六足机器人的步态分析与稳定行走

张子昕　韩旭东　李星泽　晏寓帅

摘要：自然界存在众多利用腿行走的生物，其各自有着独特的行走方式和步态。我们将使用 ROBOTICS PREMIUM 构建六足机器人，通过逆运动学对机器人的腿部关节值进行计算，得到所需要的舵机角度，结合 MATLAB 仿真，实现常见的三脚步态、涟漪步态和波动步态三种行走步态，以及原地旋转的步态，并对几种步态进行测试并对结果进行数据整理，最终实现机器人的平稳运动。

关键词：六足机器人；步态；逆运动学；稳定行走

中图分类号：TG156

Gait Analysis and Stable Walking of Hexapod Robot

ZHANG Zixin　HAN Xudong　LI Xingze　YAN Yushuai

Abstract：We use ROBOTICS PREMIUM to build hexapod robot, four kinds of gaits which are tripod gait, ripple gait planning, wave gait, In-situ rotating are designed and analyzed. Through inverse kinematics analysis, we get the value of robotic leg joint and the required steering angle, combined with MATLAB simulation, we test the several gaits, then sort and analyze the data gained from the result, finally to achieve the smooth movement of the robot.

Key words：Hexapod Robot；Gaits；Inverse kinematics analysis；stability

1　项目背景

在现实中，存在许多不适合人类工作的场合，如行星表面、自然灾害现场等，对于这些具有高危险性、高不确定性的场合，机器人便成为了代替人类的十分适合的选择。一般情况下，这些场景中都含有复杂的非结构化地形，同时包含各种各样的障碍物。此时，轮式和履带机器人具有一定的局限性，而足式机器人则在这种场景下体现出了强大的移动能力。同时，足式机器人还具有一些极其特殊的优势。例如在发生了核泄露的核电站中，充满了放射性的尘埃遍布整个环境中。当使用轮式或者履带式机器人进行场景勘探的时候，会产生较大的扬尘，从而导致机器人很快就会出现故障。但是足式机器人与地面的接触面积小，接触时间短，在行走过程中几乎不会产生扬尘，同时还可以很好地应对楼梯这样的地形。从某种角度来说，足式机器人就是为极端环境而生的。

常见的足式机器人包括双足机器人、四足机器人以及六足机器人。在常见的四足机器人和六足机器人中，每条腿通常包含三个自由度，多足之间相互配合，可以实现稳定的行走步态，以及身体姿态（如重心位置、高度、身体角度等）的自由调整。

六足机器人利用六条腿行走，由于机器人可以在三条或更多条腿上静态稳定，所以六足机器人在移动方面具有很大的灵活性。如果其中双腿不能运动，机器人仍然可以走路。同时，这也意味着六足机器人的步态设计拥有较大的自由度。根据以往的研究表明，六足机器人是最有效的稳定行走机器人，Preumount 等人在 1991 年的研究中得出，超过六足后，更多的腿并不会增加足式机器人步行的效率。此外，六腿的优势是并不是所有的机器人的腿都需要用来维持稳定性; 某些腿可以自由地达到新的位置或操纵有效载荷。六足机器人的运动激发了许多六足式机器人的生物学启发。六足动物也可以用来测试关于昆虫运动，运动控制和神经生物学的生物学理论。

在此次课程项目中，我们使用 ROBOTIS 机器人开发套件，搭建了一套六足机器人测试平台。同时，我们使用运动学知识，设计了三种六足行走步态以及一种原地旋转步态，并且保证机器人在以上的运动过程中身体部分重心在 z 轴上的理论起伏为 0，达到理想中的最稳定行走状态。最后，我们利用 ROBOTIS 配套软件进行了硬件上的步态实现，以验证我们的步态计算结果。

2 运动学分析

2.1 机器人腿部模型分析

首先，我们需要对六足机器人的每条腿进行运动学建模。通过使用 ROBOTIS 官网提供的 SOLIDWORKS 零件模型，我们可以在软件中搭建出机器人的完整模型。此六足机器人的每条腿的机械结构完全相同，由三个舵机够成，共包含三个自由度。当所有舵机处于零初始位置时，机器人的姿态如图 1 所示。

在进行运动学建模的过程中，若想要将关节变量直接设置为舵机从零位置转过的角度，各个关节上坐标系之间的关系将会变得异常复杂且不符合常见约定。因此，为了简化模型，我们在分析过程中直接将各个关节相连，视作一简单三连杆系统，如图 1 所示。

