双足移动机器人技术及强化实践期末报告

组别：第3组

成员：林葳洁 3130104123

许蒙蒙 3130000580

蒋雪松 3130004184

杨 喜 3130103291

郎 园 3130101703

1. **任务描述：**

**1.1 基础任务：机器人行走与踢球。**

利用视觉模块识别乒乓球的方位，通过一套运动策略集控制机器人走近乒乓球，使机器人的右脚对着乒乓球，令机器人做出踢球动作踢到乒乓球。

**1.2. 扩展任务：机器人推箱子**

利用视觉模块识别箱子的方位以及箱子前表面的法向量，通过一套新的运动策略集控制机器人走近箱子，使机器人正对箱子表面，并且处在箱子前表面的中心位置。

接着配合手部动作规划进行推箱子。使用针对机器人与箱子相互作用的全新的步态规划（平均重心下蹲并前倾）使机器人推动箱子；计算推箱子时手部压力的相对大小，当箱子重量过重机器人无法推动时，停止推动以保护机器人的电机。

扩展任务的创新点与难点：

* 准确识别箱子前表面的法向量
* 运动中调整机器人的角度使机器人正对箱子表面且处在箱子前表面的中心位置
* 推动箱子时，在机器人与箱子相互作用下的姿态调整
* 检测推动时的手部压力

1. **运动学建模**

**2.1 足部正运动学建模**

双足机器人的建模已经有很成熟的模型，如下图：

a2

a1

a4

a3

a5

a6

a7

a10

a8

a9

a11

a12

a13

b5

b6

b8

a2

a2

b2

其中，有以下几点需要特殊注意：

* a2和a8两个关节轴矢量需要特殊注意，其指向并不是空间直角坐标的轴方向。
* 相对位置矢量的方向没有什么异议，但是关节周矢量的方向需要和实际物理模型对应，满足右手螺旋，以此确定电机的选择方向。

关节轴矢量，我们以如下的方向建立：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | /2 | 0 | 1 | 1 | 1 | 0 | /2 | 0 | 1 | 1 | 1 | 0 |
| 1 | /2 | 0 | 0 | 0 | 0 | 0 | /2 | 0 | 0 | 0 | 0 | 0 |

相对位置矢量：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | b12 | b13 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | -HY | 0 | 0 | 0 | 0 | 0 | HY | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | -ThL | -TiL | 0 | 0 | 0 | 0 | -ThL | -TiL | 0 |

注：HipOffsetY，ThighLength，TibiaLength





其中每一个θ对应的机器人关节为：

|  |  |  |
| --- | --- | --- |
| θ序号 | 关节角 | 简写 |
| 1 | 0 | 0 |
| 2 | RHipYawPitch | RHYP |
| 3 | RHipRoll | RHR |
| 4 | RHipPitch | RHP |
| 5 | RKneePitch | RKP |
| 6 | RAnklePitch | RAP |
| 7 | RAnkleRoll | RAR |
| 8 | LHipYawPitch | LHYP |
| 9 | LHipRoll | LHR |
| 10 | LHipPitch | LHP |
| 11 | LKneePitch | LKP |
| 12 | LAnklePitch | LAP |
| 13 | LAnkleRoll | LAR |

坐标系j相对于i的齐次坐标变换阵：



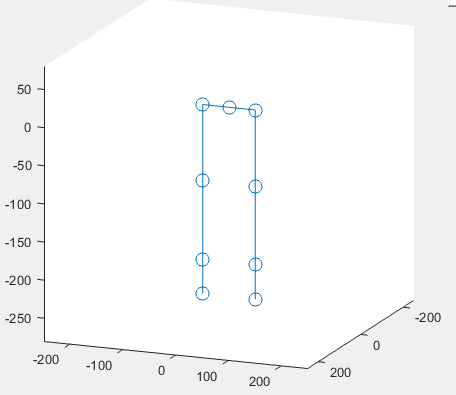
由上面的两个表格，我们可以找到所有的转移矩阵。

最后由链式法则，足底点的P4和P8可以写为：





同理，相应的其他点的坐标也可以写出来：

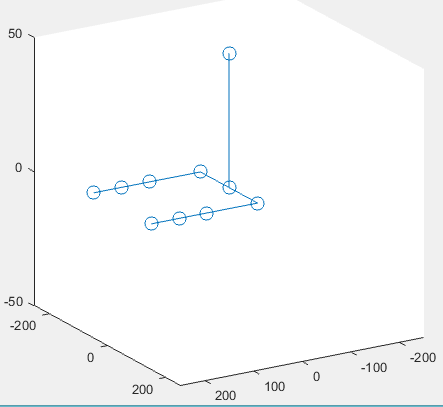


但是用于求解，我们只需足底的两个点。

**2.2 手部正运动学建模**

相应的，我们同时对手部进行正运动学建模。由于我们在计算中，没有考虑上肢运动对重心的影响，所以上肢和下肢的建模当做独立的两部分进行。

同理可得对于上肢的a和b向量，参见代码中，其初始状态如下：



**2.3 双足步态的生成**

当需要生成双足步行模式时，我们可以采用，采用三维线性倒立摆模型，或者强行将质心缓缓移动至支撑脚正上方，抬起另一只脚往前迈，着地后再将质心移动到这只脚，移动另一只脚。

