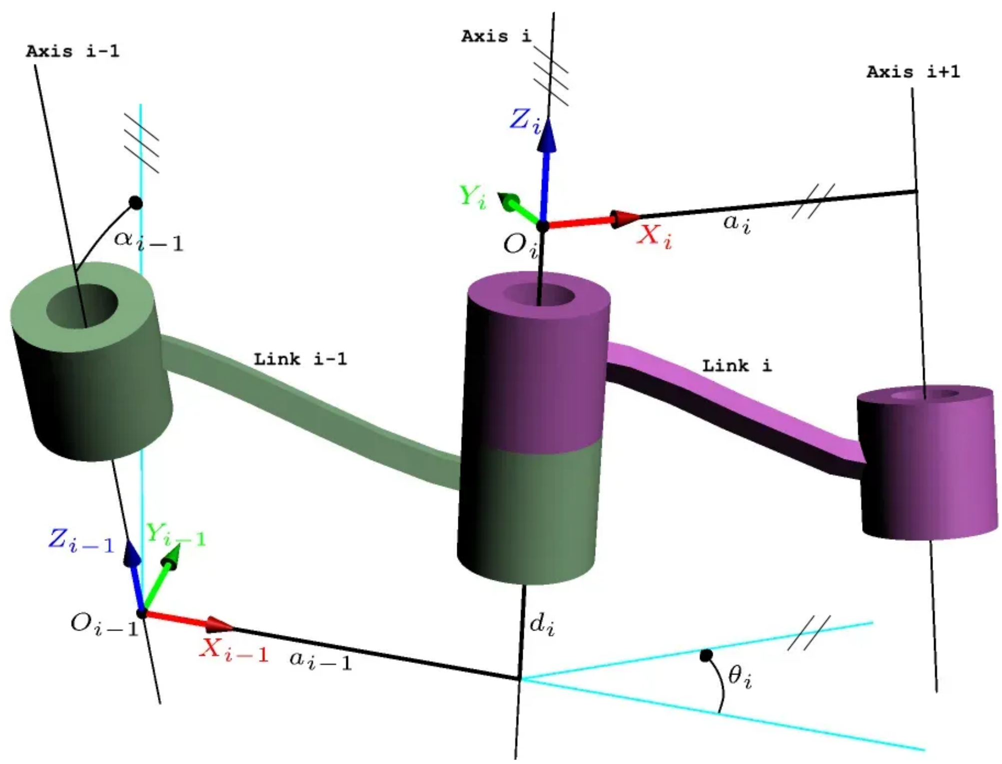
机器人机械臂控制系统

1. 控制系统组成模块

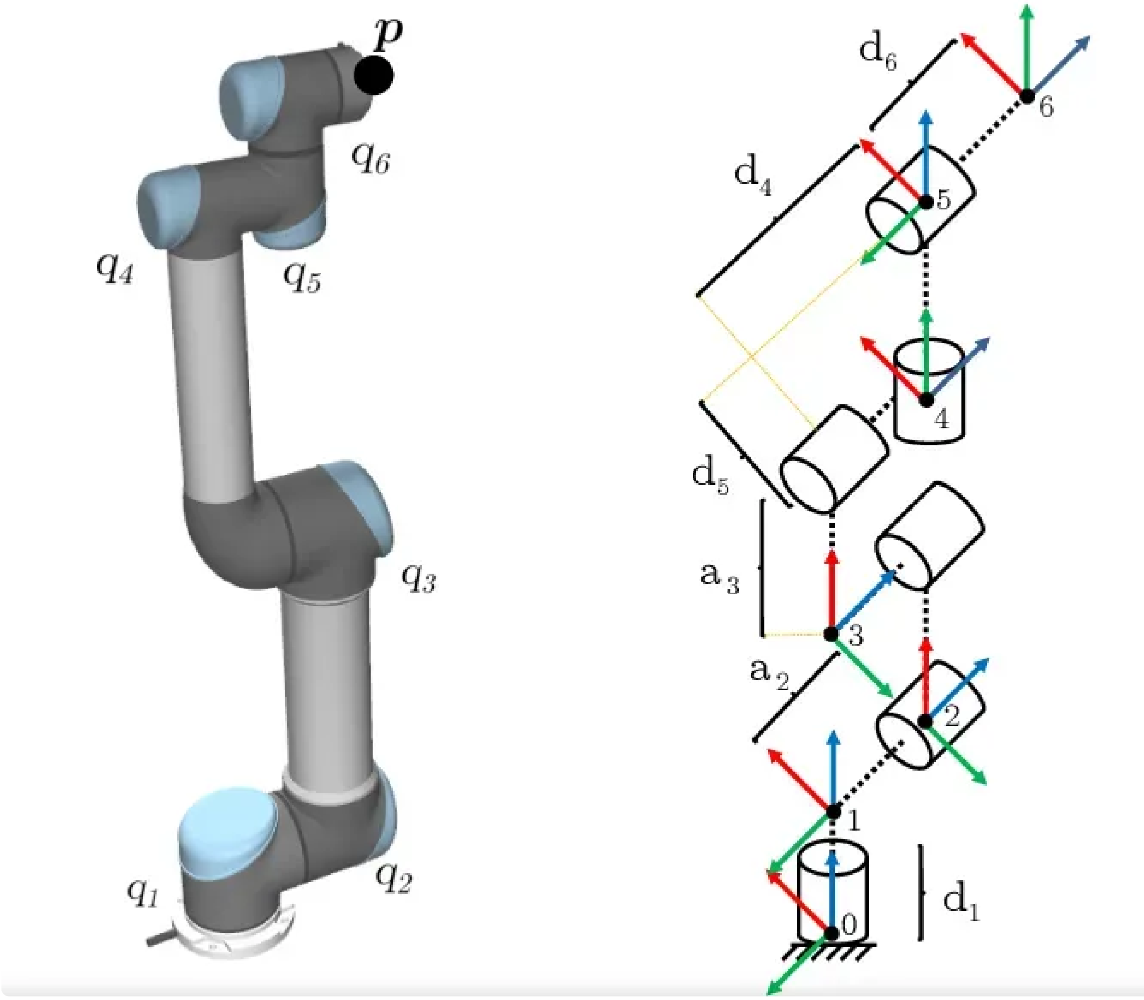
1. 正逆运动学：3平行轴构型逆解、末端3轴交于一点构型逆解
2. 动力学：牛顿欧拉方程、pinocchio库
3. 机械臂力控：导纳控制、力位混合控制
4. 机械臂避障规划：RRT*算法和FCL碰撞检测库
5. 特定轨迹插值：多项式插值

2. 运动学

