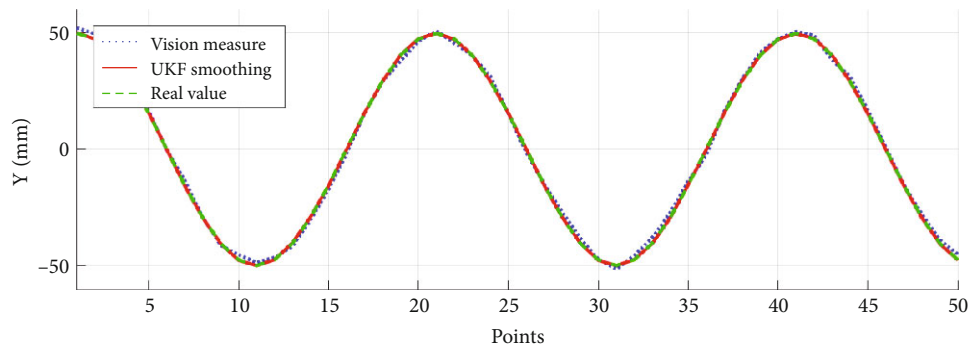


FIGURE 9: The tracking data on the Y-axis of the first 50 points in the simulation.

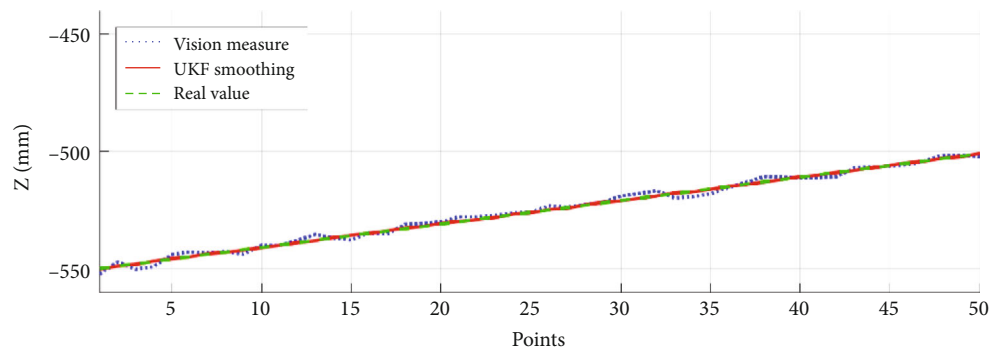


FIGURE 10: The tracking data on the Z-axis of the first 50 points in the simulation.

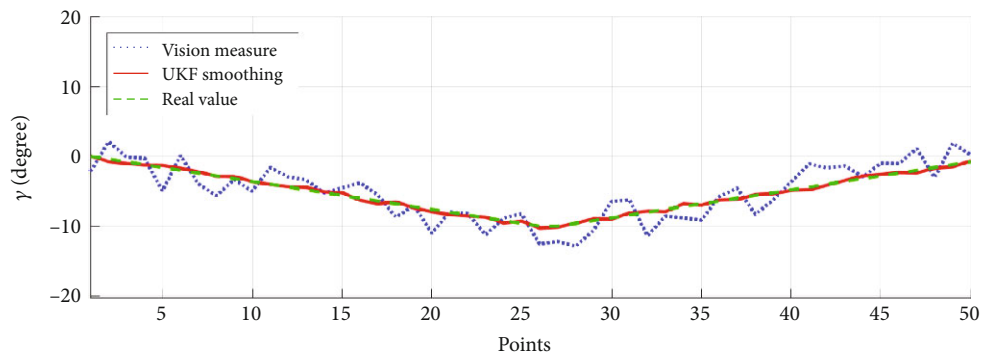


FIGURE 11: The tracking data on the rotation angle  $\gamma$  of the first 50 points in the simulation.