这里的步态生成中，我们借鉴了学长的倒立摆模型的建模方式。

**2.3 线性倒立摆模型**

质心高度恒不变，相对能够模仿正常的行走动作，因此我们真正意义上采用了线性倒立摆模型，生成双足步行模式，并取得了比较理想的效果。

为了简化，三维线性倒立摆可以直接拆分成两个正交的二维线性倒立摆分析。这两个倒立摆之间存在简单的约束：运行周期。两个倒立摆分别体现了倒立摆的两种工作状态：轨道能量充足（能越过势能最大处）和轨道能量不足（不能越过轨道能量最大处）。

* X方向线性倒立摆模型：

两只脚作为支撑脚时轨道能量表达式：



在步行的过程中，我们很自然地令轨道能量相等，于是可以解出换脚时质心速度和位置：



因此我们收获了二阶微分方程的边界条件：



通过代入求解，可以获得质心速度控制序列和位置控制序列：



其中的参数定义如下：

步长：s

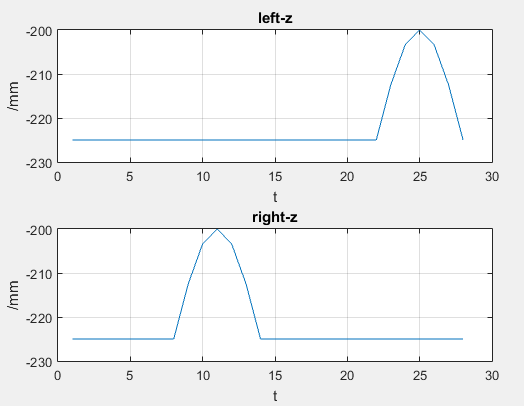
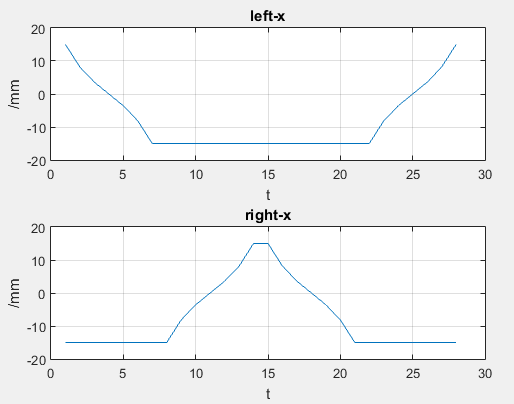
换脚速度：vf

重力加速度：g

质心高度：z

而事实上，作为完整的一步，有四个边界条件，我们用了起始的一组混合边界条件，还剩下一组终止时的混合边界条件，用来约束自由变量之间的关系，因此，当步长和换脚速度确定，则每步运行时间被确定下来。代入终止边界条件可以求得时间约束：