2.1. 改进的DH参数法描述机械臂构型

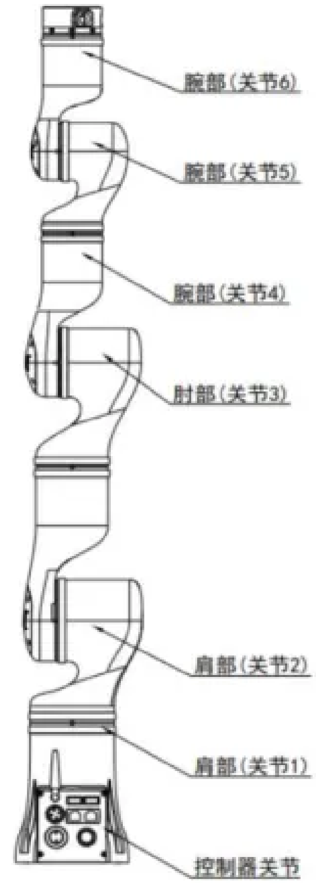


2.2. 具有三个平行轴的机械臂的运动学计算



如上图所示UR机械臂，由于关节2、3、4平行，连杆4坐标系的x、y、z轴在连杆1坐标系的y轴的投影分别为0、0、-1，并且连杆4坐标系的原点相对于连杆1的原点沿其y轴平移了-d2-d3-d4的距离。体现在矩阵中就是矩阵的第二行正好为0、0、1、-d2-d3-d4。利用第二行几个常数项，可以求解关节1、5、6的角度。