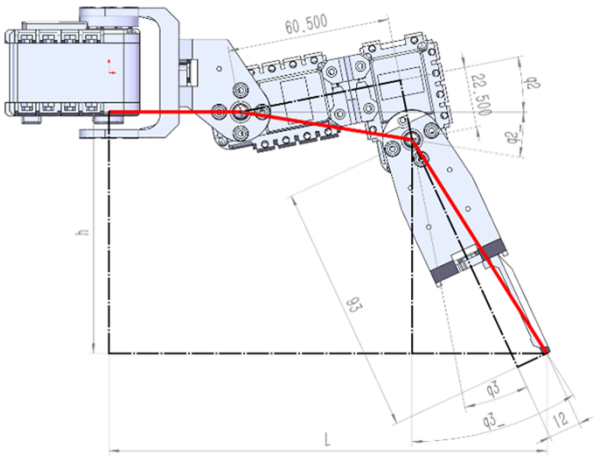


图 1　腿部结构简化后数学模型（侧视图）

分别将图 1 中从左到右的三个舵机称作髋部舵机（1 号）、大腿舵机（2 号）以及小腿舵机（3 号），分别对应关节 1、2 及 3，它们之间从左到右依次为连杆 1、2 及 3，同时我们具有以下基本约定：

1 号舵机从零位置转过的角度	q_1
2 号舵机从零位置转过的角度	q_2
连杆 2 的关节变量	q_2'
3 号舵机从零位置转过的角度	q_3
连杆 3 的关节变量	q_3'
连杆 1 起始处与足尖的垂直距离	h
连杆 1 起始处与足尖的水平距离	L

根据几何关系，我们可以得到q_2'与q_2，以及q_3'与q_3的固有关系式：

$$\begin{cases} q_2 = \arctan\dfrac{22.5}{60.5} - q_2' \\ q_3 = q_3' - q_2 - \arctan\dfrac{12}{93} \end{cases}$$

2.2 初始姿态的设计与分析

首先，我们对机器人的六条腿进行编号（1~6），并且间隔地分为两组（1、3、5 以及 2、4、6）。为了使得机器人在静止状态下具有最稳定的姿态，我们要求每一组腿的足尖的落地点构成等边三角形，如图 2 所示。图中的黑色箭头为机器人的行进方向。同时，为了简化计算，当机器人处于初始状态的时候，有$q_2' = 0$，$q_3' = 0$，此时腿部的侧视图如图 3 所示。

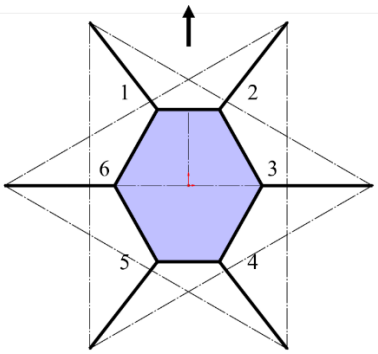


图 2　腿部初始姿态（俯视图）

为了计算初始姿态，我们需要计算出每一条腿的关节变量。首先考虑髋部舵机需要转过的角度q_1。对于 3 号与 6 号腿而言，它们在初始姿态下的$q_1 = 0$；而 1、2、4 及 5 号腿的q_1不为 0，且具有相同的绝对值。利用等边三角形三边相等的基本性质，以及在 SOLIDWORKS 软件中测量出的机器人理论固有尺寸，我们可以计算出$|q_1|$的数值。

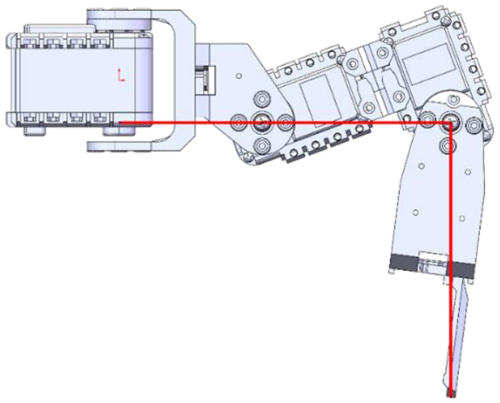


图 3　腿部初始姿态（侧视图）

接下来，我们需要计算出每一条腿的q_2与q_3。由于保持机器人身体平衡的需要，每一条腿的侧视图应当完全相同。由$q_2' = 0$，$q_3' = 0$以及它们与q_2和q_3的关系式，我们可以得到大腿以及小腿舵机转过的角度。