### Y方向线性倒立摆：

:重心

:足部与其朝向

Y方向的线性倒立摆模型处于欠驱动的状态，质心无法越过势能最高点，从而保证机器人不往支撑脚一侧摔倒。其大致流程如上图所示。

步行过程仍保证轨道能量相等，边界条件：



我们已经在前面讨论过，直接让直立时支撑脚的y坐标等于质心的y坐标就可以了。

求得质心位置控制律和速度控制律：



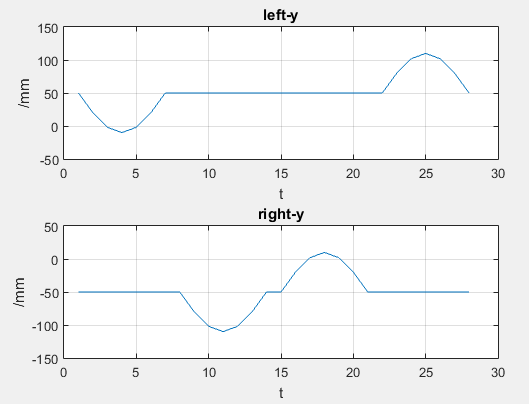
令得：

不难发现，每步运行时间是一个常数。

所以x方向的运行的时间也需要是这一个数，故换脚速度vf和步长s并不独立，满足以下约束：



通过计算，我们获得质心正在一条腿作为支撑腿的过程中，运动轨迹如下（右脚为支撑腿，横坐标y方向，纵坐标x方向，行进方向沿纸面向上）：



**2.4 非支撑脚步态生成**

对于非支撑脚的轨迹生成，我们一开始采用了取关键点（两步的换脚点和中间时刻的点），求逆运动学，然后在关节空间中进行插值。

这一步需要上机测试的时候注意其重心是否稳定，步态是否合理，没有固定的原则。

**2.5 摆臂动作的生成**

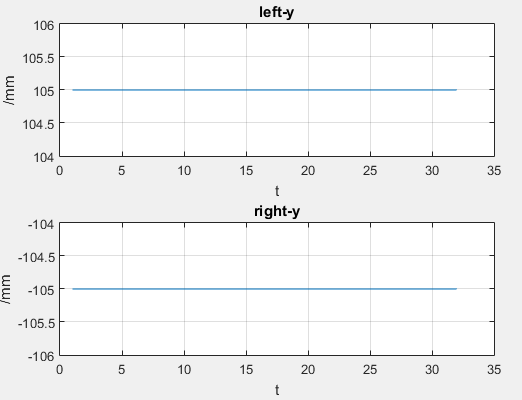
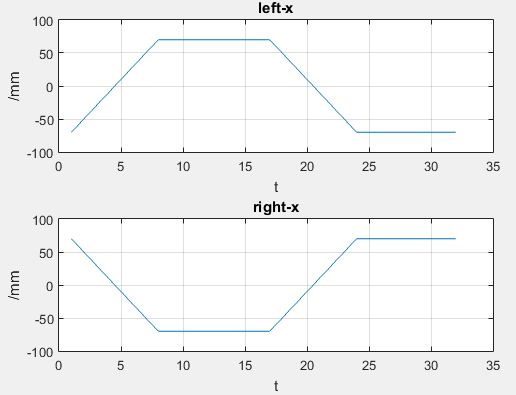
因为我们不考虑摆臂动作对重心的影响，所以我们只需要对应重心进行相应的路径规划即可：

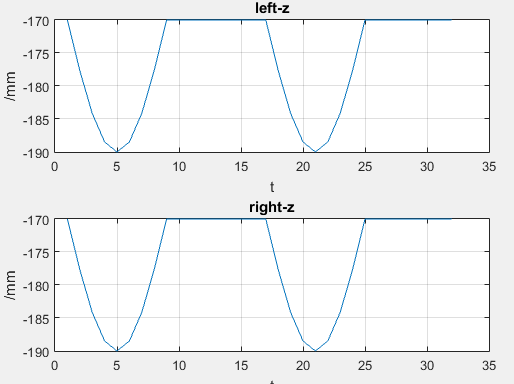
:重心

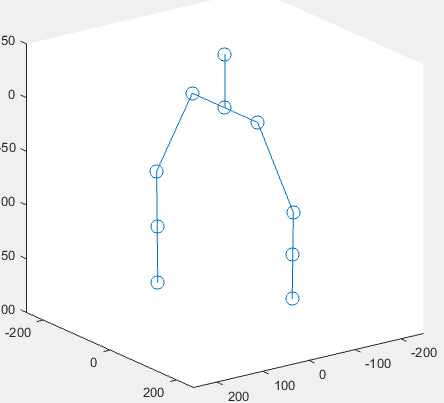
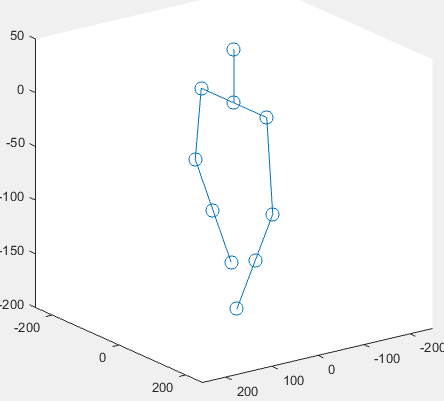
:足部与其朝向

:手部

大概的运动过程如上图所示，我们可以对齐在空间中的曲线进行插值处理，得到手部的空间轨迹，空间轨迹的三维坐标如下图：







**2.5 运动学求解**

2.5.1下肢关节角求解

从正运动学建模可以看出，如果要所有的角度都求解的话，我们需要求解RHipYawPitch、RHipRoll、RHipPitch、RKneePitch、RAnklePitch、RAnkleRoll六个角度。

