

仿蟹六足机器人步态仿真

吴梓宁 陆瑾萱 解凌一 孙健玮

摘要: 研究六足仿真螃蟹机器人的行进与转向等行为特性, 进行对真实生物螃蟹的步态分析, 并对仿生机器人进行理论分析和仿真建模。对六足螃蟹机器人的动力学和运动学模型进行了深入理论研究, 并通过应用 MATLAB 和 Webots 等高级仿真软件工具, 进一步研究了机器人的行进和转向能力。MATLAB 提供了强大的数学计算和图形绘制能力, 能够对模型进行详细的数值分析。同时, Webots 提供了真实的物理仿真环境, 为机器人提供了真实的环境去测试并优化其行为。通过理论研究与实际仿真相结合的方式, 对六足仿真螃蟹机器人的行为和性能提供了深入理解。

关键词: 仿生机器人; 六足机器人; 仿真与建模; MATLAB; WEBOTS

中图分类号: TG156

Simulation of crab-like hexapod robot based on MATLAB gait algorithm

WU Zining, LU Jinxuan, XIE Lingyi, SUN Jianwei

Abstract: This research investigates the locomotion and turning behaviors of a six-legged crab-like simulation robot, conducts a gait analysis of real crabs, and carries out theoretical analysis and simulation modeling of the biomimetic robot. A thorough theoretical study has been conducted on the dynamics and kinematics models of the six-legged crab robot. Moreover, with the application of advanced simulation software tools like MATLAB and Webots, we further studied the robot's capabilities in locomotion and turning. MATLAB provides powerful mathematical computing and graphics drawing capabilities, enabling detailed numerical analysis of the model. Meanwhile, Webots offers a realistic physical simulation environment, providing the robot a real-world environment to test and optimize its behaviors. By integrating theoretical research with practical simulation, we gain an in-depth understanding of the behaviors and performance of the six-legged crab-like simulation robot.

Key words: Biomimetic robot; six-legged robot; simulation and modeling; MATLAB; WEBOTS

0 研究动机

生物仿生学是一门研究借鉴自然生物的结构、功能和行为的学科, 而机器人技术则是研究和设计能够自主执行任务的机械装置或系统, 因此仿生机器人在机器人领域占据了重要地位。仿生机器人是指模仿生物或具备生物特点工作的机器人。在设计机器人的移动时, 我们可以想到采取轮式机器人。但要设计更多的移动方式时, 生物为机器人学提供了广阔的范本, 如六足机器人、仿生鱼机器人等。通过模仿生物运动特征, 机器人可以适应多种环境: 如机器梭子鱼, 其模仿鱼类摆尾的运动方式, 将尾部设计为弹簧状的锥形玻璃纤维线圈, 兼顾了坚硬和灵活的特点, 可以在水中自由游动; 如壁虎机器

人, 体型小, 行动灵活, 可在各种夹缝中行走, 对各种材料的墙面都能适应, 灵活性和运动速度可媲美生物壁虎。

在仿生机器人的研究过程中, 理论研究有助于对机器人的结构和运动性有深入的理解, 能够提供优化设计的理论基础。通过理论分析, 可以更准确地设计和调整机器人的各个部分, 提升其行走效率、稳定性和在复杂环境下的适应性, 这对于在仿生机器人的实际应用与未来发展有着非常重要的作用。近年来, 国内外诸多学者通过理论研究可以预测机器人的性能, 开发更有效的控制策略, 推动技术创新。BUEHLER 等^[1]讨论了四足机器人设计的基本原理, 并描述了其在行走、攀爬和跑步等方面的表现。ALTENDORFER 等^[2]介绍了由生物灵感开发的 RHex 六足机器人的设计和性能, 十分贴近昆虫的运动方式。螃蟹作为一种常见的海洋生物, 其具有独特的外形、动作和生物力学特性, 本文致力于研究六足仿生螃蟹机器人, 对六足的螃蟹机器人进

行理论研究并建立 Matlab 和 Webots 模型仿真，有助于推动其设计和控制的优化，预测和改善其性能，节约成本和时间，同时也有助于推动相关领域的技术创新，从而提升机器人的行走效率，稳定性和适应复杂环境的能力，这对于未来机器人技术的发展具有重要作用。

1 模型建立

1.1 理论分析

建立机器人的运动学与动力学模型，是对机器人的运动和控制进行数学描述的过程，是实现机器人的运动与行进轨迹进行判断、控制以及操作的必经之路。运动学主要描述了机器人各部分之间的几何关系，而动力学则描述了机器人各部分在力和运动的影响下的表现。RAIBERT 等^[3]介绍了多足机器人（特别是双足和四足机器人）的平衡和控制问题，包括运动学和动力学分析。SARANLI 等^[4]详细描述了 RHex 六足机器人的运动学与动力学模型，对机器人的设计有重大作用。本文对六足仿生蟹机器人进行运动学和动力学模型的建立，用这些模型预测螃蟹机器人在给定的环境和控制输入下的行进与转向等行为。

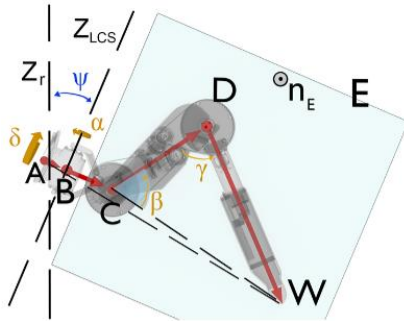


图 1 机构坐标系示意图

1.1.1 运动学分析

运动学是对物体运动时的几何、时间和空间性质进行的描述。对于六足螃蟹机器人，主要关注的是脚和机体的相对位置和速度。本文首先定义机器人的各部分的坐标系，例如机器人的身体、每个关节以及足尖。通过运动学方程，可计算出在给定关节角度下足尖位置，或者给定足尖目标位置，求解所需的关节角度，即正向与逆向运动学计算。

