第四章作业思路

主讲人 刘武

第一题

- For the OBVP problem stated in slides p.25-p.29, please get the optimal solution (control, state, and time) for partially free final state case.
- Suppose the position is fixed, velocity and acceleration are free here.

第一题

应用庞特里亚金极小值原理求解控制有约束变分问题过程：
针对一般目标函数：

$$J=\varphi(\mathbf{s}(T),T)+\int_0^T L(\mathbf{s},\boldsymbol{u},t)dt$$

求解步骤一般需要使用(参考自动控制原理第十章（胡寿松))：

- 正则方程
- 边界条件与横截条件
- 极小值条件
- H 变化率

本题的 OBVP 问题描述(**假设末端位置固定，速度和加速度自由**)：

$$\begin{cases}\min\limits_{u(t)\in\Omega} J_{sum}=\sum\limits_{k=1}^{3} J_k,\ J_k=\dfrac{1}{T}\int_0^T j_k(t)^2\,dt\\ s.t.\quad \dot{s}_k=f(s_k,u_k)=(v_k,a_k,j_k)\\ \qquad s_k(0)=(p_k^0,v_k^0,a_k^0)\\ \qquad s_k(T)=(p_k^T,free,free)\end{cases}$$

本题目标函数中：$\varphi(\mathbf{s}(T),T)=0$，$L(\mathbf{s},\boldsymbol{u},t)=\dfrac{1}{T}j^2(t)$

PPT中：

For fixed final state problem:

$$h(s(T))=\begin{cases}\boxed{0,\quad if\ \ s=s(T)}\\ \infty,\quad otherwise\end{cases}$$

Not differentiable

第一题

假设协态 $\boldsymbol{\lambda}^T=(\lambda_1,\lambda_2,\lambda_3)$，则哈密尔顿函数为:

$$H=L(\boldsymbol{s},\boldsymbol{u},t)+\boldsymbol{\lambda}^T f(\boldsymbol{s},\boldsymbol{u})$$
$$=\frac{1}{T}j^2+\lambda_1 v+\lambda_2 a+\lambda_3 j$$

通过正则方程 $\dot{\boldsymbol{\lambda}}(t)=\frac{\partial H}{\partial \boldsymbol{s}}$ 得:

$$\dot{\boldsymbol{\lambda}}(t)=\begin{bmatrix}0\\-\lambda_1\\-\lambda_2\end{bmatrix}$$

通过边界条件得:

$$\begin{bmatrix}\lambda_2(T)\\\lambda_3(T)\end{bmatrix}=\begin{bmatrix}\frac{\partial\varphi}{\partial v}\\\frac{\partial\varphi}{\partial a}\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}$$

PPT中：

For (partially)-free final state problem:

$$given\quad s_i(T),\ i\in I$$

We have boundary condition for other costate:

$$\lambda_j(T)=\frac{\partial h(s^*(T))}{\partial s_j},\quad for\ j\neq i$$

Then we solve this problem again.

第一题

$$\lambda(t)=\frac{1}{T}\begin{bmatrix}-2\alpha\\ 2\alpha(t-T)\\ -\alpha(t-T)^2\end{bmatrix}$$

应用极小值原理 $\boldsymbol{j}^*(t)=\boldsymbol{u}^*(t)=\arg\min\limits_{u\in\Omega} H(\boldsymbol{s}^*(t),\boldsymbol{u}(t),\boldsymbol{\lambda}(t))$ 得：

$$\boldsymbol{j}^*(t)=\boldsymbol{u}^*(t)=\arg\min_{j\in\Omega}[\frac{1}{T}j^2+\lambda_1 v+\lambda_2 a+\lambda_3 j]$$
$$=-\frac{\lambda_3 T}{2}=\frac{1}{2}\alpha(t-T)^2$$

根据最优控制输入 $\boldsymbol{u}^*(t)$ 和初始状态 $s(0)=(p_0,v_0,a_0)$ 可积分得最优状态：

$$\boldsymbol{s}^*(t)=\begin{bmatrix}\frac{\alpha}{120}(t-T)^5+\frac{1}{2}(a_0+\frac{\alpha}{6}T^3)t^2+(v_0-\frac{\alpha}{24}T^4)t+(p_0+\frac{\alpha}{120}T^5)\\ \frac{\alpha}{24}(t-T)^4+(a_0+\frac{\alpha}{6}T^3)t+(v_0-\frac{\alpha}{24}T^4)\\ \frac{\alpha}{6}(t-T)^3+(a_0+\frac{\alpha}{6}T^3)\end{bmatrix}$$

最后根据终点状态 $p(T)=p_f=\frac{1}{2}(a_0+\frac{\alpha}{6}T^3)T^2+(v_0-\frac{\alpha}{24}T^4)T+(p_0+\frac{\alpha}{120}T^5)$，可以求解变量 α 为：

$$\alpha=\frac{20\Delta p}{T^5},\ \Delta p=p_f-p_0-\frac{1}{2}a_0T^2-v_0T$$

综上，可得优化目标函数：

$$J=\int_0^T\frac{1}{T}j^*(t)^2dt=\int_0^T\frac{1}{T}(\frac{10\Delta p}{T^5}(t-T))^2dt$$

第二题

- Build an ego-graph of the linear modeled robot.
- Select the best trajectory closest to the planning target.

