

Visual Servo Control of the Hexapod Robot With Obstacle Avoidance

Wen-Shyong Yu and Chiau-Wei Huang

Abstract—This paper is to design a visual servo control for a hexapod robot with obstacle avoidance. The implementation of the motion control for the hexapod robot using the inverse kinematics and visual recognition system is used to achieve the trajectory tracking with obstacle avoidance. The control structure is composed of three parts: a tilt camera based on the concepts of mechanical geometry, visual servo control systems, and motion dynamics for trajectory tracking and obstacle avoidance. First, the depth between the obstacle and robot based on the proportion of the size of the area is constructed. Then, the image processing is used to identify whether there are any obstacles in the front, and make it as a feedback to the servo control system. For image recognition, we use OpenCV to process environment to the grayscale and binarization, filter noise through erosion and dilation, and then fill all of the contours using Sobel edge detection, and finally calculate the area and compare that with each other. Finally, some experiments for a hexapod robot with obstacle avoidance are used to validate the performance of the proposed control scheme.

Index Terms: Hexapod robot, visual servo control, motion dynamics, trajectory tracking, obstacle avoidance.

I. INTRODUCTION

Multi-legged robots are a hot research field in recent years since they have the advantages of higher stiffness, higher payload capacity and lower inertia to the manipulation than single legged ones at the price of a smaller workspace, more complex mechanism and inverse kinematics. Due to the properties, the motion control design of the robots are relatively complex, but their motions will be relatively flexible with a more complex terrain. The purpose of our study is to pursue multi-function, high performance, and simple control for the Multi-legged robots.

In the era of more frequently natural and manmade disasters, the robot applications which can overcome the complex terrain are particularly compelling, and it can be used in complex terrain for transportation and supply of materials or personnel rescue action. It can also be developed into small and fast multi-legged robots used in the security of surveillance, and its effectiveness is better than wheeled robots in complex terrain areas, and the hexapod robots are one of their applications that can achieve the basic functions with visual obstacle avoidance. Wilson [1] described the characteristics of movement patterns of American cockroaches, studying their gaits by using the methods of experiment and observation. Holk Cruse [2] developed artificial neural network (ANN) by studying the neural system of insect: Walknet, who successfully applied

to gait motion control of the hexapod robots. Visual-servo control is an approach that operates robots by using feedback image signals of visual system. Since the images are captured by cameras, single-eye camera has been used for different goals [3]–[5]. The main advantage of the single-eye cameras is the low cost and low computations, and can be used for real time image recognition [6]. There were numerous visual-servo controls used in robotic control. Takahama used mobile manipulator to grab a book on the table and place it into shelves. With the use of omnidirectional wheels, the upper platform was equipped with two manipulators and has CCD camera to acquire environmental informations. It divides the tasks into eight parts and then using sequential implementation method to operate two manipulators to perform the action of returning the book cooperatively. In [7], Hutchinson, Hager, and Corke proposed the classic of image feedback control, which has laid two orientations of image feedback control, i.e. position-based visual servo (PBVS) and image-based visual servo (IBVS). Under the framework of PBVS, it can estimate the coordinates of the target positions through the 3D visual system, and feedbacking them to the manipulator to perform the captured behavior. To properly capture the object, the estimation of target positions must be exact enough. IBVS framework does not require estimation of the target coordinates, but calculating error signals of the image characteristics at different time through 2D visual system, and finally feedbacking Image-Jacobian of IBVS to accomplish the conversion operation between images and manipulators. Since the image distortion problem will increase the error signals [7], the 2D visual servo system requires good image processing and tracking technologies to enhance the robustness of IBVS framework.

In this paper, we design a visual servo control for an autonomous hexapod robot with obstacle avoidance. It is an insect inspired robot with six legs that enable it to move flexibly on various terrains. The main advantage of this type of robot is its higher stiffness, higher payload capacity, and lower inertia. The control structure is composed of three parts: image recognition, visual servo control systems, and motion dynamics for trajectory tracking and obstacle avoidance. Unlike bipedal robots, insect inspired robots with six legs are statically stable and not dependent on balance mechanisms, but need feedback and positive reaction to acquire smoother walk. First, the depth between the obstacle and robot based on the proportion of the size of the area is constructed. Then, the image processing is used to identify whether there are any obstacles in the front, and make it as a feedback to the control system. For image recognition, we use OpenCV to process environment to the grayscale and binarization, filter noise

Wen-Shyong Yu and Chiau-Wei Huang are with the Department of Electrical Engineering, Tatung University, 40 Chung-Shan North Rd. 3rd. Sec., Taipei, Taiwan 10451 Taiwan, E-mail: wsyu@ctrl.ee.ttu.edu.tw.

through erosion and dilation, and then fill all of the contours using Sobel edge detection, and finally calculate the area and compare with each other. Finally, some experiments for a hexapod robot with obstacle avoidance are used to validate the performance of the proposed control scheme.

II. DESCRIPTION OF THE SYSTEM CONSTRUCTION AND HARDWARE

Consider the designed hexapod system shown in Fig. 1 in which the body is equipped with embedded systems, servo controllers, eighteen DC servos and one CCD camera. Fig. 2 is the schematic diagram of our control system flowchart.

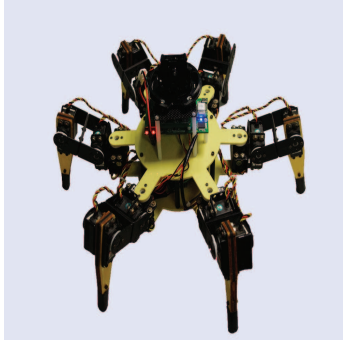


Fig. 1. The hexapod system.

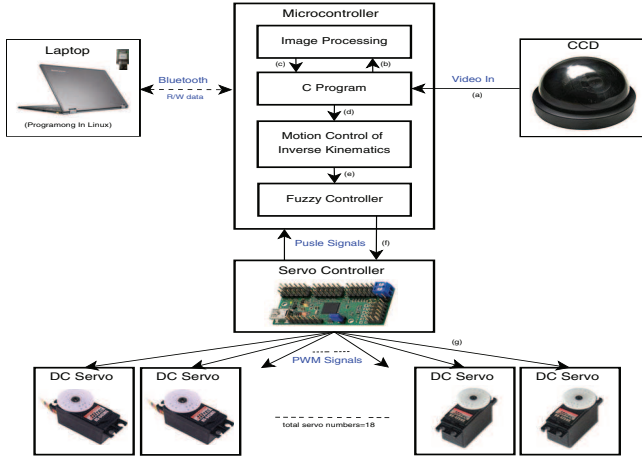


Fig. 2. Schematic diagram of the control system.

The specifications of the robot are shown in Table I and the configurations of the servos for each leg are shown in Figs. 3 and 4. HBE-KROBO is equipped with Marvell's PXA272 processor, where the PXA272 processor is a kind of XScale core with RISC (Reduced Instruction Set Computer) Embedded processors. It has the properties of the low-power environment, high-speed transmission, and it's often used in small and diversified application of the intelligent robot platform. We use this board to do the image processing and compute the inverse kinematics.

We use the UART serial communication as the communication way between microcontroller and servo controller, and

TABLE I
SPECIFICATIONS OF ROBOT.

