

Article

# MDPI

# Vision/Position Hybrid Control for a Hexa Robot Using Bacterial Foraging Optimization in Real-time Pose Adjustment

# Ba-Phuc Huynh <sup>1,2</sup>, Shun-Feng Su <sup>3</sup> and Yong-Lin Kuo <sup>1,4,\*</sup>

- <sup>1</sup> Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; hbphuc@vnkgu.edu.vn or hubaphu@gmail.com
- <sup>2</sup> Faculty of Engineering and Technology, Kien Giang University, Kien Giang 920000, Vietnam
- <sup>3</sup> Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; sfsu@mail.ntust.edu.tw
- <sup>4</sup> Center of Automation and Control, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
- \* Correspondence: kuo@mail.ntust.edu.tw

Received: 22 February 2020; Accepted: 2 April 2020; Published: 5 April 2020



Abstract: This paper presents a novel architecture of the vision/position hybrid control for a Hexa parallel robot. The 3D vision system is combined with the Proportional-Integral-Derivative (PID) position controller to form a two-level closed-loop controller of the robot. The 3D vision system measures the pose of the end-effector after the PID control. The measurement of the 3D vision system is used as a feedback of the second closed-loop control. The 3D vision system has a simple structure using two fixed symmetric cameras at the top of the robot and four planar colored markers on the surface of the end-effector. The 3D vision system detects and reconstructs the 3D coordinates of colored markers. Based on the distance and coplanarity constraints of the colored markers, the optimization problem is modeled for the real-time adjustment, which is implemented during the operation of the robot to minimize the measurement error of the 3D vision system due to both the initial calibration of the stereo camera and the external noise affecting image processing. The bacterial foraging optimization is appropriately configured to solve the optimization problem. The experiment is performed on a specific Hexa parallel robot to assess the effectiveness and feasibility of the proposed real-time adjustment using the bacterial foraging optimization. The experimental result shows that it has high accuracy and fast computation time although the experiment is conducted on a laptop with an average hardware configuration. An experimental comparison of the performance between the proposed method and another control method is also implemented. The results show the superiority and application potential of the proposed method.

**Keywords:** 3D vision system; bacterial foraging optimization (BFO); vision/position hybrid control; hexa parallel robot

# 1. Introduction

Since proposed by Stewart [1] in 1965, the parallel manipulators [2] have become popular. Several structures have been studied and proposed. A six-degree-of-freedom parallel structure was presented in [3,4], and today it is known as Hexa parallel robot. Its design is highly rigid that allows it to move and accelerate quickly with precision and stability. Therefore, it is suitable for pick and place industrial applications. However, kinematics and dynamics problems are difficult to solve [5,6]. The inverse kinematics problem is solvable in the closed-form expression while the forward kinematics problem

cannot be solved analytically [7]. The limit of the workspace [6] and the detection of singularity [8] need to be concerned.

Many control approaches have been proposed for industrial robots to make them suitable for many different applications. PID position control is commonly used in industrial robots. Force control is used to interact with the environment. Instead of performing each control individually, the force/position hybrid control combines the position data and the force/torque measurement to achieve the desired control trajectory [9–14]. For the Hexa parallel robot, a single PID position controller may not achieve the desired control effect. The complicated structure of a parallel manipulator makes it difficult to control the motion of the robot. After the PID control was performed to reach the desired position, the actuators slid a bit due to mechanical vibration and a dynamics coupling effect between the links. This causes the position error of the end-effector, which needs to be handled [15].

Today, machine vision has been widely used in the manufacturing and production industry. It plays an important role in inspection and analysis for automated applications that require high-speed, reliable, continuous, and repeat operation. In robotics, a vision-based control system is a popular feature, which provides information about the target objects or the pose (position and orientation) of the robot. In recent years, many vision-based methods have been proposed for different applications. Özgür et al. [16] introduce a vision-based and generic dynamic model for parallel kinematic manipulators. In [17], the authors focus on the uncalibrated visual tracking control for a visual servo manipulator system. A 2D vision system is used for calibrating the base frame of a mobile robotic drilling system [18]. Du et al. [19] present a teleoperated robotic 3D mapping system. It enables efficient human-guided mapping of remote environments. In [20], the authors use model predictive control for image-based visual servo control. It can stabilize a physically constrained mobile robot. In [21], a vision system is applied for the laser welding system of insulated mugs. The authors propose a vision-based seam tracking system using a fuzzy logic control method. Instead of using markers or sensors, Liang et al. [22] developed a marker-less vision system to detect the pose of a robot. It is based on the stacked hourglass network. However, it needs further modification to improve low accuracy.

In this paper, a vision/position hybrid controller is proposed and applied to a Hexa parallel robot. The 3D vision system (3DVS) is combined with the PID position controller to form a two-level closed-loop controller of the robot. The 3DVS measures the pose of the end-effector after the PID control. The measurement of the 3DVS is used as a feedback of the second closed-loop control, which ensures achieving the desired trajectory of the robot.

Machine vision based on the stereo camera provides more information about the target, making the control process more accurate. However, calibration is a matter of concern. A small deviation in the initial calibration may also affect the operation. Even though the initial calibration is done very well, the noise in image processing can cause measurement errors. Many calibration methods have been proposed to improve the quality of the 3DVS. In the calibration method proposed by Sun et al. [23], the interior, distortion, and exterior parameters of the cameras can be simultaneously obtained without known information of cameras. The measurement accuracy is high, but the structure of the calibration system is big and complex. Zhang et al. [24] propose a stereo vision-based self-calibration procedure for a robot arm. It uses a complex shape marker to estimate the pose of the robot. It also uses an external toolkit to identify the marker, so the accuracy is not controlled from within the program. Another 3DVS is built with a camera and two laser projectors in a robot controller for surface welding [25]. It is expensive and difficult to apply widely.

