การสร้างแบบจำลองและการควบคุมหุ่นยนต์หกขา

Modeling and Control of a Hexapod Robot

เกียรติสิน กาญจนวนิชกุล,[1*] ปริวัฒน์ ผารัตน์,[2] ทยาวัต คณะใน,[3] ปริญญ์ ชุปวา[4]

Kiattisin Kanjanawanishkul,[1*] Pariwat Pharat,[2] Tayawat kananai,[3] Prarin Chupawa[4]

บทคัดย่อ

หุ่นยนต์ 6 ขานั้น รู้จักกันดี ในด้านความสามารถในการจัดการและในการเคลื่อนที่ มันสามารถทำงานได้ในสภาพแวดล้อมที่ซับซ้อน เช่น ภูมิประเทศขรุขระ หรือ พื้นที่ไม่เรียบสม่ำเสมอ มันเหมาะกับงานในสภาพแวดล้อมเช่นนี้ เนื่องจาก มันสามารถเลือกตำแหน่งที่จะวางขาได้ อย่างไรก็ตาม การควบคุมหุ่นยนต์ 6 ขาที่จะสามารถเดินบนภูมิประเทศใดๆ ได้นั้น ต้องมีกรรมวิธีการควบคุมที่มีประสิทธิภาพและน่าเชื่อถือ เพื่อที่จะประสานงานการเคลื่อนที่ของทุกๆ ขา ดังนั้น ในบทความนี้ จึงได้ทำการศึกษาและพัฒนาแบบจำลองและการควบคุมหุ่นยนต์ 6 ขา โครงสร้างของหุ่นยนต์นั้น จะประกอบไปด้วย 6 ขา โดยที่แต่ละขา จะมี 3 องศาอิสระ ฮาร์ดแวร์และซอฟต์แวร์นั้น ได้ถูกออกแบบอย่างระมัดระวัง เพื่อที่จะบรรลุวัตถุประสงค์ทางด้านการศึกษาและการวิจัยในอนาคต ไมโครคอนโทรลเลอร์ LPC2148 ได้ถูกใช้เพื่อควบคุมท่าทางการเดินของหุ่นยนต์ และได้มีการลงมือทำการทดลองเบื้องต้นกับหุ่นยนต์ 6 ขา เพื่อที่จะรับรองว่าโครงสร้างการควบคุมนั้นถูกต้อง จากผลลัพธ์ที่ได้นั้น พบว่าระบบที่ได้นำเสนอนั้นสามารถสร้างท่าเดินที่น่าพอใจบนพื้นราบและระบบควบคุมสามารถควบคุมการเคลื่อนที่ของหุ่นยนต์ได้อย่างประสบความสำเร็จ

คำสำคัญ: หุ่นยนต์ 6 ขา, กรรมวิธีท่าเดิน, จลนศาสตร์แบบไปข้างหน้า, จลนศาสตร์แบบผกผัน

Abstract

A hexapod robot is well-known for its high manipulability and mobility. It can operate in complex environments, such as, rough terrain or uneven floors. It is suitable for work in such environments because it can select any landing point. However, control of a hexapod robot walking on any terrain demands the development of efficient and reliable control algorithms to coordinate the movement of every leg. Therefore, in this paper, modeling and control of a hexapod robot are studied and developed. The structure of the robot consists of six limbs with 3 degrees of freedom each. Hardware and software are carefully designed to meet our educational and research purposes in the future. An LPC2148 microcontroller is used to control walking gaits of the robot. The preliminary experiments with a real hexapod robot have been conducted in order to validate our control framework. As seen in the results, the proposed system is able to generate satisfactory gaits on flat ground and the control system can control the movement of the robot successfully.

Keywords: Hexapod robots, gait algorithms, forward kinematics, inverse kinematics

---

[1] อาจารย์, [2,3] นิสิตปริญญาตรี, [4] นิสิตปริญญาโท, สาขาวิศวกรรมเมคาทรอนิกส์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหาสารคาม อำเภอกันทรวิชัย จังหวัดมหาสารคาม 44150

[1] Lecturer, [2,3] Undergraduate student, [4] Graduate student, Department of Mechatronic Engineering, Faculty of Engineering, Mahasarakham University, Kantharawichai District, Mahasarakham 44150, Thailand

* Corresponding author: Kiattisin Kanjanawanishkul, Department of Mechatronic Engineering, Faculty of Engineering, Mahasarakham University, Kantharawichai District, Mahasarakham 44150, Thailand

Introduction

In general, robot locomotion with wheels and tracks are preferred where terrain is even, however when terrain becomes more uneven the advantages of wheeled locomotion becomes useless. Instead the advantages of legs make a walking robot more useful on uneven terrain because the number of joints on the leg provides the robot more degrees of freedom (DoFs), which can help the robot to place its feet on a better or more precise location in order to avoid obstacles or holes.

