智能技术与系统综合实验
实验手册
实验 3：youBot 麦克纳姆轮智能车控制算法

1. 实验背景

根据第 1 周仿真实验 2 所编写的生成轨迹，编写反馈控制算法：配备机械臂的麦克纳姆轮智能车运动学任务空间的前馈+PI 反馈控制算法：

$$\mathcal{V}(t) = [Ad_{X^{-1}X_d}]\mathcal{V}_d(t) + K_p X_{err}(t) + K_i \int_0^t X_{err}(t)\,dt \tag{3-1}$$

仿真的主函数为 control=FeedbackControl(X, Xd, Xdnext, Kp, Ki, dt)，其输入输出为如下定义所示：

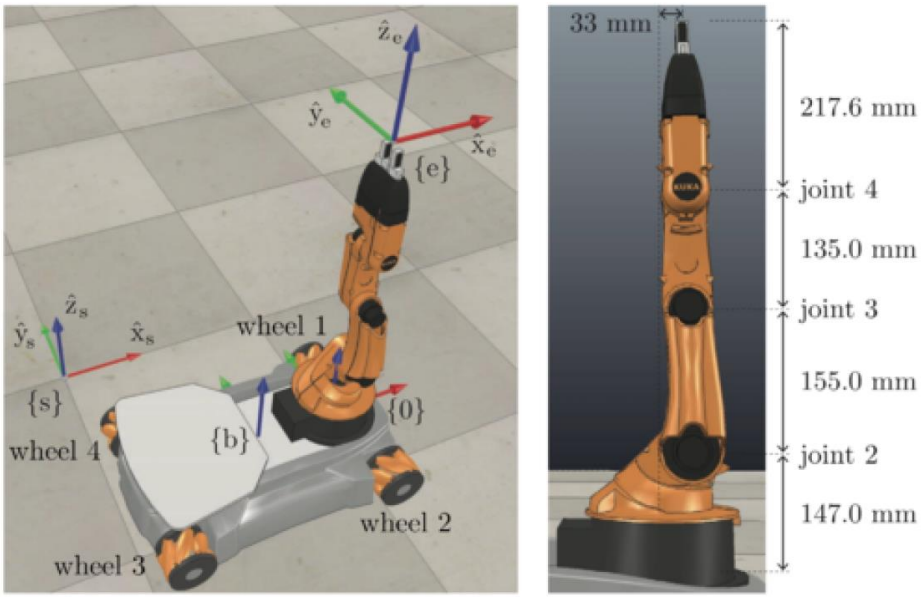

图 1：机械臂+智能车初始位形（所有关节角度为 0），注意图中的空间参考坐标系$\{s\}$，智能车体坐标系$\{b\}$，机械臂基座坐标系$\{0\}$，以及末端执行机构坐标系$\{e\}$。

2. 输入

1) 当前时刻，实际的末端执行器的位型X（也可写作T_{se}）；
2) 当前时刻，参考轨迹上期望的末端执行器的位型X_d（也可写作$T_{se,d}$，d 表示 desired）；
3) 下一时刻（Δt时间后），参考轨迹上的末端执行器的位型$X_{d,next}$（也可写作$T_{se,d,next}$）；
4) PI 控制器的增益矩阵K_p和K_i，均为对角阵（例如$K_p = K_i = eye(6)$，即6×6的单位矩阵）；
5) 两个相邻参考轨迹位形（X_d和$X_{d,next}$）间的时间间隔，即步长Δt（例如

$0.01sec$）。

3. 输出

在末端执行器坐标系$\{e\}$中表示的末端执行器控制旋量$\mathcal{V}$。

4. 具体步骤

1）机器人（智能车+机械臂，下统称机器人）当前的位形$X = X(q)$来自于第 1 周仿真实验中的$q = [\phi, x, y, J1, J2, J3, J4, J5, \theta1, \theta2, \theta3, \theta4]$。