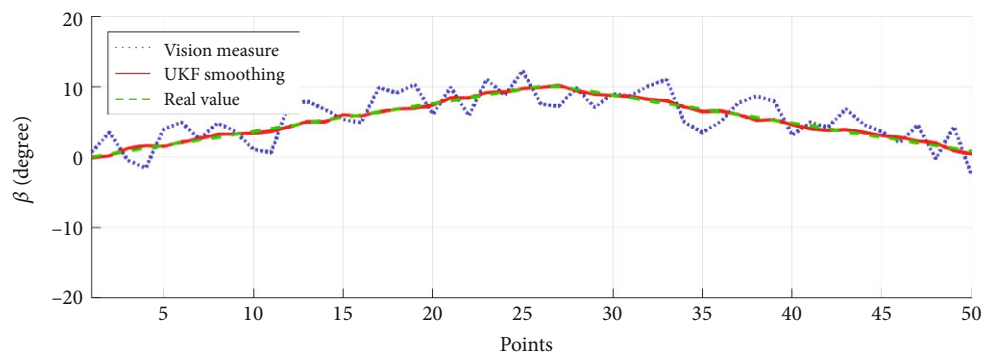


FIGURE 12: The tracking data on the rotation angle  $\beta$  of the first 50 points in the simulation.

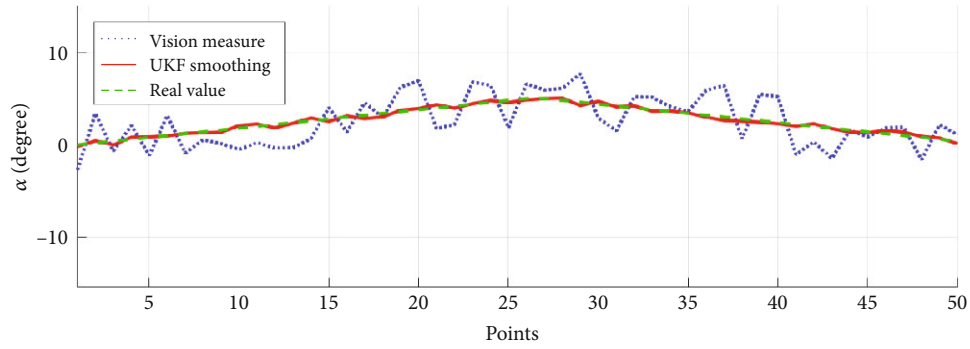


FIGURE 13: The tracking data on the rotation angle  $\alpha$  of the first 50 points in the simulation.

TABLE 2: The summary of errors in the simulation.

Pose	MAE		MSE	
	Vision	UKF	Vision	UKF
X (mm)	1.258	0.118	2.047	0.019
Y (mm)	1.304	0.139	2.292	0.024
Z (mm)	1.280	0.138	2.219	0.025
$\gamma$ (degrees)	1.639	0.229	3.647	0.067
$\beta$ (degrees)	1.952	0.227	4.954	0.067
$\alpha$ (degrees)	1.719	0.206	3.900	0.057

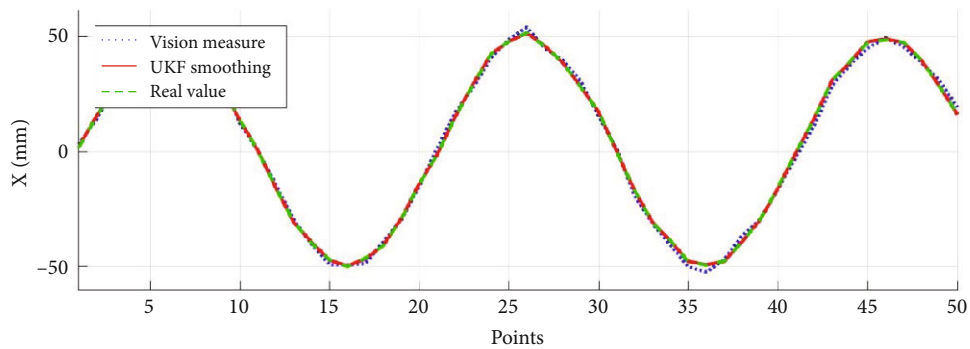


FIGURE 14: The tracking data on the X-axis of the first 50 points in the experiment.