在得到理论数值之后，我们需要将其转化为舵机控制数值。舵机的可旋转角度范围为300°，被均匀地分割为 1024 等份，其中零位置对应的控制量数值为 512。因此，若电机相较于零位置转过的角度为$|q|$，我们可以得到输入到程序中的控制量数值c与$|q|$之间的关系：

$$c = 512 \pm \frac{1024|q|}{300}$$

其中，512 后是加号或减号取决于舵机的默认设置以及旋转的方向。实际初始姿态如图 4 所示。

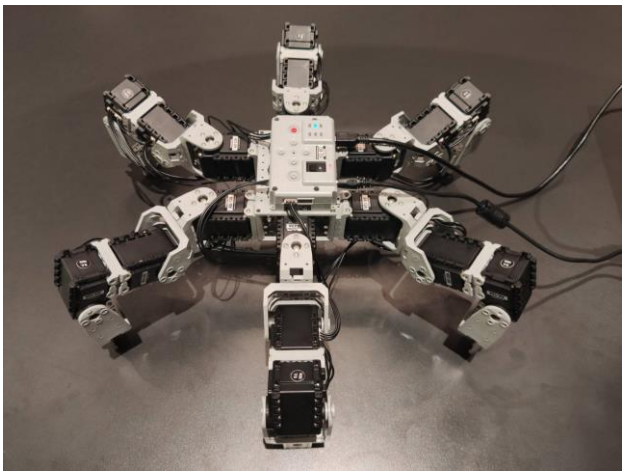


图 4　机器人初始姿态

2.3　腿部的运动分析

在对行进过程中的六足机器人进行腿部运动学分析之前，需要明确我们对机器人运动提出的要求：机器人的身体部分在行进过程中，重心的 z 轴起伏为 0，同时支撑腿的足尖位置保持不动。机器人的没一条腿的共同的运动特征是，当机器人的身体部分向前进行直线运动时，每一条支撑腿的足尖相对于身体重心的运动轨迹都是一条与运动方向相反的直线。

在机器人进行直线行走的过程中，我们可以根据运动特征将腿分成前后腿与侧面腿两种类型。前后腿在运动过程中，会相对于初始位置进行前后的不对称摆动；而侧面腿则会相对于初始位置进行对称的摆动，如图 5 和图 6 所示。为了便于计算，我们具有以下约定（其中的“前”和“后”为相对于运动方向而言）：

前后腿 1 号舵机相对身体以及初始姿态向前摆动的角度	α
前后腿 1 号舵机相对身体以及初始姿态向后摆动的角度	β
侧面腿 1 号舵机相对身体以及初始姿态向前或后摆动的角度	γ
足尖运动轨迹的长度，即机器人身体的向前行进距离	x
前后腿 1 号舵机在初始姿态下相对于零位置转过的角度	θ
前后腿初始姿态时的L	L_1
前后腿相对于身体向前摆动时的L	L_2
前后腿相对于身体向后摆动时的L	L_3
侧面腿相对于身体向前或后摆动时的L	L_4

通过几何关系以及三角学知识，我们可以列出以下式子计算L_2、L_3、L_4、α、β以及γ：

$$\begin{cases} L_2 = \sqrt{-2xL_1\cos\theta + x^2 + L_1^2} \\ \alpha = arc\sin\dfrac{x\sin\theta}{L_2},\ q_1 = \theta + \alpha \end{cases}$$

$$\begin{cases} L_3 = \sqrt{L_2^2 + 4x^2 - 4L_2x\,cos(\pi - \alpha - \theta)} \\ \beta = arc\,sin\dfrac{x\,sin(\pi - \theta)}{L_3},\ q_1 = \theta - \beta \end{cases}$$

$$\begin{cases} L_4 = \sqrt{x^2 + L_1^2} \\ \gamma = arc\,tan\dfrac{x}{L_1},\ q_1 = \gamma \end{cases}$$

其中，α、β以及γ与 1 号舵机的旋转角度q_1以及初始姿态角θ的关系分别为：

$$\begin{cases} \theta + \alpha + q_1 = \dfrac{\pi}{2} \\ \theta - \beta + q_1 = \dfrac{\pi}{2} \\ q_1 = \gamma \end{cases}$$