Other than the above studies, a novel real-time adjustment method is proposed in this paper to handle the error of the 3DVS due to both the initial calibration of the stereo camera and the external noise affecting image processing. The 3DVS has a simple structure with two fixed cameras at the top of the robot and four planar colored markers on the surface of the end-effector. The stereo camera detects and measures the coordinates of colored markers. Based on the distance and coplanarity constraints of the colored markers, the optimization problem is modeled for the real-time adjustment process, which is implemented during the operation of the robot to minimize the measurement error of the 3DVS. This

is a multi-dimensional optimization problem with a wide search space. Therefore, a suitable algorithm needs to be selected to achieve high efficiency.

Recently, bio-inspired algorithms [26] have been a popular approach for solving complex optimization problems in science and engineering. Ni et al. [27] present a survey of a bio-inspired intelligent algorithm (BIA) and applications for robot control. Particle swarm optimization, a kind of BIA, has been proposed for some applications of robot control [28–31]. Bacterial foraging optimization (BFO) for distributed optimization and control was introduced by Passino [32] in 2002. This is an evolutionary optimization algorithm, which emulates the foraging strategy of the E. Coli bacteria. It is suitable to solve non-gradient optimization problems. In this paper, BFO is used to solve the optimization problem of the real-time adjustment of the 3DVS. The experimental results show that the BFO algorithm has high accuracy and fast computation time. Therefore, this may be a promising solution for performing simultaneous intelligent adjustments in robot control.

Based on the above discussion, the primary contributions of this paper are summarized as follows:

- Although there are many studies of hybrid control in the literature, it is not often mentioned in applications of Hexa robots due to difficulty in control related to the complex structure of parallel manipulators. In this paper, a vision/position hybrid controller for Hexa parallel robot has been proposed. The 3DVS is combined with the PID position controller to form a two-level closed-loop controller of the robot. The 3DVS is designed with a simple structure consisting of one stereo camera at the top of the robot and four planar colored markers on the surface of the end-effector. Based on the colored markers, the 3DVS measures the pose of the end-effector after the PID control. The measurement of the 3DVS is used as a feedback of the second closed-loop control, which ensures achieving the desired trajectory of the robot.
- Based on the distance and coplanarity constraints of the colored markers, the optimization
  problem is modeled to minimize the measurement error of the 3DVS due to the error of camera
  parameters and the external noise affecting image processing. This is the real-time adjustment,
  which is implemented right in the operation of the robot to self-correct errors and adapt to
  environmental impacts.
- The BFO algorithm is appropriately configured to solve the optimization problem of the real-time
  adjustment process. The experimental results show that it has high accuracy and fast computation
  time although the experiment is conducted on a laptop with an average hardware configuration.

The remainder of this paper is arranged as follows: Section 2 describes the system and vision/position hybrid controller of the Hexa robot. Section 3 presents the 3DVS. Section 4 describes the real-time adjustment using BFO. Section 5 presents the calculation of the pose of the active plate. Section 6 describes the setup of the experiments. Section 7 discusses the experimental results. Section 8 is the conclusion.

#### 2. System Description

Figure 1 illustrates the model of the Hexa robot used in this study. The active plate is connected to the base by six pairs of arms and rods [15]. Two different coordinate systems are set on the robot. The fixed coordinate system {*O*} is located at the center of the base, which has the *Z*-axis perpendicular to the base and a positive direction facing up. The positive direction of the *X*-axis is towards the middle of the edge  $A_1A_2$ . The direction of the *Y*-axis is determined by the right-hand rule. The moving coordinate system {*P*} is located at the center of the active plate, which has the *Z*-axis perpendicular to the active plate and the positive direction facing up. The positive direction of the *X*-axis is towards the middle of the edge  $C_1C_2$ . The direction of the *Y*-axis is determined by the right-hand rule. The position of the *i*<sup>th</sup> arm is specified by the  $\theta_i$  angle. The position of the active plate is specified by the center point  $P(X_P, Y_P, Z_P)$  and the Euler angles  $(\alpha, \beta, \gamma)$ . Inverse kinematics equations of the Hexa robot [15] must be determined to convert the desired position of the active plate into the angles of arms.



Figure 1. The structure of a Hexa parallel robot and its model in SimMechanics.

Figure 2 is the architecture of the hybrid controller in which a vision system is combined with a PID position controller to form a two-level closed-loop controller of the Hexa robot.  $x_d$ ,  $x_i$ ,  $x_c$ , and  $x_r$  are vectors (6 × 1), which include both position and orientation information.  $\theta_c$ ,  $\theta_i$ , and  $\theta_r$  are vectors (6 × 1), which represent the rotation angles of six arms as shown in Figure 1.  $\tau$  is a vector (6 × 1), which represents torques applied to the arms.



Figure 2. The vision/position hybrid control architecture.

The rotation angles of arms are calculated as follow:

$$\theta_r = \mathbf{H}(x_r) \tag{1}$$

where H is the conversion function of the inverse kinematics problem [15];  $x_r$  is the reference position of the active plate;  $x_d$  is the desired position;  $x_c$  is the real position measured by the 3DVS.

The torque  $\tau$  can be expressed as follows:

$$\tau = \mathrm{T}(\theta_r) \tag{2}$$

where T is a transfer function of the position compensator.

