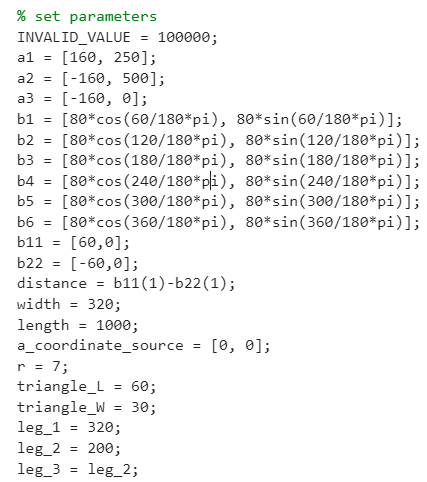
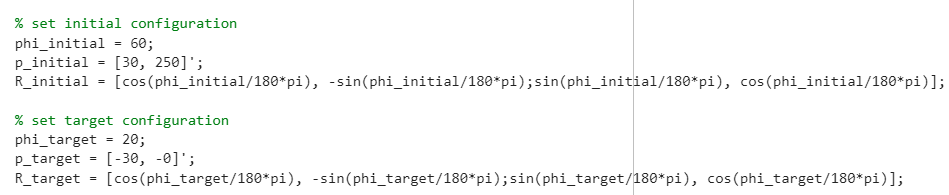
motion\_simulation\_strategy1.m

（1）设置参数



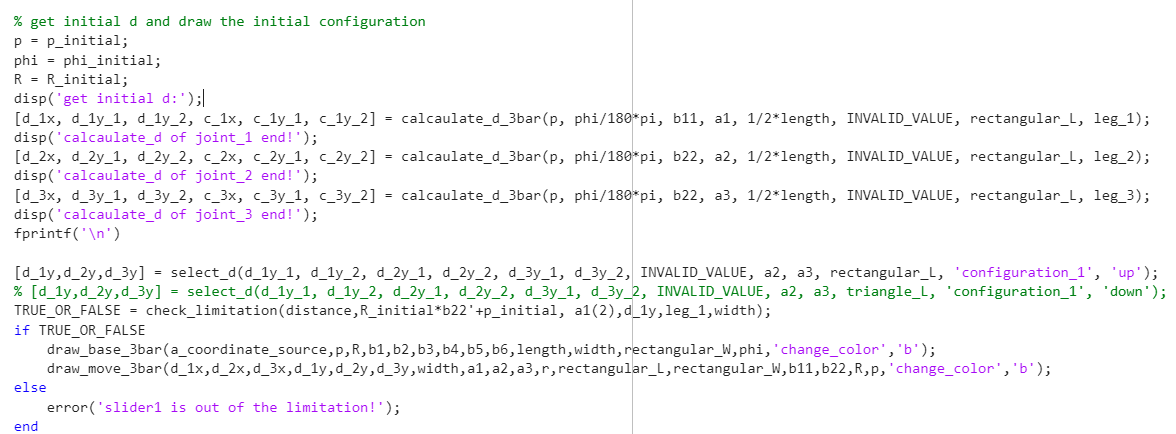
|  |  |
| --- | --- |
| INVALID\_VALUE | 无效值，若当前位型对应的滑块位置已经超出了导轨的范围，则令对应的d、c值等于INVALID\_VALUE。 |
| a1、a2、a3 | 滑块1、2、3的d的纵坐标的参考起点（a系） |
| b1、b2、b3、b4、b5、b6 | 末端平台的六边形坐标（外接圆r=80）（b系） |
| b11、b22 | 六边形上的两个球轴承的坐标（b系） |
| distance | 六边形上的两个球轴承的距离 |
| width | 两导轨之间的宽度 |
| length | 导轨长度（左导轨×2的长度） |
| a\_coordinate\_source | a系原点 |
| r | 轴承的半径 |
| triangle\_L | 滑块长度 |
| triangle\_W | 滑块宽度 |
| leg\_1 | 杆1长度 |
| leg\_2 | 杆2长度 |
| leg\_3 | 杆3长度 |