单足简化后的结构见图 2，下先写出其 DH 约定，规定其唯一角度变量是 θ 。

表 1 单足机构的 DH 参数

连杆	a_i	a_i	d_i	θ_i
1	a_1	0	0	θ_1^*
2	a_2	0	0	θ_2^*

* 变量

如表所示，通过将原点 o_2 放置在足臂 2 的末端来确定坐标系 $o_2 x_2 y_2 z_2$ 。则齐次变换矩阵 A 可被确定为

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

因此，T 矩阵由下式给出

$$T_1^0 = A_1$$

$$T_2^0 = A_1 A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1 c_1 + a_2 c_{12} \\ s_{12} & c_{12} & 0 & a_1 s_1 + a_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中

$$x = a_1 c_1 + a_2 c_{12}$$

$$y = a_1 s_1 + a_2 s_{12}$$

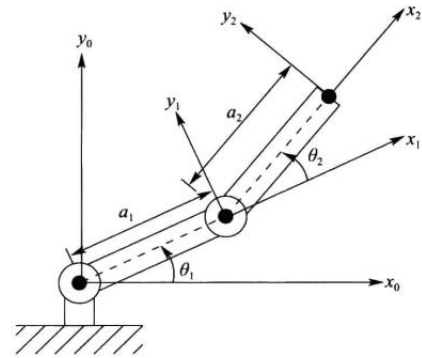


图 2 单足动力学分析图

1.1.2 动力学分析

动力学则考虑了力和扭矩的影响，运动学方程可准确地描述力与运动之间的关系，本文将引入欧拉-拉格朗日方程式（Euler-lagrange equation）。

单足简化后的结构见图 3，用

Denavit-Hartenberg 关节变量作为广义坐标。

$$v_{c1} = J_{v_{c1}} \dot{q}$$

其中，

$$J_{v_{c1}} = \begin{bmatrix} -\ell_c \sin q_1 & 0 \\ \ell_{c1} \cos q_1 & 0 \\ 0 & 0 \end{bmatrix}$$

类似，

$$v_{c2} = J_{v_{c2}} \dot{q}$$

其中，

$$J_{v_{c2}} = \begin{pmatrix} -\ell_1 \sin q_1 - \ell_{c2} \sin(q_1 + q_2) & -\ell_{c2} \sin(q_1 + q_2) \\ \ell_1 \cos q_1 + \ell_{c2} \cos(q_1 + q_2) & \ell_{c2} \cos(q_1 + q_2) \\ 0 & 0 \end{pmatrix}$$

因此，动能为

$$\frac{1}{2} m_1 v_{c1}^T v_{c1} + \frac{1}{2} m_2 v_{c2}^T v_{c2} = \frac{1}{2} \dot{q} \{ m_1 J_{v_{c1}}^T J_{v_{c1}} + m_2 J_{v_{c2}}^T J_{v_{c2}} \} \dot{q}$$

对于惯性矩阵 $D(q)$:

$$D(q) = m_1 J_{v_{c1}}^T J_{v_{c1}} + m_2 J_{v_{c2}}^T J_{v_{c2}} + \begin{pmatrix} I_1 + I_2 & I_2 \\ I_2 & I_2 \end{pmatrix}$$

其中，

$$d_{11} = m_1 \ell_{c1}^2 + m_2 (\ell_1^2 + \ell_{c2}^2 + 2\ell_1 \ell_{c2} \cos q_2) + I_1 + I_2$$

$$d_{12} = d_{21} = m_2 (\ell_{c2}^2 + \ell_1 \ell_{c2} \cos q_2) + I_2$$

$$d_{22} = m_2 \ell_{c2}^2 + I_2$$

计算 Christoffel 符号:

$$c_{111} = \frac{1}{2} \frac{\partial d_{11}}{\partial q_1} = 0$$

$$c_{121} = c_{211} = \frac{1}{2} \frac{\partial d_{11}}{\partial q_2} = -m_2 \ell_1 \ell_{c2} \sin q_2 = h$$

$$c_{221} = \frac{\partial d_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial d_{22}}{\partial q_1} = h$$

$$c_{112} = \frac{\partial d_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial d_{11}}{\partial q_2} = -h$$

$$c_{122} = c_{212} = \frac{1}{2} \frac{\partial d_{22}}{\partial q_1} = 0$$

$$c_{222} = \frac{1}{2} \frac{\partial d_{22}}{\partial q_2} = 0$$

对于每个足臂，势能是其质量乘以中立加速度以及质心高度。

$$P_1 = m_1 g \ell_{c1} \sin q_1$$

$$P_2 = m_2 g (\ell_2 \sin q_1 + \ell_{c2} \sin(q_1 + q_2))$$

总势能为

$$P = P_1 + P_2 = (m_1 \ell_{c1} + m_2 \ell_1) g \sin q_1 + m_2 \ell_{c2} g \sin(q_1 + q_2)$$

函数 g 为

$$g_1 = \frac{\partial P}{\partial q_1} = (m_1 \ell_{c1} + m_2 \ell_1) g \cos q_1 + m_2 \ell_{c2} g \cos(q_1 + q_2)$$

$$g_2 = \frac{\partial P}{\partial q_2} = m_2 \ell_{c2} g \cos(q_1 + q_2)$$

最终得到系统动力学方程

$$\begin{aligned} d_{11} \ddot{q}_1 + d_{12} \ddot{q}_2 + c_{121} \dot{q}_1 \dot{q}_2 + c_{211} \dot{q}_2 \dot{q}_1 + c_{221} \dot{q}_2^2 + g_1 &= \tau_1 \\ d_{21} \ddot{q}_1 + d_{22} \ddot{q}_2 + c_{112} \dot{q}_1^2 + g_2 &= \tau_2 \end{aligned}$$

对于矩阵 $C(q, \dot{q})$:

$$C = \begin{pmatrix} h\dot{q}_2 & h\dot{q}_2 + h\dot{q}_1 \\ -h\dot{q}_1 & 0 \end{pmatrix}$$

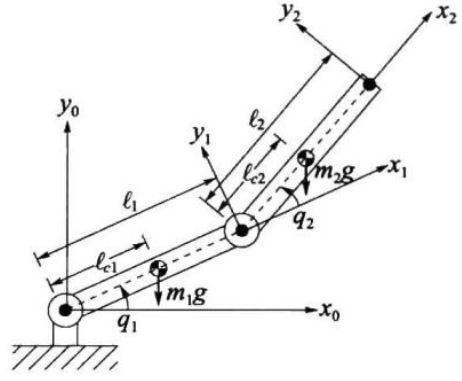


图 3 单足动力学分析图

1.2 通过 Matlab 仿真推导关节运动参数

通过上文的理论分析，我们将理论推导中得到的逆运动学方程转换成 Matlab 代码。首先进行的是单腿的模拟，我们模拟了一个周期（转一圈）的足尖运动轨迹，设置总转角 $T=2\pi$ ，并将总转角分成 100 份，再把每个转角的足尖位置写出来：

$$\begin{aligned} x &= h + a \cos(t) \\ y &= k + b \sin(t) \end{aligned}$$

t 为当前的转角， h 、 k 分别为椭圆轨迹中心的 x 和 y 坐标， a 和 b 是椭圆的半长轴和半短轴。

通过足尖位置求出两个关节的角度参数的逆运动学计算公式如下：

$$\theta_2 = -\arccos\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1 L_2}\right)$$

$$\theta_1 = \text{atan2}(y, x) - \text{atan2}(L_2 \sin \theta_2, L_1 + L_2 \cos \theta_2)$$