同时，为了保证机器人的重心在 z 轴上的起伏为 0，我们必须保证每一条支撑腿在运动过程中的h保持不变。由此，我们可以获得以下方程式，计算

出：

$$\begin{cases}\sqrt{60.5^2+22.5^2}\cos q_2' + \sqrt{93^2+12^2}\sin q_3' = L_i \\ \sqrt{60.5^2+22.5^2}\sin q_2' + \sqrt{93^2+12^2}\cos q_3' = h\end{cases}$$

根据q_2'与q_2，以及q_3'与q_3的固有关系式，我们进而可以计算出q_2以及q_3，即舵机转过的角度。

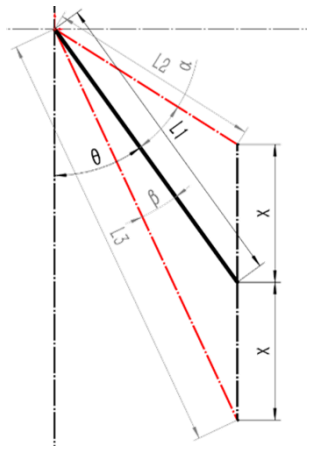


图 5　前腿和后腿简化后数学模型（俯视图）

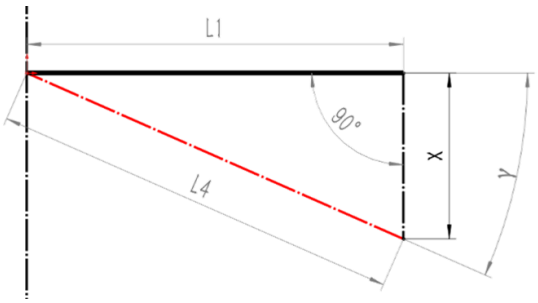


图 6　侧腿简化后数学模型（俯视图）

在以上过程中，我们使用机器人身体前进的距离x，以及机器人的固有几何尺寸，对前后腿以及侧面腿分别进行逆运动学分析。首先，我们利用x分别计算出每一条腿运动到目标位置时相对于地面的投影长度，即L，以及髋部（1 号）舵机的关节变量；然后，我们首先使用L计算出简化的连杆模型的关节变量q_2'以及q_3'，然后利用几何关系，进而计算出 2 号和 3 号舵机转过的角度q_2以及q_3；最后，我们将每一条腿的q_1、q_2以及q_3转化为对应的舵机控制量数值。

2.4　MATLAB 程序实现

我们将以上逆运动学的计算过程写入到了一个 MATLAB 程序当中（代码见附录）。我们只需要输入机器人的前进距离x，即可运算出所有舵机运动到目标状态所需要的角度。运行结果如图 7 所示。

```
theta = 35.960769 degree = 634.746091 or 389.253909

Side leg:
q1 = 16.360155 degree = 567.842662 or 456.157338
q2 = 20.291467 degree = 581.261539 or 442.738461
q3 = -24.714759 degree = 427.640288 or 596.359712

Support front leg or swing back leg (move inward relative to body):
q1 = 48.702005 degree = 678.236178 or 345.763822
q2 = 17.457007 degree = 571.586585 or 452.413415
q3 = -40.088106 degree = 375.165931 or 648.834069

Swing front leg or support back leg (move outward relative to body):
q1 = 28.031029 degree = 607.679247 or 416.320753
q2 = 16.462485 degree = 568.191949 or 455.808051
q3 = -6.127529 degree = 491.084700 or 532.915300
```

图7　MATLAB程序运行结果

3　步态规划

在本节中，我们规划实现了原地旋转、三脚步态、涟漪步态和波动步态四种步态。

3.1　原地旋转

对于原地旋转，我们采用了三条腿抬起旋转落下后，交换另外三条腿抬起旋转落下的策略。这种方式简单高效，最快可在 10 秒内旋转一圈。原地旋转示意图如图 8 所示。

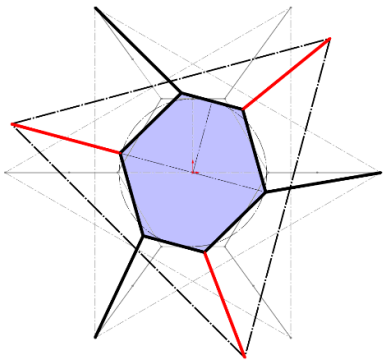


图 8　原地旋转示意图

3.2　三脚步态

在三脚步态中，最多有三只脚同时离开地面。以三只脚为一组，交替向前行进。图 7 展示了三脚步态中各条腿的时序描述。这一步态是三种行走步态中速度最快的，效率较高，适合高速轻荷载情况下使用。