设定初始位型与期望末端位型：



（2）闭运动学，确定开始位型、期望位型对应的滑块位置（关节变量c、d）

获得初始位型对应的关节参数d、c等，并绘制图像：



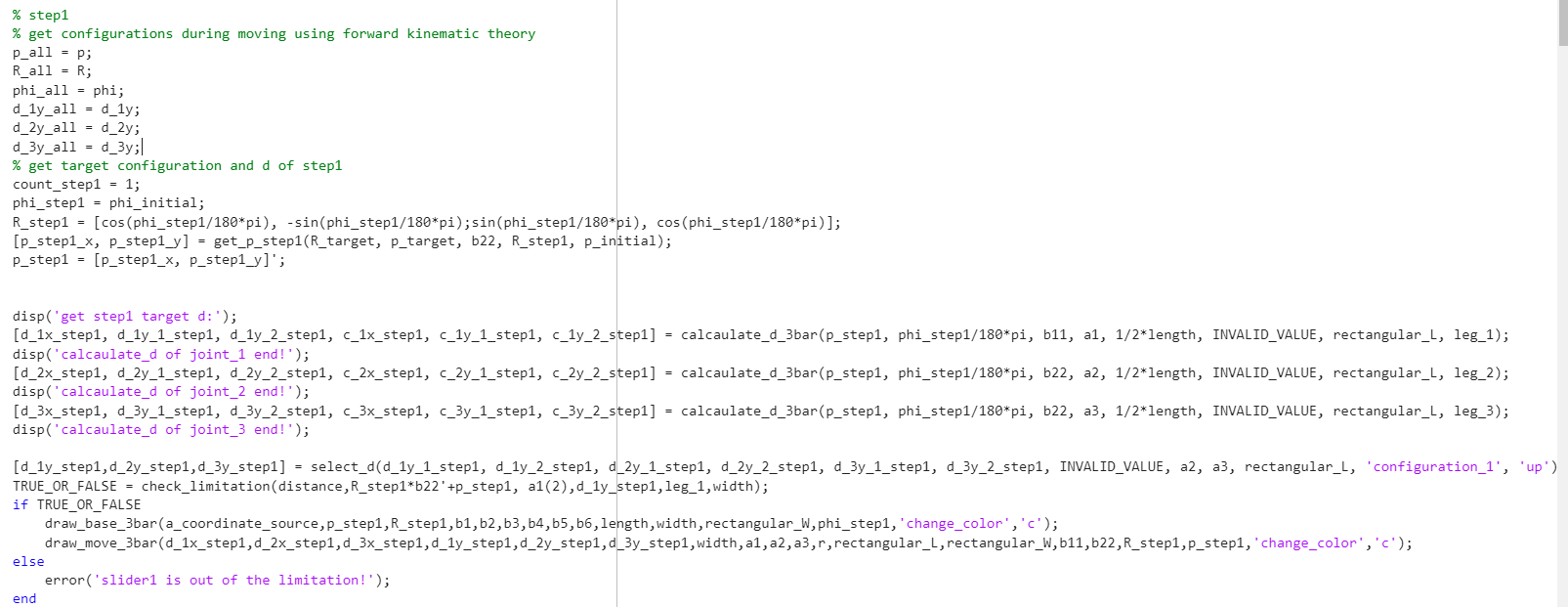
Calculate\_d\_3bar:根据闭链规律+杆长限制算出d、c（请转跳至calculate\_d\_3bar.m）

Select\_d:根据闭运动学算出来的可能的两种情况作出排除（请转跳至select\_d.m）

Check\_limitation:原意是为了检查当前杆1是否在角度限制内，但目前只能确定当前杆1位置满足下限要求（下限：杆水平的位置）

1. 第一步：水平移动

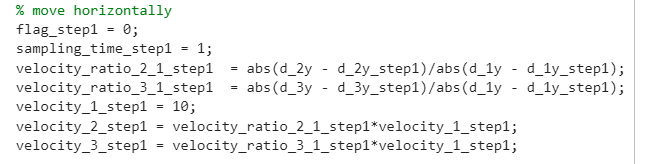
先绘制出第一步的最终状态的情况（就是平移后的状态），用逆运动学求关节变量再画图。



p\_all、R\_all、phi\_all、d\_1y\_all、d\_2y\_all、d\_3y\_all把step1过程中的p、R等记录在列表中，先让他们等于初始状态值。

get\_p\_step1:得到第一步平移后的最终位型对应的p（请转跳至get\_p\_step1.m）

得到第一步最终状态的p、R、phi等信息后，画出来。



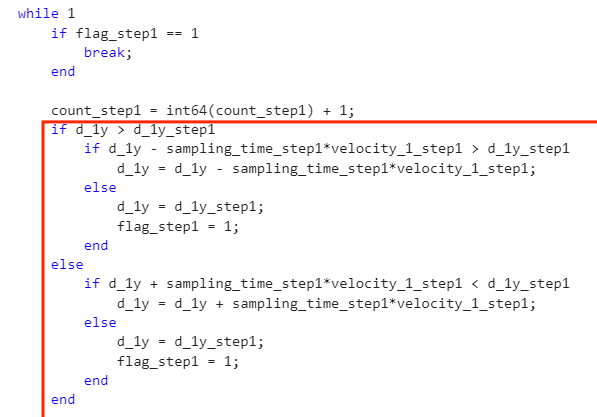
|  |  |
| --- | --- |
| flag\_step1 | 作为退出下面的循环的标志，当滑块到达最终状态时，令为1并退出循环 |
| sampling\_time\_step1 | 在step1中的采样时间，即画图时的时刻。 |
| velocity\_ratio\_2\_1\_step1 | 在step1中，滑块2与滑块1的速度比值（和距离成正比，设是匀速运动，同时开始、同时到达） |
| velocity\_ratio\_3\_1\_step1 | 在step1中，滑块3与滑块1的速度比值（和距离成正比，设是匀速运动，同时开始、同时到达） |
| velocity\_1\_step1 | 滑块1的速度 |
| velocity\_2\_step1 | 滑块2的速度 |
| velocity\_3\_step1 | 滑块3的速度 |

进入循环：

·先判断是否到达终点，flag\_step1=1意味着到达，退出。

·否则，改变d的数值（即滑块继续滑动），红框内的代码在判断滑块正确的滑动方向（向上/下），依次类推得到滑块1、2、3的滑动位置情况。

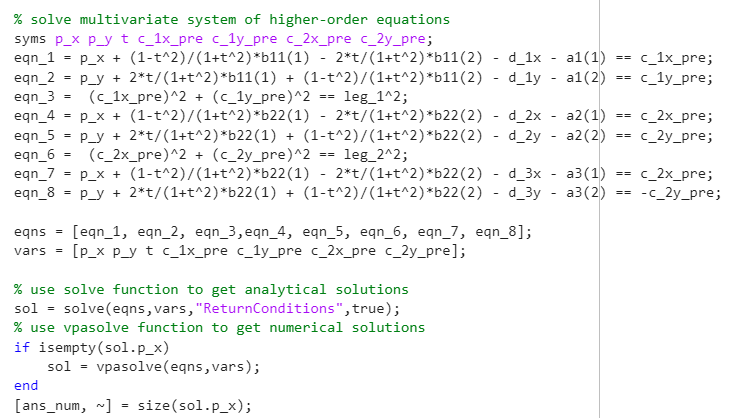
·Count\_step1统计step1一共记录了多少个状态，以便后面输出每个状态。



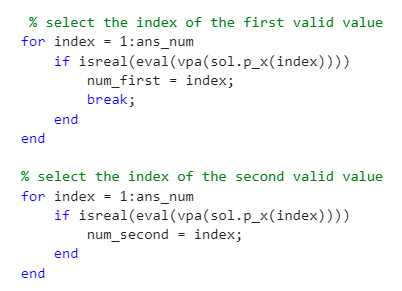
解经典的三条式子（每个杆三条，一共九条），此处因为杆2与杆3是对称的，杆长一致，所以省掉了一些方程与未知数。

解析式法：尝试利用solve解出这些式子的解析解，若失败会出现“警告”（在输出窗口可以看到）；

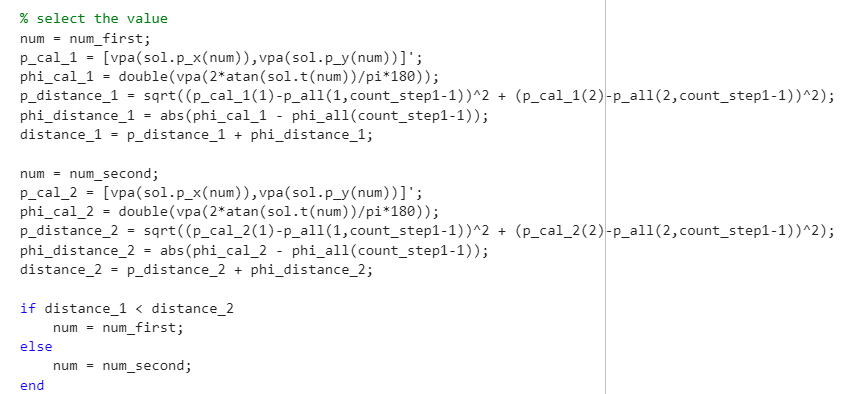
解析式法失败后，数值法：用vpasolve解数值解，解出数值解，并用ans\_num得到解的个数。



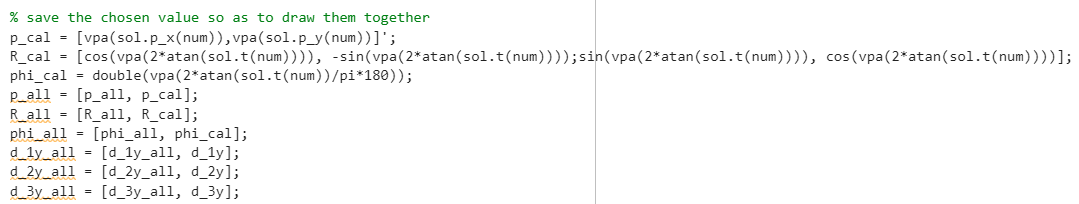
经过多次尝试，发现只有两个是实数解，用num\_first代表第一个实数解的index，用num\_second代表第二个实数解的index。



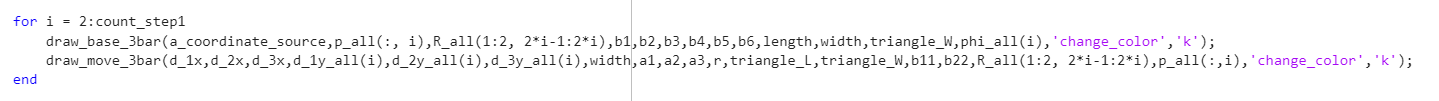
分别计算两种解的位型与上一个位型的差别，具体等于p的差别（平方差）+phi的差别（差的绝对值）（这样表示“位型相近”可能不够科学，当采样时间间隔过长时，可能得到另外的一个错误位型）



把上述位型的参数如p、R等记录在p\_all、R\_all等列表中，以便循环结束后一次性画出来。



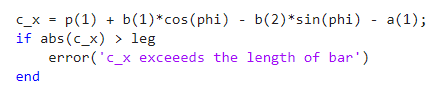
循环结束，一次性画出整个step1过程的位型变化。



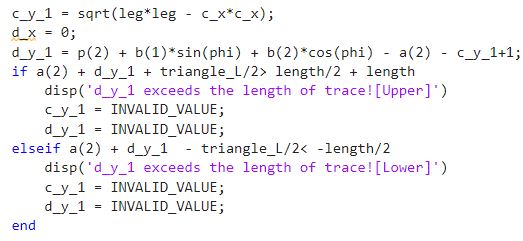
Step2、step3同step1类似，最终模拟整个过程。

calculate\_d\_3bar.m

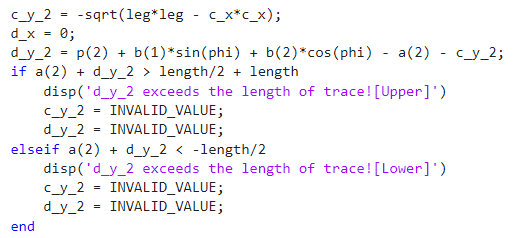
逆运动学时，c的横坐标若大于杆长，则不合理，报错退出程序。



当c的纵坐标>0，开平方取正，算出d的纵坐标后，若此时滑块已经（向上/下）滑出导轨，则令c、d的纵坐标为无效值。



当c的纵坐标<0，开平方取负，算出d的纵坐标后，若此时滑块已经（向上/下）滑出导轨，则令c、d的纵坐标为无效值。

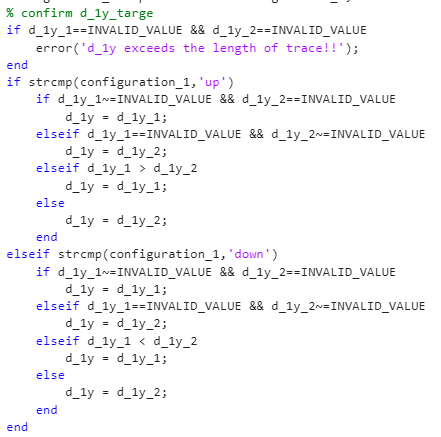


select\_d.m

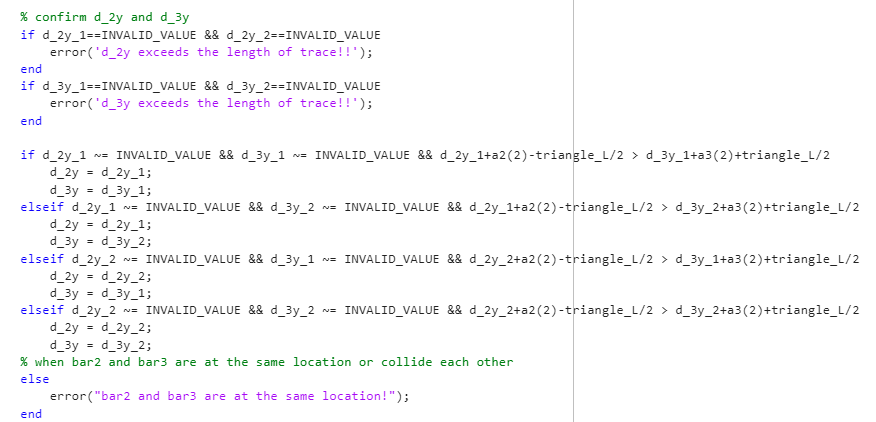


“up”参数：当逆运动学接触两个解时（注意：和正运动学的唯一解问题无关，一个c\_y大于0，一个c\_y小于0），“up”表示c\_y小于0的那个，这个函数针对的是杆1的逆运动学形态。

对杆1的d进行筛选：

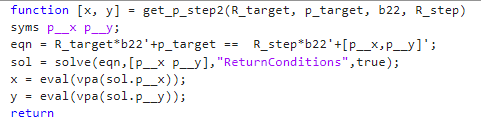


对杆2、杆3进行筛选：



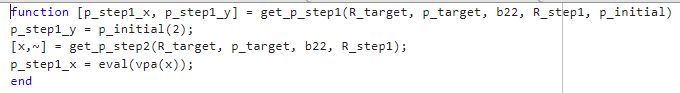
get\_p\_step2.m

算出step2对应的终点位型的p（s系原点指向b系原点），此时左定位点与预设位型的左定位点位置相同。



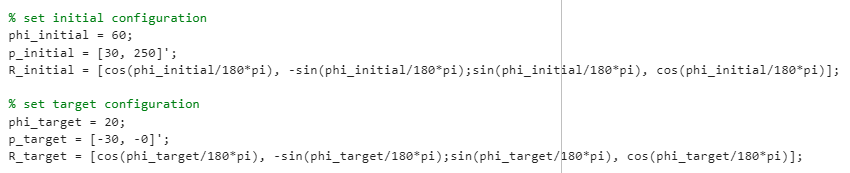
get\_p\_step1.m

算出step2对应的终点位型的p（s系原点指向b系原点），此时左定位点与预设位型的左定位点x坐标相同，b系原点的y坐标和原来初始位型下的b系原点的y坐标相同（所谓“水平移动”）。



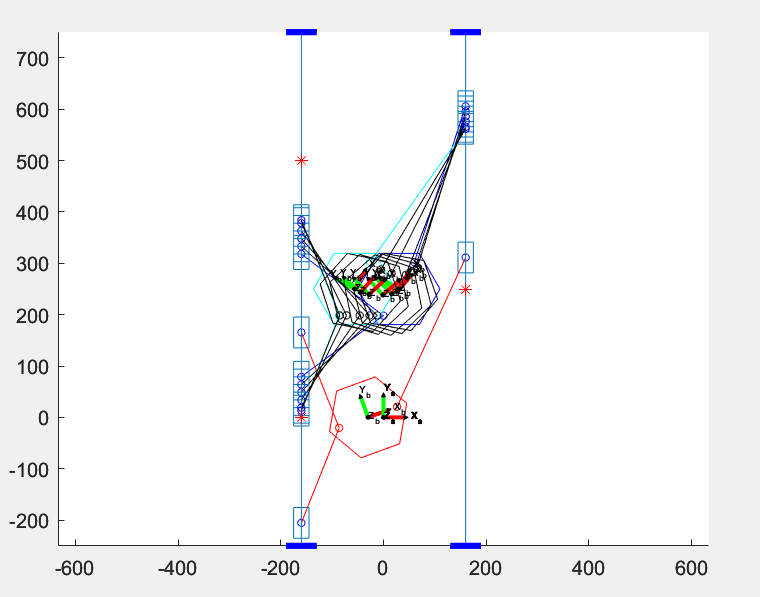
main\_parallel\_robot\_3bar.m的文件不能运行，有些调用的函数已改

策略的重要性：



以上的数据，策略一无法到达，策略二可以到达

策略一：



策略二：