2）$\mathcal{V}_d(t)$部分：将X_d经由步长时间Δt后移动到$X_{d,next}$的前向参考旋量$\mathcal{V}_d$，由如下公式计算：

$$[\mathcal{V}_d] = \frac{1}{\Delta t} log\left(X_d^{-1} X_{d,next}\right) \tag{3-2}$$

其中$[\mathcal{V}_d] \in \mathbb{R}^{4\times4}$为$\mathcal{V}_d \in \mathbb{R}^{6\times1}$对应的$se(3)$形式，$\omega$为旋量中角速度分量，$v$为旋量中线速度分量，$\omega^{\times} \in \mathbb{R}^{3\times3}$为$\omega \in \mathbb{R}^{3\times1}$的反对称矩阵，数学关系如下：

$$\mathcal{V} = [\boldsymbol{\omega} \quad \boldsymbol{v}]^T = [\omega_x \quad \omega_y \quad \omega_z \quad v_x \quad v_y \quad v_z]^T \in \mathbb{R}^{6\times1}, [\mathcal{V}] = \begin{bmatrix} \boldsymbol{\omega}^{\times} & \boldsymbol{v} \\ 0 & 0 \end{bmatrix} \in se(3) \tag{3-3}$$

$$\boldsymbol{\omega} = [\omega_x \quad \omega_y \quad \omega_z]^T, \boldsymbol{\omega}^{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3-4}$$

代码实现

i. 计算$X_d^{-1} X_{d,next}$：不需要求出X_d的逆矩阵，可以直接进行“矩阵左除”，即

```
Xd\Xdnext
```

ii. 使用MatrixLog6（附录1）计算$log\left(X_d^{-1} X_{d,next}\right)$，即

```
MatrixLog6(Xd\Xdnext)
```

iii. 使用se3ToVec（附录2）将$[\mathcal{V}_d] \in se(3)$变换为$\mathcal{V}_d \in \mathbb{R}^{6\times1}$，即

```
Vd = se3ToVec(1/dt*MatrixLog6(Xd\Xdnext))
```

3）$[Ad_{X^{-1}X_d}]$前向控制分量部分：$[Ad_X]$（这种数学操作的表达本身就含中括号）表示求解$X \in SE(3)$的伴随矩阵$[Ad_X] \in \mathbb{R}^{6\times6}$的运算，数学关系如下：

$$X = \begin{bmatrix} \boldsymbol{R}^{3\times3} & \boldsymbol{p}^{3\times1} \\ \boldsymbol{0}^{1\times3} & 1 \end{bmatrix} \in SE(3), [Ad_X] = \begin{bmatrix} \boldsymbol{R} & 0 \\ \boldsymbol{p}^{\times}\boldsymbol{R} & \boldsymbol{R} \end{bmatrix} \in \mathbb{R}^{6\times6} \tag{3-5}$$

代码实现

i. 计算$X^{-1}X_d$，即

```
X\Xd
```

ii. 使用 Adjoint（附录 3）计算$[Ad_{X^{-1}X_d}]$，即

```
Adjoint(X\Xd)
```

iii. 计算$[Ad_{X^{-1}X_d}]\mathcal{V}_d(t)$，即

```
AdXinvXdVd = Adjoint(X\Xd)*Vd
```

4) $X_{err}(t)$部分：将X经由单位时间后移动到X_d的误差控制量，由如下公式计算：

$$[X_{err}] = log(X^{-1}X_d) \tag{3-6}$$

代码实现

i. 计算$X^{-1}X_d$，即

```
X\Xd
```

ii. 使用 MatrixLog6（附录 1）计算$log(X^{-1}X_d)$，即

```
MatrixLog6(X\Xd)
```

iii. 使用 se3ToVec（附录 2）将$[X_{err}] \in se(3)$变换为$X_{err} \in \mathbb{R}^{6\times1}$，即

```
Xerr = se3ToVec(MatrixLog6(X\Xd))
```

