

基于水下六足机器人腿部运动控制问题研究

魏超, 刘卫东, 俞健, 高立娥, 李乐

(西北工业大学 航海学院, 西安 710072)

摘要: 水下爬游六足机器人相比于传统的水下控制器 ROV (Remote Operated Vehicle), AUV (Autonomous Underwater Vehicle) 具有造价低, 运行和维护成本相对较低的特点, 以及最主要的可以实现定点作业的问题, 设计了能够在海底爬行的水下六足机器人; 基于以往的多足机器人腿部控制方法使得机器人在运动时存在严重的互斥力问题, 尤其当机器人自身重量大时, 对关节损伤严重, 对如何解决分布于机体两侧各腿处于支撑态行走一步的过程中产生的互斥力问题进行了深入研究, 并对控制算法进行改进, 使得机器人的两侧腿处于支撑态时走出的轨迹为两条平行线; 给出了机器人在不同爬行状态下正逆运动学解, 并进行了 Simulink 仿真验证; 此外, 对机器人如何爬行, 即控制策略进行了描述。

关键词: 爬游六足机器人; 平行间距控制算法; 正逆运动学解; Simulink 仿真; 机器人控制策略

Research on Leg Motion Control of Underwater Hexapod Robot

Wei Chao, Liu Weidong, Yujian, Gao Li'e, Li le

(College of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Underwater crawling hexapod robots compared to traditional underwater controllers ROV (remote operated vehicle), AUV (autonomous underwater vehicle) has the characteristics of low cost, relatively low operation and maintenance costs, and the most important advantage that can achieve fixed-point operations. However, the used leg's control method of multi-legged robot has caused the robot to have a serious mutex problem during the movement, especially when the robot itself has heavy weight, which causes serious damage to the joints. Therefore, how to solve the mutual repulsive force caused by the two legs of the body in the process of supporting one step is necessary for further study and improve the control algorithm to solve this problem. then conduct Simulink simulation verification. In addition, described the strategy on how to control the robot crawls.

Keywords: crawling and roving hexapod robots; parallel spacing control algorithm; positive and inverse kinematics; simulink simulation; robot control strategy

0 引言

进行水下探索, 开发海洋资源已经成为各国的主要目标, 传统的 ROV (Remote Operated Vehicle) 由于自带线缆的影响, 使得它只能在一定区域内进行巡航, 作业范围大大受限; AUV (Autonomous Underwater Vehicle) 依靠自身的自治能力来管理和控制自己完成使命, 在实时通信上存在缺陷。此外, 两者最大的共同点是只能巡航作业, 无法实现精确定位。文献 [1-4] 提出了陆地六足机器人的设计与研究, 但不能应用于水下。基于此, 研究了一种能在水下爬游相结合的机器人, 如图 1 所示。水下爬游机器人综合了两者各自的特点, 既能水中巡航, 又能在海底爬行, 实现精确定位。它以巡游的方式到达作业区域后, 再以爬行方式到达作业点, 在探索地形, 水下救援, 打捞等方面具有独特的优势。

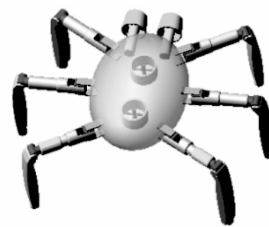


图 1 爬游机器人的整体外观图

针对机器人在海底爬行过程中需要解决的主要问题探索: 第一, 机体两侧腿在处于支撑态行走过程中出现的互斥力问题; 第二, 如何进行逻辑控制, 使机器人运动。

针对第一个问题, 文献 [5], 文献 [6] 提出的目前大多数样机采用的方法得到的运动轨迹如图 2 所示, 即机器人爬行过程中, 在跟关节处从俯视图看, 各条腿处于支撑态时走过的轨迹, 当机体质量较大时, 会对机体两侧腿部各关节的挤压力导致关节损伤严重, 因此根据实际应用上的需求, 在文献 [5], 文献 [6] 所提出的算法上进行了改进, 在机器人行走过程中足端距机体位置变化进行了约束, 将这种约束产生的影响分布在每条腿的三个关节上, 利用这三个关节协同作用, 使得每条腿在处于支撑态时, 足端