其中， L_1 是大腿的长度， L_2 是小腿的长度。在得到关节角度后，我们便可以计算出每个关节当前时刻的位置，并将其写成一个 1×2 的矩阵，分别表示关节的 x 和 y 坐标：

$$\begin{aligned} \text{joint1} &= [x_{\text{base}} + L_1 \cdot \cos \theta_1, y_{\text{base}} + L_1 \cdot \sin \theta_1] \\ \text{joint2} &= [\text{joint1}[1] + L_2 \cdot \cos(\theta_1 + \theta_2), \\ &\quad \text{joint1}[2] + L_2 \cdot \sin(\theta_1 + \theta_2)] \end{aligned}$$

其中， x_{base} 和 y_{base} 分别为螃蟹大腿根部的 x 和 y 坐标，即螃蟹腿这个二连杆机械臂系统的基座， $\text{joint1}[1]$ 表示关节 1 的 x 坐标， $\text{joint1}[2]$ 表示关节 1 的 y 坐标。同时，为了使足尖始终保持在地面，我们需要给两个关节的 y 坐标加入一个可能存在的补偿 dy ：

$$\begin{aligned} dy &= -\min\{joint2[2], 0\} \\ joint1[2] &= joint1[2] + dy \\ joint2[2] &= joint2[2] + dy \end{aligned}$$

当足尖在地面以下时，即 $joint2[2] < 0$ ，通过 dy 的补偿将足尖移回地面。据此，我们可以绘制出单腿的运动动画：

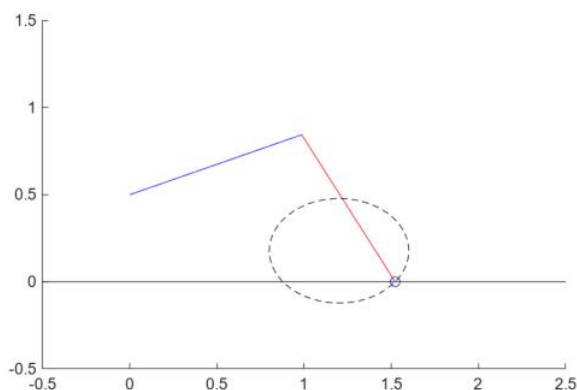


图 4 单腿运动动画（截图）

绘制双足时，仅需要在求得每个转角所对应的足尖位置时，用 $t + \pi$ 代替 t ，即：

$$\begin{aligned} x &= h + a \cos(t + \pi) \\ y &= k + b \sin(t + \pi) \end{aligned}$$

再使用同样的步骤求出另一条腿的关节角度、关节位置，即可得到另一条腿的运动动画：

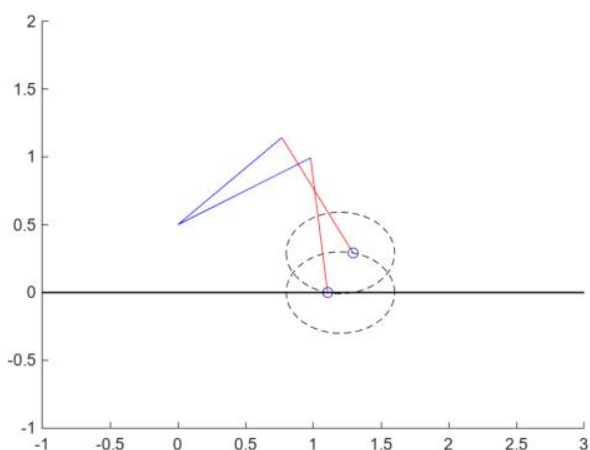


图 5 双腿运动动画（截图）

在进行四条腿的配合时则更加简单，我们只需要修改椭圆轨迹的参数，将椭圆中心 x 坐标 h 改为 $-h$ ，半长轴 a 改为 $-a$ ，半短轴 b 改为 $-b$ ，并修改计算关节角度参数的逆运动学公式：

$$\theta_2 = \arccos\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}\right)$$

即可得到四条腿的运动动画：

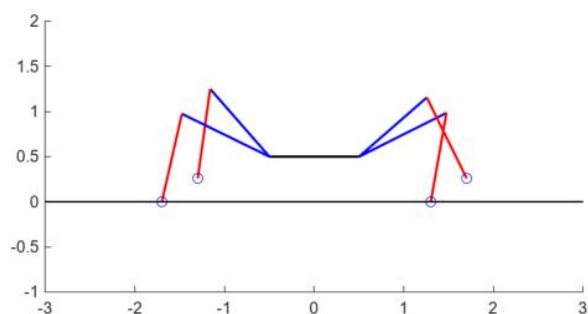


图 6 四腿运动动画（截图）

1.3 双腿配合中各个关节角度-时间函数

机器人仿真软件的原理是设计各个关节的电机运动参数，即设置每个关节的角度时间-函数，以此来控制仿蟹六足机器人腿部的配合运动。根据上文关节角度参数的逆运动学公式，分别实时输出四个关节的角度数据，得到一系列的角度-时间的数据。

其中，第一个角度数据表示初始状态下仿蟹六足机器人的关节角度，整个数据总和时间为两足配合完成一整个周期运动的时间 T ，每个关节每走完 $2\pi/100$ 的角度时提取一次对应的运动时间。

接着，利用 Matlab 软件将上述数据画在时间为横轴，角度为纵轴的二维散点图上。经过观察发现，散点随时间的分布近似三角函数的分布结果，因此利用 Matlab 软件中的 ‘fitttype’ 函数创建自定义的拟合模型-sin 函数来进行关节角度-时间函数的拟合。

拟合数据主要利用最小二乘法，用于寻找数据和拟合函数之间的最佳匹配。建立 \sin 函数模型 $A \sin(Bt + C) + D$ 的形式作为拟合模型，并假设模型参数为 A, B, C, D 。通过最小化实际数据与拟合模型之间的残差平方和，确定拟合模型中的参数。残差是指每个数据点的实际值与拟合函数的预测值之间的差异。最小二乘法的目标是找到使残差平方和最小的参数值，使用数学优化方法，如求导或矩阵运算，来最小化残差平方和。这可以通过最小二乘法的闭式解（解析解）或迭代算法（如梯度下降法）来实现。最后使用确定的拟合参数，将拟合模型应用于新的输入数据，以得到拟合曲线或预测值。

