ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

Design of Simulink-based 3D Hexapod Robot Model and Implementation of Hexapod Robot Foothold Planning

Fanzhi Meng

University of Manchester, Manchester, United Kingdom

Abstract: *In this world, there are many dangerous and complicated environments in which humans cannot work. Mobile robots are an important method that people use to solve these problems. Hexapod robots are known for their excellent static and dynamic stability, as well as strong environmental adaptability in mobile robots. Hexapod robots play a critical role in overcoming complex and challenging terrain. The paper attempts to finish two different parts of the work. The first part is to construct a Simulink-based CORIN robot model, and the second part is to develop a foothold planning algorithm for hexapod robots. The highly visual robot model can be used to display the results of related algorithms. It can also be applied in the Simulink physical environment to simulate robot overturning recovery, robot wall walking and other advanced robot motions through the use of the contact force blocks. Moreover, for the foothold planning algorithm, a collision detection algorithm based on the distance between the feature points and the obstacles is implemented. 18 feature points are set on the CORIN robot to ensure that the robot optional footholds are all collision-free points. Based on the path length, the distance from the target point and the collision state, a cost function is established, which is used in the CORIN robot foothold path planning algorithm. Results show that based on the kinematics of CORIN robot leg and modeling method in the Simulink, a complete CORIN hexapod robot model that can be controlled by specifying the position of the end effector or setting three different revolute joint angles has been designed. Furthermore, a radical foothold planning algorithm that can avoid collisions and focus on reaching the target point is implemented.*

Keywords: *Hexapod robots, Foothold planning, Modeling*

1. Introduction

1.1 Background and motivation

In this world, there are many specific situations or extremity environment places that are too risky for people, such as search and rescue in disaster [1], detect other planets surface [2], explore and emergency handling in the nuclear environment [3], anti-terrorism [4] and so on. Nowadays, the most popular method to solve these problems is to use ground mobile robotics [5]. Based on the different locomotion systems, there are three types of ground mobile robotics: wheeled robot, tracked robot and legged robot [5][6].

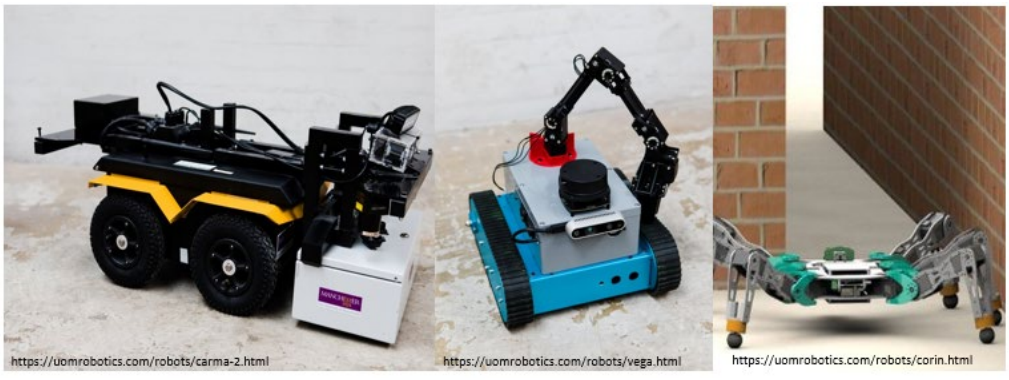

Figure 1: Three kinds of ground mobile robotics with different locomotion systems.

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

Three different categories of land-based mobile robots are shown in the Figure 1, the picture (a) is a wheeled robot which is one of the most critical branches of mobile robots. Compared with other mobile robots, not only the locomotion system and control system of the wheeled robot are much simpler, but also the price is lower and the hardware design is easier. Due to the use of wheels, the stability of the robot is better and the speed is faster when the environment is flat and less small rugged [7][8]. However, wheeled robots cannot be used in too rugged and complex terrain. Tracked robot, which applies caterpillar band to take place of wheels, is presented in the picture (b). Because of the caterpillar band, tracked robots can reduce the impact of rugged terrain and get over some gentle obstacles, which can also maintain fast moving speed and high mobility [5]. But when it comes to flat and smooth terrian, tracked robots will have errors in moving and steering due to insufficient friction [9]. Therefore, tracked robots are suitable for high friction and complex terrain.

The robot that presented in the Figure 1 picture (c) is called as legged mobile robot. Unlike wheeled robots and tracked robots, the model design of legged robots is complicated, which also means that its motion planning and control systems are more complicated. Although legged robot is more expensive and the system is more difficult to design, it has greater advantages when dealing with extremity situations and extremely rough terrains. The first advantage is that legged robot has a strong ability to step over obstacles or climb steep terrain. The second one is the legged robot can do some advanced motions like across the cracks or walk with the help of walls [10]. When the robot needs to work in a soft terrain, the legged robot can move normally without being as difficult to work as the wheeled robot, which is the third advantage. A legged robot can move smoothly and stably in rugged and complex environment is the fourth one. The last advantage is the legged robot has good ground protection ability [11]. Based on the number of legs, legged robots can be classified into different categories, such as pogo stick robots (one-legged robots), bipedal robots, tripodal robots, quadrupedal robots, hexapod robots and octopod robots [12].

As future work will consider robots using advanced motions such as wall walking and chimney walking to pass through extremely complex environments, hexapod robots are selected to be used in this project because they have better static and dynamic stability than bipedal and quadrupedal robots, and are easier to design and control than octopod robots. Some functions written by using the Matlab can be used directly in the Simulink, and the Simulink model can be used to verify the related algorithms completed with Matlab. Therefore, it is very critical to complete the 3D modeling of the CORIN robot in the Simulink for the use of the Matlab for algorithm programming.

After completing the 3D modeling and kinematics modeling of the CORIN robot, the next step is to realize the path planning of the CORIN robot. In order to solve this problem, two methods are provided. The first one is body path planning and the second method is foothold path planning. The first method is implemented by applying A* algorithm. This paper will adopt the second method. This paper will introduce how to build a CORIN robot model in the Simulink and explain how to implement the hexapod robot foothold path planning algorithm.

1.2 Report structure

The structure of the paper is arranged as follows. Chapter 2 will introduce the background knowledge of path planning for mobile robots and background information about hexapod robots, and review the relevant literature on the use of legged robot path planning algorithms. Chapter 3 is concerned with the method of physical modeling in the Simulink, the calculation of the positive and inverse kinematics equations of the CORIN hexapod robot and the detail of robot collision detection algorithm and foothold planning algorithm. Chapter 4 presents the CORIN robot model and the results of the grid-based foothold planning algorithm, and then discusses in detail about cost function in the algorithm, the ‘elbow-up’ and ‘elbow-down’ problem in the kinematics. Chapter 5 concludes the whole paper.

2. Literature review

2.1 Legged robots path planning

2.1.1 Overview of legged robots path planning

A series of points or curves that are used to connect the start point and target point is called path. The creation of a collision-free path with the optimization of one or more issues of the mobile robot movement such as energy consumption, time or distance based on the cost function is path planning of mobile robots

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

[13]. According to the different understanding of environmental information, mobile robot path planning can be divided into two different categories. One is local path planning [14], and the other is global path planning [15].

Local path planning is a kind of online planning method, which is also called dynamic planning [16]. Before the local path planning, the mobile robot knows nothing about environmental information [17], so it needs to analyze the environment in real-time by applying visualization sensors [18][19] or other positioning sensors [20][21]. The more popular local path planning methods for mobile robots are the Artificial Potential Field method [22], dynamic windows method [23], and vector field histogram method [24].

Contrary to the local path planning, the global path planning, also known as static path planning, is an off-line planning method [25]. All environmental information needs to be provided before global planning, which means the position of the obstacles are fixed and has been known by the mobile robot [26]. The genetic algorithm (GA) [27], search algorithm [28], Ant Colony Optimization method [29] are the general methods of robot global path planning. The global path planning problem for legged robots consists of two parts [10], the robot body global path planning and the robot foothold global planning algorithm. More relevant information will be introduced in section 2.1.2.