5) 前馈+PI 反馈控制算法，即公式（3-1）。

代码实现

```
V = AdXinvXdVd + Kp*Xerr + Ki*(Xerr*dt)
```

6) 机械臂运动学部分

代码实现

i. 列出机械臂体坐标系旋量轴(Body Screw Axes)，即

```
Blist = [[0;  0;  1;  0;  0.033;  0],...
         [0; -1;  0; -0.5076; 0;  0],...
         [0; -1;  0; -0.3526; 0;  0],...
         [0; -1;  0; -0.2176; 0;  0],...
        [0;   0;  1;  0;      0;  0]];
```

ii. 列出机械臂各关节角度，这里测试下述角度

```
thetalist = [0;  0;  0.2; -1.6;  0];
```

iii. 使用 JacobianBody（附录 4）计算机械臂在体坐标系下的雅克比矩阵，即

```
Jarm = JacobianBody(Blist, thetalist);
```

7) 智能车运动学部分

代码实现

i. 列出机械臂基座坐标系$\{0\}$与智能车体坐标系$\{b\}$的相对位形，即

```
Tb0 = [ 1  0  0  0.1662;
        0  1  0  0;
        0  0  1  0.0026;
        0  0  0  1];
T0b = inv(Tb0);
```

ii. 列出机械臂位于初始位形时（所有关节角度为 0），末端执行器坐标系{e}与机械臂基座坐标系{0}的初始位形，即

```
M0e = [ 1  0  0  0.033;
        0  1  0  0;
        0  0  1  0.6546;
        0  0  0  1];
```

iii. 使用 FKinBody（附录 5），计算正向运动学，即关节旋转 thetalist 后，端执行器坐标系{e}与机械臂基座坐标系{0}的相对位形

```
T0e = FKinBody(M0e, Blist, thetalist);
Te0 = inv(T0e);
```

iv. 定义麦克纳姆轮智能车几何参数

```
r = 0.0475; % radius of whell
l = 0.47/2; % L的小写l，与下面的1作区分
w = 0.3/2;  % w
```

```
F6 = [ 0          0          0          0      ;
       0          0          0          0      ;
    -1/(l+w)   1/(l+w)    1/(l+w)   -1/(l+w);
       1          1          1          1      ;
      -1          1         -1          1      ;
       0          0          0          0      ;]*r/4;
```

v. 得到智能车的雅可比矩阵

```
Jbase = Adjoint(Te0*T0b)*F6;
```

vi. 组合机械臂和智能车的雅克比矩阵

```
Je = [Jbase Jarm]
```

vii. 最终，计算控制量（机械臂关节速度和智能车轮子速度，这里与第 1 周实验时的控制量稍有区别，即第 1 周$u = [j1, j2, j3, j4, j5, \dot{\theta}1, \dot{\theta}2, \dot{\theta}3, \dot{\theta}4]$，这里改为$u = [\dot{\theta}1, \dot{\theta}2, \dot{\theta}3, \dot{\theta}4, j1, j2, j3, j4, j5]$，即简单地交换顺序）

```
control = pinv(Je)*V
```

5. 测试主函数 FeedbackControl

测试所编写的仿真函数，若输入为

$$X_d = \begin{bmatrix} 0 & 0 & 1 & 0.5 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, X_{d,next} = \begin{bmatrix} 0 & 0 & 1 & 0.6 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3-7}$$

$$X=\begin{bmatrix}0.170 & 0 & 0.985 & 0.387\\ 0 & 1 & 0 & 0\\ -0.985 & 0 & 0.170 & 0.570\\ 0 & 0 & 0 & 1\end{bmatrix}, K_p = K_i = eye(6), \Delta t = 0.01$$

