



深蓝学院  
shenlanxueyuan.com

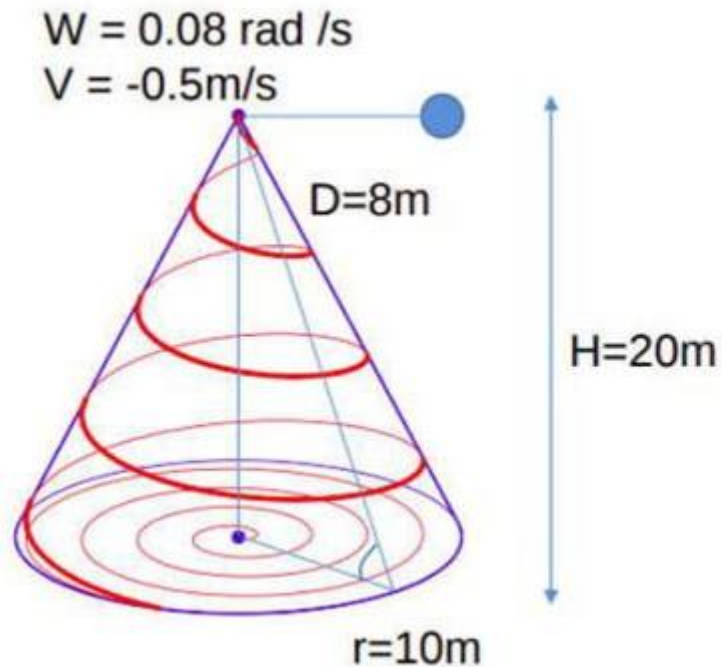
## 第八章作业思路分享



主讲人 丁成科



# 第一题：MPC



$$-6 \leq v_{x,y} \leq 6$$

$$-1 \leq v_z \leq 6$$

$$-3 \leq a_{x,y} \leq 3$$

$$-1 \leq a_z \leq 3$$

$$-3 \leq j_{x,y} \leq 3$$

$$-2 \leq j_z \leq 2$$

# 第一题：MPC

计算MPC时依赖于当前状态后4s的目标状态：

因此每一次计算MPC前都需要计算目标状态

(1). 计算跟踪目标即螺旋线的p, v, a

(2). X, Y方向为圆周运动

(3). z方向为直线运动

```
%% Construct the reference signal
```

```
for i = 1:20
```

```
    tref = t + i*0.2;
```

```
    r=0.25*tref;
```

```
    w = 0.2;
```

```
    pt(i,1) = r*sin(w*tref);
```

```
    vt(i,1) = r*cos(w*tref);
```

```
    at(i,1) = -r*sin(w*tref);
```

```
    pt(i,2) = r*cos(w*tref);
```

```
    vt(i,2) = -r*sin(w*tref);
```

```
    at(i,2) = -r*cos(w*tref);
```

```
    pt(i,3) = 20 - 0.5*tref;
```

```
    vt(i,3) = -0.5;
```

```
    at(i,3) = 0;
```

# 第一题：MPC

在MPC的主流程中(以x, y方向为例, 即xy\_axis\_mpc.m)

本次跟踪的目标状态会随时间变化 (例题目标恒定为0)

因此cost function 并不是

$$\min_J w_1 P^\top P + w_2 V^\top V + w_3 A^\top A + w_4 J^\top J$$

需要把P改写成 $(P - P_{\text{target}})$ , V改写成 $(V - V_{\text{target}})$ 等, 重新写出新的optimization target

这里框架已经给出答案

```
%% Construct the optimization problem
```

```
H = w4*eye(K)+w1*(Tp'*Tp)+w2*(Tv'*Tv)+w3*(Ta'*Ta);
```

```
F = w1*(Bp-pt)'*Tp+w2*(Bv-vt)'*Tv+w3*(Ba-at)'*Ta;
```

```
A = [Tv;-Tv;Ta;-Ta];
```

```
b = [6*ones(20,1)-Bv; 6*ones(20,1)+Bv; 3*ones(20,1)-Ba; 3*ones(20,1)+Ba];
```

# 第一题：MPC

在作业的框架下，需要填写的部分只有下面三行

分别是x, y, z三个方向的MPC

```
%% Please follow the example in linear mpc part to fill in the code here to do the tracking
j(1) = xy_axis_mpc(K, dt, p_0(1), v_0(1), a_0(1), pt(:, 1), vt(:, 1), at(:, 1));
j(2) = xy_axis_mpc(K, dt, p_0(2), v_0(2), a_0(2), pt(:, 2), vt(:, 2), at(:, 2));
j(3) = z_axis_mpc(K, dt, p_0(3), v_0(3), a_0(3), pt(:, 3), vt(:, 3), at(:, 3));
```