2.1.2 Related works of legged robots path planning

Čížek, Masri and Faigl [30] introduced a method of foothold planning for a hexapod robot. First, an RGB-D camera was used to record the working environment and converted to a 2.5D terrain map. The traditional A* search algorithm was applied to generate the hexapod robot body path and verify the reachability of the target point. When the actual workspace of the hexapod robot was determined, the greedy algorithm was applied to get foothold planning based on the set cost function which includes kinematics cost, position cost and robustness cost. The method of using the A* algorithm to determine the path of the robot body and verifying whether the robot can reach the target point and the method of using the greedy algorithm for foothold planning can be learned. The construction of this cost function is focused on the stability of the hexapod robot, which will perform well in complex environments. However, ignoring the robot forward speed may not be aggressive enough in an austere environment.

Belter et al. [31] proposed a method to obtain the optimal body path by combining two different level planners. Firstly, the high-level planner which uses the A* algorithm was applied to generate a suitable coarse path. Then the low-level planner using the guided-RRT algorithm was used to traverse other feasible paths based on the coarse path generated by the high-level planner. Finally, the optimal path that determined by the low-level planner was applied to make hexapod robot follow a path with a shorter distance and higher passability. The foothold selection algorithm was based on the thought of walking stability of the robot, avoiding falling down and avoiding slippages. The robot was trained in the simulation environment and the relevant decision ability of the robot was developed to make the robot analyze feasible footholds based on the map. The use of the two body path planning algorithms in this method is a good try. The use of the RRT algorithm will face a large amount of calculation and randomness. However, the A-star algorithm is applied first to guide the RRT algorithm. A more suitable path can be obtained.

Winkler et al. [32] proposed a method for motion planning on challenging terrain. The grid-based Anytime-Repairing A* (ARA*) search algorithm was used as the algorithm of robot body path planning. In the cost function design of foothold planning, terrain cost, collision cost, body orientation cost and stability cost were all considered. The ARA* algorithm is different from the traditional A* algorithm. After the A* search algorithm is used to obtain the path for the first time, ARA* will repeatedly run the A* algorithm to ensure that the optimal path is generated. This method is more complicated than traditional methods, but the results are more accurate than those using traditional methods. The cost function of the foothold planning in this method is designed comprehensively, and a cost between the robot body path with the foothold is also established. Since this method is applied to challenging terrain and does not require to reach the target point fast, the distance cost is not considered in the cost function.

Belter [33] introduced a method of using Gaussian Mixtures (GM) to model constraints in legged robot path planning to improve the foothold planning algorithm. For the robot body path planning algorithm, the RRT-Connect method was applied, which is a kind of advanced RRT algorithm. It could design the robot gait at each step of the legged robot while planning the body path. In the foothold planning part, Belter did not use the greedy algorithm, which was a common foothold planning method. Instead, a machine learning method was used, which could make the robot select the foothold by itself. The robot was trained continuously in the simulation environment until the robot could select the foothold point where would not be slippages. However, robots could only classify footholds based on stability.

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

Therefore, the GM model was used in foothold planning. The model consists of the leg workspace constraint, kinematics restriction and self-collision restriction. Based on the GM model, the best foothold could be selected. In this method, many constraint conditions for the foothold selections are separated, which reduces the algorithm complexity and computational complexity of the robot learning process for foothold selection. This is an excellent way to simplify difficult tasks. The RRT-Connect algorithm used here can simultaneously obtain the gait design in the process and body path, which is a good choice for body path planning.

Mastalli et al. [34] proposed a method for optimizing the body path planning and foothold planning of a quadruped robot at the same time based on ground environment information. For the foothold planning part, the foothold was selected based on the terrain cost map. For the robot body planning part, the robot target point, the actual state of the robot and the cost map were used to calculate the optimal control value. And used this value to generate the desired optimal path of the robot body. This method is a novel path planning method, which requires a deep understanding of robot gait control and kinematic constraints. The footholds and different robot gaits can be applied to keep the quadruped robot stable. Constraints can be used to change the robot foothold selection range in the cost map.

Wermelinger et al. [35] introduced a cascaded hierarchical planning structure for quadruped robots in challenging terrain. The robot body path planning used the RRT* method. A greedy algorithm was applied to the robot foothold planning. Different from traditional planning, the special planning structure could search fastly in a simple environment. In complex terrain, the short-distance planning part of the structure could speed up robot body path generation. This structure was equivalent to performing global path planning and local path planning at the same time. When the global planning could not find a suitable path, the local path would be used until the global path appeared again. Although the RRT* body path planning algorithm and the greedy foothold selection algorithm are common methods, the use of the hierarchical planning structure is a good way to speed up the path generation.

2.2 Hexapod robot

2.2.1 Background information of hexapod robot

As a part of the legged robot, the hexapod robot is a kind of bionic robot [36], which is built with insects as a model. The kinematic principle of most hexapod robots is the same as the insects, such as spiders [37] and cockroaches [36]. The leg structure of most hexapod robots is also similar to the insects, with three links and three joints on one leg [38]. The Degree of Freedom (DoF) of most hexapod robots for one leg is 6. However, some hexapod robots do not have such a leg structure and do not have 6 DoF, such as the RHex robot [39] and six-parallel-legged robot [40]. The movement of the hexapod robot is controlled by robot gait. The tripod gait and the wave gait are two common hexapod robot gaits. Since the robot can keep static stability on three or more legs [11], both tripod gait and wave gait have stability. The walking speed of the tripod gait is faster than that of the wave gait, but the wave gait has better stability [12].

2.2.2 Application of hexapod robot

GAINA [41] introduced the application of using a hexapod robot to detect plane fuel tanks. The fuel tank of a plane is a toxic environment, which is not suitable for humans to work in. Using hexapod robots instead of humans to work in plane fuel tanks can solve this problem. The hexapod robot was chosen because of its stability and ability to cross obstacles. The hexapod robot needs to use a video camera to record and inspect the quality of the fuel tank. The robot does not need to complete these tasks alone. The human will control the walking of the hexapod robot through a smartphone, and observe the condition of the fuel tank through the camera carried by the robot. In this application, the advantages of the hexapod robot are not fully reflected. It only needs to be able to remain stable and cross some obstacles through the control of the robot motion by engineer when walking in the fuel tank.

Kang et al. [42] introduced a hexapod wheeled robot for planetary surface detection. The robot has the mobile efficiency of a wheeled robot, the stability of a hexapod robot, and its adaptability to the environment. Many sensors are installed on the legged-wheeled robot, including LiDAR, GPS, IMU, Integrated Navigation System, etc. The robot is designed to be an exploration robot that can be used to carry instruments or equipment, and walk on the surface of the planet. Having a high load capacity means that more accurate sensors can be used to explore the planet. And the robot has the flexibility of a hexapod, more places that the wheeled exploration robot cannot reach can be further observed. The hybrid control method is used on the legged-wheeled robot to ensure that the robot can smoothly go through irregular terrain. At the same time, the stability of the robot movement when carrying the instrument is also

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

considered. Give priority to ensuring the safety of the instrument. A simulated planet surface environment is built by using soft soil, gravel and rocks. A robotic arm for grasping objects is put on the robot for testing the robot stability and observing the robot movement on the soft terrain. The simulation results show that using wheels to move in the complex environment of the planetary surface cannot maintain the stability of a loaded robot. When switching to a hexapod robot, the stability of the robot using the tripod gait movement will also be disturbed by the terrain. In this complex situation, using a more stable hexapod robot wave gait for movement is a good choice.

Deng et al. [43] introduced a hexapod robot called PH-Robot and its control method used in disaster rescue. Since disaster terrain is generally complex terrain, PH-Robot with better stability is used. Moreover, the disaster rescue robot also has the ability to grasp objects, which means it can use four legs to stand and walk, and the remaining two legs are used to complete the task of grasping objects. The design of this action is also derived from insects. When insects are eating, the first two legs are used to grab food, and the back four legs are used to keep stability. This design is beneficial for rescue missions, which other types of robots cannot achieve.