但是，我们只有x、y、z三个坐标，也即三个方程，通过直接求解方程的方式很难计算我们想要的解。

于是这里我们进行了这样的运算：

RHipYawPitch=0

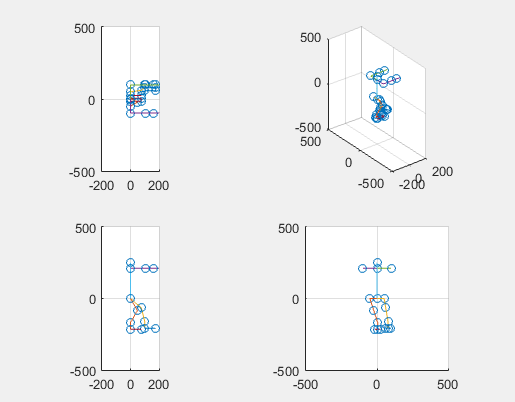
RAnkleRoll= -RHipRoll

RAnklePitch= - (RHipPitch+ RKneePitch)

其中，第一个为了简化固定为0，且经过检验不会影响其求解。

后两个限制条件为脚步的两个关节角，其目的是使脚底板与地面始终平行。可以看到，我们使脚步的Pitch等于腿部的Pitch相反数，抵消俯仰角，脚步的Roll等于腿部Roll的相反数，抵消翻转。

除此之外，其求解初值也是要考虑的问题，我们只需将已知在运动轨迹上的一组角度做初值即可求解我们想要得到的那一组解。



2.5.2上肢关节角求解

同理，最后在Matlab中对模型进行逆运动学求解，可以得到最终的结果。

这里我们在逆运动学的过程，只分别求解 RShoulderPitch，RShoulderRoll，REllbowRoll， LShoulderPitch，LShoulderRoll，LEllbowRoll（没边三个角）这几个关节角，其它的关节角锁定为0或者+-pi/2.

1. **运动策略：**

**3.1 步态集合**

在解决踢球和推箱子的任务之前，我们要专门设计几种特殊功能的步态，以备使用。

因为不允许实时在线解非线性方程，所以我们必须在实际测试之前，将一个周期的关节角参数存储到NAO中，封装成策略集合，在最后调用即可。

### 快步走和慢步走

快步走和慢步走是直线行走步态在给定速度不同时的两种解，具体解释在上面步态规划之中。

### 左右偏移

偏移指令要保证机器人在x方向上的位移为0，所以这种步态我们只需要调整好重心，使其始终在一只脚上。

:重心

:足部与其朝向

具体执行参见附件。

### 旋转

这一步步态的生成，是通过将上面直行步态规划的左右脚x方向偏移量设置成不同值来完成的，也就是说这里用的是差速的思想进行旋转。

我们尝试我进行理论计算角度以及旋转周期，但是受限于环境因素影响较大，且可用周期只有4个，所以我们采用了测量的方法。

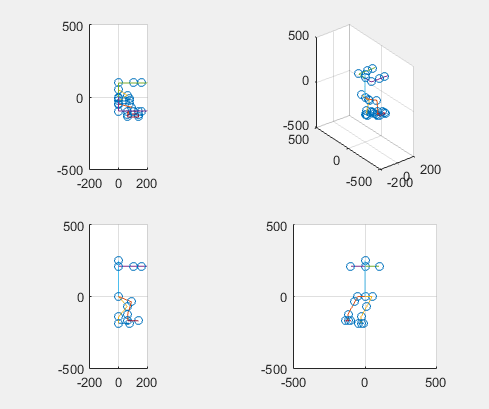
我们具体的测量值如下：

|  |  |
| --- | --- |
| 周期数 | 旋转角度 |
| 1 | 5°~15° |
| 2 | 15°~25° |
| 3 | 25°~35° |
| 4 | 35°~45° |
| 5 | 45°~55° |

转弯半径35cm。

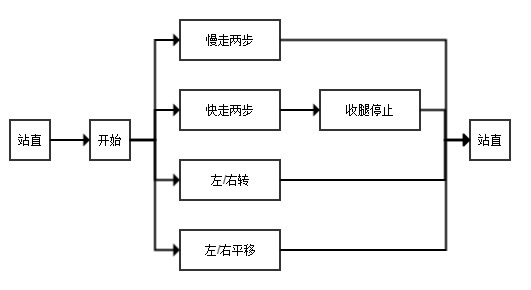
### 踢球

踢球是一个动作，只执行一个周期，其设计不具有一般性。简单来说，就是将重心固定在固定脚之上，对移动脚的空间曲线进行设计，实际测试保证重心稳定即可。



**3.2 步态之间的衔接**

经过实地调试，总结出推荐的步态衔接，如下图所示：



注意，不允许出现的步态衔接有：

* 左/右转弯->快走两步
* 收腿停止->快走两步

**3.3 推箱子的特殊步态**

3.3.1 推箱子路径规划

在机器人从起点开始探测到箱子后，记机器人所在点为坐标原点，机器人所面向方向为X轴正方向。视觉部分可以返回探测到箱子的中心坐标，和箱子被推面的法向量 。基于此，我们建立模型规划机器人接近箱子的路径。