The PID position control is the first closed-loop for controlling the rotation angle  $\theta_i$  of arms to reach the desired angle  $\theta_r$ . It works efficiently and has very high accuracy. However, because of disturbance, the position  $\theta_i$  is slid a bit to  $\theta_c$  resulting in the position error of the active plate [15]. The disturbance may be due to the following reasons:

- Dynamics are ignored in the system, so inertia and gravity affect positioning accuracy.
- The gearbox backlash and the mechanical vibration of the system also affect the accuracy.
- Slippage of the position of actuators due to the dynamics coupling effect between the links.

To eliminate the position error due to this disturbance, the double-loop PID control was proposed in [15]. In this paper, a 3DVS is used to measure the pose of the active plate after the first control loop. The high-accuracy measurement of the vision system is used as a feedback of the second closed-loop control, which ensures achieving the desired trajectory of the robot at high accuracy.

Figure 3 shows the 3DVS in the experimental model of the robot used in the study. The 3DVS includes a stereo camera fixed at the top of the robot and the colored markers on the surface of the active plate. The stereo camera points at the active plate, captures photos, and calculates the 3D coordinates of makers. The pose of the active plate  $x_i$  is specified by the 3D coordinates of the markers as shown in Section 5. The measurement process of the 3DVS is presented in Figure 4.

After the measurement, the pose of the active plate  $x_i$  is adjusted to minimize the error due to the initial calibration of the stereo camera and noise affecting the image processing in the measurement process. This adjustment is performed in real time using BFO as presented in Section 4.

The inverse kinematics problem and PID position control were detailed in [15]. This paper focuses on the 3DVS and real-time adjustment using BFO. After that, the experiment of hybrid control was implemented to verify the proposed method as presented in Section 6. A comparison with the traditional PID control and the double-loop PID control in [15] was also conducted to show the feasibility and superiority of the proposed method.



Figure 3. The experimental model.



**Figure 4.** The measurement process of the 3D vision system (3DVS), where HSV (hue, saturation, value) is a color model.

#### 3. The 3D Vision System

#### 3.1. The 3D Reconstruction of the Stereo Camera

Figure 5 shows the so-called pinhole camera model of the symmetric vision system. For each camera, a 3D point of the mark (X, Y, Z) is projected onto a point (u, v) in the image plane using a perspective transformation [33] in which the intrinsic and extrinsic parameters of cameras are determined by the initial calibration using checkerboard and the stereo camera calibrator of *MATLAB*. These parameters are used to compute rectification and undistortion transformations for each head of a camera in the 3DVS [33]. Then image frames become undistorted images, which can be used to calculate the depths and coordinates of the mark as follows:

$$Z = \frac{D_c \cdot f_x^l}{disparity} = \frac{D_c \cdot f_x^l}{u_l - u_r}$$

$$X = Z \cdot \left(\frac{u_l - c_x^l}{f_x^l}\right)$$

$$Y = Z \cdot \left(\frac{v_l - c_y^l}{f_y^l}\right)$$
(3)

where  $D_c$  is the distance between two cameras (baseline) expressed in millimeter unit.  $(u_l, v_l)$  and  $(u_r, v_r)$  are the projection points of the left and right cameras, respectively, expressed in the pixel unit.  $(c_x^l, c_y^l)$  is the principal point of the left camera.  $f_x^l$  and  $f_y^l$  are focal lengths of the left camera. All parameters here are based on the undistorted image frames.

Note that, in image processing and marks detection, the projection points  $(u_l, v_l)$  and  $(u_r, v_r)$  are counted by real numbers instead of integers. The goal is to achieve high accuracy in the adjustment process and avoid errors due to pixel rounding because deviations per pixel can lead to a very large calculation error. The projection points of the markers in the left and right images are determined by shape, color, and area.



Figure 5. The so-called pinhole camera model of the symmetric vision system.

# 3.2. The Constraints of the Colored Markers

The 3DVS uses four planar colored markers to create two constraints:

- Distance constraints: the distances from the red, yellow, and green markers to the blue mark are known constants.
- Coplanarity constraint: four colored markers are on the surface of the plate, so their central points must be on the same plane.

These constraints will be used for the optimization problem in real-time adjustment. The coordinates of the colored markers after the adjustment are used to calculate the Euler angles of the active plate.

# 3.2.1. Distance Constraints

From (3), the coordinates of the red, yellow, and green markers are calculated as follows:

$$Z^{j} = \frac{D_{c} \cdot f_{x}^{l}}{disparity} = \frac{D_{c} \cdot f_{x}^{l}}{u_{l}^{j} - u_{r}^{j}}$$

$$X^{j} = Z^{j} \cdot \left(\frac{u_{l}^{j} - c_{x}^{l}}{f_{x}^{l}}\right)$$

$$Y^{j} = Z^{j} \cdot \left(\frac{v_{l}^{j} - c_{y}^{l}}{f_{y}^{l}}\right)$$
for  $j = R, Y, G$ 

$$(4)$$

where the notations R, Y, and G indicate the colors red, yellow, and green of markers, respectively.

The distances from the blue marker to the red, yellow, and green markers are calculated respectively as follow:

$$D^{j} = \sqrt{(X^{j} - X^{B})^{2} + (Y^{j} - Y^{B})^{2} + (Z^{j} - Z^{B})^{2}}$$
for  $j = R, Y, G$ 
(5)

The distances  $D^{j}$  are compared with the real distance  $D_{r}$  and the errors are calculated as follows:

$$\varepsilon^{j} = \left| D^{j} - D_{r} \right|, \text{ for } j = R, Y, G$$
(6)

In optimal calibration problem, the distance errors  $\varepsilon^{j}$  are as small as possible.