收稿日期: 2018-06-28; 修回日期: 2018-07-25。

基金项目: 国家重点研发计划(2016YFC0301700)。

作者简介: 魏超(1992-), 男, 河南桐柏人, 硕士研究生, 主要从事计算机控制, 机器人运动控制方向的研究。

走过一步的运动轨迹为一条直线，平行于前进方向，即图 3 所示，这样分布于机体两侧的腿处于支撑态走过一步的轨迹相互平行，这样就不存在互斥力。

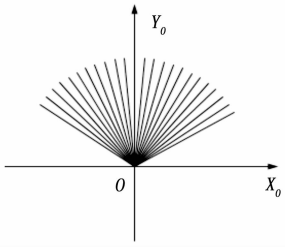


图 2 改进前的单腿支撑态走过的轨迹

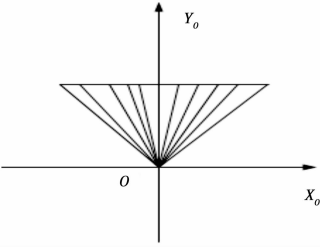


图 3 改进后单腿支撑态走过的轨迹

针对第二个问题，目前六足机器人爬行过程中有三足步态，四足步态和六足步态。相比于其他步态，三足步态具有控制相对简单，爬行速度快的优点，本文采用三足步态控制机器人运动。

根据机器人自身结构特点，参考文献 [7]，文献 [8] 提出的 D-H 坐标系的建立方法，建立机体坐标系和单腿坐标系。推导出正逆运动学方程，参考文献 [9] 中对机器人的运动仿真方法，对机器人处于支撑态行走一步所需要的角度进行仿真，给出仿真曲线。

1 单腿坐标系的建立

本文设计的六足机器人共 18 个关节，每条腿上安装 3 个关节，对每一条单步行腿，根据文献 [10]，文献 [11]，文献 [12]，采用 D-H 法建立坐标系，D-H 法的具体规则描述如下：

- 1) 确定 z_{ij} 轴，均垂直于轴向，各关节围绕此轴转动；
- 2) 确定 x_{ij} 轴， x_{ij} 通过 z_{ij-1} 和 z_{ij} 的公法线方向，由 z_{ij-1} 指向 z_{ij} ；
- 3) 确定 y_{ij} 轴，用右手定则判定。

其中 i 表示腿 1~6 中具体的某一条腿， j 表示每条腿上的关节。依据此方法，对单腿进行如下图 4 所示的坐标系建立：

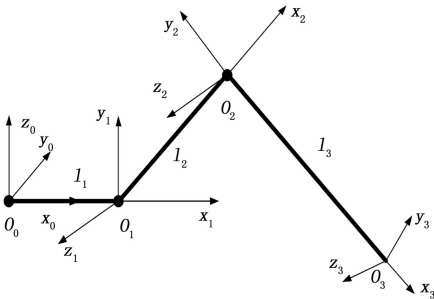


图 4 单步行腿各关节坐标系示意图

图 4 中 O_0, O_1, O_2 分别表示跟关节，髌关节，膝关节的位置， O_3 的存在是为了便于分析足端位置，所以也在足端建立一个坐标系。在 O_0, O_1, O_2 处分别绕 z_0, z_1, z_2 旋转即形成 $\theta_1, \theta_2, \theta_3$ 。

2 系统硬件设计

2.1 正运动学解

以机体中心为坐标原点建立坐标系，所有的腿部及关节坐标均以机体坐标系为参考，定义参数如下：

L_0 为身体中心到第一关节，即跟关节的距离；

θ_0 为身体中心到第一关节（跟关节）连线与 Y_b 的夹角，

由机器人本身结构参数决定；

$\theta_1, \theta_2, \theta_3$ 为机器人每条腿的第一关节角，第二关节角和第三关节角；

l_1 为第一关节（跟关节）到第二关节，即髌关节的距离；

l_2 为第二关节（髌关节）到第三关节，即膝关节的距离；

l_3 为第三关节（膝关节）到脚底，即足端的距离。

根据文献 [13]，文献 [14]，文献 [5]，腿 i 跟关节坐标系到髌关节坐标系的旋转矩阵 ${}^iR_{i_1}$ ，髌关节坐标系到膝关节坐标系 ${}^iR_{i_2}$ ，膝关节坐标系到足端坐标系的旋转矩阵 ${}^iR_{i_3}$ 分别为：

$${}^iR_{i_1} = \begin{pmatrix} \cos\theta_{i1} & 0 & \sin\theta_{i1} \\ \sin\theta_{i1} & 0 & -\cos\theta_{i1} \\ 0 & 1 & 0 \end{pmatrix} \quad (1)$$

$${}^iR_{i_2} = \begin{pmatrix} \cos\theta_{i2} & -\sin\theta_{i2} & 0 \\ \sin\theta_{i2} & \cos\theta_{i2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$${}^iR_{i_3} = \begin{pmatrix} \cos\theta_{i3} & -\sin\theta_{i3} & 0 \\ \sin\theta_{i3} & \cos\theta_{i3} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