由于模块化的运动学处理后，我们的机器人运动只考虑由快走，慢走，定半径差速左右向转弯与左右平移组成。由于左右平移动作较难看，也不利于机器人走出平滑的最短路径，因此我们决定采用定半径差速转弯与快走、慢走的组合来规划路径。

如图，以箱子位于坐标系第一象限为例。记箱子所在点为，箱子被推面的法向量为 。显然我们可以通过先左转弧度为的定半径圆弧，接着右转弧度为的定半径圆弧，最后直行到达面对箱子的正中心。其中最后的直行可以放到最开始，但是考虑到直行的误差较转弯小校正更容易，因此放到最后一段。

箱子

KX

θX

βX

α

Y

XX

θX

O

N

M

记转弯半径为R,由几何关系，易得满足下列条件。

解上式可得

利用合一变换，记

则有

于是，我们解得了一个在已有动作集上较优路径的规划，即先向左转弯弧度，接着向右转弯弧度，最后直走KN段。比较KN段长度与设定阈值的关系，若KN较大，则执行先快走后慢走的策略，若KN较小，则直接执行慢走策略。但是在实际执行中由于地面的打滑和机器人控制等原因，我们的定半径差速转弯误差较大，因此并没有完全按照此优化路径进行，而且加入了横向平移等校正策略，以更好地推到箱子。

3.3.2 手部姿态

推箱子时，为了保证箱子被平稳地推动，对手部的姿态有一定的要求。若抬手过高，则箱子会发生倾倒，若抬手过低，则足部会触碰到箱子。经过实地调试，测试出最合适的抬手姿态如下：

|  |  |  |
| --- | --- | --- |
|  | 角度 | 弧度 |
| LSP | 50 | 0.872665 |
| LSR | -3 | -0.05236 |
| LEY | -90 | -1.5708 |
| LER | -50 | -0.87266 |
| RSP | 50 | 0.872665 |
| RSR | 3 | 0.05236 |
| REY | 90 | 1.570796 |
| RER | 50 | 0.872665 |

3.3.3 躯体姿态

现象：当机器人推到箱子后，如果箱子过重，机器人的双手会受到较大的反作用力，造成ZMP投影点向足跟处移动，增加机器人向后摔倒的风险。

策略：

为了修回ZMP的位置，需要改变机器人在行走中的重心位置变化，使机器人受到的的重力力矩大于外力力矩，地面的支撑起到作用。

* 降低重心。通过足部规划时的经验可以知道，机器人重心距离地面越低。并且由力矩的定义可知，手臂接触点越低，受到的力矩越小。故我们使机器人中心下降，来获取更大的解空间和更小的力矩
* 重心前移。对应减小阻力力矩，我们增大了重力力矩，即通过将机器人的双足向后规划，并且脚掌向上翻转微小角度，使机器人的重心平均位置向前移动。
* 利用箱子的摩擦力矩。当有相对位移的趋势时，就会产生静摩擦力。机器人在移动中会受到较大的阻力，而在切换重心的时候收到的阻力较小。所以在对机器人前移的做足部轨迹规划时，使机器人的手部的高度略微上升但是，并且这里的上升要考虑到前脚掌向上旋转后，地面与机器人躯干不再垂直。通过实验，我们得到了机器人前移时前脚额外的竖直方向的移动速度参数。
* 对应躯干的旋转，机器人的手臂也要与旋转一定的角度，防止机器人的手臂的卡紧与变形。

3.3.4 箱子阻力的估计与角度传感器

现象：经过步态的重新规划，机器人不会因为推到过重的箱子而摔倒，但如果箱子过重，机器人会发生打滑或齿轮错位，这两种情况的出现会极大地损伤机器人。

策略：我们首先考虑到对机器人足部的压力传感器进行监视，但发现该值对箱子的重量不敏感，后来将监视对象改为肩部传感器。

当肩部收到挤压之后，机器人肩部发生形变，肩部的实际旋转角度也会发生变化。我们实际测量了肩膀收到一些压力时，角度的变化范围，并确定了当实际角度与目标角度相差0.2弧度左右时，会发生齿轮错位，故通过此条件反应机器人是否可以推动箱子。