Particularly, a significant interest in a hexapod robot has been motivated due to its physical ability of static stability, i.e., only three legs on the ground allowing for a very fast walking gait known as the "tripod gait". Furthermore, a hexapod robot can be seen as a biologically inspired robot since it imitates limb structure and motion control of insects. In addition, since the hexapod has redundant legs compared to other legged robots, it is possible that this sort of a walking robot is able to continue operation in the event of losing limbs.[1] Thus, the use of a hexapod robot is interesting for several applications, including planetary exploration[2], search and rescue operations[3], entertainment (e.g., the 2010 Hexapod Championship held in Hagenberg, Austria), etc., to name some.

However, it suffers from severe limitations, for example, it requires a large number of actuators to move the legs with multiple DoFs, its movement speed is low compared to wheeled locomotion, it consumes high energy and it is difficult to build. Furthermore, its control algorithms are complex since it must have to deal with lifting and placing its feet and to handle unpredictable dynamics during the contact of a foot with the ground. Therefore, over the past few decades, there has been a growing interest in the area of hexapod robots, especially in gait generation algorithms and control of walking mechanisms.

Many research groups have studied approaches for hexapod locomotion control. For example, the Rhex[4] consists of a rigid body with six compliant legs, each one with only one DoF. Thus, it has only six actuators, resulting in a mechanical simplicity to achieve reliable and robust operation in real-world tasks. The Lauron III hexapod robot[5] is more complex. It consists of three DoFs for each leg and each foot has a three-axis force sensor. The motion of a large and highly mobile six-legged lunar robot called ATHLETE was developed by the Jet Propulsion Laboratory[2]. This robot can roll rapidly on rotating wheels over flat smooth terrain and can walk on fixed wheels over irregular and steep terrain. A hexapod robot called RiSE[6] is a new concept of a hexapod robot that is able to climb on a variety of vertical surfaces.

The objectives of this work are to develop a model and to build a real hexapod robot with control architecture able to generate stable gait locomotion. This hexapod platform will be further used for educational and research purposes. To accomplish these objectives, the hexapod robot is designed and built. Then, a forward kinematic model and an inverse kinematic (IK) model of the hexapod robot are derived and gait generation algorithms on flat ground are implemented. Finally, our developed robot framework is validated through experiments.

Robot Description

Since motion of hexapod robots on natural terrain presents several complicated problems (e.g., obstacle avoidance, locomotion stability, and foot placement) that must be taken into account in both mechanical design and development of control strategies. Therefore, they are complex and expensive machines. In particular, each leg is composed of several rigid elements connected by articulated joints.

Mechanical Structure

There are several kinds of hexapod robots with different number of DoFs, joints and articulations in the literature. However, after several iterations, a geometric model is achieved, as it is shown in Figure 1. The mechanical structure of the hexapod robot consists of one rigid, load carrying frame with six legs, similar and symmetrically distributed. It has three DoFs in its legs, i.e., each leg is composed of three links, interconnected by three revolute joints, resulting in a total of 18 DoFs.

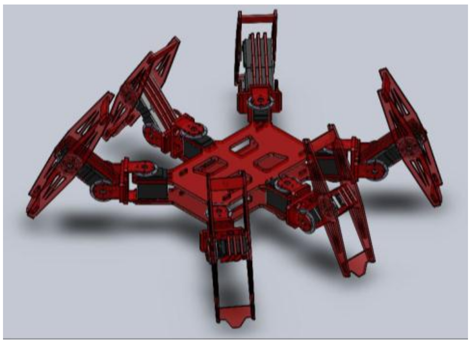

Figure 1 SolidWorks model showing the structure of our hexapod robot.

Each joint is manipulated by one MG995R servo motor. This servo motor can provide 10 kg-cm torque at 6 V. As seen in Figure 1, there is a servo motor connecting the entire leg to the base chassis through a vertical axis, allowing the leg to rotate sideways in relation to the body. The other two servo motors manipulate the other two joints of the leg, with rotation about horizontal axes as seen in Figure 2. The names are adopted from insects, as their legs have a similar structure. The innermost joint is dubbed coxa joint. The middle joint is dubbed femur, and the outermost joint is named tibia.

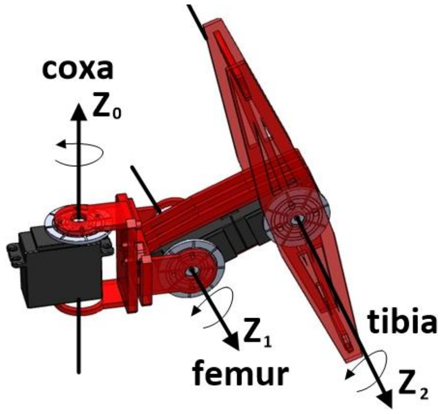

Figure 2 Illustration of the leg model, with rotational axes added and their joints are named coxa, femur, and tibia.