可得腿 i 足端坐标系原点在跟关节坐标系中的旋转矩阵为：

$${}^iR_{i_0} = {}^iR_{i_1} * {}^iR_{i_2} * {}^iR_{i_3} = \begin{pmatrix} \cos\theta_{i1} \cos(\theta_{i2} + \theta_{i3}) & -\cos\theta_{i1} \sin(\theta_{i2} + \theta_{i3}) & \sin\theta_{i1} \\ \sin\theta_{i1} \cos(\theta_{i2} + \theta_{i3}) & -\sin\theta_{i1} \sin(\theta_{i2} + \theta_{i3}) & -\cos\theta_{i1} \\ \sin(\theta_{i2} + \theta_{i3}) & \cos(\theta_{i2} + \theta_{i3}) & 0 \end{pmatrix} \quad (4)$$

进而可以推导出机器人腿 i 跟关节，髌关节，膝关节坐标系间的齐次变换矩阵如下：

$${}^iT_{i_1} = \begin{pmatrix} \cos\theta_{i1} & 0 & \sin\theta_{i1} & l_1 \cos\theta_{i1} \\ \sin\theta_{i1} & 0 & -\cos\theta_{i1} & l_1 \sin\theta_{i1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

$${}^iT_{i_2} = \begin{pmatrix} \cos\theta_{i2} & -\sin\theta_{i2} & 0 & l_2 \cos\theta_{i2} \\ \sin\theta_{i2} & \cos\theta_{i2} & 0 & l_2 \sin\theta_{i2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

$${}^iT_{i_3} = \begin{pmatrix} \cos\theta_{i3} & -\sin\theta_{i3} & 0 & l_3 \cos\theta_{i3} \\ \sin\theta_{i3} & \cos\theta_{i3} & 0 & l_3 \sin\theta_{i3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

由式 (5), (6), (7) 可得, 跟关节到足端的变换矩阵为:

$${}^i\mathbf{T}_{i_3} = {}^i\mathbf{T}_{i_1} \times {}^i\mathbf{T}_{i_2} \times {}^i\mathbf{T}_{i_3} = \begin{pmatrix} {}^i\mathbf{R}_{i_3} & {}^i\mathbf{P}_{i_3} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & 0 & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8)$$

其中:

$$\begin{aligned} a_{11} &= \cos\theta_{i1} \cos(\theta_{i2} + \theta_{i3}); \\ a_{12} &= -\cos\theta_{i1} \sin(\theta_{i2} + \theta_{i3}); \\ a_{13} &= \sin\theta_{i1}; \\ a_{14} &= (l_1 + l_2 \cos\theta_{i2} + l_3 \cos(\theta_{i2} - \theta_{i3})) \cos\theta_{i1}; \\ a_{21} &= \sin\theta_{i1} \cos(\theta_{i2} + \theta_{i3}); \\ a_{22} &= -\sin\theta_{i1} \sin(\theta_{i2} + \theta_{i3}); \\ a_{23} &= -\cos\theta_{i1}; \\ a_{24} &= (l_1 + l_2 \cos\theta_{i2} + l_3 \cos(\theta_{i2} - \theta_{i3})) \sin\theta_{i1}; \\ a_{31} &= \sin(\theta_{i2} + \theta_{i3}); \\ a_{32} &= \cos(\theta_{i2} + \theta_{i3}); \\ a_{34} &= l_2 \sin\theta_{i2} + l_3 \sin(\theta_{i2} - \theta_{i3}). \end{aligned}$$

根据 (8) 式可得足端坐标原点在其跟关节坐标系中的位置矢量为:

$${}^i\mathbf{P}_{i_3} = \begin{pmatrix} (l_1 + l_2 \cos\theta_{i2} + l_3 \cos(\theta_{i2} - \theta_{i3})) \cos\theta_{i1} \\ (l_1 + l_2 \cos\theta_{i2} + l_3 \cos(\theta_{i2} - \theta_{i3})) \sin\theta_{i1} \\ l_2 \sin\theta_{i2} + l_3 \sin(\theta_{i2} - \theta_{i3}) \end{pmatrix} \quad (9)$$

机体坐标系 $X_b O Y_b$ 到跟关节的旋转矩阵 (绕 z 轴旋转) 为:

$${}^0\mathbf{T}_{i_1} = \begin{pmatrix} \cos\theta_0 & -\sin\theta_0 & 0 & d_2 \\ \sin\theta_0 & \cos\theta_0 & 0 & d_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (10)$$