3. Methods

3.1 Hexapod (CORIN) robot model

As it is shown in the Figure 2, there are five parts in the CORIN robot model: World Setting, Body and Ground, CORIN robot leg components, Force with the ground and Inverse Kinematic part. The first four parts belong to the CORIN physical modeling which will be introduced in this section and the Inverse Kinematic will be detailed in section 3.2.

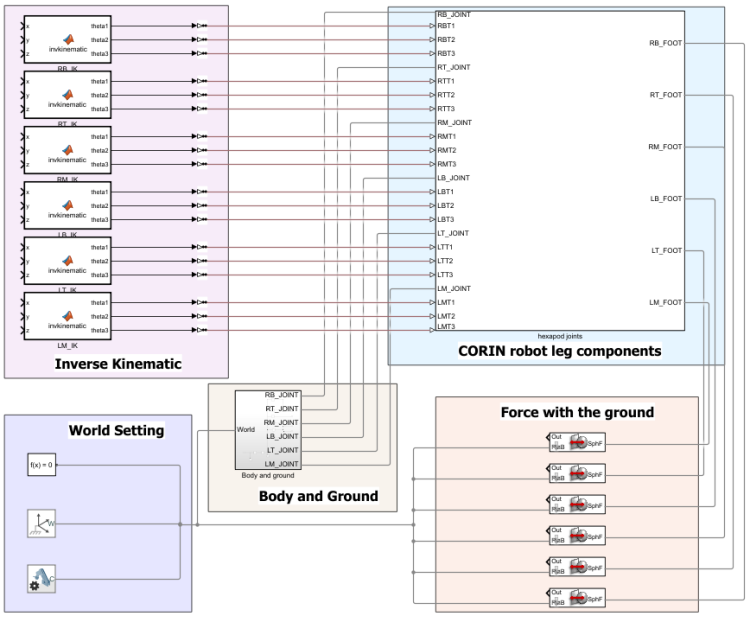

Figure 2: Overview structure of the CORIN robot model in Simulink.

The role of the World Setting part is to establish a physical simulation environment for the CORIN hexapod robot simulations. Defined the size of the ground model, the position of the CORIN robot body model relative to the ground and the position of the six coxa revolute joints relative to the body model are the function of the Body and Ground part. In CORIN Robot Leg Components part, connected the four components of each leg of the hexapod robot to the body part of the hexapod robot, and defined the positions of the remaining two rotation axes of each leg. The last part of CORIN physical modeling is the Force With The Ground, which is to define the contact force between the robot and the ground.

In order to set up a 3D model in the Simulink, a new Simscape Multibody model should be used. Three critical components of the world frame block, mechanism configuration block and solver configuration block that make up the World Setting part of the CORIN robot model are provided in the Multibody model.

Controlling the physical simulation environment solver is the function of the solver configuration block. The world frame block is used to define the origin of the world in the physical simulation space

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

and the mechanism configuration block is applied to define critical parameters such as gravity in the physical simulation space, which initial setting is $9.8m/s^2$ in the -z axis direction. Connect these three blocks together to form a complete physical simulation environment.

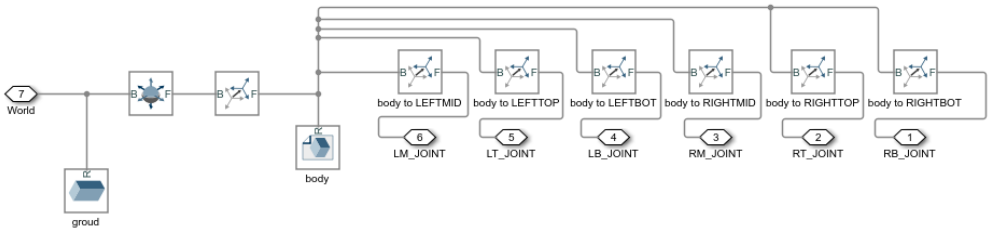

Figure 3: The detail of the Body and Ground subsystem.

The subsystem of the Body and Ground part are shown in the Figure 3 and there are four different kinds of blocks in this subsystem. The first block connected to the World Setting part is brick solid block, so it will generate a solid with the world coordinate origin as the center, and the parameters of this block are set to 10m×1m×0.1m (L×W×H) as the ground in the physical simulation space. In order to ensure that the simulated robot can make translation and rotation movements in three different dimensions relative to the ground, the 6-DOF Joint block is connected between the CORIN robot model and the ground model. Then, a rigid transform block, which function is to define the transformation between two different coordinate frames, is used to set the distance between the coordinate frame of the CORIN robot body model and the center of the ground. The CORIN body center frame is 0.25m translation of the ground center to the z-axis direction.

After confirming the relative position of the center of the CORIN robot body, import the STL file of the CORIN body through file solid block. The next step is to define the six rotation center frames on the body model by using rigid transform blocks. Based on teaching materials and parameter settings on the Simulink, the relevant specifications of the CORIN robot are recorded in the Table 1. According to the parameters provided in the Table 1, the relationship between the coordinates of each rotation center and the center of the body can be calculated. For example, the translation of the center of rotation on the robot left-top corner relative to the center of the body is (0.115, 0.09, 0).

Table 1: CORIN hexapod robot specifications [44].

Description	Data
Dimensions (nominal stance)	0.55m×0.622m×0.184m (L×W×H)
Link lengths and weights	Coxa (L1): 0.060m, 0.025kg
	Femur (L2): 0.150m, 0.280kg
	Tibia (L3): 0.150m, 0.090kg
End effector (foot) size	A sphere with a diameter of 0.032m
Joint range of motion	q1=±48deg, q2=±126deg, q3=±160deg
Leg orientation offset (front and rear legs)	q_{off}=±45deg
Distance between joints	Lateral: 0.18m, Longitudinal: 0.115m

The Figure 4 presents the structure of the CORIN robot leg components, after the center of rotation is determined by applying the rigid transform block, a revolute joint block is connected to the two links. This revolute joint block can make the following link rotate with the z-axis direction of the previously set rotation center as the rotation axis, and you can set the sensor function, drive method, etc. in the block properties list. The key point to make the revolute joint can rotate is to keep the follower frame of the rigid transform blocks before and after the joint the same.

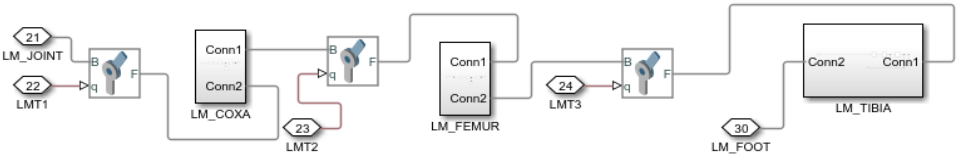

Figure 4: The structure of the CORIN robot leg components subsystem.

In this part, in order to control the joint rotation, the actuation properties in the revolute joint block need to be set, where the motion is set to be provided by input, and the torque is set to be automatically

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

computed. Three subsystems related to the import and connection of the robot leg structure model are very similar.

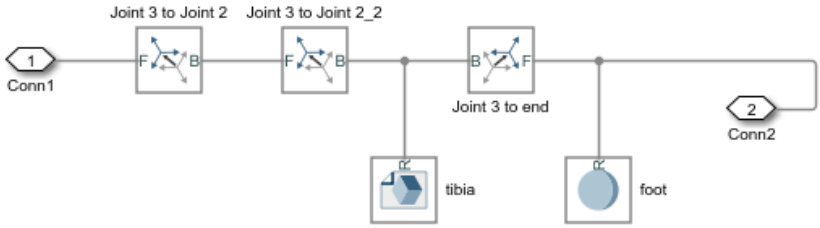

Figure 5: The detail of the CORIN robot tibia subsystem.

Therefore, the more complex CORIN tibia subsystem will be introduced below, and the CORIN tibia subsystem is presented in the Figure 5. As it is mentioned before, the position of the follower frame should be the same as the previously defined rotation center frame. Therefore, the first rigid transform block is turned over here to keep the follower frame fixed, and the tibia model is correctly connected to other robot leg structures by adjusting the rotation and translation of the base frame. The reason for using two rigid transform blocks here is that the correct placement of the robot model may require one or two rotations in different directions, and one rigid transform block can only rotate in one direction.