In the robot construction, body chassis is made of acrylic due to light weight. Also, the legs are combined with the acrylic brackets to create the desired leg shape. The complete robot is illustrated in Figure 3. The main dimension of the robot is approximately 450x450x220 mm (width x length x height) and it weighs about 2.6 kg.

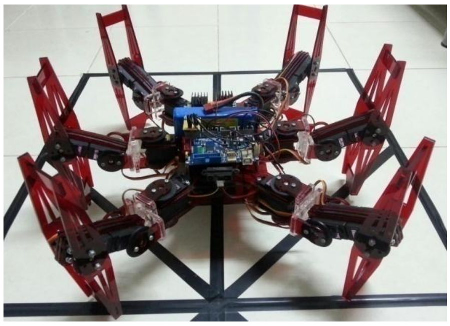

Figure 3 The real hexapod robot

Control Hardware

Control hardware consists of an LPC2148 microcontroller board and two ZX-SERVO16U servo controller boards linking over a serial port. The microcontroller calculates and sends the angle command to the servo controller and then the servo controller generates PWM signals to control the angle of each servo motor that powers each joint of the leg. One ZX-SERVO16U servo controller can control 16 servo motors simultaneously with the resolution of 2 μs. It can also control the speed and time of the servo motion which is very useful when several servo motors have to be controlled at the same time.

Furthermore, two infrared sensors mounted in front of the robot are used to scan the front area for obstacles, while a compass sensor is used to measure a heading angle of the robot. The robot can also be commanded manually via a TV remote control and it can communicate with other devices through Zigbee wireless networks. For example, a PC can be used to handle all high-level control and to monitor all events. A GUI written in MATLAB will be implemented in the future for visualization and for a tele-operated manner. For power source, a 2000 mAh Lithium polymer battery is used to supply power to overall system. The overall system diagram is summarized in Figure 4. It is important to note that all components in this work are locally available.

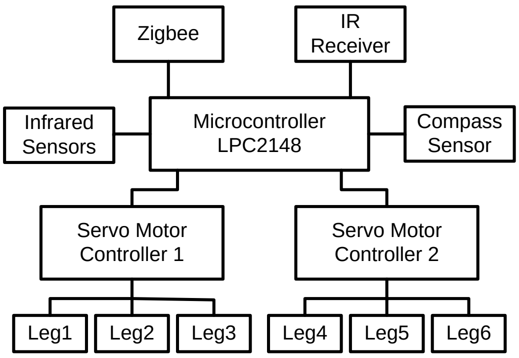

Figure 4 Control hardware of the hexapod robot

Software Structure

Software framework as illustrated in Figure 5 is programmed in C and contains the following modules:

- Network module is used for communicating between a PC and a microcontroller board.
- Servo module receives the angular position of each leg, converts to time unit and then sends to the servo controller via a serial communication protocol.
- Sensor module receives information from sensors and processes. At the moment, there are only two infrared sensors and one compass sensor.
- User interface and remote control module is used to interface with the user.
- Gait generation module is used to generate a walking gait.
- Kinematic and inverse kinematic module deals with a kinematic model of the robot.

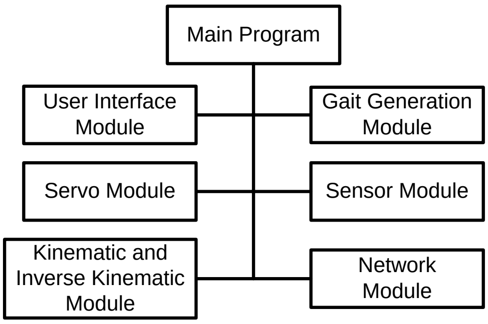

Figure 5 Software architecture

A Kinematic Model and Dynamic Considerations

The control system shown in Figure 6 requires the user to give a desired pose, $[x, y, z, \alpha, \beta, \gamma]^T$ in the global frame and a desired walking gait to the gait generator. Then the gait generator sends leg coordinates to the inverse kinematic solver for each leg. The inverse kinematic solver calculates those given coordinates into its angular coordinates for all the joints. Then all these joints are sent to the servo controller.

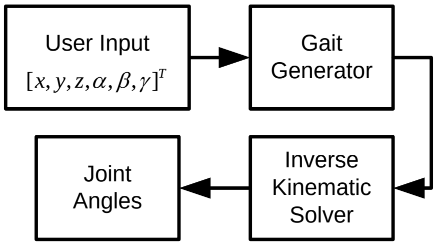

Figure 6 The overview of the locomotion control structure of our hexapod robot.