其中: θ_0 由机体本身结构所决定, d_1 绕 y 轴平移的距离, d_2 为绕 x 轴平移的距离, 当腿 i 确定时, 对应的参数即可确定。

各条腿 i 足端坐标原点 x_p, y_p, z_p 在其机体中心 O 处的坐标为:

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = {}^0\mathbf{T}_{i_1} \times {}^i\mathbf{P}_{i_3} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (11)$$

其中:

$$\begin{aligned} b_1 &= d_2 + l_1 \sin(\theta_0 + \theta_{i1}) + l_2 \cos\theta_{i2} \sin(\theta_0 + \theta_{i1}) + l_3 \sin(\theta_0 + \theta_{i1}) \cos(\theta_{i2} - \theta_{i3}); \\ b_2 &= d_1 + l_1 \cos(\theta_0 + \theta_{i1}) + l_2 \cos\theta_{i2} \cos(\theta_0 + \theta_{i1}) + l_3 \cos(\theta_0 + \theta_{i1}) \cos(\theta_{i2} - \theta_{i3}); \\ b_3 &= l_2 \sin\theta_{i2} + l_3 \sin(\theta_{i2} - \theta_{i3}); \end{aligned}$$

其中: z_p 在这里被当做标量处理, 只作数值, 不考虑正负。

比如要确定腿 2 的足端位置时, 做出其俯视图, 如图 5 所示, 从俯视图看, $X_0 O Y_0$ 为机身坐标, $X_b O Y_b$ 表示腿 2 跟关节处坐标。Z 轴都垂直平面向上。由于每条腿具有相同的关节构造, 当具体到某一条腿时, 为了公式简化, i 的作用可以忽略。

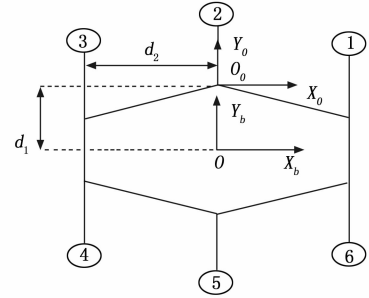


图 5 坐标旋转示意图

机体坐标系向第二条腿跟关节旋转平移, 此时 $\theta_0 = 0^\circ$, $d_1 = L_0, d_2 = 0$, 所以:

$${}^0\mathbf{T}_{i_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \quad (13)$$

其中:

$$\begin{aligned} c_1 &= l_1 \sin\theta_1 + l_2 \cos\theta_2 \sin\theta_1 + l_3 \sin\theta_1 \cos(\theta_2 - \theta_3); \\ c_2 &= L_0 + l_1 \cos\theta_1 + l_2 \cos\theta_2 \cos\theta_1 + l_3 \cos\theta_1 \cos(\theta_2 - \theta_3); \\ c_3 &= l_2 \sin\theta_2 + l_3 \sin(\theta_2 - \theta_3); \end{aligned}$$

2.2 机体逆运动学解

所谓逆运动学解, 即由坐标位置, 反解出每条腿各 θ 角的大小。在求解机体逆运动学解时, 采用几何法进行求解。

在图 6 中, 本文给出的是腿 2 的坐标示意图, (x, y) 表示腿 2 足端在机体坐标系中坐标, 机体在沿 X_b 轴前进的过程中, D 保持不变。

$$\theta_1 = \text{atan}\left(\frac{y - d_2}{x - d_1}\right) \quad (14)$$

对于 θ_2, θ_3 , 如图 7 所示, 由余弦定理:

$$\cos(\theta_2 + \alpha) = \frac{l_2^2 + z^2 + (b - l_1)^2 - l_3^2}{2l_2 \sqrt{z^2 + (b - l_1)^2}} \quad (15)$$

在这里 z 的值为取绝对值之后的量, 在腿实际行走且处于支撑相的过程中, z 一定是一个负数, 所以 θ_2 的实际角度应为:

$$\theta_2 = \text{atan}\left(\frac{\sin(\theta_2 + \alpha)}{\cos(\theta_2 + \alpha)}\right) + \text{atan}\left(\frac{z}{b - l_1}\right) \quad (16)$$

$$\cos\theta_4 = \frac{l_2^2 + l_3^2 - (z^2 + (b - l_1)^2)}{2l_2 l_3} \quad (17)$$

$$\theta_4 = \text{atan}\left(\frac{\sqrt{1 - \cos^2\theta_4}}{\cos\theta_4}\right) \quad (18)$$

$$\theta_3 = \pi - \theta_1 \quad (19)$$

式中, b 的大小由行走时支撑腿足端与跟关节在前进方向上所形成的两条平行线间的距离 D 来决定, D 的大小为 $y - d_1$, 当 θ_1 变化时, $b \cos \theta_1 = D$. $Z_b O Y_b$ 为机体坐标系, l_1 为第一关节长度, 大小可为 0, (y, z) 为腿足端在机体坐标系下的坐标。

