

# Quadruped\_Robot

## 目录

0 Brief Introduction.....	1
0.1 项目效果.....	1
0.2 参考资料.....	1
1 Parameters.....	1
2 Leg Part.....	2
2.1 Leg Kinematics & Dynamics.....	2
2.2 Leg Controller.....	3
3 Body Part.....	3
3.1 Body Kinematics & Dynamics.....	3
3.2 Body Controller.....	3
4 仿真.....	4

## 0 Brief Introduction

### 0.1 项目效果

本项目为四足机器人设计了简易的 MPC 平衡控制器，腿部则采用 PID 控制器。目前可实现原地段时间踏步，或者长时间原地站立。

待改进：

- 踏步过程 MPC 的姿态控制很有问题，无法有效收敛到目标姿态。
- 只能短时间原地踏步，无法长时间踏步或者在平面上平动。

### 0.2 参考资料

- 《机器人学导论》：了解基本运动学和动力学
- 《四足机器人控制算法》（宇树出版）：较为详细的四足机器人控制实现过程，有上本书基础的话阅读难度较低。
- 数学基础：线性代数 & 高等线性代数
- 控制理论学习（经典控制、现代控制等）：b 站 DR\_CAN: [https://space.bilibili.com/230105574?spm\\_id\\_from=333.788.0.0](https://space.bilibili.com/230105574?spm_id_from=333.788.0.0); 《控制之美》
- MPC 基础：[https://www.bilibili.com/video/BV19v4y1c7ya/?spm\\_id\\_from=333.999.top\\_right\\_bar\\_window\\_custom\\_collection.content.click&vd\\_source=d53a375084b7127af1236ed8b485b543](https://www.bilibili.com/video/BV19v4y1c7ya/?spm_id_from=333.999.top_right_bar_window_custom_collection.content.click&vd_source=d53a375084b7127af1236ed8b485b543)
- 倒立摆实践：参考视频：[https://www.bilibili.com/video/BV19M4y1E799/?spm\\_id\\_from=333.788.top\\_right\\_bar\\_window\\_custom\\_collection.content.click&vd\\_source=d53a375084b7127af1236ed8b485b543](https://www.bilibili.com/video/BV19M4y1E799/?spm_id_from=333.788.top_right_bar_window_custom_collection.content.click&vd_source=d53a375084b7127af1236ed8b485b543)；同目录下 “Inverted\_Pendulum” 推荐自己实现一边，可以对 simulink 的工作流有一个基本了解

```
clear; clc;
```

## 1 Parameters

定义机器人的一些身体参数：

```

param = struct;

param.T = 0.01;
param.Np = 8;

param.l1 = 0.2;
param.l2 = 0.8;
param.l3 = 0.8;

param.Ib = [5 0 0;0 5 0;0 0 5];
param.m = 20;

param.n_in = 12;

model = struct;

```

## 2 Leg Part

本部分主要实现腿部运动学和动力学建立，并设计用于腿部控制的 PID 控制器。

### 2.1 Leg Kinematics & Dynamics

[腿部模型建立](#)

```
run("./Leg_modeling.mlx");
```

对腿部模型进行实例化：

```

fileID = fopen("./LesMatrices.txt","w");
fprintf(fileID, '%s\n', "Matrixs for leg part");

model.P_ans = subs( ...
    Leg.P_ans, ...
    {Leg.l1, Leg.l2, Leg.l3}, ...
    {param.l1, param.l2, param.l3});

fileID = fopen("./LesMatrices.txt","a");
fprintf(fileID, '%s\n', "P_ans:");
fprintf(fileID, '%s ', "[" );
fprintf(fileID, '%s;%s;%s;\n', model.P_ans);
fprintf(fileID, '%s\n', "]" );

model.q_des = subs( ...
    Leg.q_des, ...
    {Leg.l1, Leg.l2, Leg.l3}, ...
    {param.l1, param.l2, param.l3});

fileID = fopen("./LesMatrices.txt","a");
fprintf(fileID, '%s\n', "q_des:");
fprintf(fileID, '%s ', "[" );
fprintf(fileID, '%s;%s;%s;\n', model.q_des);

```

```

fprintf(fileID, '%s\n', "J");

model.J = subs( ...
    Leg.J, ...
    {Leg.l1, Leg.l2, Leg.l3}, ...
    {param.l1, param.l2, param.l3});

fileID = fopen("./LesMatrices.txt", "a");
fprintf(fileID, '%s\n', "J:");
fprintf(fileID, '%s ', "[");
fprintf(fileID, '%s,%s,%s;\n', model.J');
fprintf(fileID, '%s\n', "J");

model.torques = subs( ...
    Leg.torques, ...
    {Leg.l1, Leg.l2, Leg.l3}, ...
    {param.l1, param.l2, param.l3});

fileID = fopen("./LesMatrices.txt", "a");
fprintf(fileID, '%s\n', "torques:");
fprintf(fileID, '%s ', "[");
fprintf(fileID, '%s;%s;%s;\n', model.torques);
fprintf(fileID, '%s\n', "J");

```

## 2.2 Leg Controller

```

% leg controller PID
leg_controller = struct;
leg_controller.kp = [1000;1050;1050];
leg_controller.ki = [2;2.5;1.5];
leg_controller.kd = [20;85;80];

```

## 3 Body Part

本部分 主要实现机身的运动学和动力学建立。对于全身，我们采用单刚体模型（SRB）进行简化。

### 3.1 Body Kinematics & Dynamics

[机体模型建立](#)

```

run("./Body_modeling.mlx");

```

### 3.2 Body Controller

首先对状态空间进行实例化，再进行 MPC 的预测部分：

```

model.A = Body.A;
model.B = Body.B;
model.C = Body.C;
model.D = Body.D;

% sys = ss(model.A,model.B,model.C,model.D);

```

```

% sys_d = c2d(sys,0.01);
% model.A = param.T.* model.A + eye(13);
% model.B = param.T .* model.B;

fileID = fopen("./LesMatrices.txt","a");
fprintf(fileID, '%s\n', "Matrixs for body part");
fprintf(fileID, '%s\n', "A:");
fprintf(fileID, '%s ', "[" );
fprintf(fileID, '%s,%s,%s,%s,%s;\n', model.A');
fprintf(fileID, '%s\n', "]" );

fprintf(fileID, '%s\n', "B:");
fprintf(fileID, '%s ', "[" );
fprintf(fileID, '%s,%s,%s,%s;\n', model.B');
fprintf(fileID, '%s\n', "]" );

```

拼接矩阵，对后续状态进行预测：

```

model.q = [1000,2000,4000,200,200,200,3000,3000,3000,200,200,200,0];
model.Q = diag(repmat(model.q,1,param.Np));
model.R = 0.0001 * eye(param.n_in*param.Np);
% 二次型应该在仿真控制器中计算，因为与系统的实时状态有关
% 后面的实现在 simulink 中完成

```

## 4 仿真

仿真程序

```

% run("./MarchMPC.slx");

```