Forward kinematics

A forward kinematic model of the hexapod robot is used to determine how the robot can be moved in a global coordinate system. First of all, the origin of the robot body frame is attached to the center of the bottom plane of the central robot structure with the z-axis pointing up, the x-axis pointing right and the y-axis pointing forward, as illustrated in Figure 7.

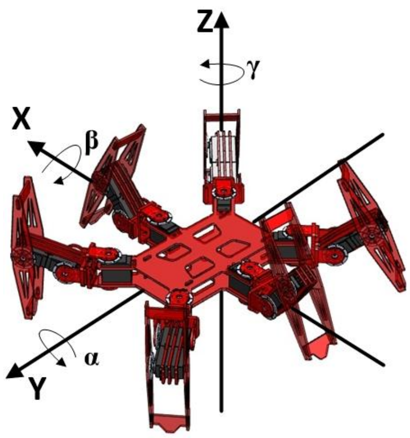

Figure 7 Location of the robot body frame relative to the robot hardware.

Then, the coordinate frames for the robot legs are assigned as shown in Figure 8. The assignment of link frames follows the Denavit-Hartenberg (D-H) notation[7]. The robot leg frames starts with link 0 which is the point on the robot where the leg is attached, link 1 is the coxa, link 2 is the femur and link 3 is the tibia. Link 4 is the end point of the leg and coincides with link 3.

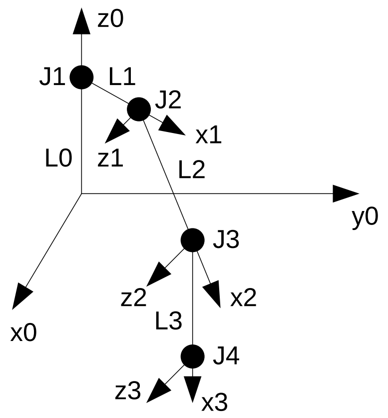

Figure 8 Illustration of leg frame and link frames.

Table 1 shows the D-H parameters of our hexapod robot shown in Figure 2.

Table 1 D-H parameters for one robot leg

Link	α_i	a_i	d_i	θ_i
1 (coxa)	90°	60 mm	25 mm	θ_1
2 (Femur)	0	62 mm	0	θ_2
3 (Tibia)	0	112 mm	0	θ_3 - 90°

The forward kinematics are a set of equations consisting of transformation matrix, transforming coordinates of one link frame to coordinates of another link frame. The general form for the transformation matrix from link *i* to link *i-1* is given in the following equation[7]:

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The specific leg transformation matrices, transforming the coordinates from frame {3} to frame {0}, is shown in (2) – (5):

$$T_3^0 = T_1^0 T_2^1 T_3^2 \quad (2)$$

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 60\cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 60\sin\theta_1 \\ 0 & 1 & 0 & 25 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 62\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 62\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_3^2 = \begin{bmatrix} \sin\theta_3 & \cos\theta_3 & 0 & 112\sin\theta_3 \\ -\cos\theta_3 & \sin\theta_3 & 0 & -112\cos\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Next, the leg frames are positioned on the robot body as illustrated in Figure 9. The leg frame needs a rotation about their z-axis and a translation along their x-axis and y-axis to be positioned correctly in the robot body frame.

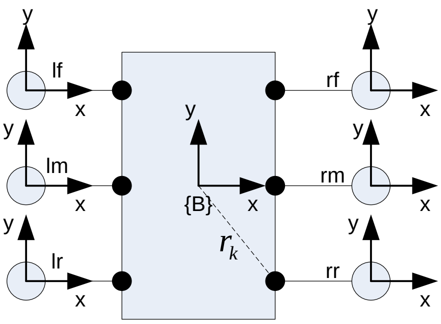

Figure 9 Illustration of the position of the leg frame relative to the robot body frame.

The general leg to body transformation T_L^B is written in (6). Subscript *L* is referred to a specific leg, for example, the transformation is denoted T_{lf}^B for the transformation from the left front (lf) leg frame to the body {B} frame and T_{rm}^B for the transformation from the right middle (rm) leg frame to the body {B} frame and so on.

$$T_L^B = \begin{bmatrix} \cos\gamma_k & -\sin\gamma_k & 0 & r_k\cos\gamma_k \\ \sin\gamma_k & \cos\gamma_k & 0 & r_k\sin\gamma_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

where the legs are distributed symmetrically along two sides with distance r_k to the body frame and angle γ_k for the *k*th leg, relative to the body frame.