After connecting the CORIN tibia model, the end effector (foot) model is the only missing part of the CORIN robot. According to the Table 1, the CORIN robot end effector is a sphere with a diameter of 0.032m. By using the rigid transform block to determine the position of the center of the sphere relative to the center of the CORIN tibia model, the spherical solid block is applied as the robot foot to complete the whole CORIN robot 3D model.

When the first three parts were finished, it means the 3D model of the CORIN robot has been established. The fourth part, named force with the ground, contains only one block that is sphere to plane force block. Link the SphF port of this block to the end effector model of the CORIN robot, and connect the PlaB port to the ground model. Applying this block can generate interaction forces between the set sphere and the set plane, ensuring that the previously constructed CORIN hexapod robot can stand stably on the ground instead of crossing the ground.

After implementing the fourth part, it means that the physical modeling of the CORIN robot has been completed. However, now the only way to control the movement of the CORIN robot is constantly changing the angle of each joint of the CORIN robot model, which is inconvenient and hard to know the robot position. The next step is to calculate and install the inverse kinematics of the CORIN hexapod robot so that the robot can be moved by setting different goal points for each leg.

3.2 Kinematics of the CORIN robot leg

As it is mentioned in the Literature Review section, each of the CORIN hexapod robot leg contains three revolute joints, six Degree of Freedom (DoF). Therefore, three different coordinate systems are set up to match three revolute joints. The parameter setting of the links and rotation angle for each joint are shown in the Table 1. The Simulink-based single leg structure of the CORIN hexapod robot is shown in the Figure 6.

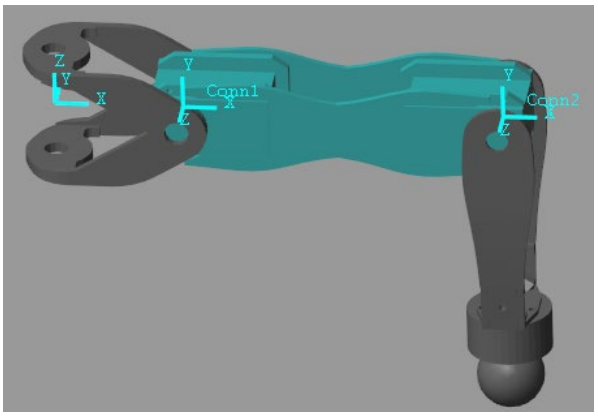

Figure 6: The simulated single leg structure of the CORIN robot.

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

In the Figure 6, the direction of the world coordinate frame is the same as the first coordinate frame on the left, and the direction of the robot foot (end effector) coordinate system is the same as the previous coordinate system which is the third coordinate system on the left. Based on the single leg structure of the CORIN robot and the relationship between three joints coordinate systems provided in the Figure 6, the Denavit-Hartenberg (D-H) parameters method, which is a standard method for robotics [45], is used to calculate the CORIN robot leg kinematics. The D-H parameters of the CORIN robot leg are presented in the Table 2.

Table 2: The D-H parameters of the CORIN robot leg.

k	a_{k-1}	α_{k-1}	d_k	θ_k
1	L1	90°	0	q1
2	L2	0	0	q2
3	L3	0	0	q3

Where d_k and θ_k are the variables of prismatic joint distance and revolute joint angle. The value of a_{k-1} depends on the distance between z_k and z_{k-1} along the axis x_k. The angle between z_k and z_{k-1} about the axis x_k decides the data of α_{k-1}. According to Equation 1 and the parameters in Table 2, three different transformation matrices ${}^{0}_{1}T$, ${}^{1}_{2}T$ and ${}^{2}_{3}T$ are obtained.

$$ {}^{k-1}_{k}T = \begin{bmatrix} \cos\theta_k & -\sin\theta_k\cos\alpha_k & \sin\theta_k\sin\alpha_k & a_k\cos\theta_k \\ \sin\theta_k & \cos\theta_k\cos\alpha_k & -\cos\theta_k\sin\alpha_k & a_k\sin\theta_k \\ 0 & \sin\alpha_k & \cos\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

$$ {}^{0}_{3}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

After calculating three different transformation matrices, using the characteristics of the homogeneous transformation matrix, which is shown in the Equation 2, the transformation matrix from coordinate system 0 to coordinate system 3 can be computed. The positive kinematics of the CORIN robot leg can be found directly from the new transformation matrix. The positive kinematics of the CORIN robot leg refers to the calculation of the specific position of the end effector (foot) based on the different rotation angles of the three revolute joints.

$$ \begin{bmatrix} \cdots & p_x \\ \cdots & p_y \\ \cdots & p_z \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cdots & \cos q1\,(L1 + L3\cos(q1+q2) + L2\cos q2) \\ \cdots & \sin q1\,(L1 + L3\cos(q1+q2) + L2\cos q2) \\ \cdots & L3\sin(q1+q2) + L2\cos q2 \\ 0 & 1 \end{bmatrix} \tag{3} $$

$$ x = \cos q1\,(L1 + L3\cos(q1+q2) + L2\cos q2) \tag{4} $$

$$ y = \sin q1\,(L1 + L3\cos(q1+q2) + L2\cos q2) \tag{5} $$

$$ z = L3\sin(q1+q2) + L2\cos q2 \tag{6} $$

The Euqation 3 shows the result of the fourth column of the transformation matrix ${}^{0}_{3}T$, the elements in the Equation 3 are the translation part of the CORIN robot, and the element in each row of this column are sorted out to get the Equation 4, 5 and 6, which are the position kinematics of the CORIN robot. The next step is to work out the inverse kinematics of the CORIN robot. When the target positon of the end effector that needs to be reached is known, the calculation of the rotation angle of the CORIN robot's three different revolute joints by using the position information is called inverse kinematics of the robot. The rotation angle for the first joint can be computed by using the Equation 5 to divide the Equation 4 and the answer of the first rotation angle is shown in the Equation 7. In order to get the result of the q2 and q3, the transformation matrix ${}^{0}_{1}T$ should be inversed and then multiplied the matrix ${}^{0}_{3}T$, the new matrix is equal to the transformation matrix ${}^{1}_{3}T$. After getting the two matrices, only the robot translation section in the last column needs to be considered, which is presented in the Equation 9. The Equation 8 shows the inverse martix of ${}^{0}_{1}T$, which is determined by applying the maths method: $(AI) = (IA^{-1})$.

$$ q1 = arctan2(y, x) \tag{7} $$

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

$$({}^{0}_{1}T)^{-1} = \begin{bmatrix} \cos q1 & \sin q1 & 0 & -L1 \\ 0 & 0 & 1 & 0 \\ \sin q1 & -\cos q1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} \cdots & x\cos q1 + y\sin q1 - L1 \\ \cdots & z \\ \cdots & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cdots & L3\cos(q2+q3) + L2\cos q2 \\ \cdots & L3\sin(q2+q3) + L2\sin q2 \\ \cdots & 0 \\ 0 & 1 \end{bmatrix} \tag{9}$$

In the Equation 9, in order to calculate the result of the rotation angle of q2, the square of the element in the first row and the square of the element in the second row should be added, and the equation related to the cosine function of q3 can be found in the Equation 10. To simplify the result, $sin^2\theta + cos^2\theta = 1$ is given and the final answer of q3 is provided in the Equation 11. Finally, the rotation angle q2 is determined by using the second row element in the Equation 9 to divide the first row element in the Equation 9 the last column, which is shown in the Equation 12.

$$\cos q3 = \frac{((x\cos q1)+y\sin q1-L1)^2+z^2-L3^2-L2^2}{2\times L2\times L3} \tag{10}$$

$$q3 = arctan2(\sqrt{1-cos^2 q3}, cosq3) \tag{11}$$

$$q2 = \arctan\left(\frac{z-\frac{L3\sin q3}{L2+L3\sin q3}\times(x\cos q1+y\sin q1-L1)}{z\left(\frac{L3\sin q3}{L2+L3\sin q3}\right)+(x\cos q1+y\sin q1-L1}\right) \tag{12}$$

The inverse kinematics of the CORIN robot leg for three different rotation angles are computed and shown in the Equation 7 and 10 to 12. Based on the CORIN robot inverse kinematics equations, an inverse kinematics function is designed and the function is then imported into the MATLAB function block in the Simulink. The three angle output nodes of the inverse kinematics function block are connected with the rotation angles of the three revolute joint points of the CORIN robot model. Now, the movement of the CORIN robot or other advanced motions can be easily controlled by designing the position of each foothold of the CORIN robot. Section 3.3 will introduce the overview design of the CORIN robot global path planning and introduce in detail the CORIN robot foothold planning algorithm implemented on the Matlab.