则输出为(如果代码正确，结果将极其接近)

$$\mathcal{V}_d = [0 \quad 0 \quad 0 \quad 20 \quad 0 \quad 10] \tag{3-8}$$

$$[Ad_{X^{-1}X_d}]\mathcal{V}_d = [0 \quad 0 \quad 0 \quad 21.409 \quad 0 \quad 6.455]$$

$$\mathcal{V} = [0 \quad 0.171 \quad 0 \quad 21.488 \quad 0 \quad 6.562]$$

$$X_{err} = [0 \quad 0.171 \quad 0 \quad 0.080 \quad 0 \quad 0.107]$$

$$control = [157.2 \quad 157.2 \quad 157.2 \quad 157.0 \quad 0 \quad -654.3 \quad 1400.9 \quad -746.8 \quad 0]$$

附录 1

Matlab 代码：输入属于的齐次变换矩阵$T \in SE(3)$，输出对应的指数坐标$[S]\theta \in se(3)$。

```
function expmat = MatrixLog6(T)
% Takes a transformation matrix T in SE(3).
% Returns the corresponding se(3) representation of exponential
% coordinates.
% Example Input:
%
% clear; clc;
% T = [[1, 0, 0, 0]; [0, 0, -1, 0]; [0, 1, 0, 3]; [0, 0, 0, 1]];
% expmat = MatrixLog6(T)
%
% Output:
% expc6 =
%        0        0        0        0
%        0        0   -1.5708    2.3562
%        0    1.5708        0    2.3562
%        0        0        0        0

[R, p] = TransToRp(T);
omgmat = MatrixLog3(R);
if isequal(omgmat, zeros(3))
    expmat = [zeros(3), T(1: 3, 4); 0, 0, 0, 0];
else
    theta = acos((trace(R) - 1) / 2);
    expmat = [omgmat, (eye(3) - omgmat / 2 ...
                  + (1 / theta - cot(theta / 2) / 2) ...
                    * omgmat * omgmat / theta) * p;
              0, 0, 0, 0];
end
end
```

附录 2

Matlab 代码：输入指数坐标$[\mathcal{V}] \in se(3)$，输出对应的列向量$\mathcal{V} \in \mathbb{R}^{6\times1}$。

```
function V = se3ToVec(se3mat)
% Takes se3mat a 4x4 se(3) matrix
% Returns the corresponding 6-vector (representing spatial velocity).
% Example Input:
%
% clear; clc;
% se3mat = [[0, -3, 2, 4]; [3, 0, -1, 5]; [-2, 1, 0, 6]; [0, 0, 0,
0]];
% V = se3ToVec(se3mat)
```

```
%
% Output:
% V =
%     1
%     2
%     3
%     4
%     5
%     6

V = [se3mat(3, 2); se3mat(1, 3); se3mat(2, 1); se3mat(1: 3, 4)];
end
```

附录 3

Matlab 代码：输入属于的齐次变换矩阵$T \in SE(3)$，输出其伴随变换矩阵$[Ad_T] \in \mathbb{R}^{6\times6}$。

```
function AdT = Adjoint(T)
% Takes T a transformation matrix SE3.
% Returns the corresponding 6x6 adjoint representation [AdT].
% Example Input:
%
% clear; clc;
% T = [[1, 0, 0, 0]; [0, 0, -1, 0]; [0, 1, 0, 3]; [0, 0, 0, 1]];
% AdT = Adjoint(T)
%
% Output:
% AdT =
%     1     0     0     0     0     0
%     0     0    -1     0     0     0
%     0     1     0     0     0     0
%     0     0     3     1     0     0
%     3     0     0     0     0    -1
%     0     0     0     0     1     0

[R, p] = TransToRp(T);
AdT = [R, zeros(3); VecToso3(p) * R, R];
end
```