The transformation matrices from the leg end point frames to the body frame can now be written as in (7) – (12). The 3 in the indices indicates that the transformation transforms coordinate frame {3}, in leg frame, to the body frame.

$$T_{rm3}^B = T_{rm}^B \cdot T_3^0 \quad (7)$$

$$T_{rf3}^B = T_{rf}^B \cdot T_3^0 \quad (8)$$

$$T_{rr3}^B = T_{rr}^B \cdot T_3^0 \quad (9)$$

$$T_{lm3}^B = T_{lm}^B \cdot T_3^0 \quad (10)$$

$$T_{lf3}^B = T_{lf}^B \cdot T_3^0 \quad (11)$$

$$T_{lr3}^B = T_{lr}^B \cdot T_3^0 \quad (12)$$

Next, the robot body frame is transformed to the global frame. The roll, pitch and yaw angles (α, β and γ) rotates the body around the y-axis, the x-axis and the z-axis, respectively. The rotation of the body frame consists of three rotations, one about each axis. In this case the rotations occur in the yxz (roll-pitch-yaw) order. The transformation from the robot body frame to the global frame is defined as in (13).

$$T_B^G = \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\cos\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & X \\ \cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & \cos\beta\cos\gamma & \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & Y \\ -\cos\beta\sin\alpha & \sin\beta & \cos\alpha\cos\beta & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Now the position of the leg end point with respect to the global frame can be expressed as in (14).

$$p_3^G = T_B^G T_L^B T_3^0 p_3 \quad (14)$$

where $p_3 = [0\ 0\ 0\ 1]^T$ is the leg end point in frame {3}.

Inverse Kinematics (IK)

To be able to find the angles of all the joints on the robot (i.e., θ_1, θ_2, and θ_3) it is necessary to know the position of the end points, and also the roll, pitch, yaw, and position of the robot body, in the global frame. In general, solving the IK equations can present some challenges. Some positions cannot be reached at all, e.g., the position is too far away from the robot. This is called kinematic singularities. Some end point positions could have more than one solution, and not all the solutions are equally desirable.

Before the IK can be solved for the individual legs, the leg end point coordinates with respect to the global frame needs to be transformed to the individual leg frames. This inverse transformation is the pseudo inverse of the leg to body transformation T_L^B and body to global frame transformation T_B^G which are shown in (15) and (16).

$$T_G^B = \left(T_B^G\right)^{-1} = \begin{bmatrix} \left(R_B^G\right)^T & -\left(R_B^G\right)^T \cdot d_B^G \\ 0 & 1 \end{bmatrix} \quad (15)$$

$$T_B^L = \left(T_L^B\right)^{-1} = \begin{bmatrix} \left(R_L^B\right)^T & -\left(R_L^B\right)^T \cdot d_L^B \\ 0 & 1 \end{bmatrix} \quad (16)$$

where

R_B^G is the rotational transformation from the body to the global frame.

d_B^G is the translational transformation from the body to the global frame.

R_L^B is the rotational transformation from the leg to the body frame.

d_L^B is the translational transformation from the leg to the body frame.

For this particular robot system, the IK equations have to be solved for each leg separately. After transforming the leg end point coordinates from the global frame to the leg frame, the coxa joint angle θ_1 can be found using the atan2(y_3^0, x_3^0) function, where x_3^0 and y_3^0 are the x-component and y-component of the position of the leg end point in the leg frame. The relation between coxa angle and robot body, is illustrated on Figure 10. It has to be noted that we do not consider the end point that is positioned directly below the coxa joint or closer to the center of the robot than the coxa joint.

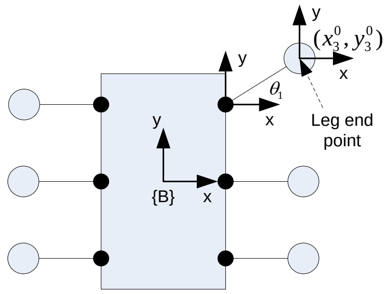

Figure 10 Illustration of the coxa joint angle, in the leg frame. It is equivalent to determine the angle of the end point relative to the x-axis and y-axis of the leg frame.

To find the femur and tibia angles, the leg end point coordinates are transformed to the femur frame, by the transformation in (18) to obtain x_3^1 and y_3^1:

$$T_0^1 = \left(T_1^0\right)^{-1} \quad (17)$$

$$T_0^1 = \begin{bmatrix} \left(R_1^0\right)^T & -\left(R_1^0\right)^T \cdot d_1^0 \\ 0 & 1 \end{bmatrix} \quad (18)$$

Then, the angles can be found by looking at the angles in the triangle with vertices in the origins of the femur, the tibia, and the leg end frame as shown in Figure 11. This triangle lies in the xy-plane of the femur link, as illustrated in Figure 12.

In Figure 11, the angle θ_2, which is the angle relating to the femur angular position, can be derived directly from the triangle as follows:

$$\theta_2 = \phi_2 \quad (19)$$

The angle ϕ_1 can be found by looking at the angle between the line b and the x-axis. This angle can be easily found as shown in (20)

$$\phi_1 = \operatorname{atan2}(y_3^1, x_3^1) \quad (20)$$

where x_3^1 and y_3^1 are the x- and y-components of the leg end point coordinates in the femur frame. Note that ϕ_1 is always negative.