2.3. 末端三轴交于一点的机械臂的运动学计算



如上图所示睿尔曼RM65机械臂，由于关节4、5、6交于一点，这3个轴的转动不会影响交点的位置，所以我们可以通过给定的目标位置确定该交点的位置，然后通过三角函数计算出前三个轴的角度。根据前三个轴计算得到的姿态可以计算得到达到目标姿态还需要末端三个轴需要达到的姿态。末端三个轴转动形成的姿态可以用zyz欧拉角矩阵表示，然后通过代数运算得到后三个轴的转动角度。

3. 动力学

3.1. 牛顿欧拉方程动力学推导流程：

1. 已知机械臂各关节的角速度 $\dot{\theta}_i$ 和角加速度 $\ddot{\theta}_i$ ，从机械臂基座到机械臂末端正向推导各个连杆在自身坐标系下的角速度 ω_i 和角加速度 α_i （注意区分关节角速度和连杆角速度），其中 α_i 表示连杆 $i-1$ 坐标系在连杆 i 坐标系的旋转矩阵:

$$\omega_i = R_{i-1}^{i}\omega_{i-1} + \dot{\theta}_i\vec{Z}$$

$$\alpha_i = R_{i-1}^{i}\alpha_{i-1} + R_{i-1}^{i}\omega_{i-1} \times \dot{\theta}_i\vec{Z} + \ddot{\theta}_i\vec{Z}$$

2. 计算各连杆坐标系原点在自身坐标系下的线速度 v_i 和线加速度 a_i ，其中 p_i^{i-1} 表示连杆 i 坐标

系原点在连杆 $i-1$ 坐标系的位置坐标:

$$v_i = R_{i-1}^i v_{i-1} + R_{i-1}^i \omega_{i-1} \times p_i^{i-1}$$

$$a_i = R_{i-1}^i \left[a_{i-1} + \alpha_{i-1} \times p_i^{i-1} + \omega_{i-1} \times \left(\omega_{i-1} \times p_i^{i-1}\right)\right]$$

3. 计算各连杆在质心处的线速度 v_{ci} 和线加速度 a_{ci} ，其中连杆 i 质心在连杆 i 坐标系下的位置为 r_{ci} ：

$$v_{ci} = v_i + \omega \times r_{ci}$$

$$a_{ci} = a_i + \alpha_i \times r_{ci} + \omega_i \times (\omega_i \times r_{ci})$$

4. 计算各连杆所受合力 F_i 及合力矩 N_i ，其中连杆质量为 m_i 连杆 i 坐标系下的转动惯量为 I_i （不是质心处的转动惯量）：

$$F_i = m_i a_{ci} = m_i a_i + \alpha_i \times m_i r_{ci} + m_i \omega_i \times (\omega_i \times r_{ci})$$

$$N_i = I_i \alpha_i + \omega_i \times I_i \omega_i$$

5. 从机械臂末端向机械臂基座反向推导各个关节的力矩，连杆 i 在其自身坐标系中所受合力为 F_i ，所受合力矩为 N_i ， f_i 表示连杆 $i-1$ 施加给连杆 i 的力， n_i 表示连杆 $i-1$ 施加给连杆 i 的力矩:

$$f_i = F_i + R_{i+1}^i f_{i+1}$$

$$n_i = N_i + R_{i-1}^i n_{i+1} + p_{i+1}^i \times R_{i+1}^i f_{i+1}$$

6. 最终可以得到，连杆所受力和力矩的计算方程:

$$f_i = m_i a_i + \alpha_i \times m_i r_{ci} + m_i \omega_i \times (\omega_i \times r_{ci}) + R_{i+1}^i f_{i+1}$$

$$n_i = I_i \alpha_i + \omega_i \times I_i \omega_i + R_{i+1}^i n_{i+1} + p_{i+1}^i \times R_{i+1}^i f_{i+1}$$

7. 机器人末端所受外力可以表示为 f_7 和 n_7 （在连杆6坐标系中表示，计算 f_6 和 n_5 时 R_7^6 为单位矩阵）：

$$f_7 = [x_1, y_1, z_1]^T, n_7 = [x_2, y_2, z_2]^T$$

3.2. 动力学库pinocchio

1. pinocchio实现的主要算法：

   正向运动学（forward kinematics）：给定机器人构型，计算每个关节的空间位置并将其存储为坐标变换。如果给定速度或加速度，将计算每个关节的空间速度（以局部坐标系表示）。

   运动学雅可比矩阵（kinematic jacobian）：在机械臂运动学中用来计算机械臂末端执行器的速度与各个关节运动速度之间的关系。

逆动力学（inverse dynamics）：采用Recursive Newton-Euler Algorithm (RNEA) 计算逆动力学。即给定所需的机器人构型、速度、加速度，计算并存储执行该运动所需的扭矩。

关节空间惯性矩阵（Joint space inertia matrix）：采用Compostie rigid body algortihm (CRBA)算法，计算关节空间惯性矩阵。

前向动力学（forward dynamics）：采用Articulated Body Algorithm（ABA）计算无约束的前向动力学。即给定机器人构型、速度、扭矩和外力的情况下，可以计算出由此产生的关节加速度。

4. 机械臂力控

4.1. 导纳控制（Admittance control）

4.1.1. 基本原理：

导纳模型：导纳控制的核心是把机械臂抽象成一个质量-阻尼-弹簧系统，通过调整三个参数控制机械臂的柔顺性，用方程表示为：

$$F = M\ddot{x} + D\dot{x} + Kx$$

其中， F 是作用力， M 、 D 和 K 分别表示质量、阻尼和弹簧系数， x 是位移。

难点：在高频控制的情况下，每个控制周期机械臂的增量都很小，此时机械臂的响应误差和获取的位置误差与增量相比就会很大。

4.1.2. 实际机械臂控制步骤：

1. 根据对力控响应速度，设置机械臂的控制参数M、D、K的值（写成6X6对角矩阵的形式）；
2. 导纳控制通常需要末端六维力传感器来实时检测机械臂受到的外力 $F_{external}$ （写成6维向量的形式，向量前三个值表示合力的方向，后三个值表示绕基座标系x、y、z三个方向的力矩）；
3. 获取机械臂末端当前的位置 $P_{current}$ 、当前速度 $V_{current}$ （写成6维向量的形式，向量前三个数为位置x、y、z，后三个值为轴角表示的旋转轴乘以角度得到的三个值）；
4.
5. 根据导纳模型计算出的加速度 $A_{current}$ ，向机械臂输出目标位置 $P_{goal} = P_{current} + A_{current}$ ，使其响应外界作用力（**注意其中的 $P_{current}$ 最好取上一次的 P_{goal} 值，因为机械臂反馈的当前位置一般会有延迟**）；

4.2. 力位混合控制

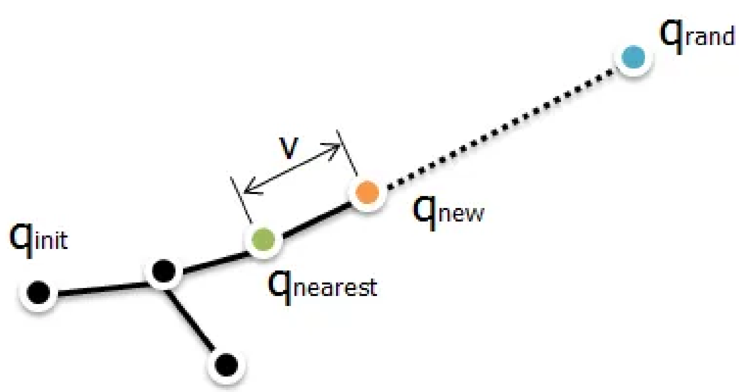
4.2.1. 基本原理：

1. 力位混合控制首先要把进行力控的方向从机械臂的运动方向解耦出来，位置控制只控制解耦出来的运动控制部分，力控则只针对预定的力控方向，最后向机械臂输出合成的运动控制目标。

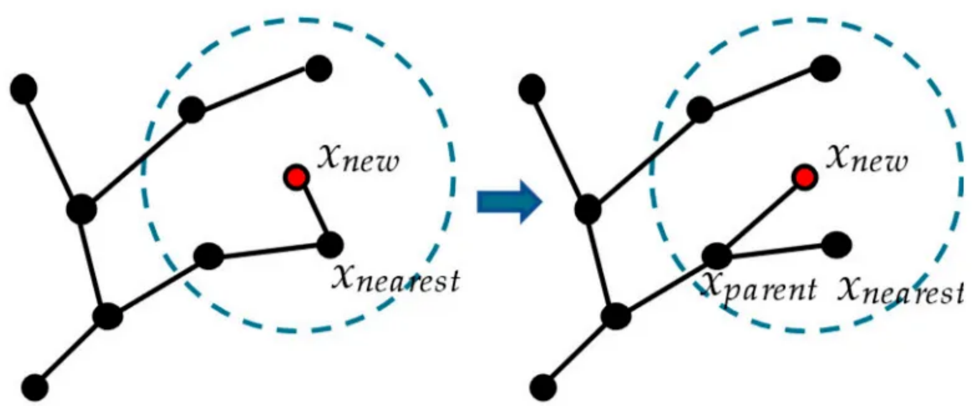
5. 避障规划

5.1. RRT及其优化算法

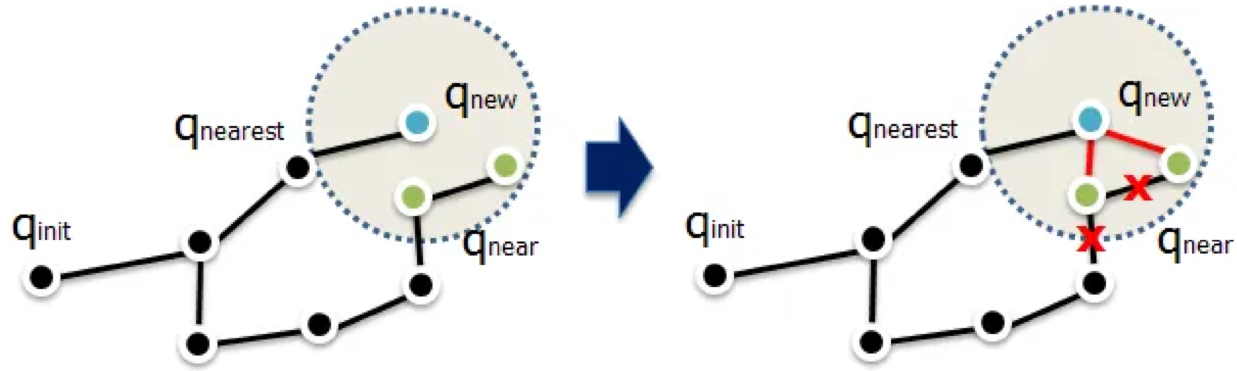
1. RRT原始算法



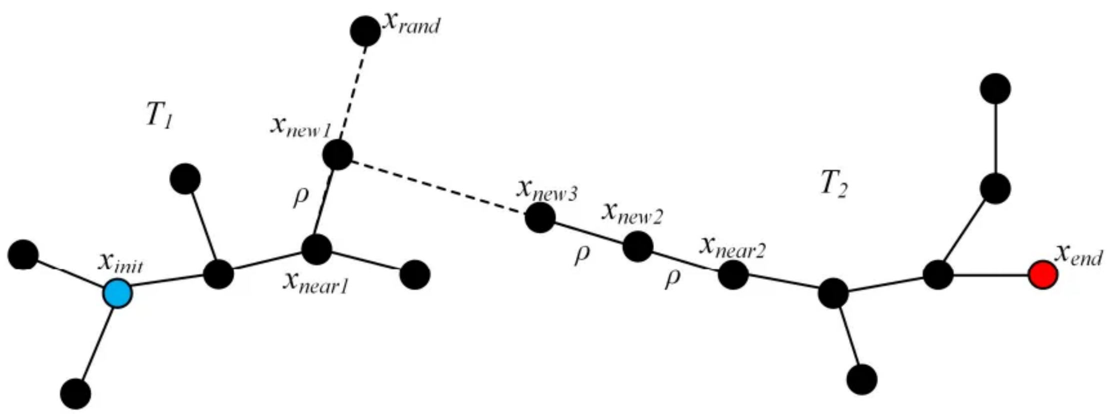
2. RRT*算法
    a. 重选父节点过程：



    b. 重布线过程：



3. RRT connect算法



5.2. FCL碰撞检测库

1. FCL 是一个对由三角形面（mesh文件表示的模型）组成的一对几何模型执行三种类型的邻近查询的库，它有以下功能：

- 碰撞检测：检测两个模型是否重叠，以及检测所有重叠的三角形。
- 距离计算：计算一对模型之间的最小距离，即最近一对点之间的距离。
- 公差验证：确定两个模型是否比公差距离更近或更远。
- 连续碰撞检测：检测两个运动模型在移动过程中是否重叠，以及接触的时间。
- 接触信息：对于碰撞检测和连续碰撞检测，可以返回接触信息（包括接触法线和接触点）。

6. 关节轨迹多项式插值

6.1. 三次多项式插值推导

6.1.1. 三次多项式插值

1. 已知空间中序列点： $(x_0, y_0), (x_1, y_1) \ldots (x_i, y_i) \ldots$ ，两序列点间插值曲线的三次多项式可

以表示为 $S_0(x), S_1(x)...S_i(x)...$ ，令 $h_i = x_{i+1} - x_i$ ， x_i 为第 i 段曲线的起始横坐标，则有：

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$
$$S_i(x)' = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2$$
$$S_i(x)'' = 2c_i + 6d_i(x - x_i)$$

2. 为了保证各段插值曲线的位置连续性、速度连续性和加速度连续性，每段曲线有以下4个约束：
   a. 根据曲线首尾位置固定，可以得到两个方程：
   $$S_i(x_i) = a_i + b_i(x_i - x_i) + c_i(x_i - x_i)^2 + d_i(x_i - x_i)^3 = y_i \quad \Rightarrow \quad a_i = y_i$$
   $$S_i(x_{i+1}) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_{i+1}$$
   b. 根据相邻曲线的速度连续性，可以得到方程：
   $$S_i(x_{i+1})' = S_{i+1}(x_{i+1})' \quad \Rightarrow \quad b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1}$$
   c. 根据相邻曲线的加速度连续性，可以得到方程：
   $$S_i(x_{i+1})'' = S_{i+1}(x_{i+1})'' \quad \Rightarrow \quad 2c_i + 6d_i h_i = 2c_{i+1}$$
3. 用 m_i 表示各个序列点处的加速度 $S_i(x_i)''$ 也即 $2c_i$ ，则上面的加速度连续方程可以改写为：
$$d_i = \frac{m_{i+1} - m_i}{6h_i}$$
4. 现在 a_i 、 c_i 和 d_i 都可以用 y 、 m 和 h 表示，将其带入位置方程可以得到：
$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{2} m_i - \frac{h_i}{6}(m_{i+1} - m_i)$$
5. 将 a_i 、 b_i 、 c_i 和 d_i 带入速度连续方程可以得到只包含加速度 m 、时间 h 和位置 y 的方程：
$$h_i m_i + 2(h_i + h_{i+1}) m_{i+1} + h_{i+1} m_{i+2} = 6\left[\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}\right]$$