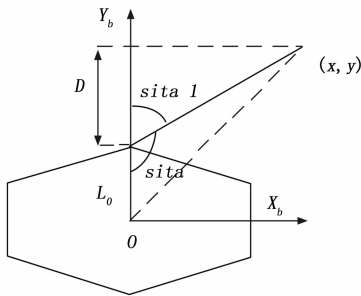


图 6 机体俯视图

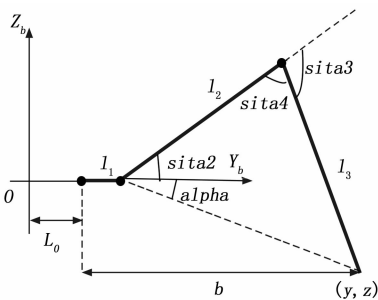


图 7 关节平面图

3 系统软件设计

3.1 设计思路和编程方法

机器人行走的步态主要包括前进/后退, 左移/右移, 左转/右转。在图 5 中, ①, ②, ③, ④, ⑤, ⑥ 代表机器人的六条腿, 为了便于描述将其依次命名为左前腿, 左中腿, 左后腿, 右后腿, 右中腿, 右前腿。三足步态下 ①, ③, ⑤ 为一组, ②, ④, ⑥ 为一组交替行走。下面具体说明机器人行走的控制策略。对前进/后退, 一共分为三个阶段进行控制, 分别为初始化阶段, 循环阶段和停止阶段。

初始化阶段:

- 1) ①, ③, ⑤ 抬起, ②, ④, ⑥ 不动。这一步是为了让 ①, ③, ⑤ 腿为下一步抬起后前进/后退做准备, ②, ④, ⑥ 支撑机体保持稳定。
- 2) ①, ③, ⑤ 抬起后前进/后退, ②, ④, ⑥ 不动。这一步是为了让 ①, ③, ⑤ 腿处于目标位置, 便于下一步下落后机体做好前进/后退准备。②, ④, ⑥ 仍支撑机体保持稳定。
- 3) ①, ③, ⑤ 下落, ②, ④, ⑥ 不动。此时机器人六条腿都支撑在地面, ①, ③, ⑤ 在目标位置。
- 4) ①, ③, ⑤ 保持不动, ②, ④, ⑥ 抬起。这一步是为了下一步机体前进/后退做准备。
- 5) ①, ③, ⑤ 处于支撑态前进/后退一步, ②, ④,

⑥ 抬起后不动。通过这一步, 机体完成前进/后退动作。

6) ①, ③, ⑤ 保持不动, ②, ④, ⑥ 抬起后前进/后退。此举是为了让 ②, ④, ⑥ 腿达到目标位置, 为下一步前进/后退移动做好准备。此时, ①, ③, ⑤ 在前/后, ②, ④, ⑥ 在后/前, 因此这一步也是机械腿最有可能造成干涉的一步, 如果此步没有干涉, 那么其他步骤也不可能造成干涉; 如果此步有干涉, 将前进/后退一步的距离变小即可。