	Hexapod's information
Size	39.5cm(L)×39.5cm(W)×30.5cm(H)
Weight	2 kg
Material	3mm ABS / 3mm PF / 1mm Carbon
DC servos	Hitec HS-475HB × 6 Hitec HS-645MG × 12
Camera	SHINEHAW 3.6mm CCD
Controller	HBE-KROBO-Marvall XScale PXA272 ARM Mini Maestro 24-Channel Servo Controller
Bluetooth 2.4GHz	Work range < 10 meters
Battery pack	7.4V Li-poly 900mAh × 2 11.1V Li-poly 1600mAh × 1

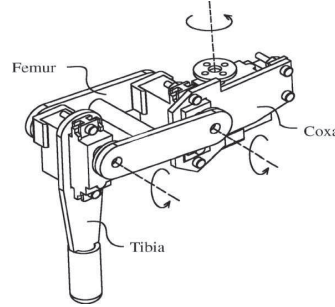


Fig. 3. Model of leg.

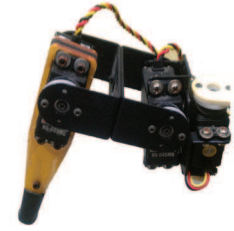


Fig. 4. Appearance of leg.

TABLE II
HARDWARE SPECIFICATIONS.

	Mini Maestro Polulu 24
Channels	24
Analog Input Channels	12
Digital Input Channels	12
Configurable Pulse Rate	1-333 Hz
Pulse Range	64-4080 μ s
Control Resolution	0.25 μ s Output Pulse Width
Script Size	8KB
Power	5V Battery or USB Mini Cable
PCB Size (L)×(W)×(H)	58.4×27.9×1.6 (mm)

the baud rate is 9600 bps. We want to accurately compute the movements with assembling plurality of servos, so we need to select a mode which can simultaneously transmit commands to the servo controller to avoid the delays between servos. The following is a kind of protocol we used, as shown in Fig. 5: mode, number of targets, first channel number, first target low bits, first target high bits, second target low bits, second target high bits, ... For example, if channel 1 is configured as a servo and one wants to set its target to 1500 μ s ($1500 \times 4 = 6000 = 0101110 1110000$ in binary), then one can send the following byte sequence:

- low bits: 01110000
- high bits: 00101110

The eighteen servos produced by the Hitec company are twelve HS-645MG and six HS-475HB, and each servo has three wires that stick out from the servo casing, one is for power, another is for ground, and the other is for signal input line. There are three servos for each limb connected by serial. Servo with reduction gears is controlled by pulses of variable

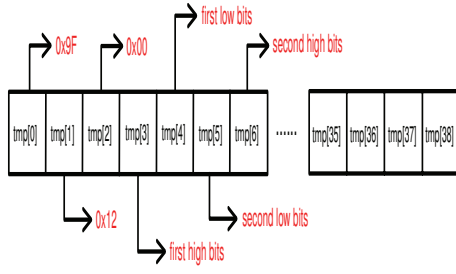


Fig. 5. Command protocols of our servo control.

width (pulse width modulation (PWM)) from the servo control, and feedback the position from potentiometer to test if it has reached the desired position.

By sending the image captured by the CCD camera with Pixels $640(H) \times 480(V)$ to the microcontroller to perform image recognition, the controller will generate the movement commands for the hexapod.

III. PROGRAMMING ENVIRONMENT

Fig. 6 is the schematic diagram of our programming. In programming, we use VMware player to simulate Linux operating system in Windows operating system, and write the servo control program in Fedora. Then, transmit the compiled file from Fedora to Windows, and use HyperTerminal to transfer it to the microcontroller by the bluetooth.

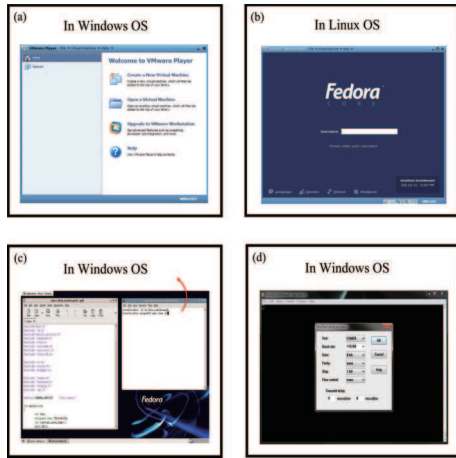


Fig. 6. Schematic diagram of the programming.

In recent years, in addition to ultrasonic, infrared sensors, and laser range finders, many studies use visual image process and recognition to handle environmental obstacle in the trajectory tracking, because the hardware cost of the imaging process devices such as CCD camera is cheaper than the others and can capture more detailed features of the environment. So, the digital image processing method can be applied to recognize and process by the microcontroller if there are any obstacles on the path.

There are six steps for the image processing: Capture images from camera, Pre-Processing, Image Binarization, Mathematical Morphology, Edge Detection, and Area Comparison after filling contours as shown in Fig. 7.

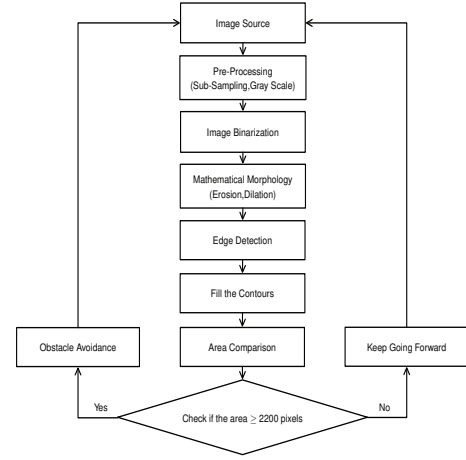


Fig. 7. Flowchart of image processing

Furthermore, representation of YUV color space is used in luminance and chrominance information, i.e. Y represents luminance, UV represents chrominance. According to the resolution and transmission of the image of YUV format, it can be divided into $YPbPr$ and $YCbCr$. $YPbPr$ is mainly used in HDTV TV, and $YCbCr$ is in SDTV TV. In HBE-KROBO, representation of each pixel is used for RGB mode. But the output format from the video decoder is $YCbCr$. Therefore, they must be able to convert into each other. The equations that transform $YCbCr$ into RGB are given by:

$$\begin{cases} R = 1.164Y - 16 + 1.596Cr - 128 \\ G = 1.164Y - 16 - 0.813Cr - 128r \\ B = 1.164Y - 16 + 2.017Cb - 128 \end{cases} \quad (1)$$

The equations for RGB to $YCbCr$ are:

$$\begin{cases} Y = 0.257R + 0.504G + 0.098B + 16 \\ Cb = -0.148R - 0.291G + 0.439B + 128 \end{cases} \quad (2)$$

The dynamic image processing flowchart of HBE-KROBO is shown in Fig. 8. First, the dynamic image data will be sent to the video decoder (SAA7113), then video processor(FPGA), finally CPU(PXA272) for computation.

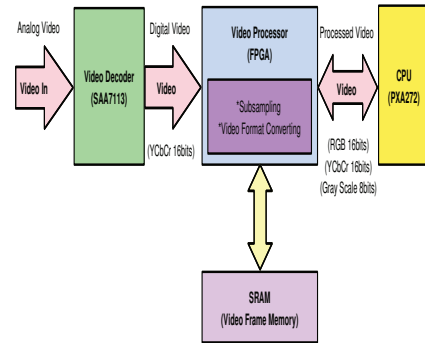


Fig. 8. Dynamic image processing flowchart of HBE-KROBO.