#### 3.2.2. Coplanarity Constraints

The plane  $\Pi$  passing through the red, yellow, and green markers at points  $(X^j, Y^j, Z^j)$  can be described by an equation of the form:

$$\Pi : ax + by + cz + 1 = 0 \tag{7}$$

with:

$$a = \frac{-1}{\Delta} \begin{vmatrix} 1 & Y^{R} & Z^{R} \\ 1 & Y^{Y} & Z^{Y} \\ 1 & Y^{G} & Z^{G} \\ X^{R} & Y^{R} & 1 \\ X^{G} & Y^{G} & 1 \end{vmatrix} \qquad b = \frac{-1}{\Delta} \begin{vmatrix} X^{R} & 1 & Z^{R} \\ X^{Y} & 1 & Z^{Y} \\ X^{G} & 1 & Z^{G} \\ X^{R} & Y^{R} & Z^{R} \\ X^{Y} & Y^{Y} & Z^{R} \\ X^{G} & Y^{G} & Z^{G} \end{vmatrix}$$

The distance from the blue marker to the plane  $\Pi$  is:

$$\delta = \frac{\left|aX^{B} + bY^{B} + cZ^{B} + 1\right|}{\sqrt{a^{2} + b^{2} + c^{2}}}$$
(8)

In an optimal calibration problem, the distance  $\delta$  is as small as possible.

#### 4. Real-time Adjustment Using BFO

Even though the stereo camera was initially calibrated very well, the coordinates calculated from (4) still have errors. The errors are not only due to the camera parameters but also due to the fluctuation of the light around the camera, which leads to a wrong positioning of the colored markers in image processing. The error of image processing significantly affects the results. Assume that  $D_c = 120$  mm,  $f_x = 700$  pixels, *disparity* = 140 pixels, and depth Z = 600 mm. If the *disparity* is changed only one pixel after the image processing, the depth will be Z = 595.745 mm. The error is about 4.26 mm (0.71% of depth). This error is still good when most stereo cameras on the market today have an error of about 2% of the measuring distance. However, it is not good enough for applications that require high accuracy like Hexa robots.

To eliminate the errors, a real-time adjustment is executed during the robot moving. This process will adjust the projection points to change the 3D coordinates of the colored marks into the correct values. It takes place after image processing, so it is possible to handle both the remaining error due to the initial calibration of the stereo camera and the error of image processing due to light source noise. First, the optimization problem is established. Then, the BFO algorithm is used to solve it.

#### 4.1. The Optimization Problem

Assuming that all projection points are shifted, the calculation in (4) is adjusted as follows:

$$Z^{j} = \frac{D_{c} \cdot f_{x}^{l}}{disparity} = \frac{D_{c} \cdot f_{x}^{l}}{(u_{l}^{j} + \delta u_{l}^{j}) - (u_{r}^{j} + \delta u_{r}^{j})}$$

$$X^{j} = Z^{j} \cdot \left(\frac{(u_{l}^{j} + \delta u_{l}^{j}) - c_{x}^{l}}{f_{x}^{l}}\right)$$

$$Y^{j} = Z^{j} \cdot \left(\frac{(v_{l}^{j} + \delta v_{l}^{j}) - c_{y}^{l}}{f_{y}^{l}}\right)$$
for  $j = B, R, Y, G$ 

$$(9)$$

where the notations *B*, *R*, *Y*, and *G* indicate the colors blue, red, yellow, and green of marks, respectively.  $\delta u_1^j$ ,  $\delta u_r^j$ , and  $\delta v_1^j$  are correction values of  $u_1^j$ ,  $u_r^j$ , and  $v_1^j$ , respectively.

The real-time adjustment is modeled as a multi-objective optimization problem based on the distance and coplanarity constraints of the colored markers as:

$$\min: \begin{cases} F_1(\delta u_l^i, \delta v_l^i, \delta u_r^i) = \sum_j \varepsilon^j \\ F_2(\delta u_l^i, \delta v_l^i, \delta u_r^i) = \delta \\ for \ i = B, R, Y, G \ and \ j = R, Y, G \end{cases}$$
(10)

subject to : 
$$\begin{cases} L_{u} \leq \delta u_{l}^{i} \leq U_{u} \\ L_{v} \leq \delta v_{l}^{i} \leq U_{v} \\ L_{u} \leq \delta u_{r}^{i} \leq U_{u} \\ \varepsilon^{j} \leq \varepsilon_{d} \text{ and } \delta \leq \delta_{d} \\ for \ i = B, R, Y, G \text{ and } j = R, Y, G \end{cases}$$

where the errors  $\varepsilon^j$  are calculated by (5) and (6). Note that, the coordinates  $(X^j, Y^j, Z^j)$  of the colored marks in (5) are calculated by (9).  $L_*$  and  $U_*$  are lower and upper boundaries of the parameters, respectively.  $\varepsilon_d$  and  $\delta_d$  are the desired distance errors.

To determine the goodness of a solution, in programming, a total cost function will be calculated as follow:

$$J = w_R \cdot \varepsilon^R + w_Y \cdot \varepsilon^Y + w_G \cdot \varepsilon^G + w_\delta \cdot \delta \tag{11}$$

Note that the weighting values  $w_*$  would be adjusted experimentally to get the best solution.

#### 4.2. Introduction to BFO

The bacterial foraging algorithm for distributed optimization and control was introduced by Passino [32] in 2002. This is an evolutionary optimization algorithm, which emulates the foraging strategy of the E. Coli bacteria. It is suitable to solve non-gradient optimization problems. In this method, each solution will be expressed as a bacterium determined by a position vector  $\theta$ . The set of solutions for the problem is expressed as a bacterial population. In foraging for food (optimizing the problem), the main action of bacteria, including gathering to a nutrient region (good solution) and avoiding noxious region (not good solution) by changing the moving direction, is called chemotaxis. Other processes, including reproduction, elimination, and dispersal, represent natural evolutionary rules: good individuals will be retained while bad ones are at risk of being eliminated. In this paper, BFO is used to solve the multi-objective optimization problem in Section 4.1.

#### 4.3. Applying BFO for Real-time Adjustment

The flowchart of the BFO algorithm is shown in Figure 6. The position of the *i*<sup>th</sup> bacterium  $\theta^i$  is defined by a vector  $M \times 1$ , where  $i = 1 \dots S$ , M = 12 is the number of parameters to be optimized including  $\delta u_1^B$ ,  $\delta v_1^B$ ,  $\delta u_1^R$ ,  $\delta v_1^P$ ,  $\delta u_1^Q$ ,  $\delta v_1^G$ ,  $\delta u_r^G$ ,  $\delta u_r^G$ ,  $\delta u_r^G$ , and  $\delta u_r^G$  as presented in Section 4.1.

The BFO parameters need to be defined including the number of bacteria *S*, swimming length *Ns*, number of chemotaxis steps *Nc*, number of reproduction steps *Nr*, number of elimination and dispersal events *Ne*, and the probability of elimination/dispersal *Ped*. Step size of swimming *C* for bacteria is a vector  $M \times 1$  corresponding to the vector  $\theta$ .



Figure 6. The flowchart of the bacterial foraging optimization (BFO) algorithm.

# 4.3.1. Initialization

This step sets the values of *S*, *Ns*, *Nc*, *Nr*, *Ne*, *Ped*, and *C*. The initial values of  $\theta(i)$ ,  $i = 1 \dots S$ , are randomly created in the boundaries of parameters. A function J(i) is defined to evaluate the cost of the *i*<sup>th</sup> bacterium. A function F(i) is defined to evaluate the fitness of the *i*<sup>th</sup> bacterium as follows:

$$F(i) = J(i) + I_a(i) + I_r(i)$$
(12)

where  $I_a(i)$  and  $I_r(i)$  are attractant and repellant, respectively, between the *i*<sup>th</sup> bacterium and other bacteria.

The attractant and repellant can be calculated as follows:

$$I_{a}(i) = \sum_{n=1}^{S} \left[ -d_{attract} \exp\left(-w_{attract} \sum_{m=1}^{M} \left(\theta^{i}[m] - \theta^{n}[m]\right)^{2}\right) \right]$$

$$I_{r}(i) = \sum_{n=1}^{S} \left[ h_{repellant} \exp\left(-w_{repellant} \sum_{m=1}^{M} \left(\theta^{i}[m] - \theta^{n}[m]\right)^{2}\right) \right]$$
(13)

where  $d_{attract}$  and  $w_{attract}$  are the depth and width of the attractant, respectively.  $h_{repellant}$  and  $w_{repellant}$  are the height and width of the repellant, respectively.

In the beginning, the index of the elimination and dispersal step is set to zero (l = 0).

#### 4.3.2. Chemotaxis

A direction vector  $\Delta$  is randomly generated for each bacterium.  $\Delta(i)$  is a vector  $M \times 1$  corresponding to the vector  $\theta$ . Each element of the vector  $\Delta(i)$  is a random integer in the range [-1,1]. The movement (swimming) of a bacterium is done and the new position is updated as follows:

$$\theta(i) = \theta(i) + C \cdot \Delta(i) \tag{14}$$

In each movement, the new cost J is calculated and compared with the old values to keep the best position  $\theta^i$ . The new fitness F is calculated and compared with the old values. If the fitness F is improved, the swimming process will continue; otherwise, it is stopped. The health of  $i^{th}$  bacterium H(i) is the cumulative sum of the fitness F in a chemotaxis step. The maximum iterations of the swimming process are Ns. After the swimming process, the best cost J(i), health H(i), and position vector  $\theta^i$  of the  $i^{th}$  bacterium are stored to use in the reproduction step.

## 4.3.3. Reproduction

The position vectors  $\theta^i$  of bacteria are sorted in ascending order of the health *H*. The first half of the position vectors  $\theta^i$  will be copied and overwritten into the second half. This is to remove bad elements and retain good elements. The total number of bacteria remains the same.

#### 4.3.4. Elimination and Dispersal

This step reviews all bacteria. At each position, a random number is generated. If this number is less than the probability of elimination/dispersal *Ped*, the elimination and dispersal event is executed. The easiest way to execute this event is to replace the current position vector with a new random vector as in the initialization step.

# 4.3.5. The Finish

After the BFO stopped, the position vector  $\theta^i$  with the highest health *H* is the optimal adjustment, which is used to calculate the correct coordinates of the colored markers by (9).

#### 5. The Pose of the Active Plate

The pose is the position and orientation. After the real-time adjustment, the correct coordinates of colored markers with respect to the coordinate system of the stereo camera are transformed to the world coordinates of the Hexa robot as presented in Figure 1. The position of the active plate is specified by the world coordinate of the blue marker. The orientation (Euler angles) of the active plate is specified as follows:

• The angle  $\alpha$  (rotation around the *z*-axis): is the angle between the *x*-axis and the line passing through the blue and red markers.