7) ①, ③, ⑤ 保持不动, ②, ④, ⑥ 下落。在这一步中, ②, ④, ⑥ 下落达到目标位置, 为下一步前进/后退做好准备。

8) ①, ③, ⑤ 抬起, ②, ④, ⑥ 保持不动。此举是为下一步前进/后退做准备。

9) ①, ③, ⑤ 抬起保持不动, ②, ④, ⑥ 处于支撑态前进/后退一步。通过这一步, 机器人达到运动目的。

以上 9 步即为初始化阶段过程, 机器人走过这个阶段, 进入循环阶段。

循环阶段: 在这一阶段, 机器人通过不断循环走这一步, 可以达到前进/后退的目的, 循环阶段又分为以下 8 个步骤。

- 1) ①, ③, ⑤ 抬起后前进/后退, ②, ④, ⑥ 保持不动。这一步是让 ①, ③, ⑤ 在下一步下落后达到目标位置, 为下一步动作做好准备。
 - 2) ①, ③, ⑤ 下落到地, ②, ④, ⑥ 保持不动。此时 ①, ③, ⑤ 到达目标位置。
 - 3) ①, ③, ⑤ 不动, ②, ④, ⑥ 抬起。此时机体由 ①, ③, ⑤ 腿支撑, 为下一步动作做好准备。
 - 4) ①, ③, ⑤ 处于支撑态前进/后退一步, ②, ④, ⑥ 不动, 通过这一步, 机器人达到前进/后退效果。
 - 5) ①, ③, ⑤ 处于支撑态不动, ②, ④, ⑥ 抬起后前进/后退一步。这一步是为了让 ②, ④, ⑥ 腿处于目标位置, 便于下一步下落后机体做好前进/后退准备。
 - 6) ①, ③, ⑤ 不动, ②, ④, ⑥ 下落, 此时 ②, ④, ⑥ 在目标位置就绪。
 - 7) ①, ③, ⑤ 抬起, ②, ④, ⑥ 保持不动。此时机体由 ②, ④, ⑥ 腿支撑, 且 ②, ④, ⑥ 腿处于目标位置。
 - 8) ①, ③, ⑤ 抬起后保持不动, ②, ④, ⑥ 处于支撑态前进/后退一步。通过这一步, 机体完成前进/后退动作。
- 可以发现, 在循环阶段第 8 步的结束状态即为第 1 步开始前的状态, 因此通过程序中设置循环的次数即可控制机器人前进步数。
- 停止阶段: 即在循环中, 接到停止命令, 使机器人回到最初状态, 为后续左右移动, 转弯做好准备。该阶段的控制策略与初始化阶段步骤相似, 只需要按照与初始化阶段相反的执行动作即可。
- 对于左移/右移, 左转/右转的控制策略同前进/后退, 按照机器人实际运动方式进行控制即可, 在此不再赘述。
- 编程时, 以足端坐标的变化范围为控制变量, 将行走过程分解为多步, 每一步对应相应的坐标变化, 使得程序

能够按顺序执行下去。以其中一步为例说明:

```
while((z1>10)&&(z3>10)&&(z5>10))
{
    semTake(semIdForward, WAIT_FOREVER); //创建信号量
    formula(y1,z1,y2,z2,y3,z3,y4,z4,y5,z5,y6,z6); //调用算法函数,y1,z1表示第一条腿足端坐标,以此类推
    fuzhi(); //解算关节值赋给结构体
    sendNum=send(clientSock_duozhou,(char*)&crawl, sizeof(crawl),0); //网络发送
    DataRecv_duozhou(); //接收电机返回值
    z1--,z3--,z5--; //坐标变化
}
```

3.2 软件实现流程图

本文以前进/后退为例,说明三足步态下机器人的控制方式,具体流程如图 8 所示。

4 实验结果与分析

本文对正逆运动学算法采用 Matlab 仿真,利用其中的 Simulink 模块,将正逆算法公式进行搭建,输入为各腿在机体坐标系下的坐标,输出为对应坐标下各关节角度,具体步骤为:

1) 确定各腿在机体坐标系下 x, y, z 坐标,该坐标的确定方法为:根据腿部各关节长度大小,以及期望各关节中心角度,利用正运动学解公式,大概解算出各坐标大小,在该坐标附近确定一个值,通过逆运动学仿真得到该腿行走一步所经过的抬起,平移,落地,支撑推进阶段所需要的关节角度为平滑曲线,则该坐标可以作为机器人动作点。

2) 在机器人行走时,某些点在其行走过程中可以作为不动点,比如前进/后退时,各腿 y 坐标保持不动;左移/右移时, x 坐标保持不变。仿真时,变化的坐标可以通过系统按时间生成,所得角度即为机器人运动控制所需的角度。

3) 机器人运动时点的选取原则为理论和实验结合,保证机器人运动所需的角度合理即可,同时要满足支撑态时足端距跟关节在水平面的投影大于设定的平行间距。

4) 步长的大小要根据机器人尺寸来确定,即 d_2 的大小,若步长太大,则会超出机器人运动摆幅极限,左前腿向后摆动,左中腿向前摆动时,产生物理干涉,机器人无法运动;若步长太小,则机器人行走速度慢,电机做了大量的无用功,理论上来说,机器人支撑所走的一个步长不能超过 d_2 的大小,考虑到机械腿有一定的宽度,实际上来说机器人支撑所走的一个步长不能超过 $0.8 * d_2$ 的大小。平行间距的确定:前进/后退时,若各腿足端坐标确定,间距的大小等于对应 y 坐标值减去跟关节距机体坐标系 y 轴的值。

基于此,本文给出腿 2 处于支撑态前进一步的角度仿真曲线,设定机器人支撑行走时一个步长为 14 cm,所以腿 2 足端坐标 x_{p2} 在机体坐标系下的变化范围可以是 7 cm 至 -7 cm (坐标变化起止的范围可以不同,但为了便于观察机器人走的步长以及直观看逆运动学解的对称性,所以建议以各腿 y 轴为中心进行摆动),给定机体坐标系下 $y_{p2} =$