最小二乘法的流程如图 7。绘制出的两条腿四个关节角度-时间拟合函数与实际函数图象如下：（其中拟合误差小于 5%，符合实验要求。）

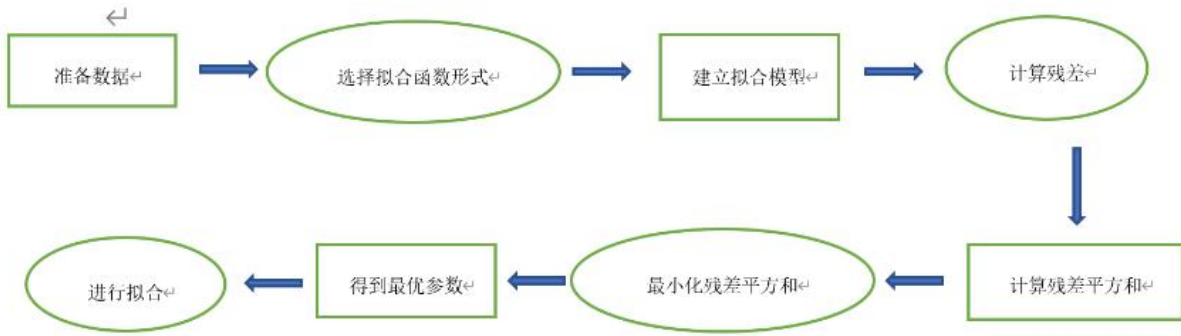


图 7 最小二乘法的流程

各个关节的角度-时间函数：

$$\begin{aligned} \text{leg1joint1} &= 0.3216 \sin(1.02t - 0.4696) + 0.4348 \\ \text{leg1joint2} &= 0.5279 \sin(0.9797t - 4.351) - 4.351 \\ \text{leg2joint1} &= -0.3162 \sin(0.9742t - 0.3374) + 0.4356 \\ \text{leg2joint2} &= 0.5209 \sin(1.011t - 1.299) - 1.664 \end{aligned}$$

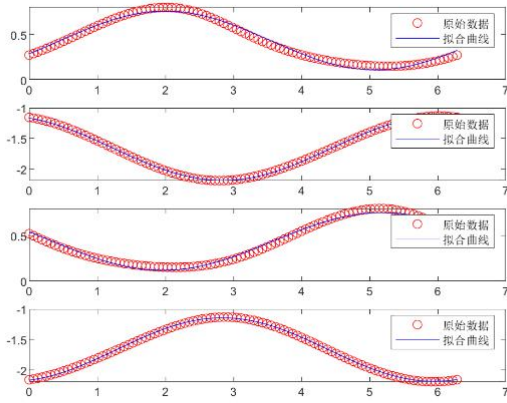


图 8 各个关节的角度-时间函数

根据拟合得到的两条腿四个关节的角度-时间函数，求时间的一阶导数和二阶导数，得到两条腿四个关节的角速度-时间函数，角加速度-时间函数，如下图：

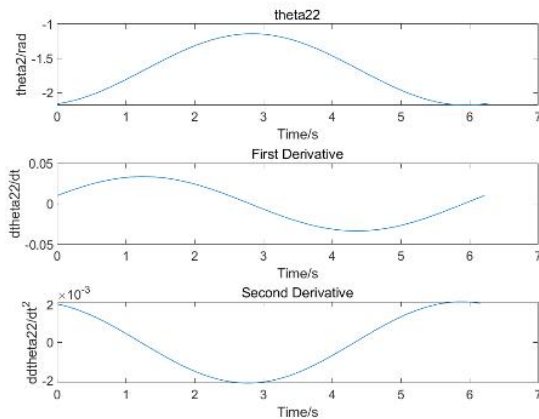


图 9 腿 1 关节 1

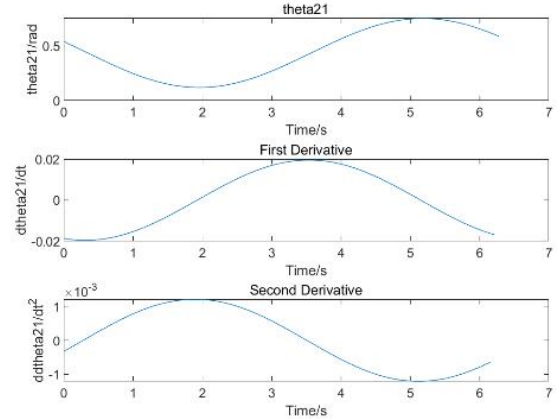


图 10 腿 1 关节 2

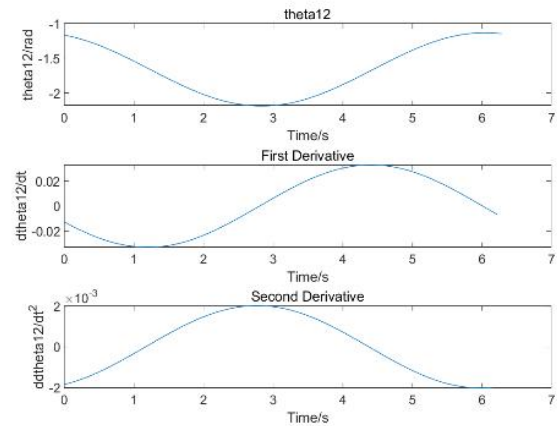


图 11 腿 2 关节 1

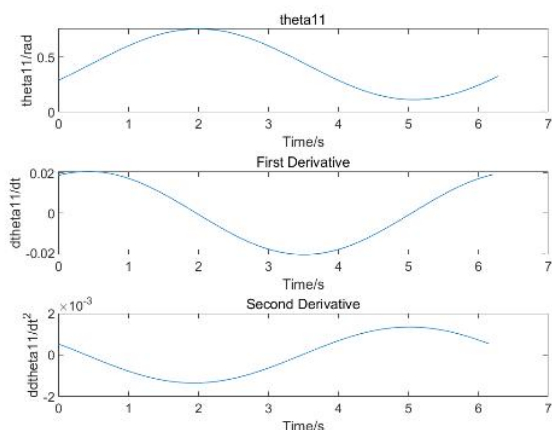


图 12 腿 2 关节 2

根据以上分析得到的各关节函数，可以作为正运动学输入的关节参数用于仿蟹六足机器人的运动仿真。

2. Webots 仿真

2.1 搭建 webots 六足仿蟹机器人

我们的机器人主体尺寸参数如图 13 所示。以一个扁平长方体块作为机器人的身体, 该长方体长为 1, 宽为 0.7, 高度为 0.1。每条腿由三个直径 0.05, 高度 0.1 的圆柱体构成旋转关节, 两个长、宽、高分别为 0.08、0.3、0.05 的长条分别组成大腿和小腿。大小腿长度相同, 使得关节可操作度较高。足尖由一个直径 0.08 的球构成。