STEP1:

$$v = v_0 + a * t$$

$$p = p_0 + v_0 * t + 0.5 * a * t^2$$

STEP2:

$$J = \int_0^T g(x,u)dt = \int_0^T (1 + u^T R u)dt = \int_0^T (1 + a_x^2 + a_y^2 + a_z^2)dt$$

$$\frac{dJ(T)}{dT} = 0$$

第二题

J:

```
1.0*(1.0*T**4 + 4.0*T**2*v_x_0**2 +
4.0*T**2*v_y_0**2 + 4.0*T**2*v_z_0**2
+ 12.0*T*p_x_0*v_x_0 -
12.0*T*p_x_f*v_x_0 +
12.0*T*p_y_0*v_y_0 -
12.0*T*p_y_f*v_y_0 +
12.0*T*p_z_0*v_z_0 - 12.0*T*p_z_f*v_z_0
+ 12.0*p_x_0**2 - 24.0*p_x_0*p_x_f +
12.0*p_x_f**2 + 12.0*p_y_0**2 -
24.0*p_y_0*p_y_f + 12.0*p_y_f**2 +
12.0*p_z_0**2 - 24.0*p_z_0*p_z_f +
12.0*p_z_f**2)/T**3
```

对J求导:

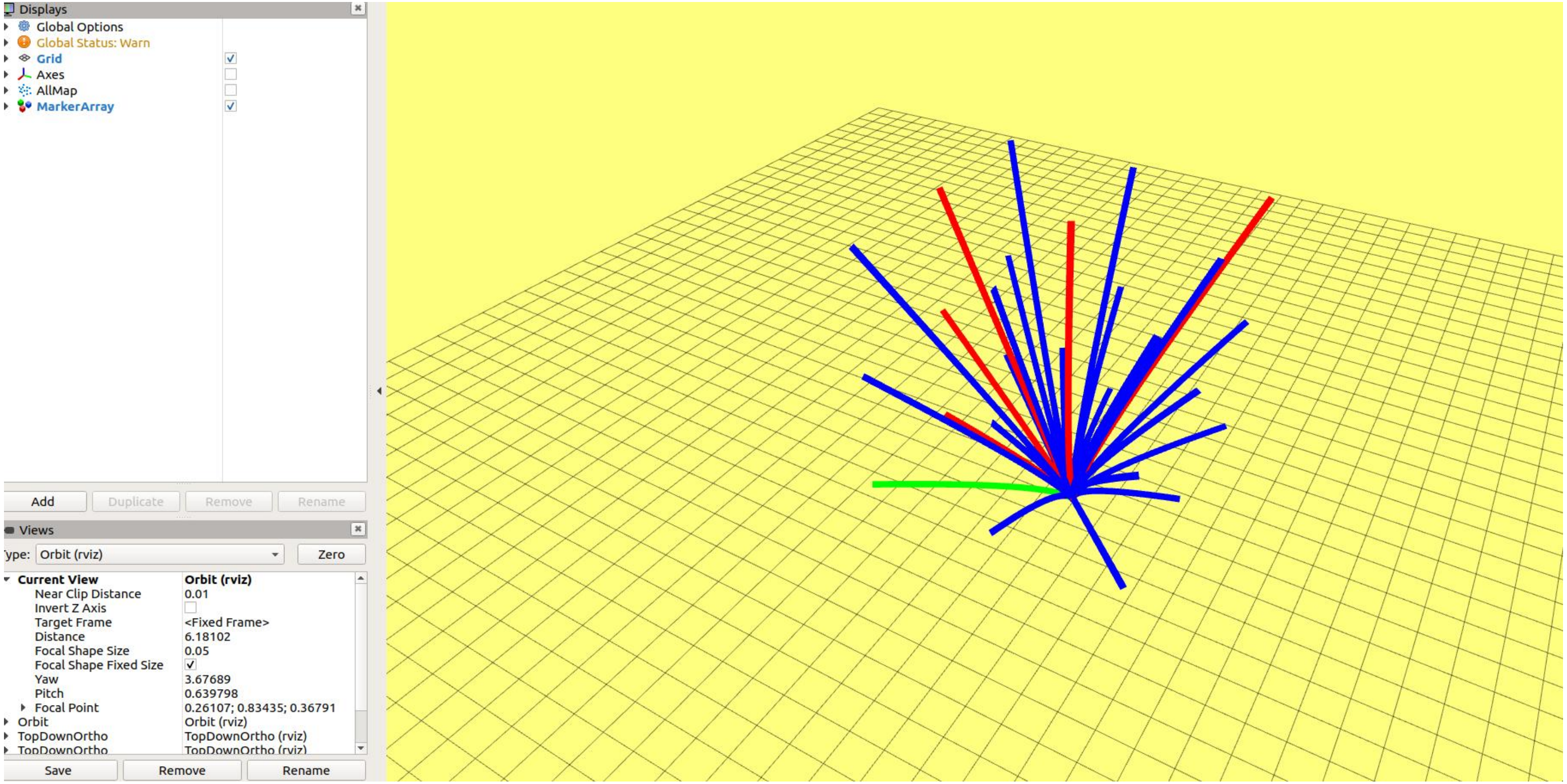
```
1.0*(1.0*T**4 - 4.0*T**2*v_x_0**2 -
4.0*T**2*v_y_0**2 - 4.0*T**2*v_z_0**2
- 24.0*T*p_x_0*v_x_0 +
24.0*T*p_x_f*v_x_0 -
24.0*T*p_y_0*v_y_0 +
24.0*T*p_y_f*v_y_0 -
24.0*T*p_z_0*v_z_0 +
24.0*T*p_z_f*v_z_0 - 36.0*p_x_0**2 +
72.0*p_x_0*p_x_f - 36.0*p_x_f**2 -
36.0*p_y_0**2 + 72.0*p_y_0*p_y_f -
36.0*p_y_f**2 - 36.0*p_z_0**2 +
72.0*p_z_0*p_z_f - 36.0*p_z_f**2)/T**4
```