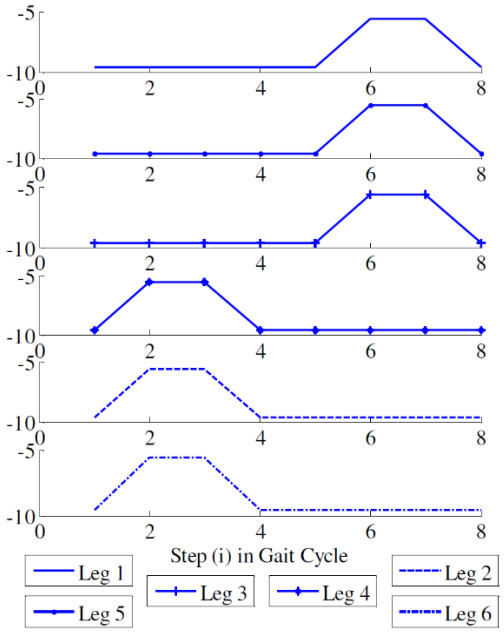


图 9　三脚步态

3.3　涟漪步态

在涟漪步态中，最多有两只脚离开地面。可以看到，涟漪步态的特点是同时离开地面的两只脚分别位于不同侧，且不会相邻。图 8 展示了涟漪步态中各条腿的时序描述。这一步态的速度介于三脚步态和波动步态之间。

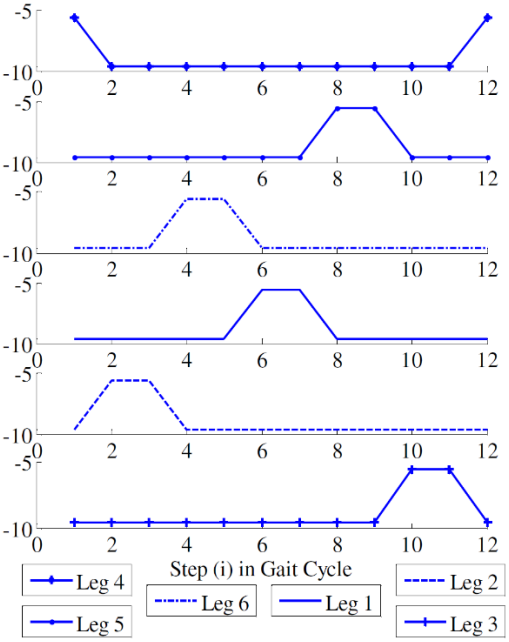


图 10　涟漪步态

3.4　波动步态

在波动步态中，最多只有一条腿离开地面，六条腿交替循环抬起、向前移动、落下，从而带动身体向前移动。图 9 展示了波动步态中各条腿的时序描述。这一步态是三种步态中速度最低的，但是稳定性是最好的，在机器人载荷较大或者慢速移动时使用。