3.3 CORIN robot foothold planning

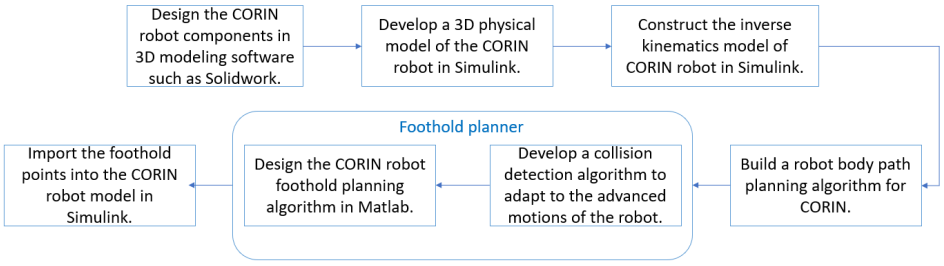

Figure 7: The overview block diagram of the CORIN robot global path planning.

As it is shown in the Figure 7, the process of developing a global path planning can be divided into six steps, the first three steps are about the CORIN robot 3D modeling, which is discussed in detail in the section 3.1 and 3.2 before. The fourth step is to establish the path planning algorithm of the CORIN robot body. This part will be completed and introduced by my colleague from the project group. Therefore, the remaining two steps will be discussed in detail in this section. As it is mentioned in Chapter 1, this is the first time that the CORIN robot has been programmed and modeled in the Matlab and the Simulink, so the foothold planner will be designed in the 2D environment first.

Developing a collision detection algorithm that can be used for advanced motions of CORIN robot in the Matlab is part of the fifth step. Different from the obstacle avoidance design of general wheeled robots, the CORIN hexapod robot can work well in complex environments and can also do some advanced motions that wheeled robots cannot. For example, a wall is an unusable obstacle for a wheeled robot, but for the CORIN hexapod robot, the area on the wall can also be selected for foothold, and then the CORIN robot can use the wall to walk through some narrow terrain. Moreover, if the CORIN hexapod robot is taken as a whole, such as a circle with the distance from the center of the body to the farthest

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

foothold point as the radius and the center of the body as the center point for collision detection, it will reduce the flexibility of the hexapod robot foothold selection. Therefore, for the CORIN hexapod robot, it is a good choice to detect the distance between multiple feature points on the robot structure and obstacles.

In this case, three feature points are set on each of the CORIN robot leg. The first point is set on the end effector of the robot, and the second point is determined on the rotation joint between femur link and tibia link. The third point is on the rotating joint that connects the CORIN robot body to the robot coxa link. Similarly, three different variables are used to distinguish the collision detection situations of different feature points.

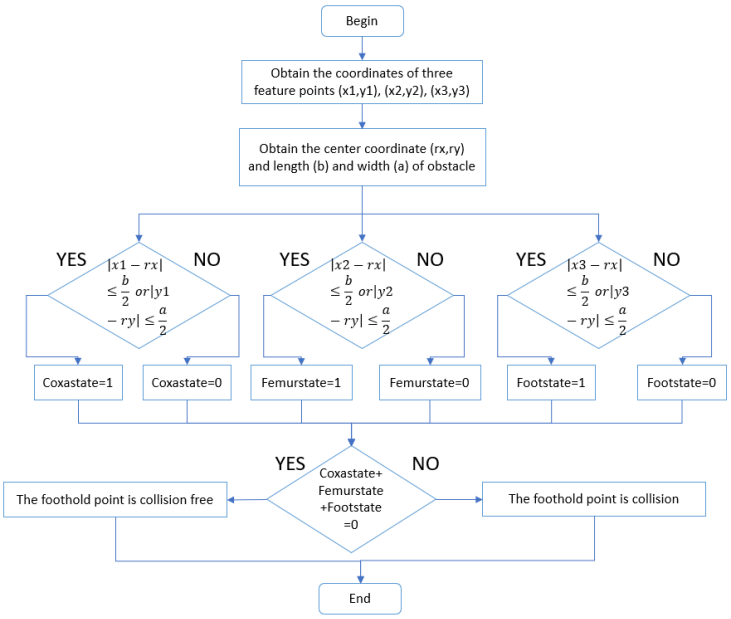

Figure 8: Flow chart of the CORIN robot collision detection algorithm for single leg.

The Figure 8 shows the flow chart of the collision detection algorithm of the CORIN robot. The first step of the algorithm is to calculate the coordinates of the three feature points set on each leg based on the coordinates of the current foothold point. The second step is to upload the obstacles information such as the center coordinates of the known obstacles and the length and width of the obstacles to the collision detection algorithm. Then, the distance between each feature point and the obstacle will be used to judge whether the feature point is inside the obstacle or not, which is the third step of the algorithm. If the feature point is on the obstacle or inside the obstacle, the value of the variable that is used to show the collision state of the feature point will become to 1, or the value of the variable will remain 0 when there is no collision. There are many advantages to using different variables to represent the state of different feature points. It can quickly determine where the collision occurs based on the values of different variables, which can effectively improve the robot motion design. When developing the advanced motions of the CORIN hexapod robot, such as using the robot to do the chimney walking, the feet of the robot will directly touch obstacles. The variables that are used to detect the collision state of the feature points of the robot feet can be ignored when judging the collision state of the robot to ensure that the robot can do chimney walking and will not affect the collision detection of normal walking. The last step is to classify the selected foothold points according to three variables values. When any of the three variables is not equal to 0, it means that the robot touch an obstacle, and the foothold point is classified as a collision point. Other selected foothold points will be considered as the collision-free points.

According to the Figure 7, another part of the foothold planner is foothold planning algorithm. Two critical parts of the foothold planning algorithm are the CORIN robot leg workspace and the cost function of the algorithm. According to the CORIN hexapod robot specifications shown in the Table 1, the maximum robot leg 2D workspace can be calculated. The limit range of the rotation angle q1 of the CORIN robot first revolute joint related to the xy plane is $\pm 48^{\circ}$.

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

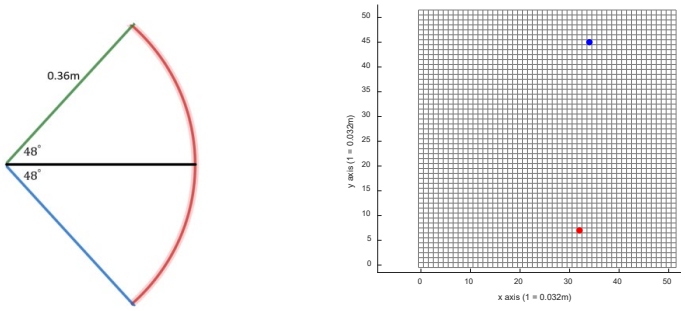

Figure 9: The maximum 2D workspace of the CORIN leg and the grid map of foothold planner.

In the ideal case of ignoring the volume of the CORIN robot links, it is assumed that the nearest robot foothold point can be the same position as the body, and the farthest point can reach the length of entire the leg. The picture on the left in the Figure 9 shows that the maximum workspace of the CORIN robot foothold selection in the ideal case. The ideal workspace is in a fan shape with a radius of 0.36m and an angle of 96 degrees. Since the CORIN robot movement may have errors such as foot sliding, the size of the hexapod robot foothold point is set to a square with a side length of 0.032m, which is the same length as the diameter of the robot feet. The grid map on the right side of the Figure 9 is constructed on the Matlab based on the size of the robot foothold point. The resolution of each cell is 0.032m×0.032m, which is helpful for coordinate conversion.