- The angle  $\beta$  (rotation around the *y*-axis): is the angle between the plane  $\Pi$  and the *OYZ*-plane, where the *OYZ*-plane has an equation of the form x = 0. The plane  $\Pi$  is described by (7).
- The angle  $\gamma$  (rotation around the *x*-axis): is the angle between the plane  $\Pi$  and the *OXZ*-plane, where the *OXZ*-plane has an equation of form y = 0.

#### 6. Experimental Setup

#### 6.1. Experimental Configuration Parameters

The experiment of the proposed method is implemented on the real robot as shown in Figure 3. The control software runs on a laptop with CPU Intel Core i5-2430M, Intel HD Graphics 3000 onboard, DDR3 8 GB, SSD Intel 120 GB. Figure 7 is the interface of the control software.



Figure 7. The control software interface.

The parameters of the BFO algorithm in the real-time adjustment are set as follows: number of bacteria S = 16, number of chemotactic iteration Nc = 16, swimming length Ns = 16, step size of swimming C[n] = 0.05, for n = 1, ..., 12, number of reproduction steps Nr = 6, number of elimination and dispersal events Ned = 4, probability of elimination and dispersal Ped = 0.26, depth of attractant  $d_{attract} = 0.1$ , width of attractant  $w_{attract} = 0.2$ , height of repellant  $h_{repellant} = 0.1$ , and width of repellant  $w_{repellant} = 10$ . The parameters of the optimization problem are  $\varepsilon_d = 1$  mm,  $\delta_d = 1$  mm,  $L_* = -2$  *pixel*, and  $U_* = 2$  *pixel*.

#### 6.2. Validation of the Proposed Method

To evaluate the effectiveness of the real-time adjustment using BFO, the active plate is moved through 60 different positions. The positions are selected along with the height of the workspace. The purpose is to evaluate the robustness of the proposed method because the measurement error is small at positions near the cameras and large at positions far away from the cameras. After each position, the pose of the active plate is measured by 3DVS. The measurement is recorded twice: before and after the real-time adjustment using BFO. The computation time of the BFO algorithm at each point is also measured for further analysis.

#### 6.3. The Comparison

As discussed in Section 2, to eliminate the position error due to the disturbance, the double-loop PID controller was proposed in [15]. In this paper, the vision/position hybrid controller has been proposed. For comparison, the same experiment is repeated, in which the active plate is moved on a helix trajectory as shown in Figure 8. The angles of motors are tracked and recorded for further analysis.



Figure 8. The helix trajectory.

# 7. The Discussion of Experimental Results

#### 7.1. Validation Result

Figures 9–14 show the tracked data of sixty positions in the validation of the real-time adjustment as discussed in Section 6.2. The blue dot lines represent the real pose of the active plate. The green dash lines represent the pose of the active plate measured by the 3DVS before the real-time adjustment using BFO. The red lines represent the pose of the active plate after the real-time adjustment using BFO. As shown in figures, the pose values of the active plate are adjusted close to the real (desired) values after doing the real-time adjustment using BFO. Table 1 summaries the errors in the validation. The mean absolute error (*MAE*), mean squared error (*MSE*), and normalized root-mean-square error (*NRMSE*) are significantly reduced after the real-time adjustment using BFO. The mean absolute error is less than 0.9305 mm in axes and less than 0.2802 degrees in Euler angles. This accuracy is good enough for the most common applications of the Hexa parallel robot.



Figure 9. Validation of the real-time adjustment using BFO in *x*-axis.



Figure 10. Validation of the real-time adjustment using BFO in *y*-axis.







**Figure 12.** Validation of the real-time adjustment using BFO in Euler angle  $\alpha$ .



**Figure 13.** Validation of the real-time adjustment using BFO in Euler angle  $\beta$ .



**Figure 14.** Validation of the real-time adjustment using BFO in Euler angle  $\gamma$ .

Table 1. Summary of the errors in the validation (using millimeter for translation and degree	for
rotation).	

Error	Before Real-Time Adjustment			After Real-Time Adjustment Using BFO		
	X	Ŷ	Ζ	X	Ŷ	Ζ
MAE	3.15805	3.47503	4.45033	0.89908	0.93045	0.77919
MSE	11.42375	13.96472	30.04942	0.85175	0.90952	0.69280
NRMSE	0.02447	0.02885	0.03758	0.00657	0.00686	0.00604
Error	Before Real-Time Adjustment			After Real-Time Adjustment Using BFO		
	α	β	γ	α	β	γ
MAE	1.88882	2.13933	1.76775	0.28015	0.25887	0.27514
MSE	4.65455	5.79043	3.72159	0.08151	0.07025	0.07843
NRMSE	0.16745	0.17822	0.12105	0.02704	0.02034	0.01929

Figure 15 shows the computation time of the real-time adjustment using BFO. The average computation time is 149.5516 ms. The maximum computation time is 272.0304 ms. This computation time is acceptable with a non-gradient optimization problem using the BFO algorithm. Normally, the real-time adjustment is performed at the end of each movement, so it has a negligible effect on the performance of the Hexa robot. In the future, when the computer is faster and the computation time is shorter, the parameters of the BFO algorithm could be expanded to increase the accuracy.



Figure 15. The computation time of the real-time adjustment using BFO.

#### 7.2. Comparison Result

Figures 16–21 presents the resulted comparisons of the rotation angles between using the double-loop PID controller [15] and using the proposed vision/position hybrid controller when the active plate is moved on the helix trajectory as discussed in Section 6.3. Table 2 shows the resulted errors of the comparison. As presented in Section 2, the effect of disturbance behind the PID control loop results in position errors. In the case of double-loop PID controller, if the active plate deviates from the desired trajectory, the double-loop PID controller adjusts it to the correct position. The maximum *MAE* is 0.34077 degrees. In the case of the proposed hybrid controller, *MSE*, *MAE*, and *NRMSE* are much lower. The maximum *MAE* is only 0.1715 degrees. This superiority is due to the real-time adjustment using the BFO algorithm.