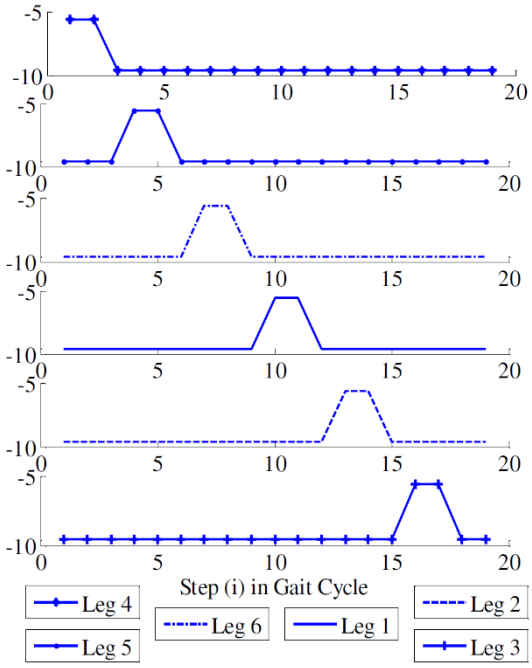


图 11　波动步态

在步态规划过程中，为方便理解三种行走步态下各条腿的运动状态，我们使用 MATLAB 对步态进行了仿真，如图 13 所示。图中，红色代表处于抬起运动状态的腿，黑色代表处于落地状态的腿。从图中，可以清晰地看出三种步态下各条腿的连续运动谱，以及相互之间的差异。我们可以清楚地看到，完成一个周期，三脚步态所需要的时间最少，因为其同时有三只腿向前移动；涟漪步态耗时次多；波动步态耗时最多。因而在设置每只腿单次前进相同距离的情况下，三脚步态可在相同时间内行走更远距离，即效率最高；涟漪步态次之；波动步态效率最低。但同时，越多腿着地，机器人越稳定，因而波动步态是最稳定的。在实际情况中，我们既要考虑效率，同时也要考虑稳定性。

为提高所有步态的稳定性，我们对各类步态进行了处理，通过反算舵机角度来较为精确地控制每条腿运动过程。经过优化的步态，在实际实验中，可以实现上下浮动不超过 5%（图 12），基本实现了平稳运动。

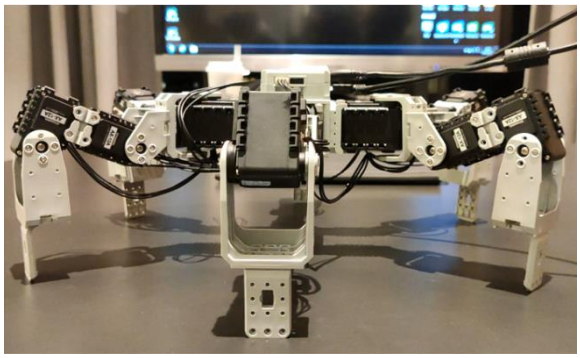
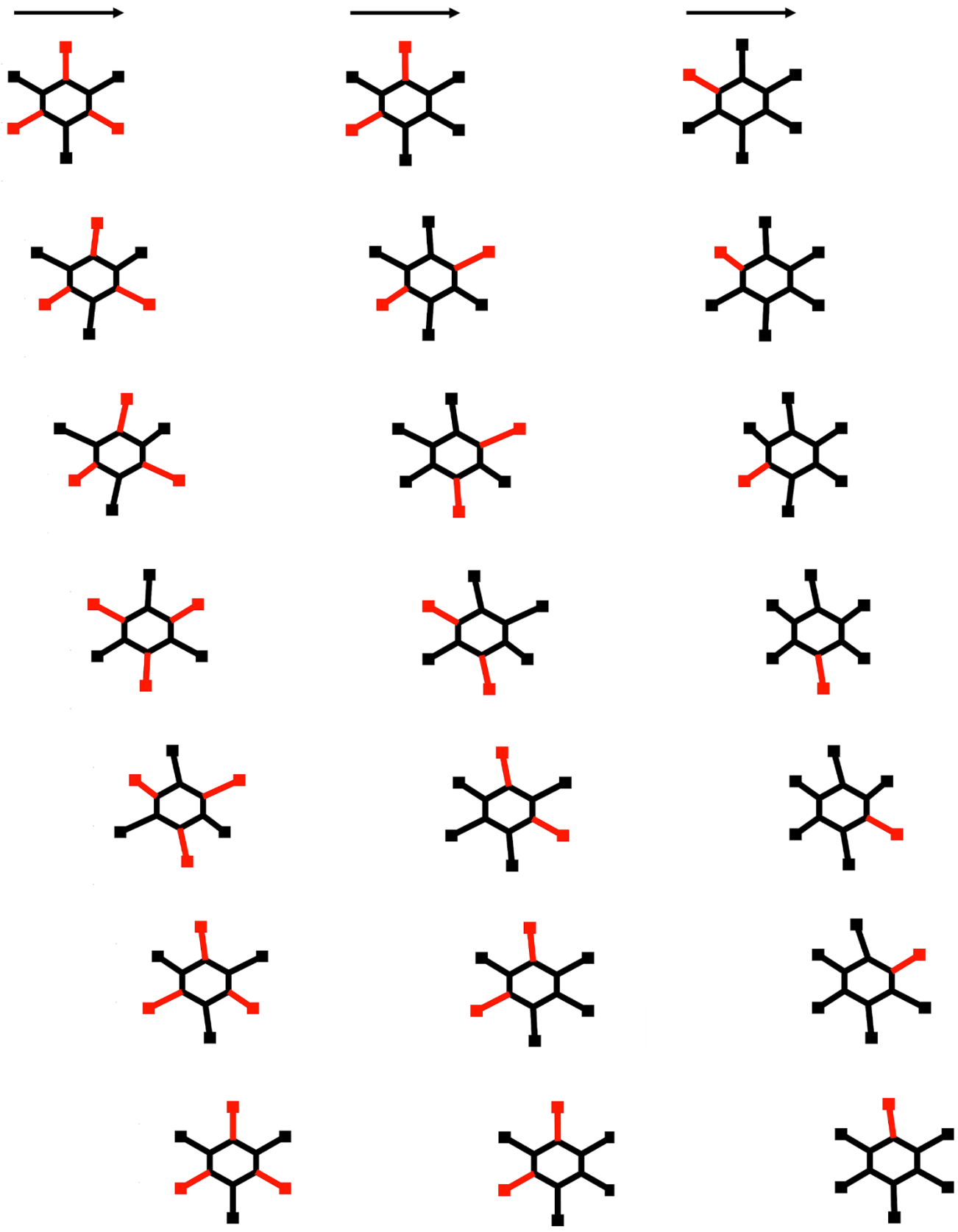