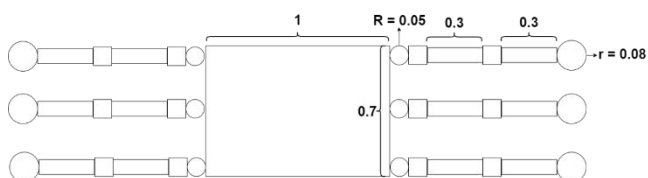


图 13 机器人主体尺寸参数

在 webots 中搭建六组机器人模型的流程如图

13 所示。首先创建一个 Robot 节点，并命名为“Hexapod”。在该节点的子节点下，创建 Shape 节点，Shape 节点构成机器人的身体。接下来需要搭建机器人的六条腿。由于六条腿在结构上相同，因此可以搭建好一条腿后通过复制该腿来创建其他的腿。下面将详述搭建一条腿的流程。

以搭建 Leg 4 为例，先在 Robot 的子节点下创建一个铰链 HingeJoint，命名为“HingeJoint 41”(“4”指第四条腿，“1”指第一关节，其余腿上各结构命令同理)，在 HingeJoint 41 的末端节点 endPoint 的子节点下创建铰链节点 HingeJoint 42，这两个铰链共同构成机器人的髋关节，控制两个方向上的旋转运动。在 HingeJoint 42 的末端节点的子节点下创建铰链节点“HingeJoint 43”，这是机器人的膝关节。在 HingeJoint 41 的末端节点的子节点中有一个 Shape 节点，这个节点定义了一个圆柱体，构成一个旋转关节。HingeJoint 42 和 HingeJoint 43 的末端节点的子节点下均创建了 Group 节点，Group 节点下有多个 Shape 节点，定义机器人的一组腿部结构。具体来讲，HIP_Group4 这个 Group 节点下有两个 Shape 节点，分别定义了腿部的第二关节 joint 42 和大腿连杆 Link 41；Knee 4 节点下则定义了三个 Shape 节点，分别是小腿连杆“Link 42”、膝关节 joint 43 及足尖 Foot 4。为每个 Shape 节点下的 geometry 节点下设置对应的几何形状构成机器人腿部的外形结构。最后为每个 HingeJoint 节点下的 device 节点添加旋转电机并按照上文规则命名，至此一条腿短的搭建工作完成。接着通过复制的方法搭建其他几条腿，但需要注意需要修改每条腿上的各个节点的命名。

最后为使我们的机器人更加美观，我们在 Robot 节点的子节点下创建了 Head、LEFT_EYE、RIGHT_EYE 三个实体节点，分别作为机器人的头部，左眼和右眼。

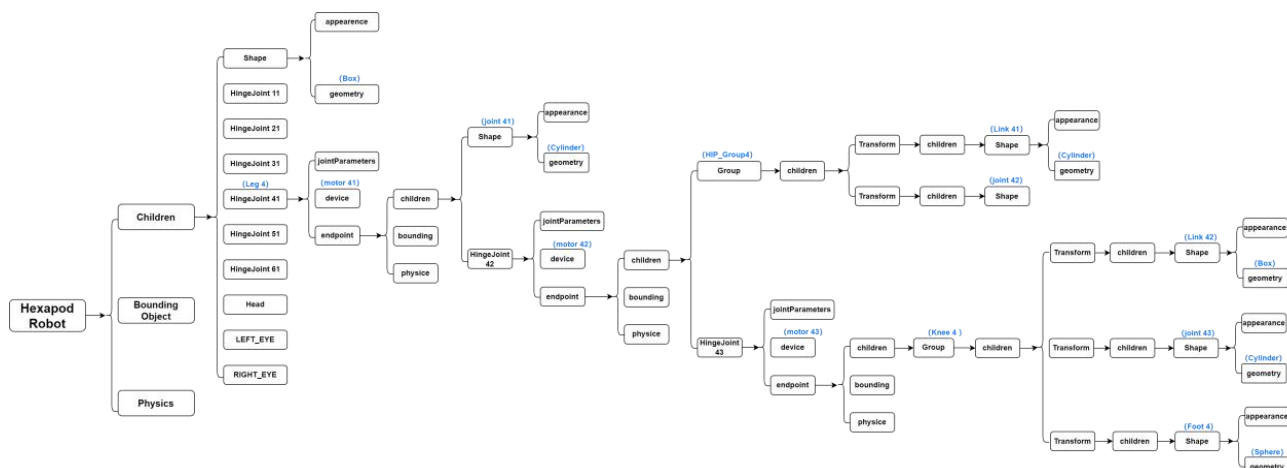


图 14 搭建一条腿的流程

需要注意到是，在完成上述步骤后，还需要进一步检查每个节点的参数设置。例如某杆件的长度可能在设计过程中因为一些操作被修改了，如果不检查并更正，可能使得后续的仿真实验得到一个错误的结果，这是过来人的一点忠告！

最后需要设置所有旋转电机的轴向并将电机位置归零。将各条腿的第一关节角度设置为 -90° ，机器人将呈现一个给定姿态，我们将这个姿态作为机器人的起始姿态，如图 15 所示。

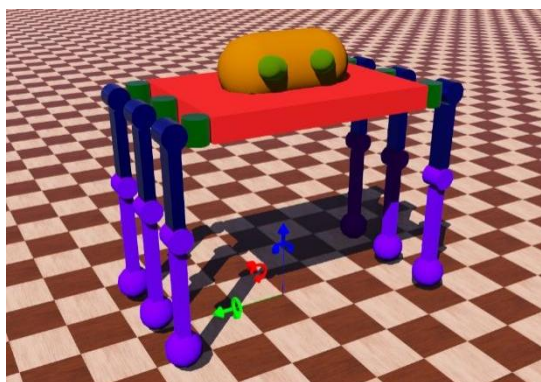


图 15 机器人的起始姿态

2.2 webots 运动控制

2.2.1 三角步态直行控制器

在所有的控制器程序中，为方便调用电机，我们在 `RobotController` 类中通过旋转电机的名称“`motorij`”获取各个电机并赋给类属性“`mij`”，其中 i 指定第几条腿， j 指定腿上第几个电机。并将每条腿上实现同一功能的关节上的电机存入一个集合，以便循环调用各个电机。

`controller_move1.py` 控制器中，变量 `aij`、`bij`、`cij`、`dij`、`a-ij`、`b-ij`、`c-ij`、`d-ij` 记录了从 MATLAB 拟合得到的运动角度随时间变化函数的各项系数。在类方法 `moveForward` 中，上述参数用于实时计算各个关节给定时刻的角度，然后将相应的关节上旋转电机的位置设定为相应角度。通过死循环控制控制传入 `moveForward` 方法的时刻 t 周期性变化，实现每条腿交替配合运动，以三角步态直线行进。