The Image Binarization uses threshold value to convert gray images value to a low and a high values. All of the pixels after the Image Binarization processing are divided into black and white colors, and the general equation $g(x, y)$ for the threshold

value is as follows:

$$g(x, y) = \begin{cases} 0, & \text{when } f(x, y) \geq T \\ 255, & \text{otherwise} \end{cases} \quad (3)$$

where T denotes the threshold value, $f(x, y)$ the original pixel value, and $g(x, y)$ the final pixel value.

The noise of Binary image is known as salt and pepper noise. The noise is distributed like sprinkling salt. By the characteristics of Binarization, we use methods of dilation and erosion to remove the noise.

Mathematical morphology uses the image component to define tool of regional patterns, such as contours and boundary. Therefore, the method of process to tackle the images through the shape is very effective and direct. For the purpose of image recognition, more informations such as complex background will result in slow recognition and low recognition rate. If we want to obtain the informations about the object in the image, such as shape, size, distribution, etc., determining from the edge is the most direct way. Therefore, the method of edge detection of the object for the image processing is an important issue since it can effectively and accurately detect the edge of the target information. The commonly used methods of edge detection in image processing are: Roberts, Prewitt, Sobel, Laplace, and Canny.

Edge detection is a first order derivative of the direction, i.e. the gradient variation in the x, y direction of the image $f(x, y)$.

$$\nabla f(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4)$$

We use Sobel edge detection to detect the edges of objects in which there's a mask operator used to find the gradient value of the image. We should first fill the contours after edge detection, then do area calculation in each pixel as a unit. Under the premise that frame background can not be too complicated, we can take the largest area as an obstacle.

IV. MODELING OF THE HEXAPOD ROBOTIC SYSTEM AND KINEMATICS DERIVATION

In this section, we should calculate the joint angles of the robot legs with respect to the positions of the robot, and then obtain the joint torques from the dynamic equations for the overall hexapod robot. It is seen from Fig. 9 that the hexapod robot has 3-dof with 6 limbs, one moving platform, 18 joints, and 18 links.

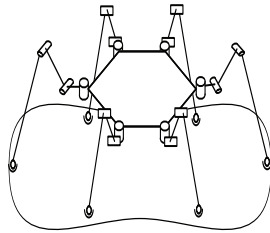


Fig. 9. The model of hexapod.

First, we derive the location of each limb in terms of the location of the moving platform. When the leg raise and the

end-effector is not on the ground, we will treat the leg as the serial manipulator with 3 links. Fig. 10 shows the link frame for each link in the leg. When all coordinate frames are defined, it is possible to derive the kinematic for each leg of the hexapod. As for limb i , the Denavit Hartengberg parameters for link 1 (coxa) are $\alpha_{i1} = -90^\circ$, $\ell_{i1} = 25mm$, q_{i1} , and $d_{i1} = 109mm$, for link 2 (femur) are $\alpha_{i2} = 0^\circ$, $\ell_{i2} = 57.5mm$, q_{i2} , and $d_{i2} = 0$, and for link 3 (tibia) are $\alpha_{i3} = 0^\circ$, $\ell_{i3} = 106.5mm$, q_{i3} , and $d_{i3} = 0$. Then, we can obtain the end-effector of the limb with respect to the joint angles as follows:

$$p_{ix} = \ell_3 c q_{i1} c q_{i23} + \ell_2 c q_{i1} c q_{i2} + \ell_1 c q_{i1} \quad (5)$$

$$p_{iy} = \ell_3 s q_{i1} c q_{i23} + \ell_2 s q_{i1} c q_{i2} + \ell_1 s q_{i1} \quad (6)$$

$$p_{iz} = -\ell_3 s q_{i23} - \ell_2 s q_{i2} + d_{i1} \quad (7)$$

where p_{ix} , p_{iy} , and p_{iz} are the end-effector parameters in the coordinate frame for limb i , and $c q_{ij} = \cos q_{ij}$, $s q_{ij} = \sin q_{ij}$, $j = 1, 2, 3$, $c q_{i23} = \cos(q_{i2} + q_{i3})$, $s q_{i23} = \sin(q_{i2} + q_{i3})$. From (5), (6), and (7), the inverse kinematics can be shown as follows:

$$q_{i1} = \arctan\left(\frac{p_{iy}}{p_{ix}}\right) \quad (8)$$

$$q_{i3} = \arccos\left(\frac{k_{i1}^2 + k_{i2}^2 - \ell_2^2 - \ell_3^2}{2\ell_2\ell_3}\right) \quad (9)$$

$$q_{i2} = \arctan\left(\frac{-k_{i1}\ell_3 s q_{i3} + k_{i2}(\ell_2 + \ell_3 c q_{i3})}{k_{i1}(\ell_2 + \ell_3 c q_{i3}) + k_{i2}\ell_3 s q_{i3}}\right) \quad (10)$$

where $k_{i1} = p_{ix} c q_{i1} + p_{iy} s q_{i1} - \ell_1$ and $k_{i2} = -p_{iz} + d_{i1}$. Next, the equations between the end-effector of each limb and the centroid of the moving frame can be obtained as

$$\begin{bmatrix} D_{x1} \\ D_{y1} \\ D_{z1} \end{bmatrix} = \begin{bmatrix} c\phi & s\phi & 0 \\ -s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x + \ell_p \\ P_y \\ P_z \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} D_{x2} \\ D_{y2} \\ D_{z2} \end{bmatrix} = \begin{bmatrix} c(\phi + 60^\circ) & -s(\phi + 60^\circ) & 0 \\ s(\phi + 60^\circ) & c(\phi + 60^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x + \ell_p \\ P_y \\ P_z \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} D_{x3} \\ D_{y3} \\ D_{z3} \end{bmatrix} = \begin{bmatrix} c(\phi + 120^\circ) & -s(\phi + 120^\circ) & 0 \\ s(\phi + 120^\circ) & c(\phi + 120^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x + \ell_p \\ P_y \\ P_z \end{bmatrix} \quad (13)$$

$$\begin{bmatrix} D_{x4} \\ D_{y4} \\ D_{z4} \end{bmatrix} = \begin{bmatrix} c(\phi + 180^\circ) & -s(\phi + 180^\circ) & 0 \\ s(\phi + 180^\circ) & c(\phi + 180^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x + \ell_p \\ P_y \\ P_z \end{bmatrix} \quad (14)$$

$$\begin{bmatrix} D_{x5} \\ D_{y5} \\ D_{z5} \end{bmatrix} = \begin{bmatrix} c(\phi - 120^\circ) & -s(\phi - 120^\circ) & 0 \\ s(\phi - 120^\circ) & c(\phi - 120^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x + \ell_p \\ P_y \\ P_z \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} D_{x6} \\ D_{y6} \\ D_{z6} \end{bmatrix} = \begin{bmatrix} c(\phi - 60^\circ) & -s(\phi - 60^\circ) & 0 \\ s(\phi - 60^\circ) & c(\phi - 60^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x + \ell_p \\ P_y \\ P_z \end{bmatrix} \quad (16)$$

where $[D_{xi} \ D_{yi} \ D_{zi}]^T$ is the new end-effector coordinate, $[P_x \ P_y \ P_z]^T$ is the original end-effector coordinate, and ℓ_p is the length between the centroid of the moving frame and coxa.