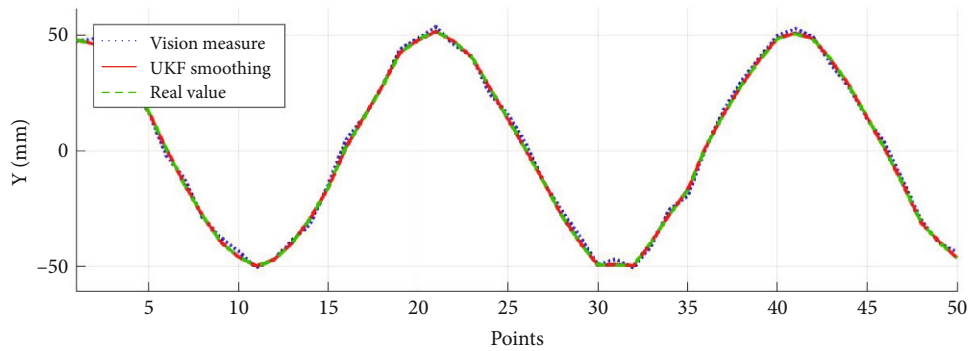


FIGURE 15: The tracking data on the Y-axis of the first 50 points in the experiment.

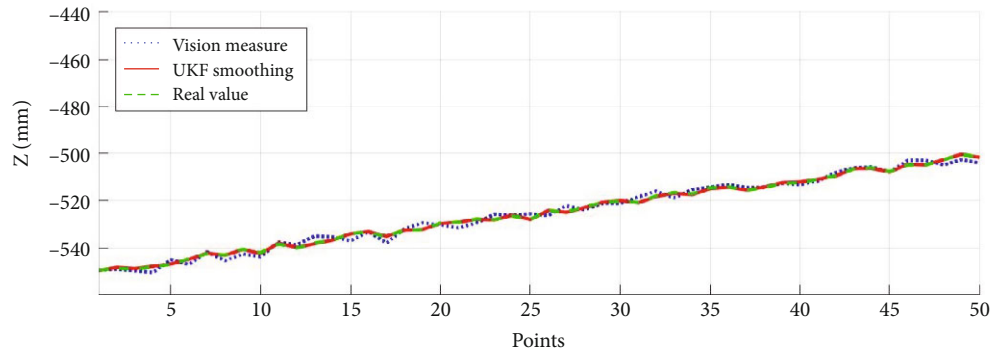


FIGURE 16: The tracking data on the Z-axis of the first 50 points in the experiment.

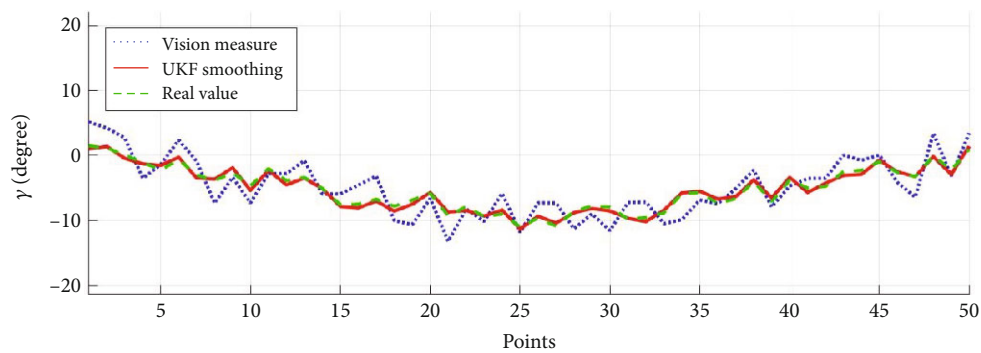


FIGURE 17: The tracking data on the rotation angle  $\gamma$  of the first 50 points in the experiment.