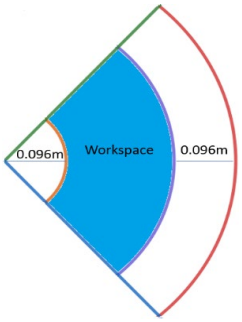

Figure 10: Actual 2D workspace of the CORIN robot leg.

However, since the leg components of the robot have their own self-collision limitations, footholds that are too far or too close will seriously affect the stability of the robot. The actual foothold selection range of the CORIN robot cannot reach the maximum workspace shown in the Figure 9. Therefore, the fan-shaped area with the radius of three footholds length on the left side of the ideal workspace is deleted, and the area in white with the distance of three footholds length on the right side of the ideal workspace is deleted. The real 2D workspace of the CORIN robot leg is shown in blue area in the Figure 10. According to calculations, a total of 19 footholds fall entirely within the actual workspace. Each time a new foothold point is selected, the value of the cost function of the 19 alternative footholds needs to be considered.

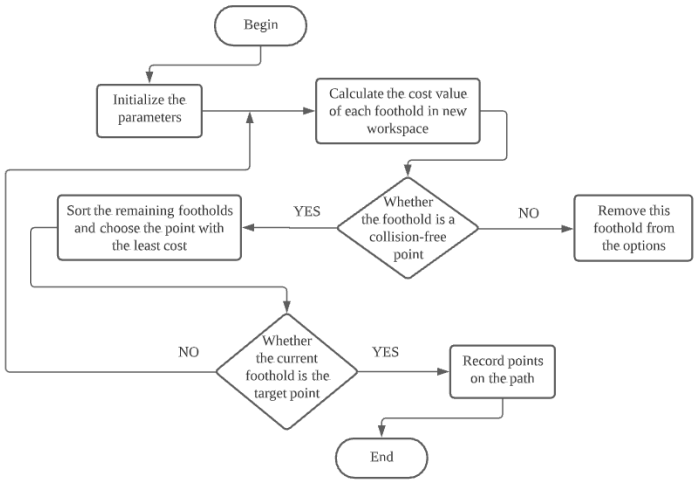

Figure 11: The flow chart of the CORIN robot foothold path planning algorithm.

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

After the workspace is determined, the next important thing that needs to be determined is the cost function. In the current situation, the cost function is set to $cost = \omega_d c_d + \omega_g c_g + \omega_c c_c$, where ω_d, ω_g and ω_c are the weight parameters of the cost function, c_d is the cost of the foothold path distance, c_g is the cost of the distance between the goal point and current point and c_c is the leg collision cost. In subsequent projects, various kinds of costs will be considered, such as the cost of distance to the body global planning path. This part will be introduced more in Chapter 4.

The flow chart of the algorithm for the foothold path planning is shown in the Figure 11. First of all, initialized the required variables and set the current foothold as the starting point. The second step is to calculate the current workspace according to the current robot position, and calculate the cost value of 19 optional footholds in the workspace. It should be noted that the workspace here is not determined based on the current foothold position like the traditional search algorithm. Instead, it is determined by the current position where the robot legs connect with the body. This is because the workspace that mentioned before is based on the first revolute joints. Then the collision status of each optional foothold is calculated according to the collision detection algorithm. The collision-free foothold points will be kept, and the foothold points that will collide with obstacles will be deleted. The fourth step is to sort the remaining available footholds in order of cost value from small to large, and select the foothold with the smallest cost value as the next foothold, and the robot legs will move to this foothold point. Set the new foothold as the current point and compare whether the current point is the same as the target point, which is the fifth step of the foothold planning algorithm. If the current point and the target point are the same, it means that the CORIN robot has reached the target position. If the current point and the target point are not the same, return to the second step of the algorithm, find a new workspace based on the new current point and new body and leg based point, and then repeat the subsequent steps until the CORIN robot reaches to the target point. The last step is to record the robot foothold path from the starting point to the target point. These foothold points can be transformed into foothold points in the CORIN robot simulation environment according to specific mathematical transformations, and then applied to the CORIN robot Simulink-based model.

4. Results and discussion

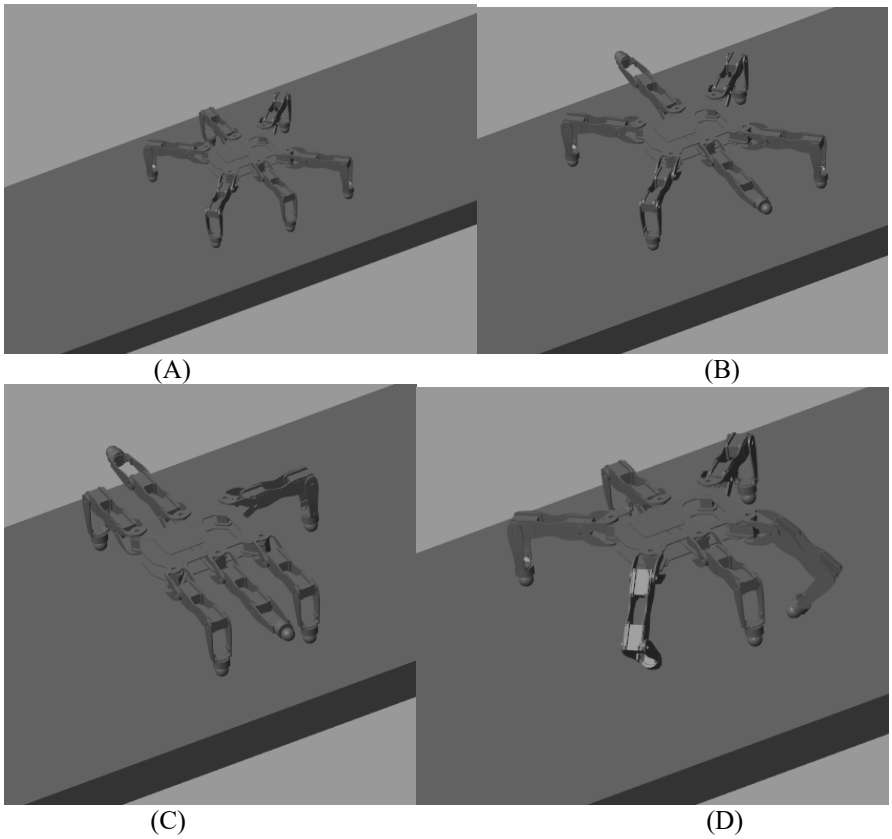

Figure 12: Four different kinds of combination of rotation angles of the CORIN robot.

Four different kinds of combination of rotation angles of the CORIN hexapod robot is shown in the

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

Figure 12. Picture A presents the standard state of the CORIN robot. Its two front legs will turn forward 45 degrees, and its rear two legs will turn back 45 degrees due to the leg orientation offset described in the Table 1. The hexapod robot in picture B keeps the two front legs and two rear legs motionless, and lifts the two middle legs to be parallel to the ground. Its motions are consistent with the input parameters, which the angle of the third revolute joint of two middle legs are set to 90 degrees.

According to the coordinate systems set in the Figure 6, the robot in picture C is based on the robot motion in picture B, and then the first revolute joint of the left rear leg and the front two legs are rotated to -45 degrees, as well as the first revolute joint of the right rear leg is turned to the 45 degrees, which perform the same as the parameters setting. In the last picture, only the second revolute joint in both the right front leg and the right rear leg are set to rotate -45 degrees, which is the same as the hexapod robot motion shown in the picture D. Therefore, it can be concluded that a complete CORIN robot physical model controlled by setting three angles of different revolute joints is developed.

Table 3: Test results of Kinematics of CORIN Robot Leg.