附录 4

Matlab 代码：计算体雅克比矩阵

```
function Jb = JacobianBody(Blist, thetalist)
% Takes Blist: The joint screw axes in the end-effector frame when
the
%            manipulator is at the home position, in the format of a
```

```
%            matrix with the screw axes as the columns,
%      thetalist: A list of joint coordinates.
% Returns the corresponding body Jacobian (6xn real numbers).
% Example Input:
%
% clear; clc;
% Blist = [[0; 0; 1;   0; 0.2; 0.2], ...
%       [1; 0; 0;   2;   0;   3], ...
%       [0; 1; 0;   0;   2;   1], ...
%       [1; 0; 0; 0.2; 0.3; 0.4]];
% thetalist = [0.2; 1.1; 0.1; 1.2];
% Jb = JacobianBody(Blist, thetalist)
%
% Output:
% Jb =
%   -0.0453    0.9950         0    1.0000
%    0.7436    0.0930    0.3624         0
%   -0.6671    0.0362   -0.9320         0
%    2.3259    1.6681    0.5641    0.2000
%   -1.4432    2.9456    1.4331    0.3000
%   -2.0664    1.8288   -1.5887    0.4000

Jb = Blist;
T = eye(4);
for i = length(thetalist) - 1: -1: 1
    T = T * MatrixExp6(VecTose3(-1 * Blist(:, i + 1) * thetalist(i +
1)));
    Jb(:, i) = Adjoint(T) * Blist(:, i);
end
end
```

附录 5

Matlab 代码：计算正向运动学

```
function T = FKinBody(M, Blist, thetalist)
% Takes M: the home configuration (position and orientation) of the
%         end-effector,
%      Blist: The joint screw axes in the end-effector frame when the
%            manipulator is at the home position,
%      thetalist: A list of joint coordinates.
% Returns T in SE(3) representing the end-effector frame when the
joints
% are at the specified coordinates (i.t.o Body Frame).
% Example Inputs:
%
```

```
% clear; clc;
% M = [[-1, 0, 0, 0]; [0, 1, 0, 6]; [0, 0, -1, 2]; [0, 0, 0, 1]];
% Blist = [[0; 0; -1; 2; 0; 0], [0; 0; 0; 0; 1; 0], [0; 0; 1; 0; 0;
0.1]];
% thetalist = [pi / 2; 3; pi];
% T = FKinBody(M, Blist, thetalist)
%
% Output:
% T =
%   -0.0000    1.0000         0   -5.0000
%    1.0000    0.0000         0    4.0000
%         0         0   -1.0000    1.6858
%         0         0         0    1.0000

T = M;
for i = 1: size(thetalist)
    T = T * MatrixExp6(VecTose3(Blist(:, i) * thetalist(i)));
end
end
```

智能技术与系统综合实验
实验手册
实验 4：youBot 麦克纳姆轮智能车物体抓取

1. 实验背景

根据前 3 次仿真实验，编写主函数（Main_Function），整合（并稍加修改）所有代码，即机械臂智能车物体抓取轨迹生成（TrajectoryGenerator）、前馈+PI 反馈控制（FeedbackControl）、位形控制（NextState），实现物体抓取。

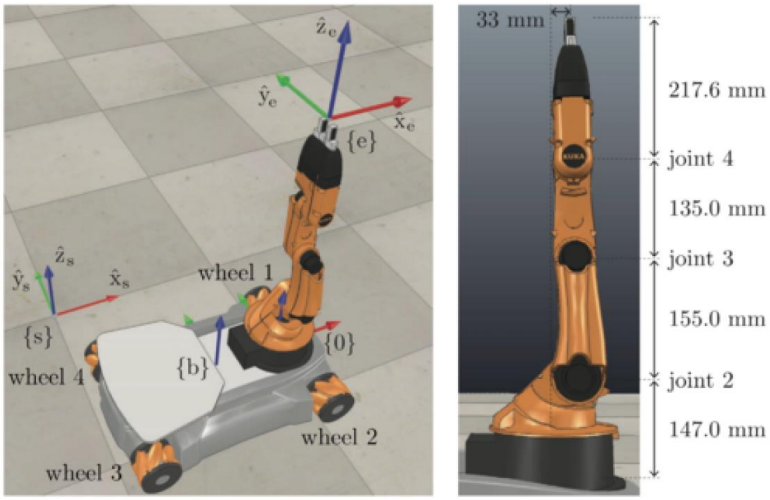

图 2：机械臂+智能车初始位形（所有关节角度为 0），注意图中的空间参考坐标系$\{s\}$，智能车体坐标系$\{b\}$，机械臂基座坐标系$\{0\}$，以及末端执行机构坐标系$\{e\}$。