图 12　六足机器人的平稳运动



（a）三脚步态　　（b）涟漪步态　　（c）波动步态

（红色代表处于抬起运动状态，黑色代表处于落地状态）

图 13　三种步态的仿真和比较

4 结论

本项目中，我们利用 ROBOTIS PREMIUM 平台构建了六足机器人，并受自然界已知的腿行走生物的启发，对该六足机器人的逆运动学分析，结合 MATLAB 仿真，实现了原地旋转和常见的三脚步态、涟漪步态和波动步态，并对三种行走步态进行了优化，实现了平稳行走。

在本项目中，大量应用了逆运动学分析，但机器人步态仍不够平滑，这主要受限于 ROBOTIS PREMIUM 平台的可操作性，但基于此平台已经实现了逆运动学分析和步态规划及验证，后续或可利用其他手段、平台获得更好效果。

总的来说，本项目的研究成果有助于评估常用步态的执行效率，并提供了多种步态下平稳行走的解决方案，这将启发在真实场景中六足机器人不同运动模式下的稳健行走。

5 时间线与组员的分工情况

5.1 时间线

第十一周：确定机器人构型，完成平台搭建，并查找相关文献资料。

第十二周：逆运动学分析计算，并进行 Robotics 平台测试。

第十三周：进行三种步态规划与模拟。

第十四周：进行机器人关节数据录入与优化。

第十五周：进行最终测试并对结果进行整理。

5.2 组员分工

张子昕：逆运动学分析和步态规划。

韩旭东：仿真和运动实现。

李星泽：关节值计算与优化。

晏寓帅：机器人模型搭建和实现。

参 考 文 献

[1] U. Asif, J. Iqbal and M. Ajmal Khan, Kinematic analysis of periodic continuous gaits for a bio-mimetic walking robot, 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, 2011, pp. 80-85, doi: 10.1109/SSRR.2011.6106784.

[2] Spong M W , Hutchinson S , Vidyasagar M . Robot Modeling and Control[J]. Industrial Robot An International Journal, 2006, 17(5):709-737.