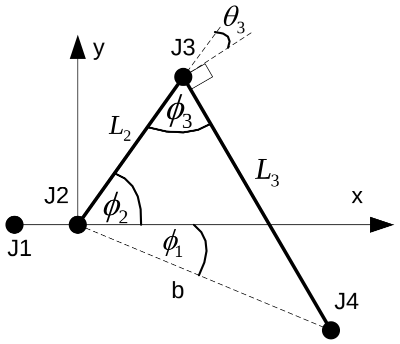

Figure 11 Illustration of the 2D triangle with vertices in the femur, the tibia, and the leg end point.

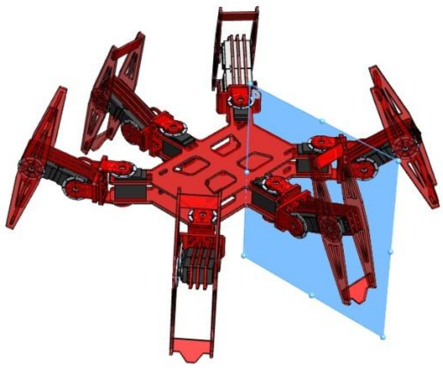

Figure 12 Illustration of the femur frame. It is always oriented so the x-axis is parallel with the coxa link.

ϕ_2 can be found by using the law of cosines as shown in (22):

$$b = \sqrt{(x_3^1)^2 + (y_3^1)^2} \quad (21)$$

$$\phi_2 - \phi_1 = \operatorname{acos}\left(\frac{(L_2)^2 + b^2 - (L_3)^2}{2L_2 b}\right) \quad (22)$$

Using the same formula as in (22) we can find ϕ_3

$$\phi_3 = \operatorname{acos}\left(\frac{(L_3)^2 + (L_2)^2 - b^2}{2L_2 L_3}\right) \quad (23)$$

ϕ_3 relates directly to θ_3, the tibia joint angle, as shown in (24)

$$\theta_3 = \phi_3 - \frac{\pi}{2} \quad (24)$$

The equations below provide a summary of the formulas needed to find the individual joint angles.

$$\theta_1 = \operatorname{atan2}(y_3^0, x_3^0) \quad (25)$$

$$\theta_2 = \operatorname{acos}\left(\frac{(L_2)^2 + b^2 - (L_3)^2}{2L_2 b}\right) + \operatorname{atan2}(y_3^1, x_3^1) \quad (26)$$

$$\theta_3 = \operatorname{acos}\left(\frac{(L_3)^2 + (L_2)^2 - b^2}{2L_2 L_3}\right) - \frac{\pi}{2} \quad (27)$$

Dynamic Considerations

Dynamics must be taken into account in order to ensure static and dynamic stability. To deal with static stability, the center of mass (CoM) of the robot must lie within the Polygon of Stability (PoS) at all times[8] as shown in Figure 13. Since the CoM can be projected onto the ground perpendicular to the global z-axis, it is possible to discard the z-component. Then the CoM can be defined as follows:

$$c^G = \frac{\Sigma m_i \cdot p_{CoMi}^G}{\Sigma m_i} = \frac{\Sigma m_i \cdot T_B^G T_L^B T_i^0 \, p_{CoMi}^i}{\Sigma m_i} \quad (28)$$

where m_i is the mass of link *i* and p_{CoMi}^i is the CoM point of link *i* positioned relative to the link frame.

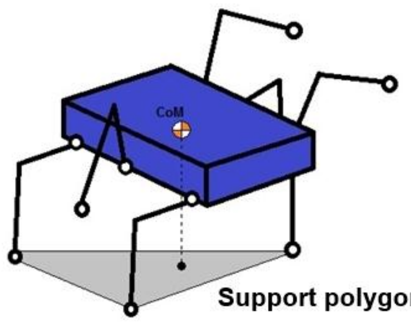

Figure 13 Support polygon (support pattern) of a multi legged robot with the CoM above. This means that when the CoM is above the support polygon the robot is

statically balanced.

Dynamic stability is needed when the CoM is outside or on the border of PoS, which makes the robot falling over. To ensure dynamic stability the Zero Moment Point (ZMP) must lie within the PoS at all times and not lie on the edges of the PoS. However, it is beyond the scope of this work. The reader is referred to reference 8 for more details.

Six-legged Gait Generation

After constructing a walking mechanism, a leg and body motion sequence is performed to make the mechanism walk. This sequence is known as the gait. The function of the gait generator for a walking robot is to select an appropriate sequence of leg and body movements so that the robot can advance with a desired speed and direction.