**3.4 踢球运动策略**



踢球的运动策略流程如上图，主要分三个阶段，首先是当乒乓球离机器人较远时，执行第一阶段，在视觉模块判定乒乓球相对机器人的水平偏移量超过相应阈值时，我们执行相应的行走转弯动作。当视觉判定垂直偏移量大于一定阈值时，执行快走动作。

在执行相应运动模块化进行视觉检测，若距离小于一定阈值，则进入第二阶段，执行慢走动作。若此时水平仍有较大偏离，将执行水平向左平移或者水平向右平移动作进行校正。在最后一个阶段中，将继续利用水平平移校准和慢走动作来接近所踢乒乓球，最后执行踢球动作

有几点需要注意：

* 由于我们默认是右脚踢球，其Y坐标的0位置要做相应偏移。
* 我们在流程中加入了很多反馈，所以必须保证视觉的准确性，以避免陷入自激振荡死循环过程。
* 同样，加入了三次反馈（踢球流程里还有一次），会使调整时间延长，所以我们的踢球任务完成需要耗费很多时间。

**3.5 推箱子运动策略**



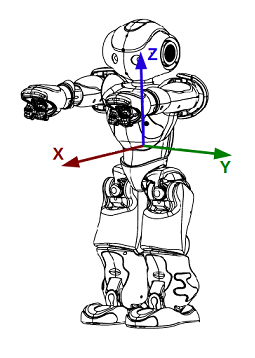
这里我们定义了等效目标点：

Xn=X-A\*math.cos(theta)-B

Yn=Y-A\*math.sin(theta)

其中A取箱子面前的点为目标点，面前距离为A，B为给x方向预留出空间。

:手部



:重心

:足部与其朝向

1. **视觉模块：**

**4.1物象空间坐标转换**

根据课上所讲的内容和Matlab已有的算法，已经有成熟的模型实现物空间到实际空间的变换，我们对这些文献和算法进行阅读和理解，并成功在python环境下实现了功能。然后，由于我们的移动策略对误差的要求比较严格，因此我们除了做了零漂修正等常见策略，也结合实际做了优化。

外参的获取优化：如果确定摄像头的外参，理想做法是对各个传感器的角度数据进行几何建模。但是，如果这些传感器稍有误差，该误差经过积累就会被放大。故我们采用了固定姿势测量的方法，在测量前先使机器人进入一个确定的姿态。遮掩，我们将外部参数内部化，直接通过标定得到了该姿势下摄像头的外参和内参。需要注意的是，一些关键的关节（如头部）对角度要求更为严格，我们使其进行两次移动，先到某固定的角度A，再到达目标角度，来取消齿轮的空程插。

物体选取优化：我们分析了单目测距精度与该物体的高度之间的关系，发现若物体靠近摄像头的光轴，则精度极低，故我们选择箱子定位时，箱子的位置特征点进行了下移，避免其与光轴接近。但是，当机器人与箱子的距离过近时，该点会离开视场。故我们的角度特征点，即“标定线”依然距离光轴比较近，使其能被检测到。值得一提的是，我们的角度计算的误差确实比方向误差较大，但是由于机器人使用了双手推箱子，这些误差会在机器人接触箱子被消除掉。

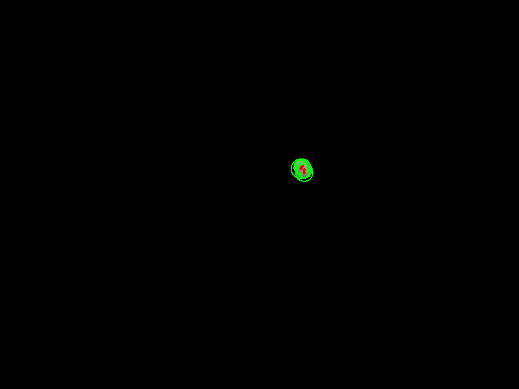
**4.2乒乓球识别**

4.2.1 函数使用说明

[x y]=FindBall(IP)