附 录

```
clear, close, clc

syms q1 q2 q3 q1_ q2_ q3_ theta real

theta1 = atan(22.5/60.5);
theta2 = atan(12/93);
x = 50/6*4; % moving distance
L1 = 49 + sqrt(60.5^2+22.5^2);

eqn = sqrt(3) * (L1 * cos(theta) + 74.5) == L1 + 2 * 32 + 44 +
      L1 * sin(theta);
theta = solve(eqn, theta, 'Real', true);
theta = double(theta(1));

fprintf('theta = %f degree = %f or %f\n\n', rad2deg(theta), 512 +
      angle2val(theta), 512 - angle2val(theta));

%% SIDE LEG
q1 = atan(x/L1);
eqn1 = 49 + sqrt(60.5^2+22.5^2) * cos(q2_) + sqrt(93^2+12^2)
      * sin(q3_) == sqrt(x^2+L1^2);
eqn2 = sqrt(60.5^2+22.5^2) * sin(q2_) + sqrt(93^2+12^2) *
      cos(q3_) == sqrt(93^2+12^2);
eqns = [eqn1, eqn2];
S = solve(eqns, [q2_ q3_]);

% Select the best q2_ and q3_
if abs(S.q2_(2)) <= abs(S.q2_(1))
    q2_ = double(S.q2_(2));
    q3_ = double(S.q3_(2));
else
    q2_ = double(S.q2_(1));
    q3_ = double(S.q3_(1));
end

q2 = theta1 - q2_;
q3 = q3_ - theta2 - q2;

fprintf('Side leg:\n');
fprintf('q1 = %f degree = %f or %f\n', rad2deg(q1), 512 +
      angle2val(q1), 512 - angle2val(q1));
fprintf('q2 = %f degree = %f or %f\n', rad2deg(q2), 512 +
      angle2val(q2), 512 - angle2val(q2));
fprintf('q3 = %f degree = %f or %f\n', rad2deg(q3), 512 +
      angle2val(q3), 512 - angle2val(q3));
fprintf('\n');

%% SUPPORT FRONT LEG or SWING BACK LEG
syms q2_ q3_ real
L2 = sqrt(-2*x*L1*cos(theta)+x^2+L1^2);
q1_ = asin(x*sin(theta)/L2);
q1 = theta + q1_;
eqn1 = 49 + sqrt(60.5^2+22.5^2) * cos(q2_) + sqrt(93^2+12^2)
      * sin(q3_) == L2;
eqn2 = sqrt(60.5^2+22.5^2) * sin(q2_) + sqrt(93^2+12^2) *
      cos(q3_) == sqrt(93^2+12^2);
eqns = [eqn1, eqn2];
S = solve(eqns, [q2_ q3_]);

% Select the best q2_ and q3_
if abs(S.q2_(2)) <= abs(S.q2_(1))
    q2_ = double(S.q2_(2));
    q3_ = double(S.q3_(2));
else
    q2_ = double(S.q2_(1));
    q3_ = double(S.q3_(1));
end

q2 = theta1 - q2_;
q3 = q3_ - theta2 - q2;

fprintf('Support front leg or swing back leg (move inward
      relative to body):\n');
fprintf('q1 = %f degree = %f or %f\n', rad2deg(q1), 512 +
      angle2val(q1), 512 - angle2val(q1));
fprintf('q2 = %f degree = %f or %f\n', rad2deg(q2), 512 +
      angle2val(q2), 512 - angle2val(q2));
fprintf('q3 = %f degree = %f or %f\n', rad2deg(q3), 512 +
      angle2val(q3), 512 - angle2val(q3));
fprintf('\n');

%% SWING FRONT LEG or SUPPORT BACK LEG
syms q2_ q3_ real
L2 = sqrt(-2*x*L1*cos(theta)+x^2+L1^2);
q1_ = asin(x*sin(theta)/L2);
q1 = theta + q1_;
L3 = sqrt(L2^2+4*x^2-4*L2*x*cos(pi-q1_-theta));
q1_ = asin(x*sin(pi-theta)/L3);
q1 = theta - q1_;

eqn1 = 49 + sqrt(60.5^2+22.5^2) * cos(q2_) + sqrt(93^2+12^2)
      * sin(q3_) == L3;
eqn2 = sqrt(60.5^2+22.5^2) * sin(q2_) + sqrt(93^2+12^2) *
      cos(q3_) == sqrt(93^2+12^2);
eqns = [eqn1, eqn2];
S = solve(eqns, [q2_ q3_]);

% Select the best q2_ and q3_
if abs(S.q2_(2)) <= abs(S.q2_(1))
    q2_ = double(S.q2_(2));
    q3_ = double(S.q3_(2));
else
    q2_ = double(S.q2_(1));
    q3_ = double(S.q3_(1));
end

q2 = theta1 - q2_;
```

```
q3 = q3_ - theta2 - q2;

fprintf('Swing front leg or support back leg (move outward
    relative to body):\n');
fprintf('q1 = %f degree = %f or %f\n', rad2deg(q1), 512 +
    angle2val(q1), 512 - angle2val(q1));
fprintf('q2 = %f degree = %f or %f\n', rad2deg(q2), 512 +
    angle2val(q2), 512 - angle2val(q2));
fprintf('q3 = %f degree = %f or %f\n', rad2deg(q3), 512 +
    angle2val(q3), 512 - angle2val(q3));
fprintf('\n');
```