Developing walking patterns is a challenging task because there are a large number of DoFs and therefore the solution must be determined in a large, multidimensional search space. The problems of gait generation have been studied for many years and there are a variety of strategies to solve this problem found in the literature[9].

Insects walk with a variety of different patterns of leg movements at different speeds. In this work, we focus only on the following periodic gaits[10], although the free gait is much more effective on rough terrain with obstacles.

- Wave Gait: the robot move one leg at a time, it starts by lifting one leg and then lowering it down until the foot touches the ground and then the next leg starts to move, as illustrated in Figure 14. Obviously, this gait sequence is rather slow but it provides maximum stability for the robot, and it enables the robot to walk on uneven terrain.

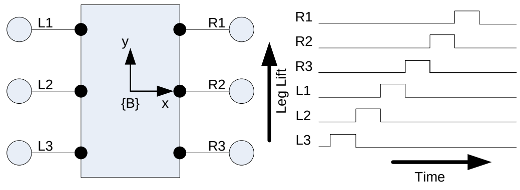

Figure 14 Single wave gait

- Ripple Gait or Two Wave Gait: the robot moves two legs at a time as shown in Figure 15 and it needs 3 phases to complete one cycle.

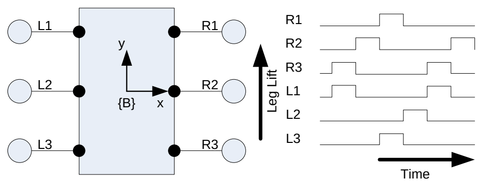

Figure 15 Two wave gait

- Tripod Gait: This gait is the fastest gait for the hexapod; it completes a cycle in two phases as seen in Figure 16. The robot lifts three legs at the same time while leaving three legs on the ground, which keeps the robot statically stable.

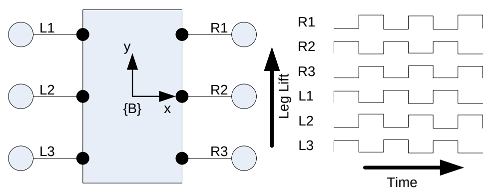

Figure 16 Tripod gait

Experimental Results

As mentioned above, there are several ways of walking (called gaits), and each gait has its own advantages. In this work, the preliminary experiments of following a straight-line path with respect to the global frame and rotating about the vertical-axis of its body frame have been conducted. Three repeats of each experiment were performed and average results were recorded. In addition, in our implementation, in order to ensure all the feet move in a correct sequence, a gait lookup table was produced and the delay between the consecutive phases was added.

In the first experiment, the robot was programmed by hand to walk with a statically stable gait: a tripod gait. This is the fastest statically stable gait for a hexapod robot, and it is one of the standard gaits of insects. Step length of a single complete cycle was set to 7 cm.

Two sets of three legs each were moved repeatedly. The robot had always at least three legs placed on the ground, and its CoM was located within the support polygon all the time. After four complete cycles, the robot arrived at the final destination point. As seen in Table 2 and Figure 17, the robot can successfully follow a straight line with small deviations. The deviations from the desired values are due to the amount of clearance between servo motors and leg structure and imprecise timing of each joint in leg coordination. The average walking speed of our hexapod robot while using this gait and walking straightforward is achieved at about 1.56 cm/s.

In the second experiment, the robot was programmed to rotate about its z-axis of the body frame with zero forward velocity and the rotation angle at one step was set to 45 degrees. As seen in Table 3 and Figure 18, the robot can rotate counterclockwise correctly with very small errors. It is capable of turning at a rate of approximately 0.11 rad/s.

Figure 17 A snapshot of experimental results during the robot follows a straight line on a flat ground using a tripod gait.

Figure 18 A snapshot of experimental results during the robot rotates about the vertical-axis of its body frame using a tripod gait.

Table 2 Experimental results during the robot follows a straight line on a flat ground using a tripod gait.

Desired Pose			Actual Pose			Error			Time (s)	Velocity (cm/s)
X (cm)	Y (cm)	θ (deg)	X (cm)	Y (cm)	θ (deg)	ΔX (cm)	ΔY (cm)	Δθ (deg)		
7	0	0	7.0	0.2	1.6	0	0.2	1.6	4.83	1.45
14	0	0	15.2	0.9	3.4	1.2	0.9	3.4	9.49	1.60

21	0	0	22.7	1.7	4.3	1.7	1.7	4.3	14.37	1.58
28	0	0	30.2	2.6	4.9	2.2	2.6	4.9	18.82	1.61

Table 3 Experimental results during the robot rotates about the z-axis of its body frame using a tripod gait.