它们的本质就是使用quadprog求解出未来四秒内的J，

并取第一项作为当前的输入

# 第一题：MPC

注意：

如果螺旋线不完整，需要调整 $\omega$ 的大小

```
%% Construct the reference signal
```

```
for i = 1:20
```

```
    tref = t + i*0.2;
```

```
    r=0.25*tref;
```

```
    w = 0.7;
```

```
    pt(i,1) = r*sin(w*tref);
```

```
    vt(i,1) = r*cos(w*tref);
```

```
    at(i,1) = -r*sin(w*tref);
```

```
    pt(i,2) = r*cos(w*tref);
```

```
    vt(i,2) = -r*sin(w*tref);
```

```
    at(i,2) = -r*cos(w*tref);
```

```
    pt(i,3) = 20 - 0.5*tref;
```

```
    vt(i,3) = -0.5;
```

```
    at(i,3) = 0;
```

```
end
```

## 第二题：PSO

首先要理解代码中粒子的结构：

```
p[i,1]    p[i,2]    p[i,3]    p[i,4]    p[i,5]    p[i,6]    p[i,7]  
%theta, v_end, v_theta, v_vend, best_theta, best_v, best_cost
```

其中：

$P[i, 1:2]$  为粒子的状态

$P[i, 3:4]$  为粒子的速度

$P[i, 5:6]$  为粒子的历史最优状态

## 第二题：PSO

因此我们可以参照伪代码去更新粒子的状态和速度

```
for each  $\theta_i \in \Theta$  do  
     $\delta_i = \delta_i + k_1 \cdot \text{rand} \cdot (\theta_i^* - \theta_i) + k_2 \cdot \text{rand} \cdot$   
         $(\theta^* - \theta_i)$   
     $\theta_i = \theta_i + \delta_i$ 
```

因此我们可以参照伪代码去更新粒子的速度和状态

```
P(i, 3) = rand * (global_best(1) - P(i, 1)) + rand * (P(i, 5) - P(i, 1)) + P(i, 3);
```

```
P(i, 4) = rand * (global_best(2) - P(i, 2)) + rand * (P(i, 6) - P(i, 2)) + P(i, 4);
```

```
P(i, 1) = P(i, 1) + P(i, 3);
```

```
P(i, 2) = P(i, 2) + P(i, 4);
```



## 第二题：PSO

其中：新的速度= rand\*全局最优速度 + rand\*历史最优速度 + 上一次的速度

$P(i,3) = \text{rand} * (\text{global\_best}(1) - P(i,1)) + \text{rand} * (P(i,5) - P(i,1)) + P(i,3);$

$P(i,4) = \text{rand} * (\text{global\_best}(2) - P(i,2)) + \text{rand} * (P(i,6) - P(i,2)) + P(i,4);$

$P(i,1) = P(i,1) + P(i,3);$

$P(i,2) = P(i,2) + P(i,4);$

$P(i,1:2) = \text{limitRange}(P(i,1:2));$

```
%% Simulation starts here
for n=1:600
    % Use the PSO algorithm to select the best velocity and angular velocity
    % target
    global_best = pso_select(theta,omega,v_c,[x y]',theta_target,v_target);
    theta_target = global_best(1);
    v_target = global_best(2);
end
```

- 伪代码中的系数w, k1, k2等可以简单设为1即可
- 同时注意题目需要限制粒子速度，因此需要加入limitRange()函数
- 加入limitRange()后，需要加长仿真时间，将仿真次数由300改为600或以上

## 第二题：PSO

最终收敛的位置在cost最小的位置附近

而在本代码中记录cost的地方在hMap中

hMap										
640x640 double										
	382	383	384	385	386	387	388	389	390	391
570	8.0711	7.6569	7.2426	6.8284	6.4142	6	6.4142	6.8284	7.2426	7.6569
571	7.0711	6.6569	6.2426	5.8284	5.4142	5	5.4142	5.8284	6.2426	6.6569
572	6.6569	5.6569	5.2426	4.8284	4.4142	4	4.4142	4.8284	5.2426	5.6569
573	6.2426	5.2426	4.2426	3.8284	3.4142	3	3.4142	3.8284	4.2426	5.2426
574	5.8284	4.8284	3.8284	2.8284	2.4142	2	2.4142	2.8284	3.8284	4.8284
575	5.4142	4.4142	3.4142	2.4142	1.4142	1	1.4142	2.4142	3.4142	4.4142
576	5	4	3	2	1	0	1	2	3	4
577	5.4142	4.4142	3.4142	2.4142	1.4142	1	1.4142	2.4142	3.4142	4.4142
578	5.8284	4.8284	3.8284	2.8284	2.4142	2	2.4142	2.8284	3.8284	4.8284
579	6.2426	5.2426	4.2426	3.8284	3.4142	3	3.4142	3.8284	4.2426	5.2426
580	6.6569	5.6569	5.2426	4.8284	4.4142	4	4.4142	4.8284	5.2426	5.6569
581	7.0711	6.6569	6.2426	5.8284	5.4142	5	5.4142	5.8284	6.2426	6.6569
582	8.0711	7.6569	7.2426	6.8284	6.4142	6	6.4142	6.8284	7.2426	7.6569

感谢各位聆听!

Thanks for Listening