2. 具体流程

1) 使用上周实验的函数（无需修改）TrajectoryGenerator 生成轨迹 `result_traj`，并记录其长度 N。

代码实现

i. 读取生成轨迹`result_traj`的第一行作为当前期望位形X_d

```
Xd =
[result_traj(1,1) result_traj(1,2) result_traj(1,3) result_traj(1,10);
  result_traj(1,4) result_traj(1,5) result_traj(1,6) result_traj(1,11);
  result_traj(1,7) result_traj(1,8) result_traj(1,9) result_traj(1,12);
  0              0              0              1];
```

ii. 读取生成轨迹`result_traj`的第二行作为下一时刻期望位形$X_{d,next}$

```
Xdnext =
[result_traj(2,1) result_traj(2,2) result_traj(2,3) result_traj(2,10);
  result_traj(2,4) result_traj(2,5) result_traj(2,6) result_traj(2,11);
   result_traj(2,7) result_traj(2,8) result_traj(2,9) result_traj(2,12);
  0              0              0              1];
```

2) 初始化系统

代码实现

i. 位形$q = [\phi, x, y, J1, J2, J3, J4, J5, \theta1, \theta2, \theta3, \theta4]$（第 13 位表示机械爪闭合，之后根据期望轨迹的第 13 位进行输入）的初始值

```
current_conf = [0 0 0 0 0 0 0 0 0 0 0 0 0]; % 1x12
```

ii. 齐次变换矩阵：车体转角为0度时，智能车体坐标系$\{b\}$相对于空间坐标系$\{s\}$的位形$T_{sb}0$

```
Tsb_0 = [cos(0) -sin(0)  0  0;
         sin(0)  cos(0)  0  0;
         0       0       1  0.0963;
         0       0       0  1];
```

机械臂基座坐标系$\{0\}$相对于智能车体坐标系$\{b\}$的位形T_{b0}

```
Tb0 = [ 1  0  0  0.1662;
        0  1  0  0;
        0  0  1  0.0026;
        0  0  0  1];
```

机械臂位于初始位形时（所有关节角度为0），末端执行器坐标系$\{e\}$相对于机械臂基座坐标系$\{0\}$的初始位形

```
M0e = [ 1  0  0  0.033;
        0  1  0  0;
        0  0  1  0.6546;
        0  0  0  1];
```

末端执行器坐标系$\{e\}$相对于空间坐标系$\{s\}$的位形

```
Tse = Tsb_0*Tb0*M0e;
  X = Tse;
```

iii. 定义机械臂体坐标系旋量轴(Body Screw Axes)

```
Blist = [[0;  0;  1;  0;  0.033;  0],...
         [0; -1;  0; -0.5076; 0;  0],...
         [0; -1;  0; -0.3526; 0;  0],...
         [0; -1;  0; -0.2176; 0;  0],...
         [0;  0;  1;  0;      0;  0]];
```

iv. 初始化机械臂各转动角度

```
thetalist = zeros(5,1);
```

v. 创建数列，以记录位形（参考实验1，我们使用位形来动画仿真）和误差（研究控制效果）

```
result_conf = zeros(N-1, 13);
result_Xerr = zeros(N-1, 6);
```