Desired pose			Actual Pose			Errors			Time (s)	$\omega = \frac{\theta}{T}$ (rad/s)
X (cm)	Y (cm)	θ (deg)	X (cm)	Y (cm)	θ (deg)	ΔX (cm)	ΔY (cm)	$\Delta\theta$ (deg)		
0	0	45	0	0	45.5	0	0	0.5	7.33	0.108
0	0	90	0	0	91.9	0	0	1.9	14.53	0.110
0	0	135	0	0	136.2	0	0	1.2	21.94	0.108
0	0	180	0	0	186.2	0	0	6.2	29.20	0.111
0	0	225	0	0	230.1	0	0	5.1	36.61	0.110
0	0	270	0	0	274.2	0	0	4.2	43.83	0.109
0	0	315	0	0	316.8	0	0	1.8	50.93	0.109
0	0	360	0	0	364.5	0	0	4.5	58.35	0.109

As seen in both experimental results, the first version of our hexapod robot can satisfactorily achieve basic requirements of a walking robot. However, it has to be noted that our hexapod robot is an open-loop system. The tracking errors can be decreased, if a localization feedback system is implemented. In addition, due to its discontinuous periodic motion and the pre-programmed leg sequence manners, it suffered from unsmooth movement during phase transition. Nevertheless, our developed robot framework is very promising and it will be further developed to achieve our ultimate goal, i.e., it will be able to navigate over uneven and unstructured environments.

Conclusions and Future Work

The remarkable advantage of hexapod robots is that they can walk over rough terrain where the wheeled locomotion is very complicated. However, their main disadvantage is the difficult control due the multiple DOFs. In this work, we achieved the following main objectives: to derive a kinematic model of a hexapod robot, to construct a hexapod robot by considering its most important features, and to control its movement. A gait generation algorithm and a controlling system for a hexapod robot were developed. We demonstrated walking straightforward and turning about the z-axis of its body frame in order to verify the effectiveness of our proposed robot system.

However, there is still much work remaining to be done. Since our long term goal is to enable the robot to navigate over uneven and unstructured environments, this ability will require more robustness to the locomotion controller. Furthermore, the robot system will be used for an educational area of mechatronic engineering, particularly in robotics. Therefore, our future work includes

- An intelligent gait generator: gait generation is very important as it directly affects the quality of movement of legged robots. It is classified as an optimization problem with constraints[9] because it has to determine the optimal position, velocity and acceleration for each DoF at any moment in time. It also directly affects the robot's dynamic stabilization, energy dissipation, actuator saturation, and so on. The controller should generate smooth trajectories, in order to result in smooth movements.

- Sensors and actuators: a gripper arm that enables grasp and manipulation of objects will be installed. A vision system will be integrated into the robot in order to obtain visual information. Sonar sensors will be mounted in order to detect the height of the hexapod. Force sensors will be attached on the tip of each leg to let the robot know when its feet touch something.

Acknowledgements

This research has been financially supported by Faculty of Engineering, Mahasarakham University, which is gratefully acknowledged.

References

1. Görner M, Hirzinger G. Analysis and evaluation of the stability of a biologically inspired, leg loss tolerant gait for six- and eight-legged walking robots. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2010 May 3-7; Anchorage, AK. p. 1525 – 1531.
2. Hauser K, Bretl T, Latombe J, Wilcox B. Motion planning for a six-legged lunar robot. In: Proceedings of the International Workshop on the Algorithmic Foundations of Robotics; 2006 Jul 16-18; New York City.
3. Karalarli E, Erkmen A, Erkmen I. Intelligent gait synthesizer for hexapod walking rescue robots. In: Proceedings of the IEEE International Conference on Robotics and Automation; 2004 Apr 26 - May 1; New Orleans, LA. p. 2177 – 2182.
4. Saranli U, Buehler M, Koditschek D. Rhex: A simple and highly mobile hexapod robot. International Journal of Robotics Research 2001; 20(7): 616–631.
5. Celaya E, Albarral J. Implementation of a hierarchical walk controller for the Lauron III hexapod robot. In: Proceedings of the International Conference on Climbing and Walking Robots; 2003 Sep 17-19; Catania, Italy. p. 409 – 416.
6. Haynes G, Rizzi A. Gait regulation and feedback on a robotic climbing hexapod. In: Proceedings of the Robotics: Science and Systems; 2006 Aug 16-19; Philadelphia, Pa.
7. Spong M, Hutchinson S, Vidyasaga M. Robot modeling and control. New York: John Wiley & Sons; 2005.
8. Siciliano B, Khatib O, editors. Springer Handbook of Robotics. Berlin: Springer; 2008. p. 361-389.
9. Gong D, Yan J, Zuo G. A review of gait optimization based on evolutionary computation. Applied Computational Intelligence and Soft Computing 2010.
10. Campos R, Matos V, Santos C. Hexapod locomotion: a nonlinear dynamical systems approach. In: Proceedings of the 36th Annual Conference on IEEE Industrial Electronics Society; 2010 Nov 7-10; Glendale, AZ. p. 1546 – 1551.