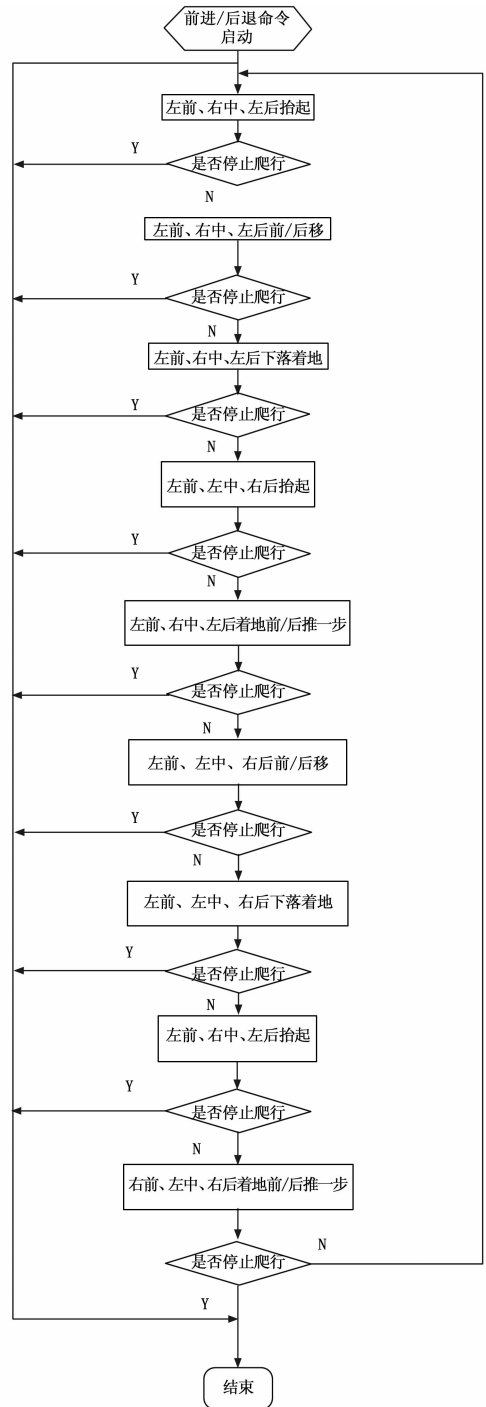


图 8 三足步态控制策略

22 cm (由于 D 的存在,走过一步轨迹平行于前进方向,所以 y_{p2} 为一个定值), $z_{p2} = -10$ cm。得到 $\theta_1, \theta_2, \theta_3$ 的仿真曲线如图 9 所示。

实验发现 Simulink 解算出的角度值与理论分析的一致,将角度再带入正运动学方程,得到值即为设定的坐标值。将此算法通过 c 语言编程控制电机运动,机器人可以实现相应的步态行走,在前进/后退实验室时,机器人两侧腿处于支撑态走出的轨迹曲线大致相互平行。

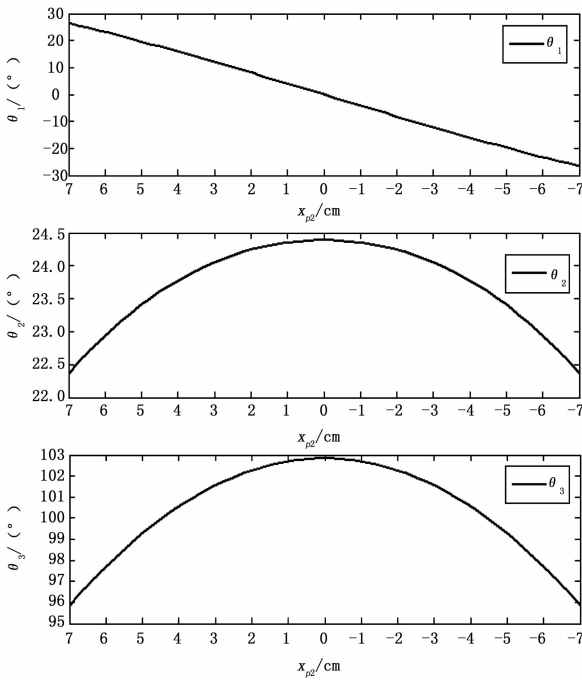


图 9 $\theta_1, \theta_2, \theta_3$ 随 x_{p2} 在机体坐标系中的变化情况

5 结论

本文建立了机器人的运动学模型，在原有基础上对腿部各角度运动控制算法进行了改进，给出了解算机器人空间位置坐标下所需要的各关节角度的方法，即机体逆运动学解；以及给出各关节角度下，足端所能达到的空间位置，即机体正运动学解。通过 Simulink 仿真实验和 c 语言编程控制机器人运动，证实了理论的正确性。

(上接第 55 页)

[2] Sivaraman S, Trivedi M M. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis [J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14 (4): 1773-1795.