误差分析：乒乓球直径的一半左右，乒乓球越近误差越大，越远，误检率越大。

直观效果：



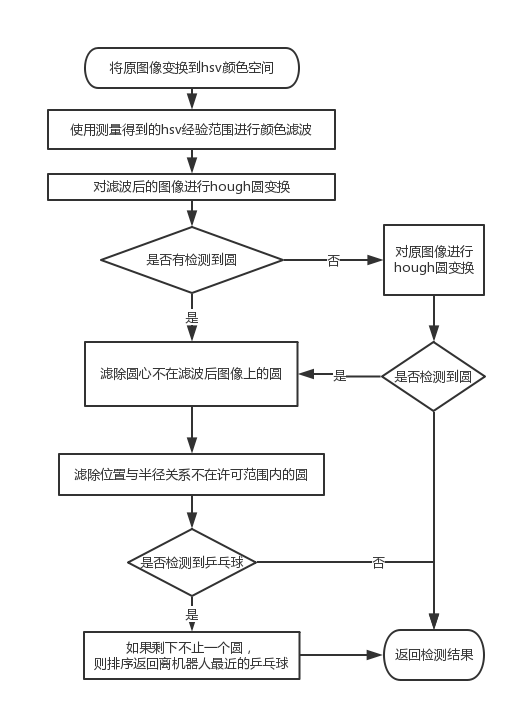
原图 结果图

最后将返回离机器人最近的乒乓球。同一个乒乓球上会检测到多个圆，但只保留在理论半径范围内的圆，这可以用来滤除其他误判的圆。

4.2.2 算法思路

首先调用底部摄像头,其视野小，识别距离近，目标清晰，如果没有检测到再调用顶部摄像头，其视野大，识别距离远，目标较为模糊。检测到乒乓球则返回坐标，否则返回结果全为-1。检测过程其核心为颜色滤波与hough圆变换。

程序流程：



**图像滤波：**

乒乓球颜色与背景色有比较明显的区别，我们使用PS读取图片中的乒乓球部分，发现其HSV有着比较清晰的范围，于是选择在*HSV颜色空间*滤除无关部分。为了获得较为良好的滤波效果，在滤波前对图像做了*中值滤波*，在滤波后做了形态学*开运算处理*。

**尽可能使乒乓被检测到（提高准确性）：**

乒乓球的检出与hough圆检测函数的参数有着直接的关系。其*最大检测半径和最小检测半径都是根据上下相机实拍确定，最小圆心距和梯度阈值也是根据不断实验确定不同相机的数值*。这样使得不同相机下都能特异性识别出乒乓球。

**尽可能使干扰被滤除掉（降低误检率）：**

Hough圆变换检测出的圆返回参数[x y r]，*如果x,y未在滤波图像上，肯定不是乒乓球*，滤除。同一个相机和乒乓球，其之间的*距离与乒乓球在图片上的半径有着一一对应关系*。我们通过在不同位置摆放乒乓球确定其不同位置的理论半径值。则在实际检测中如果发现r过于异常，则认为是误检，滤除。（理论半径通过标定有限个点，利用其半径和位置信息在matlab中curving fitting工具箱拟合出曲线函数，计算值即为理论值）

4.2.3 历史版本

初版函数没有对机器人相机进行相机，为了通用，为了能够识别各种情况下的乒乓球，多次采用不同Hough参数检测不同情况下的乒乓球，导致识别时间较长；另外因为没有通过乒乓球的坐标与直径的关系来筛选，识别准确率也没有那么高。

**4.3箱体识别**

4.3.1 函数使用说明

**[x y theta]=FindBox(IP)**

输入：机器人IP

返回：箱体的坐标（单位厘米）和方向（因为已知垂直地面，一个角度便足以表示方向，单位弧度制）。

**误差分析：**

法向量误差：像面的直线误差极值为标定线的宽/标定线的长；

箱体越近误差越大，但远远高于转弯所要求的误差精度。误检率始终较低。

物象空间变化的误差（略）；

位置误差：同乒乓球识别

**直观效果：**

****

4.3.2 算法思路

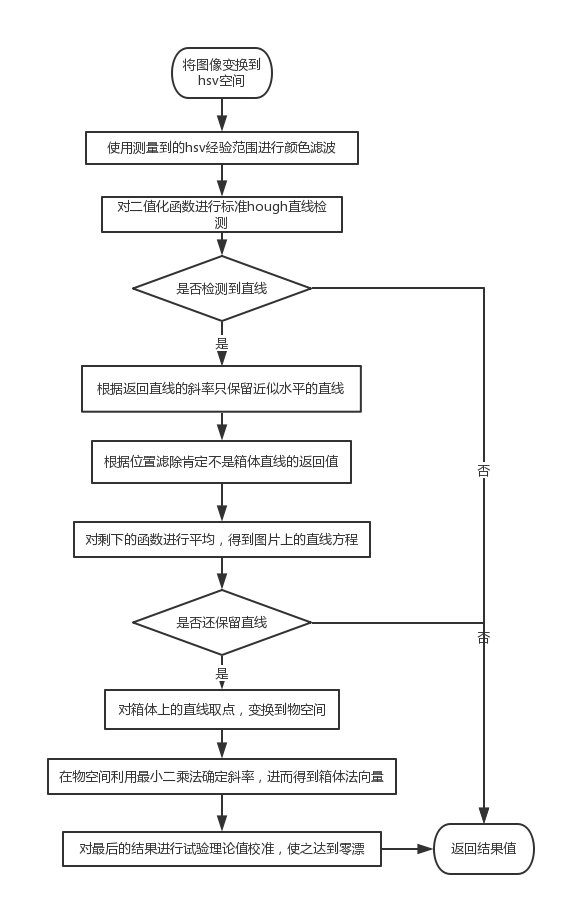
已知箱子一个面垂直于地面，那么只需在知道箱子面上一条直线的方向，便可以得出箱子的法向量。因为相机标定时需要确定的相对高度进行计算，所以选择与地面水平的直线进行运算。