`controller_move2.py` 控制器中，不再通过拟合函数计算关节角度，而是直接导入 MATLAB 分析得到的角度随时间变化的数据，通过这些数据直接控制电机的位置，如此避免了实时计算角度，占用计算资源相比 `controller_move1` 中的控制方法理论上更少一些。

2.2.2 原地转向控制

最初的转向方案是三角步态转向，即三条腿同

时抬起、转动、落下，然后扭动身体，最后转动另外三条腿实现完整转向。但我们在实践中，可能由于我们的机器人设计原因，同侧三条腿之间距离较近，导致在实现三角步态转向时触地足总是发生大的滑移，转向过程不稳定，且转向角度非常受限。因此，我们尝试了另一种转向方案：先扭动身体，使得机器人朝向预设方向，然后依次转动关于机器人中心对称的三组腿，实现转向。相关程序在 `controller_turn.py` 中实现。

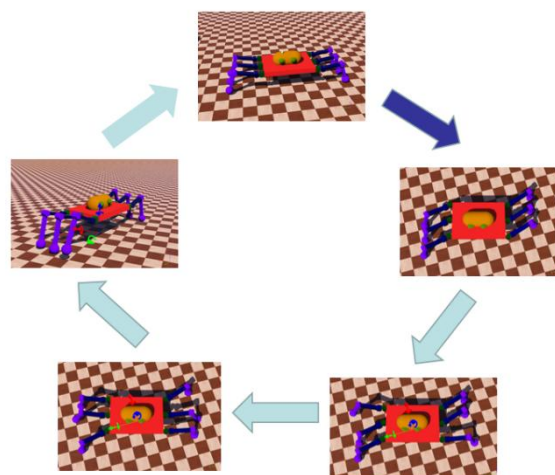


图 16 转向过程

经过仿真测试发现，在我们设计的这款六组机器人上，这一转向策略表现得更加稳定，并且转向角度也比三角步态转向更大。但这种转向由于需要三次抬腿，相比三角步态转向稍显繁琐。且许多资料中的六足机器人均是采用三角步态转向，一定程度说明这种策略实际上是更优的。但也发现许多采用三角步态转向的六组机器人其 6 条腿大致分布在六边形的六个顶点上，这与我们这款仿蟹六足机器人有明显区别。可能是由于别样的结构设计，导致我们这款机器人在实现三角步态转向时没能达到预期效果。当然，也有可能是我们的三角步态专项策略不够完善。关于仿蟹六足机器人的稳定高效转向还有待进一步研究。

2.2.3 移动中转向

在最后的的时间里，我们做了一个有趣的尝试，在 `controller_moveANDturn.py` 中将前面两种运动方式结合起来，观察仿蟹六足机器人在移动中转向的效果。我们让机器人先以三角步态直线行进，然后在某一时间收到转向指令，机器人先调整到准备姿态，然后原地转向。最终呈现效果可以预见，由于只是简单组合了两种运动方式，机器人的运动实际上直线移动和转向是相互分离的，因此这种运动方式严格来说算不上在移动中转向。想要真正实现移

动中转向，需要考虑每条腿的配合以及如何规划足尖轨迹，这一工作相比单纯的三角步态直行要更加复杂和具有挑战性。

3. 仿蟹六足机器人三角步态直行的稳定性分析

定性分析

在 webots 仿真软件中，我们收集并导出了机器人在不同速度下以三角步态方式直行时的位置与时间的数据，利用收集到的数据，在 Matlab 软件上绘制出不同速度下中心位置在 X 轴和 Z 轴方向上的位置-时间函数。