V. VELOCITY AND JACOBIAN ANALYSIS

Next, we compute the linear and angular velocities of each limb in terms of the velocity and angular velocity of the moving platform by Conventional Jacobian. To simplify the analysis, the origin of the fixed coordinate frame is located at the point of the joint coxa. Referring to Fig. 10 for the hexapod robotic system, the input vector is $\dot{\mathbf{q}} = [\dot{q}_{11} \ \dot{q}_{12} \ \dot{q}_{13} \ \cdots \ \dot{q}_{61} \ \dot{q}_{62} \ \dot{q}_{63}]_{1 \times 18}^T$, and the output vector can

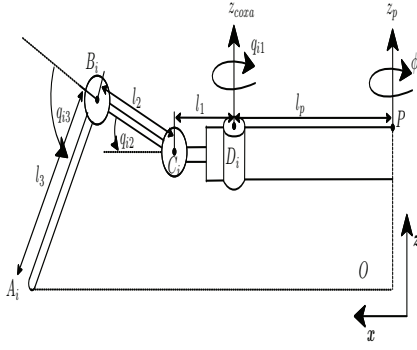


Fig. 10. Schematic diagram of the i th limb.

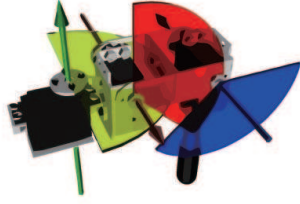


Fig. 11. The workspace of each leg.

be described by the velocity of the centroid P and the angular velocity of moving platform:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v}_p \\ \boldsymbol{\omega}_B \end{bmatrix} \quad (17)$$

A loop-closure equation for the i th limb can be written as

$$\overline{OP} + \overline{PD}_i = \overline{OA}_i + \overline{A_iB_i} + \overline{B_iC_i} + \overline{C_iD_i} \quad (18)$$

Differentiating equation (18) with respects to time yields

$$\mathbf{v}_p + \boldsymbol{\omega}_B \times \mathbf{d}_i = \boldsymbol{\omega}_{i3} \times \mathbf{a}_i + \boldsymbol{\omega}_{i2} \times \mathbf{b}_i + \boldsymbol{\omega}_{i1} \times \mathbf{c}_i \quad (19)$$

where $\mathbf{a}_i = \overline{A_iB_i}$, $\mathbf{b}_i = \overline{B_iC_i}$, $\mathbf{c}_i = \overline{C_iD_i}$, $\mathbf{d}_i = \overline{PD_i}$, and $\boldsymbol{\omega}_{ij}$ is the angular velocity of the j link of the i th limb. Expressing equation (19) in terms of the (x_i, y_i, z_i) coordinate frame, we have

$$\begin{aligned} {}^i\mathbf{a}_i &= \ell_3 \begin{bmatrix} -cq_{i1}c(-q_{i2} + q_{i3}) \\ sq_{i1}c(-q_{i2} + q_{i3}) \\ s(-q_{i2} + q_{i3}) \end{bmatrix}, \quad {}^i\mathbf{b}_i = \ell_2 \begin{bmatrix} -cq_{i1}cq_{i2} \\ sq_{i1}cq_{i2} \\ sq_{i2} \end{bmatrix}, \\ {}^i\mathbf{c}_i &= \ell_1 \begin{bmatrix} -cq_{i1} \\ sq_{i1} \\ 0 \end{bmatrix}, \quad {}^i\mathbf{d}_i = \ell_p \begin{bmatrix} -c\phi_i \\ s\phi_i \\ 0 \end{bmatrix}, \quad {}^i\boldsymbol{\omega}_{i1} = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_{i1} \end{bmatrix}, \quad {}^i\boldsymbol{\omega}_{i2} = \begin{bmatrix} 0 \\ -\dot{q}_{i2} \\ 0 \end{bmatrix}, \\ {}^i\boldsymbol{\omega}_{i3} &= \begin{bmatrix} 0 \\ -\dot{q}_{i3} \\ 0 \end{bmatrix}, \quad {}^i\mathbf{v}_p = \begin{bmatrix} v_{p,x}c\phi_i + v_{p,y}s\phi_i \\ -v_{p,x}s\phi_i + v_{p,y}c\phi_i \\ v_{p,z} \end{bmatrix} \end{aligned}$$

Substituting the above expressions into equation (19), we obtain the Jacobian of the hexapod robotic system as

$$\mathbf{J} = [\mathbf{J}_1 \ \mathbf{J}_2 \ \mathbf{J}_3] \quad (20)$$

where

$$\mathbf{J}_1 = \begin{bmatrix} (\ell_1 + \ell_2sq_{i2})cq_{i1} \\ (\ell_1 + \ell_2sq_{i2})sq_{i1} \\ \ell_1 + \ell_2sq_{i2} - \ell_p c\phi_i sq_{i1} + \ell_p s\phi_i cq_{i1} \\ -cq_{i1} \\ -sq_{i1} \\ 0 \end{bmatrix} \quad (21)$$

$$\mathbf{J}_2 = \begin{bmatrix} 0 \\ 0 \\ \ell_p s\phi_i sq_{i1} + \ell_p c\phi_i cq_{i1} \\ -cq_{i1} \\ -sq_{i1} \\ -1 \end{bmatrix}, \quad \mathbf{J}_3 = \begin{bmatrix} -\ell_p c\phi_i \\ -\ell_p s\phi_i \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \quad (22)$$

VI. DYNAMICS OF THE HEXAPOD SYSTEM

In this section, the dynamics of the hexapod robotic system is considered. We assume that every link of the robotic system is a rigid body. Consider the dynamic equation of each limb. Let D_i be the fixed point of a serial manipulator of the hexapod robotic system when one of limb's end effector was not on the ground. Assuming that all link of each limb are homogeneous with relatively small cross section, the position vector of the center of mass for limb i are given by

$${}^1\mathbf{p}_{ci1} = \left[-\frac{\ell_1}{2}, -\frac{d_i}{2}, 0\right]^\top, \quad {}^2\mathbf{p}_{ci2} = \left[-\frac{\ell_2}{2}, 0, 0\right]^\top, \quad {}^3\mathbf{p}_{ci3} = \left[-\frac{\ell_3}{2}, 0, 0\right]^\top \quad (23)$$

The link inertia matrices about their center of mass and expressed in their respective link frames as

$$\begin{aligned} {}^1I_{i1} &= \frac{1}{12}m_1\ell_1^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad {}^2I_{i2} = \frac{1}{12}m_2\ell_2^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ {}^3I_{i3} &= \frac{1}{12}m_3\ell_3^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (24)$$

The link inertia matrices about their center of mass expressed in the base frame can be obtained by substituting equations (24) along their rotation matrices, and they are

$$\begin{aligned} I_{i1} &= \frac{1}{12}m_1\ell_1^2 \begin{bmatrix} -s^2q_{i1} & -sq_{i1}cq_{i1} & 0 \\ -sq_{i1}cq_{i1} & c^2q_{i1} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ I_{i2} &= \frac{1}{12}m_2\ell_2^2 \begin{bmatrix} c^2q_{i1} & 2cq_{i1}sq_{i1}cq_{i2}sq_{i2} & 0 \\ 2cq_{i1}sq_{i1}cq_{i2}sq_{i2} & s^2q_{i1} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ I_{i3} &= \frac{1}{12}m_3\ell_3^2 \begin{bmatrix} c^2q_{i1} & sq_{i1}cq_{i1} & 0 \\ sq_{i1}cq_{i1} & s^2q_{i1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (25)$$