Setting angles			**Positive kinematics (positions)**			**Inverse kinematics (angles)**		
q1 (deg)	**q2 (deg)**	**q3 (deg)**	**X (m)**	**Y (m)**	**Z (m)**	**Theta1 (deg)**	**Theta2 (deg)**	**Theta3 (deg)**
30	0	0	0.3118	0.1800	0	30	0	0
10	15	-20	0.3489	0.0615	0.0257	10	15	-20
20	25	40	0.2437	0.0887	0.1993	20	65	-40
20	65	-40	0.2437	0.0887	0.1993	20	65	-40
-20	-25	-40	0.2437	-0.0887	-0.1993	-20	-25	-40

The Table 3 shows results of small test of both positive kinematics and inverse kinematics for CORIN hexapod robot. The first three columns in the Table 3 indicate the set angle values of three revolute joints. The three middle columns are used to record the three different positions of x,y,z, which calculated by applying the CORIN robot leg positive kinematics equation determined in section 3.2. The last three columns are applying the results of the positions of the three middle columns to calculate rotation angles of the three revolute joints by using the CORIN robot inverse kinematics equation determined in section 3.2. In order from top to bottom, the angle values set in advance in the remaining four groups of tests except for the third group of test are the same as the angles that computed by applying the inverse kinematics, which shows both the positive kinematics and inverse kinematics may be correct.

However, the third test presents the set angles are different from the angles calculated by robot inverse kinematics. Both the angles of the second and the third revolute joint are different from the setting angles. In the comparison of the third test with the fourth test, it can be found that though the setting angles q2 and q3 are different, the positions of x,y,z are all the same. But the test with a negative q3 value can make the second and third revolute joint angles that are computed by the inverse kinematics the same as the setting values. As it is mentioned in section 3.2, the coordinate system of the second revolute joint is in the same direction as the third revolute joint. Therefore, two possible solutions can be obtained for one position, which is the classic ‘elbow down’ and ‘elbow up’ problem. Due to the stability of the hexapod robot, ‘elbow up’ solution is selected, which means the second revolute joint angle should be larger or equal to zero and the third revolute joint angle should be smaller or equal to zero. This is why the negative value of q3 in the Table 3 can get the correct angle relationship.

In general, though the angle relationship obtained by applying the inverse kinematics in the third experiment does not match the setting relationship, the answer of the angle values are acceptable because the setting angle values will make the hexapod robot unstable. It can be found that the kinematics of the CORIN hexapod robot are correct.

The Figure 13 and the Figure 14 present the different foothold selections for different target points by applying the foothold path planning algorithm. The blue grids are the selection of the foothold points, and the yellow line shows the foothold path from the starting point to the target point. 19 optional footholds can be selected in the workspace. Since the robot leg has a large workspace, the choice of footholds should be discrete.

When reviewing the literature of global path planning for legged robots, the initial cost function is set as $cost = \omega_d c_d + \omega_g c_g + \omega_c c_c + \omega_h c_h + \omega_s c_s$, the first three costs of the cost function are the same as those mentioned in section 3.3. The fourth element in the cost function is terrain height cost and the last section in the cost function is the cost of robot stability. The terrain height cost is deleted due to the use of the 2D grid map.

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

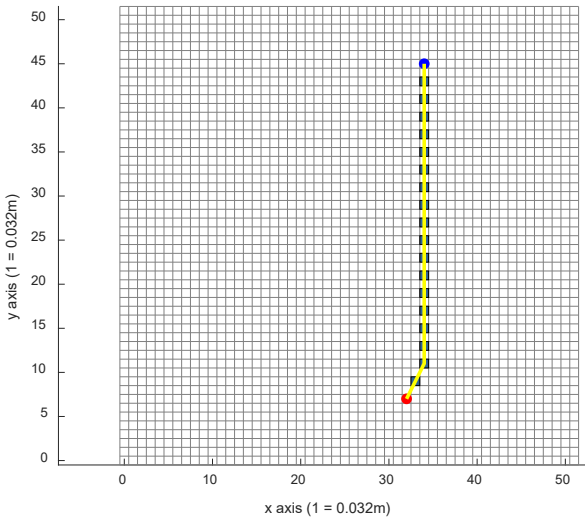

Figure 13: The result of foothold path planning algorithm for right front leg.

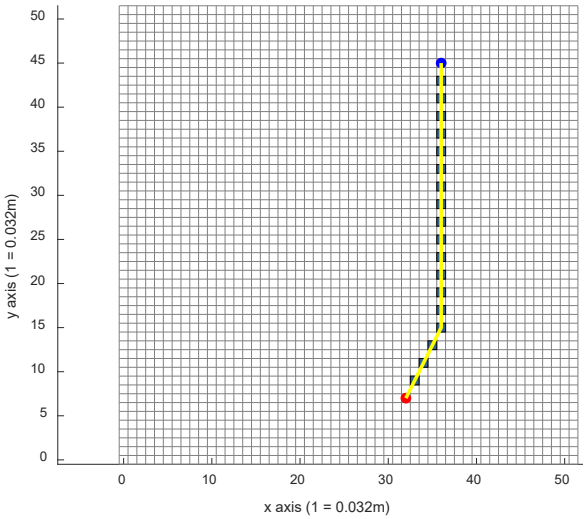

Figure 14: The result of foothold path planning algorithm for right front leg for different goal.

When the hexapod robot maintains a standard posture, the center of gravity (COG) of the robot is equivalent to the middle of a hexagon. When a certain leg of the robot is lifted up and will move to a new foothold point, the balanced shape will be changed from a hexagon to a pentagon and the movement of COG depends on the distance between the new foothold and the robot current COG position. If the foothold is far away from the robot, the COG of the hexapod robot will move out of the pentagon formed by the other five legs, which means the hexapod robot loses its static stability. On the other hand, if the distance between the new foothold point and the center of the robot is moderate, and the COG of the robot can still be maintained in the pentagon, the robot still has static stability. The shortest distance between COG with one side of the pentagon is the stability margin of the hexapod robot. Due to the lack of relevant sensors to measure the position of the COG, the cost of robot stability will also be applied in the future.

5. Conclusions

The aim of this paper is to develop a highly visualized hexapod robot (CORIN) 3D model based on the Simulink, and then design and implement the collision detection algorithm and the foothold planning algorithm of the hexapod robot. In order to build the CORIN robot 3D model in the Simulink, the parameter specifications and structure of the CORIN robot were reviewed, and kinematics of the robots as well as the method of building the model in the Simulink was presented. A complete CORIN hexapod robot model that can be controlled by specifying the position of the end effector or setting three different revolute joint angles has been designed.

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

For the design of the foothold global path planning algorithm, after a comprehensive literature review on the global path planning of legged robots, a foothold planning algorithm based on a grid map and a distance-based collision detection algorithm has been developed. Through the research of the specifications of the CORIN robot legs, the workspace of the robot legs was calculated. The appropriate footholds were selected in the workspace for comparison, and then the optimal foothold was determined based on the path length, the distance between the current point and the target point, and whether it collides or not. The algorithm was executed repeatedly until the target point was reached.