**Figure 16.** Comparison of rotation angle  $\theta_1$  using the double-loop PID controller and the proposed hybrid controller.



**Figure 17.** Comparison of rotation angle  $\theta_2$  using the double-loop PID controller and the proposed hybrid controller.



**Figure 18.** Comparison of rotation angle  $\theta_3$  using the double-loop PID controller and the proposed hybrid controller.



**Figure 19.** Comparison of rotation angle  $\theta_4$  using the double-loop PID controller and the proposed hybrid controller.



**Figure 20.** Comparison of rotation angle  $\theta_5$  using the double-loop PID controller and the proposed hybrid controller.



**Figure 21.** Comparison of rotation angle  $\theta_6$  using the double-loop PID controller and the proposed hybrid controller.

θ	MSE		MAE (Degrees)		NRMSE	
	Double-Loop PID	Proposed Method	Double-Loop PID	Proposed Method	Double-Loop PID	Proposed Method
1	0.24507	0.07827	0.33689	0.16484	0.00728	0.00413
2	0.22764	0.07742	0.33918	0.17136	0.00698	0.00408
3	0.23438	0.08534	0.33019	0.17191	0.00594	0.00358
4	0.22605	0.07466	0.33611	0.16810	0.00583	0.00335
5	0.23187	0.07403	0.33084	0.16179	0.00585	0.00331
6	0.22733	0.07490	0.34077	0.16926	0.00577	0.00332

Table 2. The resulted errors of the comparison.

#### 8. Conclusions

In this study, a novel architecture of the vision/position hybrid controller is proposed and applied to the Hexa parallel robot. The 3DVS is combined with the PID position controller to form a two-level closed-loop controller of the robot. The 3DVS is designed with a simple structure consisting of one stereo camera at the top of the robot and four planar colored markers on the surface of the end-effector. Based on the colored markers, the 3DVS measures the pose of the end-effector after the PID control. The measurement of the 3DVS is used as a feedback of the second closed-loop control, which ensures achieving the desired trajectory of the robot. To minimize the measurement error of the 3DVS due to the error of the stereo camera parameters and the external noise affecting image processing, the optimization problem is modeled based on the distance and coplanarity constraints of the colored markers. This optimization problem is solved by using BFO in the real-time adjustment process, which is implemented right in the operation of the robot. The experiment is performed on a specific Hexa parallel robot to assess the effectiveness and feasibility of the proposed real-time adjustment using the BFO. The experimental result shows that it has high accuracy. The MAE decreased significantly from about 4.45 mm for translation and 2.14 degrees for rotation to about 0.93 mm and 0.26 degrees, respectively. The computation time of the BFO algorithm is encouraging with responses in milliseconds, although the experiment was conducted on a laptop with an average hardware configuration. Besides, the experimental comparison between the proposed method and another control method was also performed. Compared to the double-loop PID control method in another research, the MAE of rotation angles of the arms is reduced by about 50 percent. The results show the superiority and application potential of the proposed method.

**Author Contributions:** Conceptualization, B.-P.H. and Y.-L.K.; implementation, B.-P.H.; validation, B.-P.H. and Y.-L.K.; formal analysis, B.-P.H. and Y.-L.K.; writing—original draft preparation and editing, B.-P.H.; review and editing, S.-F.S. and Y.-L.K.; project administration, Y.-L.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2221-E-011-135.

Conflicts of Interest: The authors declare no conflict of interest.

# References

- 1. Stewart, D. A platform with six degrees of freedom. Proc. Inst. Mech. Eng. 1965, 180, 371–386. [CrossRef]
- 2. Merlet, J.-P. Parallel Robots, 2nd ed.; Springer: Dordrecht, The Netherlands, 2006.
- 3. Pierrot, F.; Dauchez, P.; Fournier, A. Hexa: A fast six-DOF fully parallel robot. In Proceedings of the Fifth International Conference on Advanced Robotics, Pisa, Italy, 19–22 June 1991; pp. 1158–1163.
- 4. Uchiyama, M. A 6 d.o.f parallel robot HEXA. *Adv. Robot.* **1994**, *8*, 601. [CrossRef]
- Wang, X.; Tian, Y. Inverse dynamics of hexa parallel robot based on the lagrangian equations of first type. In Proceedings of the International Conference on Mechanic Automation and Control Engineering, Wuhan, China, 26–28 June 2010; pp. 3712–3716.
- Filho, S.-T.; Cabral, E. Kinematics and workspace analysis of a parallel architecture robot: The hexa. In Proceedings of the ABCM Symposium Series in Mechatronics, Ouro Preto, Minas Gerais, Brasil, 6–11 November 2005; pp. 158–165.
- Dehghani, M.; Ahmadi, M.; Khayatian, A.; Eghtesad, M.; Farid, M. Neural network solution for forward kinematics problem of hexa parallel robot. In Proceedings of the American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 4214–4219.
- 8. De Queiroz, E.-M.; Bier, C.-C.; Campos, A.; Maass, J.; Guenther, R. Direct singularity avoidance strategy for the hexa parallel robot. In Proceedings of the ABCM Symposium Series in Mechatronics, Ouro Preto, Minas Gerais, Brasil, 6–11 November 2005; pp. 182–189.
- 9. Ribeiro, R.; Hsu, L.; Costa, R.; Lizarralde, F. Passivity-based cascade hybrid pose/force robot control. In Proceedings of the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 22–26 August 2019; pp. 806–811.
- 10. Ferretti, G.; Magnani, G.; Rocco, P. Toward the implementation of hybrid position/force control in industrial robots. *IEEE Trans. Robot. Autom.* **1997**, *13*, 838–845. [CrossRef]
- 11. Raibert, M.-H.; Craig, J.-J. Hybrid position/force control of manipulators. *J. Dyn. Syst. Meas. Control* **1981**, 103, 126–133. [CrossRef]
- Chen, B.-H.; Wang, Y.-H.; Lin, P.-C. A hybrid control strategy for dual-arm object manipulation using fused force/position errors and iterative learning. In Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Auckland, New Zealand, 9–12 July 2018; pp. 39–44.
- Craig, J.-J.; Raibert, M.-H. A systematic method of hybrid position/force control of a manipulator. In Proceedings of the COMPSAC 79. Proceedings. Computer Software and The IEEE Computer Society's Third International Applications Conference, Chicago, IL, USA, 6–8 November 1979; pp. 446–451.
- 14. Fisher, W.-D.; Mujtaba, M.-S. Hybrid position/force control: A correct formulation. *Int. J. Robot. Res.* **1992**, *11*, 299–311. [CrossRef]
- 15. Huynh, B.-P.; Wu, C.-W.; Kuo, Y.-L. Force/Position hybrid control for a hexa robot using gradient descent iterative learning control algorithm. *IEEE Access* **2019**, *7*, 72329–72342. [CrossRef]
- 16. Özgür, E.; Dahmouche, R.; Andreff, N.; Martinet, P. A vision-based generic dynamic model of PKMs and its experimental validation on the quattro parallel robot. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besacon, France, 8–11 July 2014; pp. 937–942.
- 17. Liu, Z.; Wang, F.; Zhang, Y. Adaptive visual tracking control for manipulator with actuator fuzzy dead-zone constraint and unmodeled dynamic. *IEEE Trans. Syst. ManCybern. Syst.* **2015**, *45*, 1301–1312. [CrossRef]
- 18. Mei, B.; Zhu, W.; Yuan, K.; Ke, Y. Robot base frame calibration with a 2D vision system for mobile robotic drilling. *Int. J. Adv. Manuf. Technol.* **2015**, *80*, 1903–1917. [CrossRef]
- 19. Du, J.; Mouser, C.; Sheng, W. Design and evaluation of a teleoperated robotic 3-d mapping system using an rgb-d sensor. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 718–724. [CrossRef]
- 20. Ke, F.; Li, Z.; Xiao, H.; Zhang, X. Visual servoing of constrained mobile robots based on model predictive control. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 1428–1438. [CrossRef]
- 21. Fang, Z.; Weng, W.; Wang, W.; Zhang, C.; Yang, G. A vision-based robotic laser welding system for insulated mugs with fuzzy seam tracking control. *Symmetry* **2019**, *11*, 1385. [CrossRef]

- 22. Liang, C.-J.; Lundeena, K.-M.; McGeeb, W.; Menassaa, C.-C.; Leea, S.-H.; Kamat, V.-R. A vision-based marker-less pose estimation system for articulated construction robots. *Autom. Constr.* **2019**, *104*, 80–94. [CrossRef]
- 23. Sun, P.; Lu, N.-G.; Dong, M.-L.; Yan, B.-X.; Wang, J. Simultaneous all-parameters calibration and assessment of a stereo camera pair using a scale bar. *Sensors* **2018**, *18*, 3964. [CrossRef] [PubMed]
- 24. Zhang, X.; Song, Y.; Yang, Y.; Pan, H. Stereo vision based autonomous robot calibration. *Robot. Auton. Syst.* **2017**, *93*, 43–51. [CrossRef]
- 25. Idrobo-Pizo, G.-A.; Motta, J.-M.-S.-T.; Sampaio, R.-C. A Calibration method for a laser triangulation scanner mounted on a robot arm for surface mapping. *Sensors* **2019**, *19*, 1783. [CrossRef] [PubMed]
- 26. Ding, Y.; Chen, L.; Hao, K. *Bio-Inspired Collaborative Intelligent Control and Optimization*; Springer Nature: Singapore, 2018.
- 27. Ni, J.; Wu, L.; Fan, X.; Yang, S.-X. Bioinspired intelligent algorithm and its applications for mobile robot control: A survey. *Comput. Intell. Neurosci.* **2016**, 2016, 1–16. [CrossRef] [PubMed]
- 28. Zeng, N.; Zhang, H.; Chen, Y.; Chen, B. Path planning for intelligent robot based on switching local evolutionary PSO algorithm. *Assem. Autom.* **2015**, *36*, 120–126. [CrossRef]
- 29. Wang, B.; Li, S.; Guo, J.; Chen, Q.-W. Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. *Neurocomputing* **2018**, *282*, 42–51. [CrossRef]
- Zhang, G.; Jia, S.; Zeng, D.; Zheng, Z. object detection and grabbing based on machine vision for service robot. In Proceedings of the IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 1–3 November 2018; pp. 89–94.
- Farag, R.-M.-A.; Saad, M.-S.; Emara, H.; Bahgat, A. Three-dimensional localization of known objects for robot arm application based on a particle swarm optimized low end stereo vision system. In Proceedings of the IECON 2018 -44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 2736–2741.
- 32. Passino, K.-M. Biomimicry of bacterial foraging for distributed optimization. *IEEE Control Syst. Mag.* 2002, 22, 52–67.
- 33. OpenCV.org. Available online: https://docs.opencv.org/ (accessed on 16 January 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).