X 轴方向上的位置-时间函数：

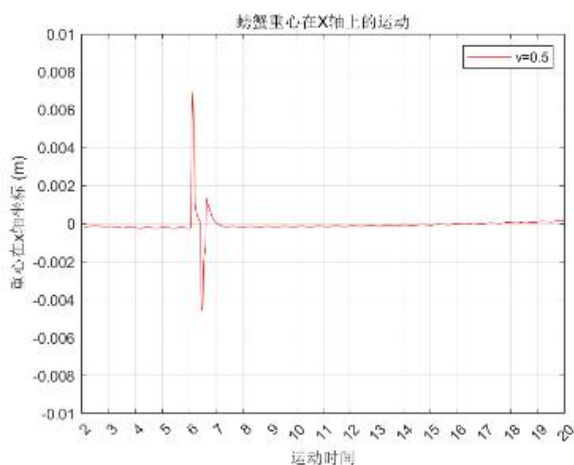


图 17 螃蟹重心在 $v=0.5\text{m/s}$ 时，在 X 轴上的运动

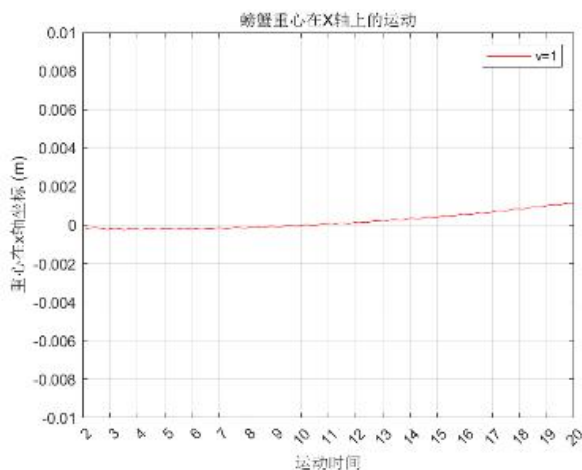


图 18 螃蟹重心在 $v=1\text{m/s}$ 时，在 X 轴上的运动

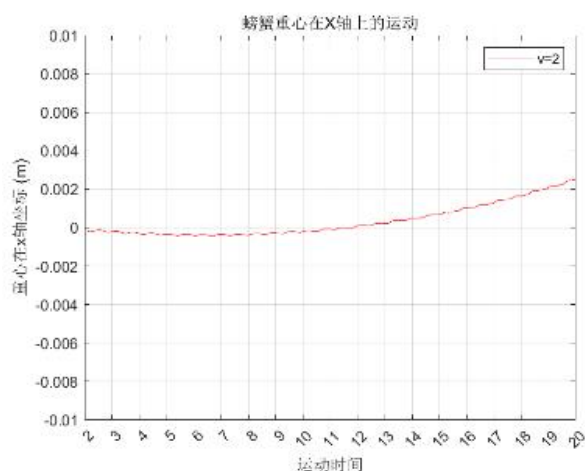


图 19 螃蟹重心在 $v=2\text{m/s}$ 时，在 X 轴上的运动

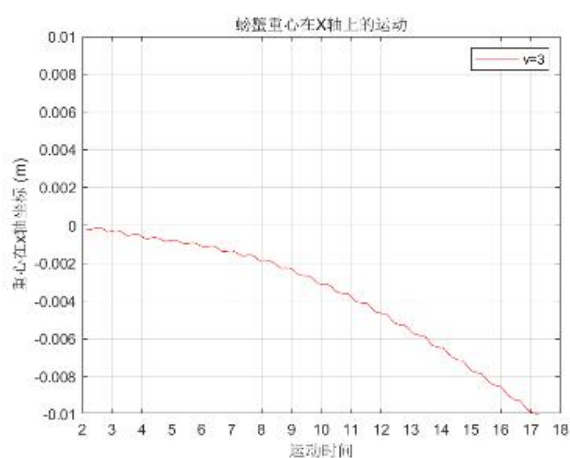


图 20 螃蟹重心在 $v=3\text{m/s}$ 时，在 X 轴上的运动

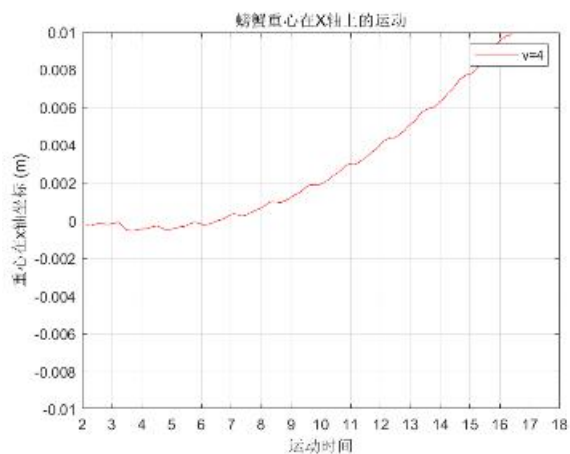
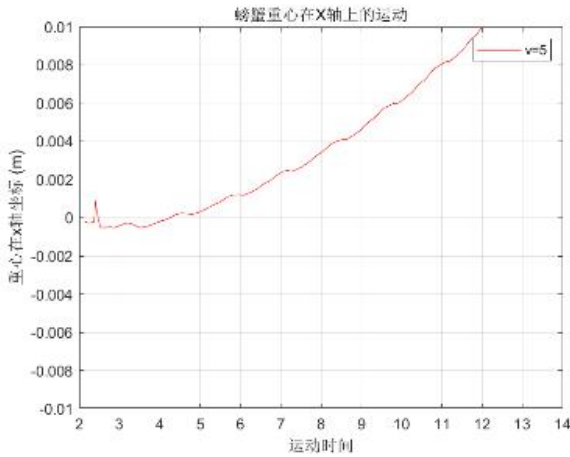
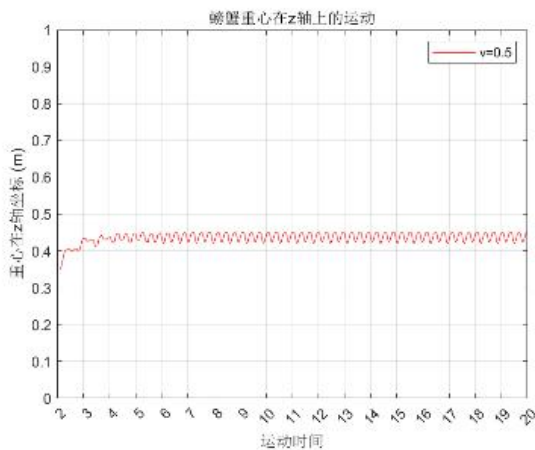
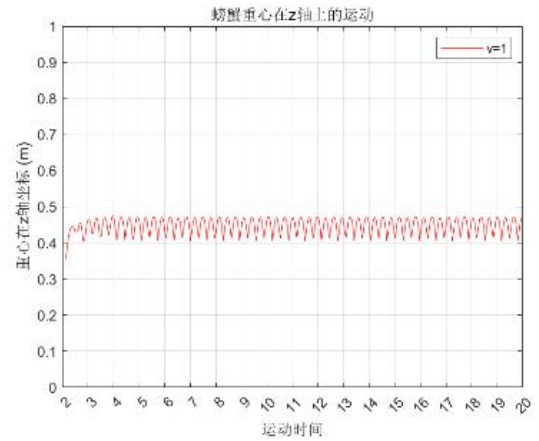
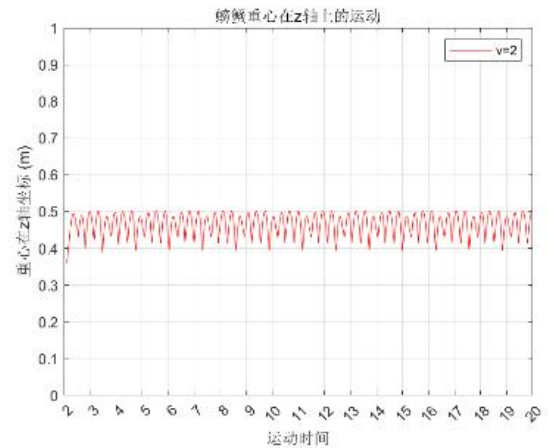
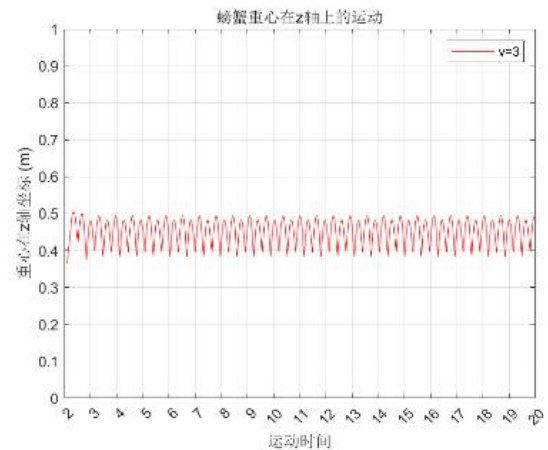