**图像滤波**：见乒乓球识别。但这里对标定线的颜色有一定要求。我们需要*标定线、兵乓球。以及背景的主色调在hsv颜色空间上相隔得尽可能的远*，这样大家的滤波可调范围便更广一些。

**误检直线滤除**：因为实际使用水平的直线，哪怕侧着放，斜率也不会变化太大，*因此滤除斜率过大的检测直线*；使用上下相机拍摄箱体时，上相机视野比较宽，越过箱体，*所以图片偏上部分为无限远，实际直线绝无可能出现*，因此滤除。

**法向量的最终确定**：对箱体直线采取50个点变换到物空间，然后进行二维的（因为直线高度已确定）*最小二乘法分析*。最后返回斜率。然而理论和实际总有出入，我们再次进行了*实验矫正*。

**箱体位置获取**：在箱体底部中央粘上乒乓球，直接调用寻找乒乓球函数。



4.3.3 历史版本

方位获取：

起初准备通过在箱体不同位置贴多个不同的圆，通过利用三点确定一个平面来找出箱面法向量，然而由于Hough圆检测的误差极大，同一个实际圆可能检测出多个圆，且会有误检。这里难以像检测乒乓球一样分别检测出多个圆心。并且在物象空间的转换过程中，误差会进一步增大，无论从实现难度和精度上，都不利用检测。

位置检测：

起初利用在箱体中央、直线上下两侧帖圆形标志，即两圆心的连线为箱体的中垂线。其与直线的交点为箱体正中央坐标。实现过程中只需考虑抑制误检即可，只要检测出的圆都来源于标定的圆，则其圆心的x值经平均后期望值应该落在箱体中央，然后带入直线方程便可得到中央点。 该方法经检验具有一定可行性，但精度尚不够高。

之后将中央两个圆改为一条其他颜色的直线，将一垂直一水平直线的理论焦点视为中央点。

最后为了精度，改为直接使用检测乒乓球的函数，检测箱子底部中央。

1. **任务完成情况**

**5.1 行走与踢球**

由于视觉模块能够根据实际情况选择上或者下摄像头，不论远近的乒乓球均能够识别。

行走踢球算法在短距离（2块格子以内，即150cm以内）中表现非常出色，几乎能够每次成功；在长距离（3块格子及以上，即150cm以上）中表现也有较好效果，虽然由于在地面不平、摩擦力不足等因素的影响下在经过较长距离后脚底可能出现打滑或阻碍现象，但是闭环反馈控制原理能够较好地在下一阶段及时校正，因此最后仍能踢到球，即使偶然没有踢到，误差也很小。

一次成功的基本任务实现展示在视频中。

值得一提的是，在长距离行走与踢球时，为了增大与地面的摩擦力，防止脚底打滑。我们就地取材，利用橡皮筋为机器人配置了下图所示的防滑底链，并收到了不错的效果。



**5.2 推箱子**

在完成了基本任务的基础上，我们把推箱子也设定在长距离。同样地，机器人的行走受到地面平整度的影响较大。我们为了尽量减小这个影响，把透明胶带贴在地面上相邻版块的衔接凸起处，实际测试表明，这么做显著地减小了机器人在经过时受到的影响。

在每次执行扩展任务时，机器人能够按照我们所期望地识别箱子、按照运动策略集进行相应的运动。我们的程序稳定性好，实验的可重复性高。

一次成功的扩展任务实现展示在视频中。

1. **小组分工：**

**林葳洁：** 项目统筹。运动指令修正，运动策略集设计，模块对接整合，承担大部分机器人运动模块的调试工作

**许蒙蒙：**

**蒋雪松：**运动学建模，逆运动学求解，运动策略集设计，包括测试实验调参

**杨 喜：**视觉模块设计及程序编写，包括乒乓球识别和箱体方位、法向量识别。视觉相关的调试。

**郎 园：** 推箱子路径规划建模、代码调试，视频拍摄及剪辑。