第二题

```
double p_x_0 = _start_position(0);
double p_y_0 = _start_position(1);
double p_z_0 = _start_position(2);
double p_x_f = _target_position(0);
double p_y_f = _target_position(1);
double p_z_f = _target_position(2);
double v_x_0 = _start_velocity(0);
double v_y_0 = _start_velocity(1);
double v_z_0 = _start_velocity(2);
double c0 =
    -36.0 * pow(p_x_0, 2) + 72.0 * p_x_0 * p_x_f - 36.0 * pow(p_x_f, 2) -
    36.0 * pow(p_y_0, 2) + 72.0 * p_y_0 * p_y_f - 36.0 * pow(p_y_f, 2) -
    36.0 * pow(p_z_0, 2) + 72.0 * p_z_0 * p_z_f - 36.0 * pow(p_z_f, 2);

double c1 = -24.0 * p_x_0 * v_x_0 + 24.0 * p_x_f * v_x_0 -
            24.0 * p_y_0 * v_y_0 + 24.0 * p_y_f * v_y_0 -
            24.0 * p_z_0 * v_z_0 + 24.0 * p_z_f * v_z_0;

double c2 = -4.0 * pow(v_x_0, 2) - 4.0 * pow(v_y_0, 2) - 4.0 * pow(v_z_0, 2);
double c3 = 0.0;

Eigen::Matrix<double, 4, 4> matrix_44;
Eigen::Matrix<complex<double>, Eigen::Dynamic, Eigen::Dynamic>
    matrix_eigenvalues;
matrix_44 << 0, 0, 0, -c0, 1, 0, 0, -c1, 0, 1, 0, -c2, 0, 0, 1, -c3;

matrix_eigenvalues = matrix_44.eigenvalues();
```

```
for (int i = 0; i < matrix_eigenvalues.size(); i++) {
  if (matrix_eigenvalues(i).imag() == 0.0 &&
      matrix_eigenvalues(i).real() > 0.0) {
    double t = matrix_eigenvalues(i).real();
    //   std::cout << "optimal_t:" << t << std::endl;
    double optimal_J = computerJ(t, p_x_0, p_y_0, p_z_0, p_x_f, p_y_f, p_z_f,
                                 v_x_0, v_y_0, v_z_0);
    if (optimal_J < optimal_cost)
      optimal_cost = optimal_J;
  }
}
```

第二题



Displays
Global Options
Global Status: Warn
Grid
Axes
AllMap
MarkerArray
Add
Duplicate
Remove
Rename
Views
Type: Orbit (rviz)
Zero
Current View
Orbit (rviz)
Near Clip Distance
0.01
Invert Z Axis
Target Frame
<Fixed Frame>
Distance
6.18102
Focal Shape Size
0.05
Focal Shape Fixed Size
Yaw
3.67689
Pitch
0.639798
Focal Point
0.26107; 0.83435; 0.36791
Orbit
Orbit (rviz)
TopDownOrtho
TopDownOrtho (rviz)
TopDownOrtho
TopDownOrtho (rviz)
Save
Remove
Rename

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening