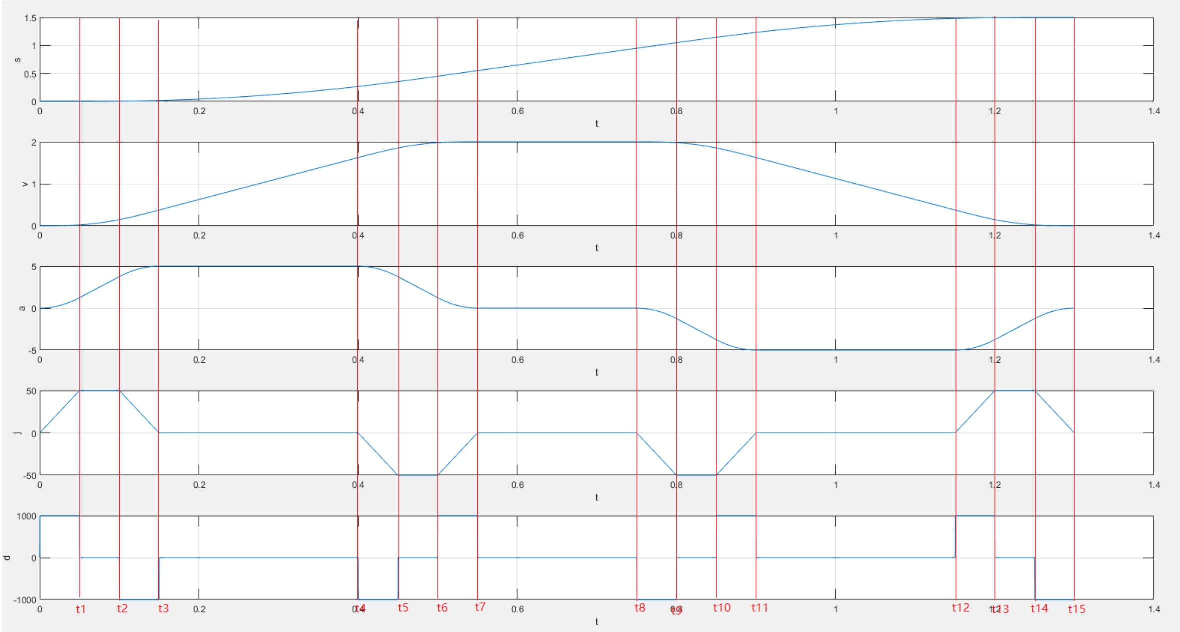
4阶轨迹规划算法介绍



上图所示的为一种典型的完整对称4阶轨迹轮廓，所示的轨迹规划可由4段时间唯一确定，最大加加速度斜率时间段t_d，最大加加速度时间段t_j，最大加速度时间段t_a，最大速度时间段t_v。确定了以上4段时间，即确定了算法实现过程中所需的所有切换点时间$t_0, t_1, t_2, \cdots t_{15}$，显然，不同的4段时间值确定不同的轨迹轮廓，15段是在4阶轨迹规划时所有时间段都存在的情况。

因此，4阶轨迹规划算法即是在保证轨迹规划实用目标的基础上，一句给定的约束条件，依次确定时间段t_d、t_j、t_a、t_v的规划过程。步骤如下：

- 最大加加速度斜率时间段t_d的确定。
- 最大加加速度时间段t_j的确定。
- 最大加速度时间段t_a的确定。
- 最大速度时间段t_v的确定。

给定4阶轨迹的约束条件为运动距离s，最大运动速度v_{max}，最大加速度a_{max}，最大加加速度j_{max}，最大加加速度斜率d_{max}.

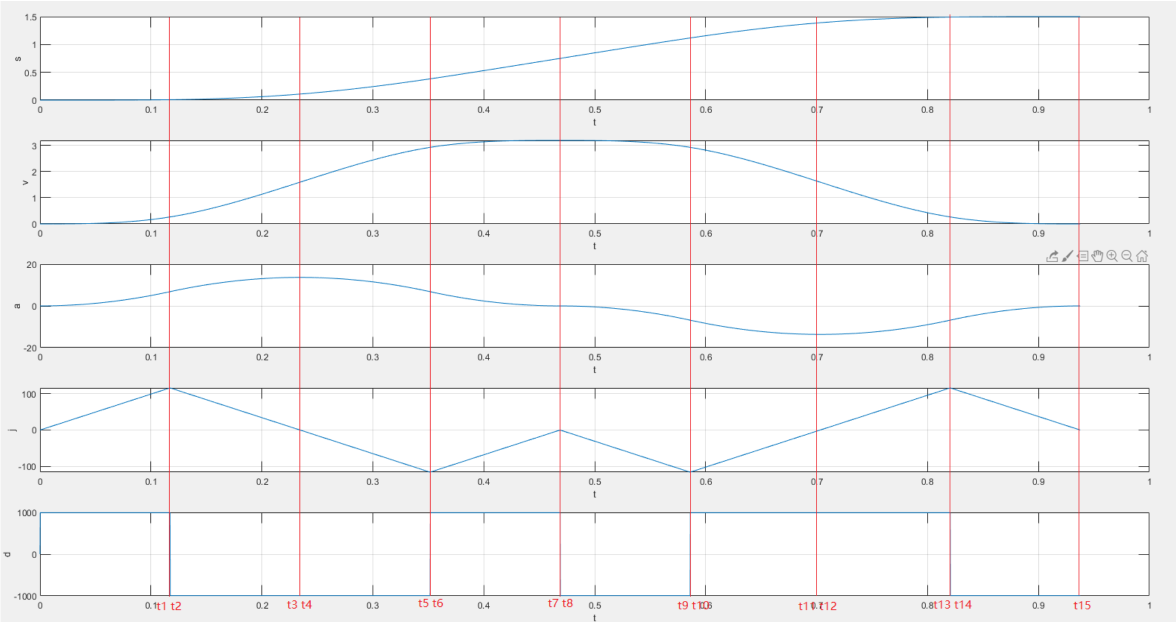
根据积分原理，我们可以得知，加加速度是加加速度斜率的一重积分，加速度为加加速度斜率的二重积分，速度为加加速度斜率的三重积分，位置则为加加速度斜率的四重积分。因此，可以得到以下的公式：

$$\left.\begin{aligned} j(t) &= j_0 + dt \\ a(t) &= a_0 + j_0 t + dt^2/2 \\ v(t) &= v_0 + a_0 t + j_0 t^2/2 + dt^3/6 \\ s(t) &= s_0 + v_0 t + a_0 t^2/2 + j_0 t^3/6 + dt^4/24 \end{aligned}\right\} \tag{1}$$

式中，j_0, a_0, v_0, s_0为初始边界条件，t为时间。

确定最大加加速度斜率时间段的最优值

- 首先不考虑最大加加速度、最大加速度和最大速度限制。在满足运动距离的前提下，建立运动距离与最大加加速度斜率的关系，因此，
$t_j = t_a = t_v = 0, t_2 = t_1 = t_d, t_4 = t_3 = 2t_d, t_6 = t_5 = 3t_d, t_8 = t_7 = 4t_d$



整个轨迹的执行时间$t_s = 8t_d$，设定初始时刻和结束时刻的加加速度、加速度和速度均为0，可以得到公式(1)中的初始条件为$j_0 = 0$，$a_0 = 0$，$v_0 = 0$，$s_0 = 0$，可以得到t_1时刻的计算公式为：

$$\left.\begin{aligned} j(t_1) &= d_{max}t_d \\ a(t_1) &= d_{max}t_d^2/2 \\ v(t_1) &= d_{max}t_d^3/6 \\ s(t_1) &= d_{max}t_d^4/24 \end{aligned}\right\} \tag{2}$$

在t_3时刻，有$d = -d_{max}$，则：

$$\left.\begin{aligned} j(t_3) &= 0 \\ a(t_3) &= d_{max}t_d^2 \\ v(t_3) &= d_{max}t_d^3 \\ s(t_3) &= 7d_{max}t_d^4/12 \end{aligned}\right\} \tag{3}$$

依次类推，可以得到t_7时刻的计算公式：

$$\left.\begin{aligned} j(t_7) &= 0 \\ a(t_7) &= 0 \\ v(t_7) &= 2d_{max}t_d^3 \\ s(t_7) &= 4d_{max}t_d^4 \end{aligned}\right\} \tag{4}$$

根据轨迹轮廓对称性，$s(t_{15}) = 2(s_7) = s$，所以可以得到由运动距离所确定的最大加加速度斜率段时间为

$$t_d = \sqrt[4]{s/(8d_{max})} \tag{5}$$

```
t1 = pow((1.0/8.0 * TP->p / d),(1.0/4.0));
if(TP->Ts > 0)
{
    t1 = ceil(t1 / TP->Ts) * TP->Ts;
    *dd = 1.0/8.0 * TP->p / pow(t1, 4.0);

}
```

- 引入最大速度限制。最大速度在t_7时刻到达，即由t_d时间所确定的最大速度值为

$$v_d = 2d_{max}t_d^3 \tag{6}$$

比较给定的最大速度v_{max}与计算得到的v_d,若$v_{max} > v_d$,则t_d满足最大速度限制要求，否则按最大速度限制重新计算t_d：

$$t_d = \sqrt[3]{v_{max}/(2d_{max})} \tag{7}$$

```
if(TP->v < 2.0 * (*dd) * pow(t1, 3.0) )
{
    t1 = pow((1.0/2.0 * TP->v / d), (1.0/3.0));
    if(TP->Ts > 0)
    {
        t1 = ceil(t1 / TP->Ts) * TP->Ts;
        *dd = 1.0/2.0 * TP->v / pow(t1, 3.0);
    }
}
```

- 引入最大加速度限制。最大加速度在t_3时刻到达，即由t_d时间所确定的最大加速度值为

$$a_d = d_{max}t_d^2 \tag{8}$$

比较给定的最大加速度a_{max}和计算得到的最大加速度a_d，若$a_{max} > a_d$，则t_d满足最大加速度限制要求，否则按照最大加速度限制重新计算t_d：

$$t_d = \sqrt{a_{max}/d_{max}} \tag{9}$$

```
if(TP->a < (*dd) * pow(t1, 2.0))
{
    t1 = pow((TP->a / d), (1.0/2.0));
    if(TP->Ts > 0)
    {
        t1 = ceil(t1 / TP->Ts) * TP->Ts;
        *dd = TP->a / pow(t1, 2.0);
    }
}
```

- 引入最大加加速度限制。最大加加速度在t_1时刻到达，即由t_d时间所确定的最大加加速度值为：

$$j_d = d_{max}t_d$$

比较给定的最大加加速度j_{max}与计算得到的最大加加速度j_d，若$j_{max} > j_d$，则t_d满足最大加加速度限制要求，否则按照最大加加速度重新计算t_d：

$$t_d = j_{max}/d_{max} \tag{11}$$

```
if(TP->j < (*dd) * t1)
{
    t1 = TP->j / d;
    if(TP->Ts > 0)
    {
        t1 = ceil(t1 / TP->Ts) * TP->Ts;
        *dd = TP->j / t1;
    }
}
```

至此，满足所有约束的最大加加速度斜率段时间已确定。

确定最大加加速度段时间的最优值

- 首先不考虑最大加速度与最大速度限制。在摆正运动距离的前提下建立运动距离与最大加加速度之间的关系。通过上文得到t_d的最优值，有$t_j = 0$，即存在以下关系$t_1 = t_2, t_6 = t_5$,并且在t_2与t_1、t_5与t_6之间$d = 0$。由此能推导出t_2时刻对应的计算公式:

$$\left.\begin{aligned}
&j(t_2) = d_{max} t_d \\
&a(t_2) = d_{max} t_d t_j + d_{max} t_d^2/2 \\
&v(t_2) = d_{max} t_d t_j^2/2 + d_{max} t_d^2 t_j/2 + d_{max} t_d^3/6 \\
&s(t_2) = d_{max} t_d t_j^3/6 + d_{max} t_d^2 t_j^2/4 + d_{max} t_d^3 t_j/6 + d_{max} t_d^4/24
\end{aligned}\right\} \tag{12}$$

根据公式(1)推导出t_3、t_7时刻对应的公式。

$$\left.\begin{aligned}
&j(t_3) = 0 \\
&a(t_3) = d_{max} t_d t_j + d_{max} t_d^2 \\
&v(t_3) = d_{max} t_d t_j^2/2 + 3d_{max} t_d^2 t_j/2 + d_{max} t_d^3 \\
&s(t_3) = d_{max} t_d t_j^3/6 + 3d_{max} t_d^2 t_j^2/4 + 7d_{max} t_d^3 t_j/6 + 7d_{max} t_d^4/12
\end{aligned}\right\} \tag{13}$$

$$\left.\begin{aligned}
&j(t_7) = 0 \\
&s(t_7) = 0 \\
&v(t_7) = d_{max} t_d t_j^2 + 3d_{max} t_d^2 t_j + 2d_{max} t_d^3 \\
&s(t_7) = d_{max} t_d t_j^3 + 5d_{max} t_d^2 t_j^2 + 8d_{max} t_d^3 t_j + 4d_{max} t_d^4
\end{aligned}\right\} \tag{14}$$

根据轨迹对称性，$s(t_{15}) = 2s(t_7)$则:

$$s(t_{15}) = 2d_{max} t_d t_j^3 + 10d_{max} t_d^2 t_j^2 + 16d_{max} t_d^3 t_j + 8d_{max} t_d^4 \tag{15}$$

公式(15)是一个关于t_j的一元三次多项式,求解得到的实根即为由运动距离所确定的最大加加速度段时间t_j值。

```
P = -1.0/9.0 * pow(t1, 2.0);
Q = -1.0/27.0 * pow(t1,3.0) - TP->p / (4.0 * d * t1);
D = pow(P, 3.0) + pow(Q, 2.0);
R = pow((-Q + sqrt(D)), (1.0/3.0));
t2 = R - P / R - 5.0/3.0 * t1;
if(fabs(t2) < tolerance)
    t2 = 0.0;
if(TP->Ts > 0)
{
    t2 = ceil(t2 / TP->Ts) * TP->Ts;
    *dd = TP->p / (8.0 * pow(t1, 4.0) + 16.0 * pow(t1, 3.0) * t2 + 10.0 * pow(t1, 2.0) * pow(t2, 2.0) + 2.0 * t1 * pow(t2, 3.0));
}
```

- 引入最大速度限制。最大速度在t_7时刻达到，即由t_j所确定的最大速度值为:

$$v_j = d_{max} t_d t_j^2 + 3d_{max} t_d^2 t_j + 2d_{max} t_d^3 \tag{17}$$

比较给定的最大速度v_{max}与计算得到的速度v_j，若$v_{max} > v_j$，则t_j满足最大速度限制要求，否则，按最大速度重新计算t_j，存在以下关系式:

$$d_{max} t_d t_j^2 + 3d_{max} t_d^2 t_j + 2d_{max} t_d^3 - v_{max} = 0 \tag{18}$$

求解以上一元二次方程，取正实根有：

$$t_j = -3t_d/2 + \sqrt{t_d^2/4 + v_{max}/(d_{max}t_d)} \tag{19}$$

```
if(TP->v < (2.0 * (*dd) * pow(t1, 3.0) + 3.0 * (*dd) * pow(t1, 2.0) * t2 + (*dd) * t1 * pow(t2, 2.0)))
{
    t2 = pow((pow(t1, 2.0) / 4.0 + TP->v / d / t1), (1.0 / 2.0)) - 3.0 / 2.0 * t1;
    if(fabs(t2) < tolerance)
        t2 = 0.0;
    if(TP->Ts > 0)
    {
        t2 = ceil(t2 / TP->Ts) * TP->Ts;
        *dd = TP->v / (2.0 * pow(t1, 3.0) + 3.0 * pow(t1, 2.0) * t2 + t1 * pow(t2, 2.0));
    }
}
```

- 引入最大加速度限制。最大速度在t_3时刻达到，即由t_j所确定的最大加速度值为：

$$a_j = d_{max}t_dt_j + d_{max}t_d^2 \tag{20}$$

比较给定的最大加速度a_{max}与计算得到的加速度a_j，若$a_{max} > a_j$，则t_j满足最大加速度限制要求。否则，按最大加速度重新计算t_j：

$$t_j = a_{max}/(d_{max}t_d) - t_d$$

```
if(TP->a < ((*dd) * pow(t1, 2.0) + (*dd) * t1 * t2))
{
    t2 = TP->a / (d * t1) - t1;
    if(fabs(t2) < tolerance)
        t2 = 0.0;
    if(TP->Ts > 0)
    {
        t2 = ceil(t2 / TP->Ts) * TP->Ts;
        *dd = TP->a / (pow(t1, 2.0) + t1 * t2);
    }
}
```

t_j计算完毕，在后续的计算中，t_j就可以作为一个常数存在。

确定最大加速度段时间的最优值

- 首先不考虑最大速度限制。在满足运动距离的前提下建立运动距离和最大加速度之间的关系。由上文可知$t_a = 0$，则$t_3 = t_4$，且t_3与t_4之间$d = 0, j = 0$。则t_{15}时刻的计算公式为：

$$s(t_{15}) = d_{max}t_dt_jt_a^2 + d_{max}t_d^2t_a^2 + 3d_{max}t_dt_j^2t_a + 9d_{max}t_d^2t_jt_a + 6d_{max}t_d^3t_a + 2d_{max}t_dt_j^3 + 10d_{max}t_d^2t_j^2 + 16d_{max}t_d^3t_j + 8d_{max}t_d^4 \tag{22}$$

为保证运动距离，令$s = s(t_{15})$，则：

$$(d_{max}t_dt_j + d_{max}t_d^2)t_a^2 + (3d_{max}t_dt_j^2 + 9d_{max}t_d^2t_j + 6d_{max}t_d^3)t_a + \\ (2d_{max}t_dt_j^3 + 10d_{max}t_d^2t_j^2 + 16d_{max}t_d^3t_j + 8d_{max}t_d^4 - s) = 0 \tag{23}$$

求取上式的正实根，即为由运动距离所确定的最大加速度段的时间t_a值。

```
c1 = pow(t1, 2.0) + t1 * t2;
c2 = 6.0 * pow(t1, 3.0) + 9.0 * pow(t1, 2.0) * t2 + 3.0 * t1 * pow(t2, 2.0);
c3 = 8.0 * pow(t1, 4.0) + 16.0 * pow(t1, 3.0) * t2 + 10.0 * pow(t1, 2.0) * pow(t2, 2.0) + 2.0 * t1 * pow(t2, 3.0);
t3 = (-c2 + sqrt(pow(c2, 2.0) - 4.0 * c1 * (c3 - TP->p / d))) / (2.0 * c1);
if(fabs(t3) < tolerance)
    t3 = 0.0;
if(TP->Ts > 0)
{
    t3 = ceil(t3 / TP->Ts) * TP->Ts;
    *dd = TP->p / (c1 * pow(t3, 2.0) + c2 * t3 + c3);
}
```

- 引入最大速度限制。最大速度在t_7时刻达到，即由t_a所确定的最大速度值为：

$$v_a = d_{max}t_dt_jt_a + d_{max}t_d^2t_a + d_{max}t_dt_j^2 + 3d_{max}t_d^2t_j + 2d_{max}t_d^3 \tag{24}$$

比较给定的最大速度v_{max}与计算得到的速度v_a，若$v_{max} > v_a$，则t_a满足最大速度限制要求，否则，按照最大速度限制重新计算t_a：

$$t_a = (v_{max} - 2d_{max}t_d^3 - 3d_{max}t_d^2t_j - d_{max}t_dt_j^2)/(d_{max} - t_dt_j + d_{max}t_d^2)$$

可以得到t_a。

```
if(TP->v < (*dd) * (2.0 * pow(t1, 3.0) + 3.0 * pow(t1, 2.0) * t2 + t1 * pow(t2, 2.0) + pow(t1, 2.0) * t3 + t1 * t2 * t3))
{
    t3 = - (2.0 * pow(t1, 3.0) + 3.0 * pow(t1, 2.0) * t2 + t1 * pow(t2, 2.0) - TP->v / d) / (pow(t1, 2.0) + t1 * t2);
    if(fabs(t3) < tolerance)
        t3 = 0.0;   //for continuous time case
    if(TP->Ts > 0)
    {
        t3 = ceil(t3 / TP->Ts) * TP->Ts;//为23148148.
        *dd = TP->v / (2.0 * pow(t1, 3.0) + 3.0 * pow(t1, 2.0) * t2 + t1 * pow(t2, 2.0) + pow(t1, 2.0) * t3 + t1 * t2 * t3);
    }
}
```

确定最大速度段时间最优值

最大速度段由运动距离确定，可由下式计算:

$$t_v = [s - 2s(t_7)]/v_{max}$$

至此，完全确定了4阶轨迹规划中的4个时间段t_d、t_j、t_a、t_v。结合初始时刻即可确定出轨迹中的所有切换点。

```
t4 = (TP->p - d * (c1 * pow(t3, 2.0) + c2 * t3 + c3)) / TP->v;
if(fabs(t4) < tolerance)
    t4 = 0.0;
if(TP->Ts > 0)
{
    t4 = ceil(t4 / TP->Ts) * TP->Ts;
    *dd = TP->p / (c1 * pow(t3, 2.0) + c2 * t3 + c3 + t4 * (2.0 * pow(t1, 3.0) + 3.0 * pow(t1, 2.0) * t2 + t1 * pow(t2, 2.0) + pow(t1, 2.0) * t3 + t1 * t2 * t3));
}
```

精度补偿

以上的处理都是连续计算，但是在计算机系统中运行就必须得离散化，所有切换点时间全部圆整为系统采样周期的整数倍，以保证轨迹规划的精度。为补偿精度损失，可用以下方法：

- 先按给定的算法确定轨迹规划的4个时间段t_d、t_j、t_a、t_v;
- 采用向上取整的圆整规则，将最大加加速度斜率段时间t_d圆整并转化为采样周期t_s的整数倍$t_{d,round}$，显然$t_{d,round} \geq t_d$，为保证运动距离，将圆整后的时间$t_{d,round}$重新带入轨迹规划算法进行计算，确定出调整后的最大加加速度斜率限制值$d_{max,round}$,显然，$d_{max,round} \leq d_{max}$，满足其物理极限限制。
- 根据重新计算时所得到的各个时间段值，采用类似方法并以t_j、t_a、t_v顺序分别将他们圆整为$t_{j,round}$、$t_{a,round}$、$t_{v,round}$并依次确定$d_{max,round}$，这样每次计算得到的$t_{d,round}$、$d_{max,round}$分别呈下降趋势，保证了调整后的对应物理量都满足给定的物理限制。

```
if(TP->Ts > 0)
{
    x = ceil(log10(*dd));
    ddq = *dd / pow(10.0, x);
    ddq = (ddq * pow(10.0, TP->s)) / pow(10.0, TP->s);
    ddq = ddq * pow(10.0, x);
    pp = ddq * (c1 * pow(t3, 2.0) + c2 * t3 + c3 + t4 * (2.0 * pow(t1, 3.0) + 3.0 * pow(t1, 2.0) * t2 + t1 * pow(t2, 2.0) + pow(t1, 2.0) * t3 + t1 * t2 * t3));
    dif = TP->p - pp;
    cnt =(dif / TP->r);
    tt = 8.0 * t1 + 4.0 * t2 + 2.0 * t3 + t4;
    ti =      (tt / TP->Ts);
    cor1 = SIGN(cnt) * floor(abs(cnt / ti)) * ti;
    cor2 = cnt - cor1;
    *dd = ddq;
}
else
{
    cor1 = 0.0;
    cor2 = 0.0;
}
```