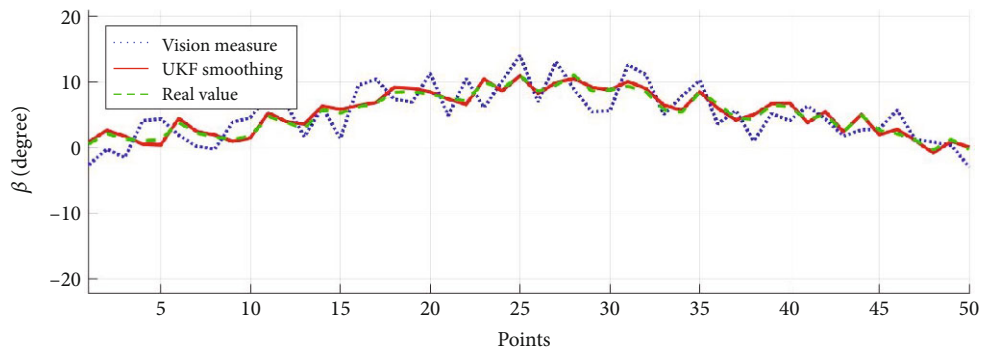


FIGURE 18: The tracking data on the rotation angle  $\beta$  of the first 50 points in the experiment.

instrument, which is used as the ground truth of the experiment.

Figures 14–19 show the tracking data of the first 50 points of the helix path in the experiment. The dot blue line indicates the pose estimation of the vision system. The dashed green line is the real pose of the plate, which is the manual measurement. The solid red line is the smoothed estimation after using the UKF. As showing in the figures, the hybrid filter corrects the pose measurement very close to the real pose although the measurement has a big noise due to the vibration of the real Hexa robot when moving. If the smoothed estimation is highly accurate, it will help the closed-loop motion control to be highly accurate. As stated in Section 3.3, motion control is not the scope of this

paper, so the experiment data is not shown in this result. For easy comparison of values, mean squared error (MSE) and mean absolute error (MAE) of the simulation are also summarized in Table 3. After using the UKF, the vision-based pose estimation is smoothed out, and the errors are also reduced significantly.

The high accurate result of the vision-based pose estimation using the proposed hybrid filter could be used as feedback in the dynamic path tracking control in conjunction with a position controller. This content will be covered in a different topic because of article's limitation.

*5.5. The Comparison.* Table 4 shows the comparison of the mean absolute error of different pose estimation methods

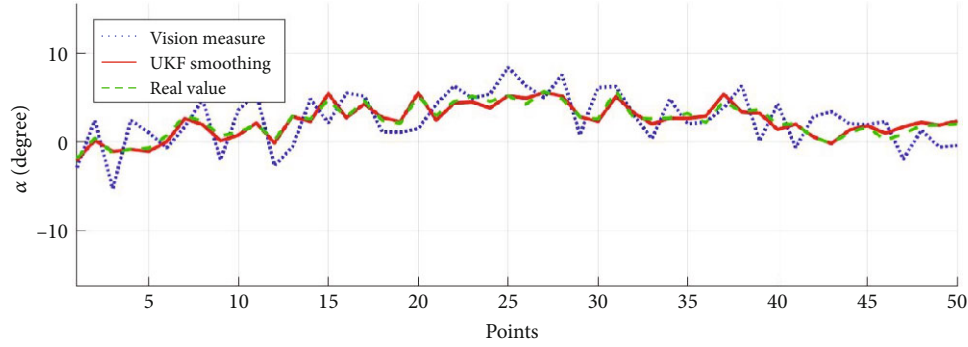


FIGURE 19: The tracking data on the rotation angle  $\alpha$  of the first 50 points in the experiment.

TABLE 3: The summary of errors in the experiment.

Pose	MAE		MSE		
	Vision	UKF	Vision	UKF	UKF
X (mm)	1.600	0.236	3.234		0.070
Y (mm)	1.614	0.230	3.458		0.068
Z (mm)	1.476	0.223	2.864		0.066
$\gamma$ (degrees)	1.917	0.423	4.894		0.227
$\beta$ (degrees)	2.128	0.396	5.719		0.207
$\alpha$ (degrees)	1.880	0.392	4.813		0.209

TABLE 4: The summary of the mean absolute error in some pose estimation methods.

Ref	Method	Robot	Mean absolute error					
			X (mm)	Y (mm)	Z (mm)	$\gamma$ (degrees)	$\beta$ (degrees)	$\alpha$ (degrees)
[12]	Sensor frame, reflectors	6-RSS parallel robot	0.319	0.393	0.360	0.086	0.074	0.043
[19]	Vision-based, AKF	PUMA 560	1.0	1.1	1.5	0.2	0.4	0.4
[20]	C-track 780 vision, AKF	KUKA KR 6-2	0.2	0.2	0.2	0.1	0.1	0.1
[21]	Vision-based, EKF	Delta Parallax LKF-2040	1.79	1.79	1.79	—	—	—
[28]	Vision-based, EKF	Robot arm	0.3	0.3	0.6	0.1	0.4	0.4
[29]	Vision-based, neural network, EKF	Tensegrity manipulator	2.568	2.204	1.663	2.269	2.128	2.054

with the proposed approach in this paper. The dynamic hybrid filter has encouraging results compared to most other methods. The pose estimation accuracy is inferior to the method in [20]. However, the method in [20] is applied to an industrial robot arm and uses a high-quality vision system. Meanwhile, the proposed method used a simple and cheap vision system that is easy to install and use. It also applies to a laboratory model of a Hexa parallel robot, which has low mechanical quality. If applied to a professional industrial robot with good hardware, this proposed approach may achieve better performance.

## 6. Conclusions

This paper proposes a novel approach for pose estimation of a Hexa parallel robot. The vision system using simple color feature points fixed on the surface of the end-effector and

Intel RealSense Camera D435i offers a cheap solution and high accuracy in pose estimation. The dynamic hybrid filter is designed to correct the vision-based pose measurement. The complementary filter is used to eliminate the noise of image processing due to environmental light source interference. The UKF is designed to smooth out the pose estimation of the vision system based on robot's kinematic parameters. The combination of two filters in the same control scheme contributes to increased stability and improved accuracy of robot's positioning. The simulation, experiment, and comparison results are encouraging. The measurement error of the vision system is reduced to 9 to 12 percent left in simulation and 14 to 21 percent left in experiment. Although the result is not too good if compared to methods using high-tech hardware, it is a feasible approach for low-cost robot systems. As a further improvement, EKF, AKF, and the proposed hybrid filter can be deployed on the same

Hexa parallel robot for more accurate comparison and evaluation of efficiency. The vision-based pose estimation using the proposed hybrid filter could be used as feedback in the dynamic path tracking control in conjunction with a position controller to analyze the performance in the real application. In this paper, the parameters of UKF are determined by tuning experimentally. In future works, an adaptive learning controller could be designed to determine those parameters automatically.

## Data Availability

The data of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was supported in part by the Ministry of Science and Technology, Taiwan, under grants MOST 109-2221-E-011-068 and MOST 109-2811-E-011-516.