References

[1] Birk, A., & Carpin, S. (2006). Rescue robotics—a crucial milestone on the road to autonomous systems. Advanced Robotics, 20(5), 595-605.
[2] Altendorfer, R., Moore, N., Komsuoglu, H., Buehler, M., Brown, H. B., McMordie, D., ... & Koditschek, D. E. (2001). Rhex: A biologically inspired hexapod runner. Autonomous Robots, 11(3), 207-213.
[3] Zhao, F., Ma, Y., & Sun, Y. (2017). Application and standardization trend of maintenance and inspection robot (MIR) in nuclear power station. DEStech Transactions on Engineering and Technology Research, (ismii).
[4] Yamauchi, B. M. (2004, September). PackBot: a versatile platform for military robotics. In Unmanned ground vehicle technology VI (Vol. 5422, pp. 228-237). International Society for Optics and Photonics.
[5] Rubio, F., Valero, F., & Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. International Journal of Advanced Robotic Systems, 16(2), 1729881419839596.
[6] Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). Introduction to autonomous mobile robots. MIT press.
[7] Gage, D. W. (1995). UGV history 101: A brief history of Unmanned Ground Vehicle (UGV) development efforts. NAVAL COMMAND CONTROL AND OCEAN SURVEILLANCE CENTER RDT AND E DIV SAN DIEGO CA.
[8] Kamel, M. A., & Zhang, Y. M. (2014). Developments and challenges in wheeled mobile robot control. In International Conference on Intelligent Unmanned Systems (ICIUS).
[9] Shiroma, N., Chiu, Y. H., Min, Z., Kawabuchi, I., & Matsuno, F. (2006, October). Development and control of a high maneuverability wheeled robot with variable-structure functionality. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4000-4005). IEEE.
[10] Cheah, W., Khalili, H. H., Watson, S., Green, P., & Lennox, B. (2018, October). Grid-based motion planning using advanced motions for hexapod robots. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 3573-3578). IEEE.
[11] Todd, D. J. (2013). Walking machines: an introduction to legged robots. Springer Science & Business Media.
[12] Online-sciences.com. (2019). Legged robots features, types, uses, advantages and disadvantages. [online] Available at: https://www.online-sciences.com/robotics/legged-robots-features-types-uses-advantages-and-disadvantages/ [Accessed 22 Aug. 2021]
[13] Patle, B. K., Pandey, A., Parhi, D. R. K., & Jagadeesh, A. (2019). A review: On path planning strategies for navigation of mobile robot. Defence Technology, 15(4), 582-606.
[14] Kanayama, Y. J., & Hartman, B. I. (1997). Smooth Local-Path Planning for Autonomous Vehicles1. The International Journal of Robotics Research, 16(3), 263–284.
[15] Xin, D., Hua-hua, C., & Wei-kang, G. (2005). Neural network and genetic algorithm based global path planning in a static environment. Journal of Zhejiang University-Science A, 6(6), 549-554.
[16] Masehian, E., & Katebi, Y. (2007). Robot motion planning in dynamic environments with moving obstacles and target. International Journal of Mechanical Systems Science and Engineering, 1(1), 20-25.
[17] Yung, N. H., & Ye, C. (1999). An intelligent mobile vehicle navigator based on fuzzy logic and reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 29(2), 314-321.
[18] Kostavelis, I., Nalpantidis, L., Boukas, E., Rodrigalvarez, M. A., Stamoulias, I., Lentaris, G., ... & Gasteratos, A. (2014). Spartan: Developing a vision system for future autonomous space exploration robots. Journal of Field Robotics, 31(1), 107-140.
[19] Azkarate, M., Gerdes, L., Joudrier, L., & Pérez-del-Pulgar, C. J. (2020, May). A GNC Architecture for Planetary Rovers with Autonomous Navigation. In 2020 IEEE International Conference on Robotics

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

and Automation (ICRA) (pp. 3003-3009). IEEE.
[20] Trăsnea, B., Gineric ă C., Zaha, M., Măceşanu, G., Pozna, C., & Grigorescu, S. (2021). OctoPath: An OcTree-Based Self-Supervised Learning Approach to Local Trajectory Planning for Mobile Robots. Sensors, 21(11), 3606.
[21] Yang, X. (2021). Slam and navigation of indoor robot based on ROS and lidar. In Journal of Physics: Conference Series (Vol. 1748, No. 2, p. 022038). IOP Publishing.
[22] Bounini, F., Gingras, D., Pollart, H., & Gruyer, D. (2017, June). Modified artificial potential field method for online path planning applications. In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 180-185). IEEE.
[23] Henkel, C., Bubeck, A., & Xu, W. (2016). Energy efficient dynamic window approach for local path planning in mobile service robotics. IFAC-PapersOnLine, 49(15), 32-37.
[24] Kazem, B. I., Hamad, A. H., & Mozael, M. M. (2010). Modified vector field histogram with a neural network learning model for mobile robot path planning and obstacle avoidance. Int. J. Adv. Comp. Techn., 2(5), 166-173.
[25] Alexopoulos, C., & Griffin, P. M. (1992). Path planning for a mobile robot. IEEE Transactions on systems, man, and cybernetics, 22(2), 318-322.
[26] Samadi, M., & Othman, M. F. (2013, December). Global path planning for autonomous mobile robot using genetic algorithm. In 2013 International Conference on Signal-Image Technology & Internet-Based Systems (pp. 726-730). IEEE.
[27] Qu, H., Xing, K., & Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing, 120, 509-517.
[28] Chaari, I., Koubaa, A., Bennaceur, H., Ammar, A., Alajlan, M., & Youssef, H. (2017). Design and performance analysis of global path planning techniques for autonomous mobile robots in grid environments. International Journal of Advanced Robotic Systems, 14(2), 1729881416663663.
[29] Che, H., Wu, Z., Kang, R., & Yun, C. (2016, July). Global path planning for explosion-proof robot based on improved ant colony optimization. In 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS) (pp. 36-40). IEEE.
[30] Čížek, P., Masri, D., & Faigl, J. (2017, September). Foothold placement planning with a hexapod crawling robot. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4096-4101). IEEE.
[31] Belter, D., Łabęcki, P., & Skrzypczyński, P. (2016). Adaptive motion planning for autonomous rough terrain traversal with a walking robot. Journal of Field Robotics, 33(3), 337-370.
[32] Winkler, A. W., Mastalli, C., Havoutis, I., Focchi, M., Caldwell, D. G., & Semini, C. (2015, May). Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain. In 2015 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5148-5154). IEEE.
[33] Belter, D. (2019). Efficient modeling and evaluation of constraints in path planning for multi-legged walking robots. IEEE Access, 7, 107845-107862.
[34] Mastalli, C., Focchi, M., Havoutis, I., Radulescu, A., Calinon, S., Buchli, J., ... & Semini, C. (2017, May). Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1096-1103). IEEE.
[35] Wermelinger, M., Fankhauser, P., Diethelm, R., Krüsi, P., Siegwart, R., & Hutter, M. (2016, October). Navigation planning for legged robots in challenging terrain. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1184-1189). IEEE.
[36] Delcomyn, F., & Nelson, M. E. (2000). Architectures for a biomimetic hexapod robot. Robotics and Autonomous Systems, 30(1-2), 5-15.
[37] Sun, C., Yuan, M., Li, F., Yang, Z., & Ding, X. (2019, February). Design and Simulation Analysis of Hexapod Bionic Spider Robot. In Journal of Physics: Conference Series (Vol. 1168, No. 2, p. 022094). IOP Publishing.
[38] Cheah, W., Khalili, H. H., Arvin, F., Green, P., Watson, S., & Lennox, B. (2019). Advanced motions for hexapods. International Journal of Advanced Robotic Systems, 16(2), 1729881419841537.
[39] Saranli, U., Buehler, M., & Koditschek, D. E. (2001). RHex: A simple and highly mobile hexapod robot. The International Journal of Robotics Research, 20(7), 616-631.
[40] Chen, Z., & Gao, F. (2019). Time-optimal trajectory planning method for six-legged robots under actuator constraints. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233(14), 4990-5002.
[41] GAINA, M. G. (2021). Practical application using the hexapod robot for interior inspection of a Boeing 737-300 aircraft fuel tank.
[42] Kang, X. U., Shoukun, W. A. N. G., Junzheng, W. A. N. G., Xiuwen, W. A. N. G., Zhihua, C. H. E. N., & Jinge, S. I. (2021). High-adaption locomotion with stable robot body for planetary exploration

ISSN 2616-5775 Vol. 5, Issue 3: 45-61, DOI: 10.25236/AJCIS.2022.050307

robot carrying potential instruments on unstructured terrain. Chinese Journal of Aeronautics, 34(5), 652-665.

[43] Deng, H., Xin, G., Zhong, G., & Mistry, M. (2017). Gait and trajectory rolling planning and control of hexapod robots for disaster rescue applications. Robotics and Autonomous Systems, 95, 13-24.

[44] Cheah, W., Khalili, H. H., Arvin, F., Green, P., Watson, S., & Lennox, B. (2019). Advanced motions for hexapods. International Journal of Advanced Robotic Systems, 16(2), 1729881419841537.

[45] Corke, P. I. (2007). A simple and systematic approach to assigning Denavit–Hartenberg parameters. IEEE transactions on robotics, 23(3), 590-594.