图 21 螃蟹重心在 $v=4\text{m/s}$ 时，在 X 轴上的运动


 图 22 螃蟹重心在 $v=5\text{m/s}$ 时，在 X 轴上的运动

机器人被设定为沿着 y 轴直行，我们获取了不同速度下机器人在 X 轴上的位置-时间信息，发现机器人的移动速度并不完全与设定的电机速度成比例。当设定电机速度大于 5 时，移动速度不再改变，猜测可能是由于电机已经达到最大力矩，因此无法再加速。此外 我们试图研究不同速度下机器人中心在 x 方向上的偏移来分析不同速度下的稳定性，发现机器人总有向一侧偏转的趋势，可能是由于机器人中心与重心不重合引起的。但是当 $v=0.5$ 和 $v=3$ 时出现了不同的情况，尚不清楚造成此现象的原因。不过总的偏移几乎都在每 10M 偏移毫米级别，大致可以忽略。

z 轴方向上的位置-时间函数：


 图 23 螃蟹重心在 $v=0.5\text{m/s}$ 时，在 Z 轴上的运动

 图 24 螃蟹重心在 $v=1\text{m/s}$ 时，在 Z 轴上的运动

 图 25 螃蟹重心在 $v=2\text{m/s}$ 时，在 Z 轴上的运动

 图 26 螃蟹重心在 $v=3\text{m/s}$ 时，在 Z 轴上的运动

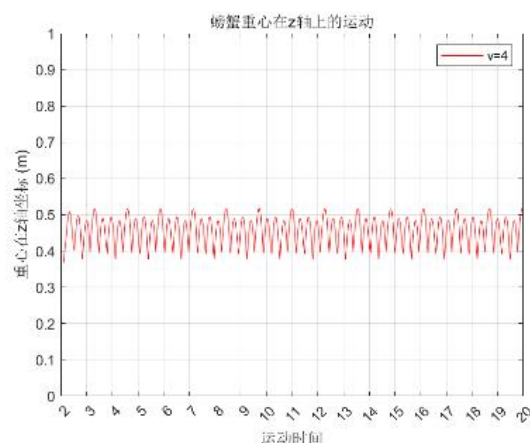


图 27 螃蟹重心在 $v=4\text{m/s}$ 时，在 Z 轴上的运动

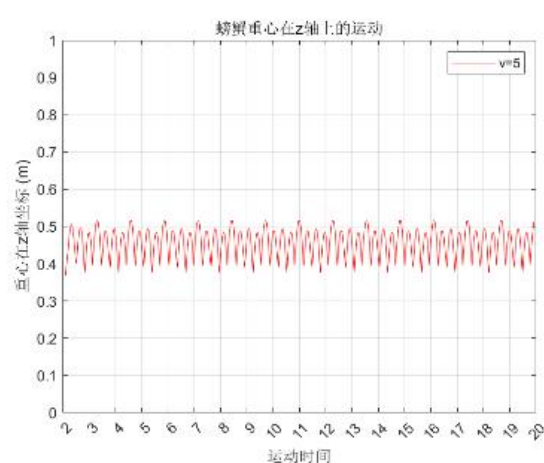


图 28 螃蟹重心在 $v=5\text{m/s}$ 时，在 Z 轴上的运动

机器人被设定为沿着 y 轴直行，我们获取了不同速度下机器人在 Z 轴上的位置-时间信息，发现机器人随着速度增大，重心上下起伏增大。当设定电机速度大于 5 时，重心波动变化趋于稳定，几乎不变，猜测可能是由于电机已经达到最大力矩，因此无法再加速，不过上下偏移几乎都在毫米级别，大致可以忽略。

由以上的数据分析得知，除了在 X 轴和 Z 轴上存在毫米级的误差，几乎可以忽略不计，且误差原因可能由于仿真得到的机器人中心与重心不重合引起的。因此，在误差允许的范围内，仿真得到的机器人运动情况与理论设计的结果一致，证明仿蟹六足机器人三角步态直行的正逆运动学的理论推导和结构设计符合预期。

4 分工合作：

陆瑾萱 (25%)：运动学理论推导、Matlab 单足仿真、Matlab 转向功能仿真

吴梓宁 (25%)：双足、四足的配合、Matlab 代码的完善

孙健玮 (25%)：webots 机器人建模、设计 webots 控制器、数据分析

解凌一 (25%)：理论推导、数据处理、数据分析

5 总结：

我们对六足仿蟹机器人的三角步态运动模式进行了理论分析，并使用 MATLAB 推导出六足机器人以三角步态直行时两组足的关节运动参数，通过最小二乘法拟合得到六足机器人三角步态运动的关节角度的函数。随后我们使用 webots 搭建了六足仿蟹机器人，并使用从 MATLAB 推导得到的函数来控制机器人的运动，最后成功实现了机器人以三角步态直线，验证了相关理论推导的正确性。

我们还尝试了在 webots 中控制机器人以三角步态模式实现转向，但由于缺少相关理论研究以及我们的机器人本身的设计局限，没能很好的实现三角步态转向。继而我们尝试了另一种转向策略，实现了机器人的原地转向。

总之，我们主要研究了六足机器人的三角步态直行运动模式并结合 MATLAB 和 webots 验证了分析的准确性和合理性，但对于机器人的三角步态转向和移动中转向则需要更进一步的理论分析和研究。

参 考 文 献

- [1] Buehler, M., Battaglia, R., Cocosco, A., Hawker, G., Sarkis, J., & Yamazaki, K. (1998). SCOUT: A simple quadruped that walks, climbs, and runs. *Journal of Robotic Systems*, 14(4), 231-244.
- [2] Altendorfer, R., Moore, N., Komsuoglu, H., Buehler, M., Brown Jr, H. B., McMordie, D., ... & Koditschek, D. (2001). RHex: A biologically inspired hexapod runner. *Autonomous Robots*, 11(3), 207-213.
- [3] Raibert, M. H. (1986). *Legged robots that balance*. MIT press.
- [4] 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.
- [5] 王刚. 仿蟹机器人步态规划及复杂地貌行走方法研究[D]. 哈尔滨工程大学, 2011.
- [6] 张伟岗, 李友子. 基于 STM32 的六足机器人控制系统设计与实现[J]. 微型电脑应用, 2023, 39(05): 81-83.
- [7] 马鑫, 吴银川. 六足机器人控制系统设计[J]. 仪器仪表用户, 2023, 30(06): 14-18.
- [8] 郭建, 梁永杰, 张晓佳等. 基于步态规划的六足机器人运动学分析与计算[J]. 机床与液压, 2023, 51(05): 66-73.
- [9] 马秀峰, 张奇峰, 孙英哲. 仿生六足机器人步态设计与运动仿真[J]. 计算机仿真, 2023, 40(02): 255-260+269.

- [10]李党超,朱爱斌,李昊城.一种适应崎岖地形的六足机器人实验平台设计[J].中国现代教育装备,2022,No.385(09):40-42.DOI:10.13492/j.cnki.cmee.2022.09.049.