## References

- [1] L. W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators, 1st Ed*, Wiley, Hoboken, NJ, USA, 1999.
- [2] S. Küçük, *Serial and Parallel Robot Manipulators: Kinematics, Dynamics, Control, and Optimization*, IntechOpen, London, UK, 2012.
- [3] Z. Pandilov and V. Dukovski, "Comparison of the characteristics between serial and parallel robots," *Acta Tehnica Corviniensis-Bulletin of Engineering*, vol. 7, no. 1, pp. 143–160, 2014.
- [4] J. P. Merlet, *Parallel Robots, 2nd Ed*, Springer, Dordrecht, The Netherlands, 2006.
- [5] D. Stewart, "A platform with six degrees of freedom," *Proceedings of the Institution of Mechanical Engineers*, vol. 180, no. 1, pp. 371–386, 1965.
- [6] F. Pierrot, P. Dauchez, and A. Fournier, "Hexa: a fast six-DOF fully parallel robot," in *Fifth International Conference on Advanced Robotics, Pisa, Italy*, pp. 1158–1163, 1991.
- [7] M. Uchiyama, "A 6 D.O.F parallel robot HEXA," *Advanced Robotics*, vol. 8, no. 6, p. 601, 1993.
- [8] J. Bohg, J. Romero, A. Herzog, and S. Schaal, "Robot arm pose estimation through pixel-wise part classification," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3143–3150, Hong Kong, China, 2014.
- [9] M. Švaco, B. Šekoranja, F. Šuligoj, and B. Jerbić, "Calibration of an industrial robot using a stereo vision system," *Procedia Engineering*, vol. 69, pp. 459–463, 2014.
- [10] M. Gaudreault, A. Joubair, and I. A. Bonev, "Self-calibration of an industrial robot using a novel affordable 3D measuring device," *Sensors*, vol. 18, no. 10, pp. 3380–3398, 2018.
- [11] M. S. Marwan, M. Simic, and F. Imad, "Calibration method for articulated industrial robots," *Procedia Computer Science*, vol. 112, pp. 1601–1610, 2017.
- [12] P. Li, R. Zeng, W. Xie, and X. Zhang, "Relative posture-based kinematic calibration of a 6-RSS parallel robot by optical coordinate measurement machine," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, article 172988141876586, 2018.
- [13] F. Hefner, S. Schmidbauer, and J. Franke, "Pose error correction of a robot end-effector using a 3D visual sensor for control cabinet wiring," *Procedia CIRP*, vol. 93, pp. 1133–1138, 2020.
- [14] M. FarzanehKaloorazi, I. A. Bonev, and L. Birglen, "Simultaneous path placement and trajectory planning optimization for a redundant coordinated robotic workcell," *Mechanism and Machine Theory*, vol. 130, pp. 346–362, 2018.
- [15] F. Janabi-Sharifi, "Visual servoing: theory and applications," in *Optomechatronic Systems Handbook, 1st Ed*, CRC Press, Cleveland, OH, USA, 2003.
- [16] C. J. Liang, K. M. Lundeen, W. McGee, C. C. Menassa, S. H. Lee, and V. R. Kamat, "A vision-based marker-less pose estimation system for articulated construction robots," *Automation in Construction*, vol. 104, pp. 80–94, 2019.
- [17] F. Zhou, Z. Chi, Z. Zhuang, and H. Ding, "3D pose estimation of robot arm with RGB images based on deep learning," in *International Conference on Intelligent Robotics and Applications*, pp. 541–553, Springer, Cham, Shenyang, China, 2019.
- [18] W. J. Wilson, C. C. Williams Hulls, and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.
- [19] M. Ficocelli and F. Janabi-Sharifi, "Adaptive filtering for pose estimation in visual servoing," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 19–24, Maui, HI, USA, 2001.
- [20] T. Shu, S. Gharaaty, W. F. Xie, A. Joubair, and I. A. Bonev, "Dynamic path tracking of industrial robots with high accuracy using photogrammetry sensor," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1159–1170, 2018.
- [21] E. J. G.-G. González, E. J. Gonzalez-Galvan, M. Maya, A. Cardenas, and D. Piovesan, "Estimation of camera-space manipulation parameters by means of an extended Kalman filter: applications to parallel robots," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, article 172988141984298, 2019.
- [22] X. Wu, E. Zakeri, and W. F. Xie, "Adaptive robust Kalman filter for vision-based pose estimation of industrial robots," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, pp. 298–302, Chengdu, China, 2019.
- [23] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Proc. SPIE, Signal Processing, Sensor Fusion, and Target Recognition VI*, pp. 182–193, Orlando, FL, United States, 1997.
- [24] N. Yadav and C. Bleakley, "Two-stage Kalman filtering for position estimation using dual inertial measurement units," in *SENSORS, 2011 IEEE*, pp. 1433–1436, Limerick, Ireland, 2011.
- [25] B. P. Huynh and Y. L. Kuo, "Dynamic filtered path tracking control for a 3RRR robot using optimal recursive path planning and vision-based pose estimation," *IEEE Access*, vol. 8, pp. 174736–174750, 2020.
- [26] S. Zhao, "Time derivative of rotation matrices: a tutorial, computer science," 2016, <http://arxiv.org/abs/1609.06088>.
- [27] L. Li and Y. Xia, "Stochastic stability of the unscented Kalman filter with intermittent observations," *Automatica*, vol. 48, no. 5, pp. 978–981, 2012.

- [28] W. J. Wilson, "Visual servo control of robots using Kalman filter estimates of relative pose," *IFAC Proceedings Volumes*, vol. 26, no. 2, pp. 633–638, 1993.
- [29] M. R. Kuzdeuov, M. Rubagotti, and H. A. Varol, "Neural network augmented sensor fusion for pose estimation of tensegrity manipulators," *IEEE Sensors Journal*, vol. 20, no. 7, pp. 3655–3666, 2020.