

1. 根据机器人动力学模型，计算出离散化输入量 a 下，机器人在给定时间 T 后的状态 (forward integration)

```
pos(0) = pos(0) + vel(0)*delta_time+1/2*acc_input(0)*delta_time*delta_time; // x1 = x0+v0*t + 1/2*a*t^2
pos(1) = pos(1) + vel(1)*delta_time+1/2*acc_input(1)*delta_time*delta_time;
pos(2) = pos(2) + vel(2)*delta_time+1/2*acc_input(2)*delta_time*delta_time;

vel(0) = vel(0) + acc_input(0)*delta_time; // v1 = v0+a*t
vel(1) = vel(1) + acc_input(1)*delta_time;
vel(2) = vel(2) + acc_input(2)*delta_time;
```

2. 以机器人在运动时间 T 后的状态为起始状态，目标状态为终末状，计算出这两个状态之间的最小动力学损失 (OBVP) **non-holonomic-without obstacle**

$$J = T + \left(\frac{1}{3} \alpha_1^2 T^3 + \alpha_1 \beta_1 T^2 + \beta_1^2 T \right) + \left(\frac{1}{3} \alpha_2^2 T^3 + \alpha_2 \beta_2 T^2 + \beta_2^2 T \right) + \left(\frac{1}{3} \alpha_3^2 T^3 + \alpha_3 \beta_3 T^2 + \beta_3^2 T \right)$$

- 2.1 求出 T ; 用 mma 软件求得 J_{dt} 表达式，然后暴力求解求出使得 $J_{dt}=0$ 的大概 T 值

```
T = 0.0001; //初始化时间
// 将已知量带入J对于t的求导后的方程，暴力求解T值
do
{
    J_dt = (-36*pow(Pfx,2) - 36*pow((Pfx - Piz),2) + pow(T,4) + 24*Pfx*(3*Pix+T*Vix) - 4*pow((3*Pix+T*Vix),2)
    - 4*pow((-3*Pfy+3*Piy+T*Viy),2) + 24*(Pfx-Piz)*T*Viz - 4*pow(T,2)*pow(Viz,2))/pow(T,4);
    if(abs(J_dt)<0.001) // 求出使得J_dt大概为0时的T值
    {
        // ROS_INFO(" break at time: %f", T);
        break;
    }
    else
        T += 0.001;
}while(T<20);
```

- 2.2 再将 T 代入方程求出 J

```
//计算出J在T时刻的cost(大约极小值)即启发方程-> 考虑运动学 不考虑障碍物的函数
J_cost = T+(1/3*pow(alpha1,2)*pow(T,3)+ alpha1*beta1*pow(T,2)+ pow(beta1,2)*T)
+ (1/3*pow(alpha2,2)*pow(T,3)+ alpha2*beta2*pow(T,2)+ pow(beta2,2)*T)
+ (1/3*pow(alpha3,2)*pow(T,3)+ alpha3*beta3*pow(T,2)+ pow(beta3,2)*T);
```

3. 用 a^* 算法求出考虑障碍物但不考虑运动学的情况下，从机器人在运动时间 T 后的位置到目标位置的 cost,

```
Astar_tool->AstarGraphSearch(pos, target_pt);
// 考虑障碍物 不考虑运动学的cost
Astar_Cost = Astar_tool->GetFinalCost();

Astar_tool->resetUsedGrids(); // reset map
```

4. 以两种代价的和作为引导，选择最优路径
Heuristic_Cost = Trajectory_Cost+Astar_Cost;

```
if(Heuristic_Cost<min_Cost && TraLibrary[i][j][k]->collision_check == false){
    a = i;
    b = j;
    c = k;
    min_Cost = Heuristic_Cost;
}
```

5. 结果展示