The position vectors of the center of mass of link, $i = 1, 2, 3$, with respect to the various link frames expressed in the base frame are

$$\begin{aligned} {}^0\mathbf{p}_{ci1} &= \begin{bmatrix} \frac{1}{2}\ell_1cq_{i1} \\ \frac{1}{2}\ell_1sq_{i1} \\ \frac{1}{2}d_{i1} \end{bmatrix}, \quad {}^0\mathbf{p}_{ci2} = \begin{bmatrix} \ell_1cq_{i1} + \frac{1}{2}\ell_2cq_{i1}cq_{i2} \\ \ell_1sq_{i1} + \frac{1}{2}\ell_2sq_{i1}cq_{i2} \\ \frac{1}{2}\ell_2sq_{i2} + d_{i1} \end{bmatrix}, \\ {}^0\mathbf{p}_{ci3} &= \begin{bmatrix} \ell_1cq_{i1} + \ell_2cq_{i1}cq_{i2} + \frac{1}{2}\ell_3cq_{i1}cq_{i23} \\ \ell_1sq_{i1} + \ell_2sq_{i1}cq_{i2} + \frac{1}{2}\ell_3sq_{i1}cq_{i23} \\ d_{i1} + \ell_2sq_{i2} + \frac{1}{2}\ell_3sq_{i23} \end{bmatrix} \end{aligned} \quad (26)$$

By substituting the above equations into

$$\mathbf{J}_{vi} = [\mathbf{J}_{vi}^1, \mathbf{J}_{vi}^2, \dots, \mathbf{J}_{vi}^i, 0, 0, \dots, 0] \quad (27)$$

$$\mathbf{J}_{\omega i} = [\mathbf{J}_{\omega i}^1, \mathbf{J}_{\omega i}^2, \dots, \mathbf{J}_{\omega i}^i, 0, 0, \dots, 0] \quad (28)$$

the link Jacobian submatrices, \mathbf{J}_{vi} and \mathbf{J}_{wi} , can be obtained as follows:

$$\mathbf{J}_{v1} = \begin{bmatrix} -\frac{1}{2}\ell_1 s q_{i1} & 0 & 0 \\ \frac{1}{2}\ell_1 c q_{i1} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{J}_{\omega 1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (29)$$

$$\mathbf{J}_{v2} = \begin{bmatrix} -\ell_1 s q_{i1} - \frac{1}{2}\ell_2 s q_{i1} c q_{i2} & -\frac{1}{2}\ell_2 c q_{i1} s q_{i2} & 0 \\ \ell_1 c q_{i1} + \frac{1}{2}\ell_2 c q_{i1} c q_{i2} & -\frac{1}{2}\ell_2 s q_{i1} s q_{i2} & 0 \\ 0 & \frac{1}{2}\ell_2 c q_{i2} & 0 \end{bmatrix} \quad (30)$$

$$\mathbf{J}_{\omega 2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \mathbf{J}_{\omega 3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad (31)$$

$$\mathbf{J}_{v3} = \begin{bmatrix} (1, 1) & -\ell_2 c q_{i1} s q_{i2} - \frac{1}{2}\ell_3 c q_{i1} s q_{i23} & -\frac{1}{2}\ell_3 c q_{i1} s q_{i23} \\ (2, 1) & -\ell_2 s q_{i1} s q_{i2} - \frac{1}{2}\ell_3 s q_{i1} s q_{i23} & -\frac{1}{2}\ell_3 s q_{i1} s q_{i23} \\ 0 & \ell_2 c q_{i2} + \frac{1}{2}\ell_3 c q_{i23} & \frac{1}{2}\ell_3 c q_{i23} \end{bmatrix} \quad (32)$$

where $(1, 1) = -\ell_1 s q_{i1} - \ell_2 s q_{i1} c q_{i2} - \frac{1}{2}\ell_3 s q_{i1} c q_{i23}$ and $(2, 1) = \ell_1 c q_{i1} + \ell_2 c q_{i1} c q_{i2} + \frac{1}{2}\ell_3 c q_{i1} c q_{i23}$. From equations (25)–(32), the kinetic energy and potential energy equations of the system can be obtained as

$$\mathcal{K} = \frac{1}{2} \dot{\mathbf{q}}^\top \left[\sum_{i=1}^3 \left(\mathbf{J}_{vi}^\top m_i \mathbf{J}_{vi} + \mathbf{J}_{wi}^\top I_i \mathbf{J}_{wi} \right) \right] \dot{\mathbf{q}} \quad (33)$$

and

$$\mathcal{U} = - \sum_{i=1}^3 \mathcal{U}_i = - \left(m_1 \mathbf{g}^\top \mathbf{p}_{c1} + m_2 \mathbf{g}^\top \mathbf{p}_{c2} + m_3 \mathbf{g}^\top \mathbf{p}_{c3} \right) \quad (34)$$

respectively. By substituting equations (23)–(32) into (33) and (34), and acquiring the Lagrange's equation of motion, we can obtain the Lagrangian as

$$\begin{aligned} & \left(\frac{1}{3} \ell_1^2 m_1 + m_2 \ell_2^2 \left(\frac{1}{12} + \frac{1}{4} c^2 q_{i2} \right) + m_2 \ell_1 \ell_2 c q_{i2} \right. \\ & + m_3 \ell_1^2 + m_3 \ell_2^2 c^2 q_{i2} + m_3 \ell_3^2 \left(\frac{1}{12} + \frac{1}{4} c^2 q_{i23} \right) + 2 m_3 \ell_1 \ell_2 c q_{i2} \\ & + m_3 \ell_1 \ell_3 c q_{i23} + m_3 \ell_2 \ell_3 c q_{i2} c q_{i23} \left. \right] \ddot{q}_1 + \left(-\frac{1}{2} m_2 \ell_2^2 c q_{i2} s q_{i2} \right. \\ & - m_2 \ell_1 \ell_2 s q_{i2} - \frac{1}{2} m_2 \ell_1 \ell_2 c (2 q_{i1}) s q_{i2} - \frac{1}{2} m_2 \ell_2^2 c (2 q_{i1}) s q_{i2} c q_{i2} \\ & - 2 m_3 \ell_2^2 c q_{i2} s q_{i2} - \frac{1}{2} m_3 \ell_3^2 s q_{i23} c q_{i23} - 2 m_3 \ell_1 \ell_2 s q_{i2} \\ & - m_3 \ell_1 \ell_3 s q_{i23} - m_3 \ell_2 \ell_3 s (q_{i2} + q_{i23}) \left. \right) \dot{q}_1 \dot{q}_2 - \frac{1}{12} m_3 \ell_3^2 s q_{i1} c q_{i1} \dot{q}_2^2 \\ & + \left(-\frac{1}{2} m_3 \ell_3^2 s q_{i23} c q_{i23} - m_3 \ell_1 \ell_3 s q_{i23} - m_3 \ell_2 \ell_3 c q_{i2} s q_{i23} \right) \dot{q}_1 \dot{q}_3 \\ & + \left(m_3 \ell_3 s (2 q_{i1}) \right] \dot{q}_2 \dot{q}_3 + \left(-\frac{1}{12} m_3 \ell_3^2 s q_{i1} c q_{i1} \right) \dot{q}_3^2 = \tau_{i1} \quad (35) \end{aligned}$$

$$\begin{aligned} & \left(m_2 \ell_1 \ell_2 s q_{i1} c q_{i1} s q_{i2} + \frac{1}{2} m_2 \ell_2^2 s q_{i1} c q_{i1} s q_{i2} c q_{i2} \right) \ddot{q}_1 + \left(\frac{1}{3} m_2 \ell_2^2 \right. \\ & + m_3 \ell_2^2 + m_3 \ell_3^2 \left(\frac{1}{4} + \frac{1}{12} s^2 q_{i1} \right) + m_3 \ell_2 \ell_3 c q_{i3} \left. \right) \ddot{q}_2 + \frac{1}{12} m_3 \ell_3^2 s^2 q_{i1} \ddot{q}_3 \\ & + \left(m_2 \ell_1 \ell_2 c (2 q_{i1}) s q_{i2} - \frac{1}{2} m_2 \ell_2^2 c (2 q_{i1}) s q_{i2} c q_{i2} - \frac{1}{4} m_2 \ell_2^2 c q_{i2} s q_{i2} \right. \\ & - \frac{1}{2} m_2 \ell_1 \ell_2 s q_{i2} - m_3 \ell_2^2 c q_{i2} s q_{i2} - \frac{1}{4} m_3 \ell_3^2 c q_{i23} s q_{i23} - m_3 \ell_1 \ell_2 s q_{i2} \\ & - \frac{1}{2} m_3 \ell_1 \ell_2 s q_{i23} - \frac{1}{2} m_3 \ell_2 \ell_3 s (q_{i2} + q_{i23}) \left. \right) \dot{q}_1^2 + \left(\frac{1}{2} m_2 \ell_1 \ell_2 s q_{i1} c q_{i1} c q_{i2} \right. \\ & + \frac{1}{4} m_2 \ell_2^2 s q_{i1} c q_{i1} c (2 q_{i2}) + \frac{1}{12} m_3 \ell_3^2 s q_{i1} c q_{i1} \left. \right) \dot{q}_1 \dot{q}_2 \\ & + \left(\frac{1}{6} m_3 \ell_3^2 s q_{i1} c q_{i1} \right) \dot{q}_1 \dot{q}_3 + \left(-m_3 \ell_2 \ell_3 s q_{i3} \right) \dot{q}_2 \dot{q}_3 + \frac{1}{2} m_2 \ell_2 c q_{i2} g_c \\ & + m_3 \ell_2 c q_{i2} g_c + \frac{1}{2} m_3 \ell_3 c q_{i23} g_c = \tau_{i2} \end{aligned}$$

$$\begin{aligned} & \left(\frac{1}{2} m_3 \ell_2 \ell_3 c q_{i3} + \ell_3^2 \left(\frac{1}{4} + \frac{1}{12} s^2 q_{i1} \right) \right) \ddot{q}_2 + \frac{1}{2} m_3 \ell_2 \ell_3 s q_{i3} \ddot{q}_2 \\ & + \left(\frac{1}{6} m_3 \ell_3^2 s q_{i1} c q_{i1} \right) \dot{q}_1 \dot{q}_2 + \left(\frac{1}{6} m_3 \ell_3^2 s q_{i1} c q_{i1} \right) \dot{q}_1 \dot{q}_3 \\ & + \left(\frac{1}{4} m_3 \ell_2 \ell_3 s q_{i3} \right) \dot{q}_2 \dot{q}_3 + \frac{1}{2} m_3 \ell_3 c q_{i23} g_c = \tau_{i3} \quad (37) \end{aligned}$$

Finally, calculating the integral dynamic equations of the hexapod robotic system and by considering the gait generations, we can simply calculate that for the three legs with the platform.

In order to design the visual servo control for $\tau = [\tau_{11} \ \tau_{12} \ \tau_{13} \ \cdots \ \tau_{61} \ \tau_{62} \ \tau_{63}]^\top$, an adaptive PID-type control law is given by

$$\tau = k_P \mathbf{e}(t) + k_I \int \mathbf{e}(t) dt + k_D \dot{\mathbf{e}}(t) \quad (38)$$

where the tracking error vector $\mathbf{e} = \mathbf{q} - \mathbf{q}_r = [e_{11} \ e_{12} \ e_{13} \ \cdots \ e_{61} \ e_{62} \ e_{63}]^\top \in \mathbb{R}^{18}$, where $\mathbf{q}_r = [q_{r11} \ q_{r12} \ q_{r13} \ \cdots \ q_{r61} \ q_{r62} \ q_{r63}]^\top \in \mathbb{R}^{18}$ is the desired input, k_P , k_I , and k_D are the proportional, integral, and derivative gains of the controller, respectively. From (38), the PID-type controller can be rewritten in a matrix as

$$\tau(\boldsymbol{\xi}|\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{\xi}(e) \quad (39)$$

where $\boldsymbol{\theta} = [k_P \ k_I \ k_D]^\top$ and $\boldsymbol{\xi} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3]^\top = [\mathbf{e} \ \mathbf{e}_I \ \mathbf{e}_D]^\top$ for which $\mathbf{e}_I = \int \mathbf{e}(t) dt$ and $\mathbf{e}_D = \dot{\mathbf{e}}(t)$. The PID-type controller (38) or (39) can uniformly approximate the controller τ^* . Hence, there exists an optimal gain vector $\boldsymbol{\theta}^*$ such that the approximation error between τ and τ^* can be bounded by a prescribed constant δ , i.e., $\tau^* = \tau_{PID}(\boldsymbol{\xi}|\boldsymbol{\theta}^*) + \delta(\boldsymbol{\xi})$ where $|\delta(\boldsymbol{\xi})| \leq \delta, \forall t$.

The fuzzy PID system is characterized by fuzzy **IF-THEN** rules and a fuzzy inference engine. The fuzzy inference engine uses the fuzzy **IF-THEN** rules to perform a mapping from an input linguistic vector $\boldsymbol{\xi}$ to an output linguistic variable $z(\boldsymbol{\xi}) \in \mathbb{R}$. Since there is no mature guidance in fuzzy set theory for the determination of the best shapes for fuzzy sets, it is suggested that different shapes for different set points need to be studied to obtain an optimum solution for various ranges of the system states. In addition, the choice of equal-width intervals entails no loss of generality, particularly in applications. The sinusoidal membership functions with equal-width intervals of the means are thus proposed to eliminate the sharp boundary and defined as

$$\mu_{i1}(e_{ij}) = \begin{cases} 0 & \text{if } e_{ij} < -\epsilon_j \\ |\cos(\frac{\pi}{2\epsilon_j} e_{ij})|, & \text{if } -\epsilon_j \leq e_{ij} \leq \epsilon_j \\ 0 & \text{if } e_{ij} > \epsilon_j \end{cases} \quad (40)$$

$$\mu_{i2}(e_{ij}) = \begin{cases} 1 & \text{if } e_{ij} < -\epsilon_j \\ |\cos(\frac{\pi}{2\epsilon_j} e_{ij} + \frac{\pi}{2})|, & \text{if } -\epsilon_j \leq e_{ij} \leq \epsilon_j \\ 1 & \text{if } e_{ij} > \epsilon_j \end{cases} \quad (41)$$

$$\mu_{i3}(e_{ij}) = \begin{cases} 1 & \text{if } e_{ij} < -\epsilon_j \\ |\cos(\frac{\pi}{2\epsilon_j} e_{ij} + \pi)|, & \text{if } -\epsilon_j \leq e_{ij} \leq \epsilon_j \\ 1 & \text{if } e_{ij} > \epsilon_j \end{cases} \quad (42)$$

for $j = 1, 2, 3$, where ϵ_j is variable of the membership functions corresponding to the fuzzy PID controller's inputs e_{ij} . Since a fuzzy system can always be approximated by a group of multi-input single-output (MISO) fuzzy systems, we assume that the fuzzy systems are MISO systems consists of $N = \prod_{j=1}^3 N_j$ rules in the following form:

$$\begin{aligned} & R_{i_1 i_2 i_3}: \text{IF } e_{i1} \text{ is } G_{i_1}^1 \text{ AND } e_{i2} \text{ is } G_{i_2}^2 \text{ AND } e_{i3} \text{ is } G_{i_3}^3 \\ & \text{THEN } z_{ij}(\boldsymbol{\xi}) \text{ is } C_{i_1 i_2 i_3}, \\ & i_1 = 1, \dots, N_1, i_2 = 1, \dots, N_2, i_3 = 1, \dots, N_3 \end{aligned}$$

where e_{ij} , $j = 1, 2, 3$, and $z_{ij}(\boldsymbol{\xi})$ denote the linguistic variables associated with the inputs and output of the fuzzy system, respectively, $G_{i_j}^j$ and $C_{i_1 i_2 i_3}$ are linguistic values of linguistic variables $\boldsymbol{\xi}$ and $z_{ij}(\boldsymbol{\xi})$ in the universes of discourse $U \in \mathbb{R}^3$ and \mathbb{R} , respectively. Let $U = U_1 \times U_2 \times U_3$, where $U_i \in \mathbb{R}$, $i = 1, 2, 3$. The defuzzifier (36) maps a fuzzy set in V to a crisp point in V . By using the singleton

fuzzifier, product inference engine and the center average defuzzifier, the inferred outputs of fuzzy system are

$$z_{ij}(\xi) = \frac{\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \sum_{i_3=1}^{N_3} \frac{\prod_{k=1}^3 \mu_{ij}(e_{ij})}{\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \sum_{i_3=1}^{N_3} \prod_{k=1}^3 \mu_{ij}(e_{ij})} c_{i_1 i_2 i_3}$$

$$= \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \sum_{i_3=1}^{N_3} a_{i_1 i_2 i_3}(\xi) c_{i_1 i_2 i_3} \quad (43)$$

where $c_{i_1 i_2 i_3}$'s are the center of the i_j th fuzzy set and are the points in V at which $c_{i_1 i_2 i_3}$ achieves their maximum values or equal to 1, and the following nonlinear mapping

$$a_{i_1 i_2 i_3}(\xi) = \frac{\prod_{j=1}^3 \mu_{ij}(e_{ij})}{\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \sum_{i_3=1}^{N_3} \prod_{k=1}^3 \mu_{ij}(e_{ij})} \quad (44)$$

Because $\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \sum_{i_3=1}^{N_3} a_{i_1 i_2 i_3}(\xi) = 1$, $a_{i_1 i_2 i_3}(\xi)$ can be viewed as a weighting function. Hence, these fuzzy inferences can be described in the form of a linear equivalent neural network:

$$\theta = \Xi c \quad (45)$$

where $c_{i_1 i_2 i_3}$'s are free (adjustable) parameters and $\Xi = \text{diag}\{a^T, a^T, a^T\}$ with dimension $3 \times (3N)$ for which $a = [a_{111} \ a_{112} \ \dots \ a_{11N_3} \ \dots \ a_{N_1 N_2 N_3}]^T$ is an $N \times 1$ fuzzy basis function vector and $a_{i_1 i_2 i_3}$'s are defined in (44), and $c = [c_1 \ c_2 \ c_3]^T$ for which $c_\ell = [c_{111\ell} \ c_{112\ell} \ \dots \ c_{11N_3\ell} \ \dots \ c_{N_1 N_2 1\ell} \ \dots \ c_{N_1 N_2 N_3\ell}]^T$, $\ell = 1, 2, 3$

Hence, the fuzzy system is used to approximate the controller τ^* in (38) by using the update laws derived to tune the adjustable parameter vector θ . Because the unknown parameters of the controlled plant represented as the parameter vector θ are absorbed partly into the fuzzy system, θ can be obtained more accurately by further estimating the unknown but constant weight vector c according the tracking error and the coefficients of the fuzzy system. Let $\theta = \Xi \hat{c}$ be the estimate of θ^* due to \hat{c} and $\tilde{c} = \hat{c} - c^*$ the error vector. Then, the certainty equivalent PID controller of (38) can be re-defined as

$$\tau_{PID}(\xi|\theta) = \xi^T \Xi \tilde{c} \quad (46)$$

VII. THE GAIT GENERATION OF HEXAPOD ROBOTIC SYSTEM

In this section, the generation of the robot gait will be described. To provide the fastest movement speed, the robot is chosen to focus on a tripod gait. The gait to be generated is based on the tripod gait in two states. State one is when legs 1, 3, and 5 are lifting while legs 2, 4, and 6 are set as the supporting whole platform. State two is the reverse where legs 2, 4, and 6 are lifting and legs 1, 3, and 5 are supporting the whole platform. The movement of the robot happens when the center of gravity is moved from one stable state to another, as shown in Fig. 12. The gait cycles are illustrated in Fig. 13.

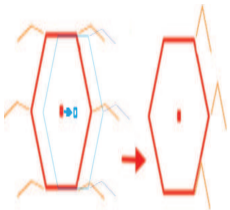


Fig. 12. The movement of the center of gravity.

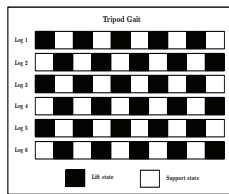


Fig. 13. The schematic diagram of tripod gait.

VIII. EXPERIMENTAL RESULTS

The control system architecture is shown in Fig. 2. There are three parts needed to be considered during the experiments:

- As for the robotic hardware, consider the mechanical limitations, such as limited joint angles, weight, and location configuration of the powers and control boards.
- As for the motion programming, drive the 18 servos for all links simultaneously and obtain inverse kinematics without any delays.
- As for the visual programming, capture source images and do some preliminary processing, and select the desired algorithms of image recognition.

The algorithm of our motion control is shown in Fig. 14. First, we do initial settings of the robot, such as the size of platforms, the length of limbs, and the position of end points. Second, we send the control commands: offsets of any direction, such as how far we want the robot to move and how high we want it to rise. Then, according to a given gait sequence, all angles of the robot joints respectively are calculated by the inverse kinematics for the left-three legs and the right-three legs. Finally, the calculated angles will be sent to servo controller to implement the movements before checking if it's satisfied by the conditions.

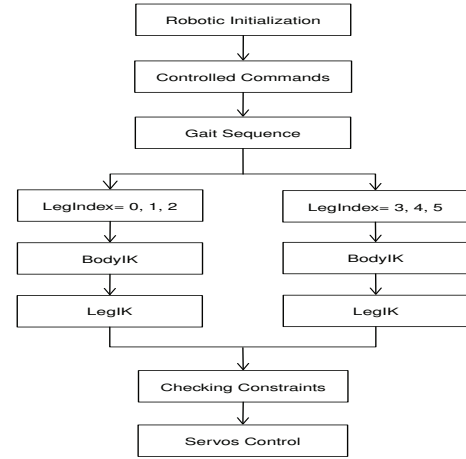


Fig. 14. The flowchart of motion control.

In these experiments, the robot goes forward with tripod gait by visual obstacle avoidance using the algorithm shown in Fig. 17. First, we capture the images, do some preliminary processing, and determine the size of the area of the contours. Then, the largest size of the contour can be seen as the obstacle. To reduce the computational burden, the image is sampled every three seconds. If there is an obstacle in front of the robot, the hexapod robot would do obstacle avoidance; otherwise, the hexapod robot would keep going forward along the default trajectory. Figs. 18 and 19 show the trajectories for obstacle avoidance and the pictures of the robot doing obstacle avoidance, respectively. Fig. 15 is the responses of the obstacle avoidance path tracking. The red line is default path, and the blue one is actual path. Fig. 16 shows the trajectory of the center of mass position error of the motion. From above, we can observe that the maximum error is in turning state, and the position errors of every 3 seconds are always below 10 mm.

IX. CONCLUSIONS

In this paper, we have accomplished visual servo control design for a hexapod robot with obstacle avoidance. We make use of the visual recognition system as feedback to achieve the capabilities of obstacle avoidance and implement the motion control of the hexapod robot by using the inverse kinematics. Except the visual servo system, no other sensors are used. The control structure is composed of three parts: image recognition, visual servo control systems, and inverse kinematics for trajectory tracking and obstacle avoidance. First, the

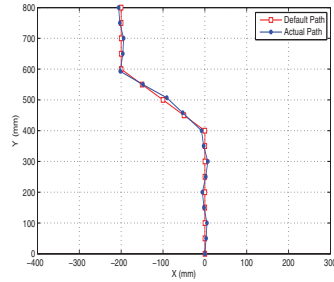


Fig. 15. The responses of obstacle avoidance path tracking.

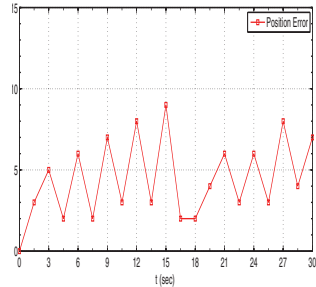


Fig. 16. Trajectory of the center of mass position error.

depth between the obstacle and robot based on the proportion of the size of the area is constructed. Then, the image processing is used to identify whether there are any obstacles in the front, and make it as a feedback to the servo control system. For image recognition, we use OpenCV to process environment to the grayscale and binarization, filter noise through erosion and dilation, and then fill all of the contours using Sobel edge detection, and finally calculate the area and compare that with each other. Finally, some experiments for a hexapod robot with obstacle avoidance are used to validate the performance of the proposed control scheme.

ACKNOWLEDGMENT

Financial support of this research by National Science Council, Taiwan, under the grant NSC 101-2632-E-036 -001 -MY3 is gratefully acknowledged.

REFERENCES

- [1] Wilson, D.M. , "Insect walking," Annual Rev. of Entomology, vol 11, pp. 103-122, 1996.
- [2] Cruse, H., Kindermann, Th., Schumm, M., Dean, J., Schmitz, J. "Walknet- a biologically inspired network to control six-legged walking," Neural Networks. vol 11, pp. 1435-1447, 1998.
- [3] C.-C. Hsu, M.-C. Lu, W.-Y. Wang, and Y.-Y. Lu, "Three-dimensional measurement of distant objects based on laser-projected CCD images," IET Sci. Meas. Technol., vol. 3, Iss. 3, pp. 197-207, 2009.
- [4] B. Williams, P. Smith, and I. Reid, "Automatic Relocalisation for a Single-Camera Simultaneous Localisation and Mapping System," IEEE International Conference on Robotics and Automation, Roma, Italy, pp. 2784-2790, 2007.
- [5] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, "A Lidar and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking," Proceedings of IEEE on Intelligent Transportation Systems Conference, Seattle, USA, pp. 1044-1049, 2007.
- [6] K. Tsiakmakis, B.P. Jordi, P.-V. Manel, and T. Laopoulos, "A Camera Based Method for the Measurement of Motion Parameters of IPMC Actuators," IEEE Trans. on Instrumentation and Measurement, vol. 58, no. 8, pp. 2626-2633, 2009.
- [7] S. Hutchinson, G. D. Hager and P. I. Corke, "A Tutorial on Visual Servo Control," IEEE Trans. on Robotics and Automation, vol. 12, pp. 651-670, 1996.
- [8] L. W. Tasi, "Robot Analysis: The Mechanics of Serial and Parallel Manipulator." New York: Wiley, 2005.

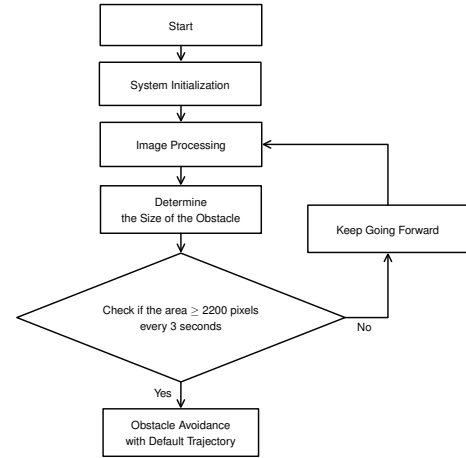


Fig. 17. The flowchart of obstacle avoidance.

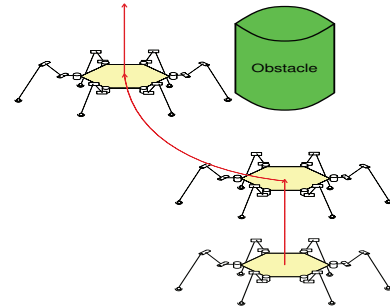


Fig. 18. The trajectories of obstacle avoidance.

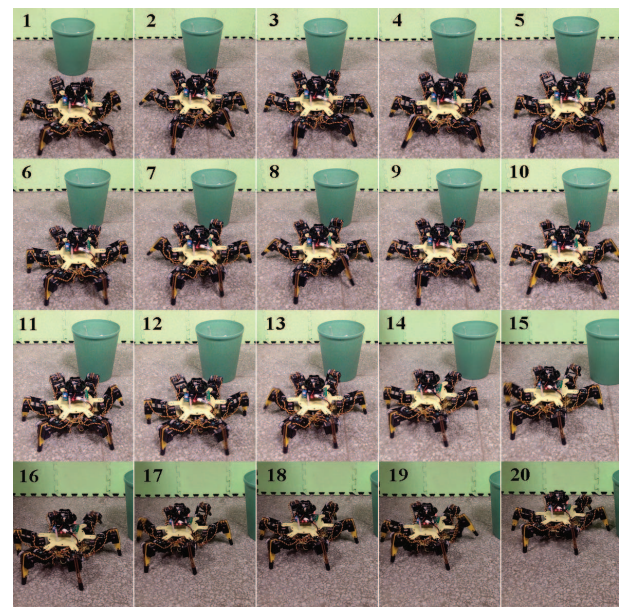


Fig. 19. Exploded view of obstacle avoidance.