3) 进入循环，循环 N-2 次。在时刻 i 时

代码实现

i. 读取当前位形的 4-8 位（参考 **2.2.i 部分**，q的表达式），即机械臂转角

```
thetalist(:,1) = current_conf(4:8);
```

ii. 利用本周实验目标 3 的 FeedbackControl 代码（这里我们需要稍加修改，添加一个输入 thetalist），计算控制量 control 和误差 Xerr

（注意定义 PI 控制的增益K_p, K_i和步长dt）

```
[control, Xerr] = FeedbackControl(X, Xd, Xdnext, Kp, Ki, dt,
                                  thetalist);
```

iii. 利用第 1 周实验目标 1 的 NextState，计算下一时刻的位形

```
[next_conf] = NextState(current_conf, control, dt, max_speed);
```

iv. 记录位形和误差（第 13 位直接根据期望的轨迹写入：0 表示机械爪开，1 表示闭）

```
result_conf(i,1:12) = next_conf;
result_conf(i,13) = result_traj(i,13);
result_Xerr(i,:) = Xerr;
```

v. 记录车体的转角和横纵坐标

```
phi = next_conf(1);
x = next_conf(2);
y = next_conf(3);
```

vi. 读取所计算出的下一时刻的位形的 4-8 位，作为机械臂转角

```
thetalist(:,1) = next_conf(4:8);
```

vii. 计算此时末端执行器坐标系$\{e\}$相对于机械臂基座坐标系$\{0\}$的位形

```
T0e = FKinBody(M0e, Blist, thetalist);
```

viii. 以及此时，车体转角为 phi（见 v）时，智能车体坐标系$\{b\}$相对于空间坐标系$\{s\}$的位形T_{sb_phi}

```
Tsb_phi = [cos(phi) -sin(phi)  0  x;
           sin(phi)  cos(phi)  0  y;
           0         0         1  0.0963;
           0         0         0  1];
```

ix. 末端执行器坐标系$\{e\}$相对于空间坐标系$\{s\}$的位形

```
Tse = Tsb_phi*Tb0*T0e;
X = Tse;
```

x. 读取生成轨迹 result_traj 的下一行作为 i+1 时刻期望位形X_d

```
Xd =
[result_traj(i+1,1) result_traj(i+1,2) result_traj(i+1,3) result_traj(i+1,10);
 result_traj(i+1,4) result_traj(i+1,5) result_traj(i+1,6) result_traj(i+1,11);
 result_traj(i+1,7) result_traj(i+1,8) result_traj(i+1,9) result_traj(i+1,12);
 0                  0                  0                  1];
```

xi. 读取生成轨迹 result_traj 的下一行的后一行作为 i+2 时刻期望位形$X_{d,next}$

```
Xdnext =
```

```
[result_traj(i+2,1) result_traj(i+2,2) result_traj(i+2,3) result_traj(i+2,10);
result_traj(i+2,4) result_traj(i+2,5) result_traj(i+2,6) result_traj(i+2,11);
result_traj(i+2,7) result_traj(i+2,8) result_traj(i+2,9) result_traj(i+2,12);
0             0             0             1];
```

xii. 将下一时刻位形作为当前位形，结束 i 时刻循环，进入 i+1 时刻循环

4) 循环完毕，记录所有位形、误差，保存为 csv 文件。

3. 测试主函数 Main_Function

1) 控制目标：调整 TrajectoryGenerator 里的 Tscfin 改变小物块最终的位置，例如

```
Tscfin = [cos(pi/4) -sin(pi/4) 0   1.5;
     sin(pi/4)  cos(pi/4) 0   1.5;
   0            0           1   0.025;
    0            0           0   1];
```

2) 运行 CoppeliaSim，打开 scene->Scene6_youBot_cube，输入 result_conf.csv 文件，观察仿真模拟，调整 PI 控制增益，以保证小物块在搬运中不会掉落。

3) 在 Matlab 中绘制 result_Xerr.csv 曲线。

并测试控制效果，观察仿真模拟。