[3] 季凡. 自助交费占道停车管理系统的关键技术 [J]. 黑龙江交通科技, 2016, 39 (11): 145-146.

[4] Kahl F, Hartley R, Hilsenstein V. Novelty detection in image sequences with dynamic background [A]. Proceedings of the Conference on Statistical Methods in Video Processing [C]. Berlin: Springer-Verlag Berlin, 2004: 117-128.

[5] Costantini G, Casali D, Perfetti R. Analogic CNN algorithm for estimating position and size of moving objects [J]. International Journal of Circuit Theory and Applications, 2004, 32 (6): 509-522.

[6] 王卫锋, 黄翰, 郝志峰, 谭志标. 仿昆虫复眼的交通视频停车事件检测方法 [J]. 计算机工程与应用, 2012, 48 (6): 246-248.

[7] 赵敏, 张波, 唐毅, 等. 露天场景下高速公路违章停车视频检测算法 [J]. 计算机测量与控制, 2014, 22 (2): 489-492.

参考文献:

[1] 陈甫. 六足仿生机器人的研制及其运动规划研究 [D]. 哈尔滨: 哈尔滨工业大学, 2009.

[2] 王倩. 六足仿生机器人步态规划与控制系统研制 [D]. 哈尔滨: 哈尔滨工业大学, 2007.

[3] 王倩, 陈甫, 臧希喆, 等. 新型六足机器人机构与控制系统设计 [J]. 机械设计与制造, 2008, 205 (3): 148-150.

[4] 刘昊. 六足机器人自主爬行步态及姿态控制研究 [D]. 哈尔滨: 哈尔滨工业大学, 2014.

[5] 刘鹏飞. 六足机器人步态分析与控制研究 [D]. 秦皇岛: 燕山大学, 2015.

[6] 徐德民. 鱼雷自动控制系统 [M]. 西安: 西北工业大学出版社, 2001.

[7] Garcia E, Jimenez M A, Santos P G D, et al. The evolution of robotics research [J]. IEEE Robotics and Automation Society, 2007, 4 (1): 90-103.

[8] 罗庆生, 韩宝玲. 现代仿生机器人设计 [M]. 北京: 电子工业出版社, 2008.

[9] 张晓峰. 基于 MATLAB 仿壁虎机器人步态规划研究与运动仿真 [D]. 南京: 南京航空航天大学, 2010.

[10] 陈学东, 孙翊, 贾文川. 多足步行机器人运动规划与控制 [M]. 华中科技大学出版社, 2006.

[11] 苏军. 多足步行机器人步态规划及控制的研究 [D]. 武汉: 华中科技大学, 2004.

[12] 徐凯. 自主步行机器人运动控制及相关研究 [D]. 北京: 中国科学技术大学, 2008.

[13] 储忠. 六足仿生机器人的运动步态研究与控制系统设计 [D]. 合肥: 合肥工业大学, 2007.

[14] 陈甫, 臧希喆, 闫继宏, 等. 适合航行的六足仿生机器人 Spider 的研制 [J]. 吉林大学学报 (工学版), 2011, 41 (3): 765-770.

[8] Carletti V, Foggia P, Greco A, et al. Automatic Detection of Long Term Parked Cars [A]. Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance [C]. Karlsruhe: IEEE Computer Society, 2015: 1-6.

[9] Wu Q, Huang C H, Wang S Y, et al. Robust parking space detection considering inter-space correlation [A]. Proceedings of the IEEE International Conference on Multimedia & Expo [C]. Hong Kong: IEEE Societies, 2007: 659-662.

[10] Huang C C, Wang S J. A hierarchical Bayesian generation framework for vacant parking space detection [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20 (12): 1770-1785.

[11] 姜明新, 王洪玉, 沐方顺. 基于视频监控的违章停车检测算法 [J]. 计算机工程, 2012, 38 (19): 151-153.

[12] 朱丽丹. 无线地磁技术在城市交通管理方面的应用 [J]. 信息安全与技术, 2013, 4 (3): 83-85.

[13] 王维锋, 万剑, 谢斌, 等. 基于矢量运算和多级阈值判断的地磁车位占用检测算法研究 [J]. 交通信息与安全, 2016, 34 (5): 53-60.

[14] 茅嘉磊. 基于视频分析的路侧停车自动计时取证 [J]. 上海船舶运输科学研究所学报, 2017 (1): 78-81.