6.1.2. 不带控制点的三次多项式插值

1. 从总体参数和约束上来看， n 个点有 $n-1$ 段三次多项式路径，所以有 $4(n-1)$ 个未知参数，而已知的约束有：每段路径两边都有位置约束共 $2(n-1)$ 个，每段路径间的速度和加速度连续有 $2(n-2)$ 个约束。如上所述，约束比未知参数还少2个，可以固定首尾点的速度或者加速度。

2. 对于 n 个序列点，共有 n 个未知加速度，根据上面的结果可以得到 $n-2$ 个方程，将各个序列点的方程列成矩阵形式可以得到：

$$\begin{bmatrix} 1 & 0 & 0 & & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & & \\ 0 & h_1 & 2(h_1+h_2) & h_2 & 0 & \\ 0 & 0 & h_2 & 2(h_2+h_3) & h_3 & \vdots \\ \vdots & & 0 & \ddots & \ddots & \ddots \\ 0 & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \cdots & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_n \end{bmatrix}$$

$$= 6 \begin{bmatrix} 0 \\ \frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0} \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3} - \frac{y_3-y_2}{h_2} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_{n-1}} - \frac{y_{n-1}-y_{n-2}}{h_{n-2}} \\ 0 \end{bmatrix}$$

3. 上面的矩阵把首尾点的加速度设置为了0，这样未知加速度就变为 $n-2$ 个，就可以求出各个点处的加速度值。根据加速度值又可以求出每段三次多项式的参数。

6.1.3. 带控制点的三次多项式插值

1. 上面的不带控制点的方法，给定了首尾点的加速度，而首尾点的速度是根据公式求出来的，不是给定的。在许多情况下我们既需要给定首尾的位置和加速度，也需要给定首尾点的速度。为了解决这个问题，可以在首点后和尾点前插入一个控制点。
2. 我们不给定该控制点的位置，这样就少一个约束。参数不变少一个约束，就需要加一个约束。我们就可以通过给定首点的速度来增加约束，这样首点的位置、速度和加速度就都可以给定了。
3. 为了方便表示，我们假设原来有 $n-2$ 个序列点，现在加了两个控制点就变为了 n 个控制点。我们需要根据给定的速度求出控制点的位置 y_1 和 y_{n-1} 。
4. 已知首点的位置、速度和加速度，首段路径的参数 a_0 ， b_0 ， c_0 都可以求出来，由上面的位置方程可以得到:

$$a_0 + b_0h_0 + c_0h_0^2 + d_0h_0^3 = y_1$$

再根据上面的式子可以得到用 m_0 和 m_1 表示的 y_1 ，取首点的速度和加速度都为0可以得到：

$$y_0+\frac{(m_1-m_0)}{6}h_0^2=y_1$$

将上式左边部分表示的 y_1 带入矩阵第二行的等式，可以得到：

$$(3h_0+2h_1+\frac{h_0^2}{h_1})m_1+h_1^2m_2=6(y_2-y_0)$$

带入矩阵的第三行可以得到：

$$(3h_0+2h_1+\frac{h_0^2}{h_1})m_1+h_1^2m_2=6(y_2-y_0)$$

5. 在首尾点的速度和加速度都为0的情况下可以得到以下矩阵方程：

$$\begin{bmatrix} 2(h_0+h_1) & h_1 & & 0 & \cdots & 0 \\ h_1 & 2(h_1+h_2) & & h_2 & & 0 \\ 0 & h_2 & & 2(h_2+h_3) & & h_3 \\ \vdots & \vdots & \ddots & & \ddots & \vdots \\ 0 & 0 & \cdots & h_{n-2} & & 2(h_{n-2}+h_{n-1}) \end{bmatrix}\begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_n \end{bmatrix}$$

$$=6\begin{bmatrix} \frac{y_2-y_0}{h_1} \\ \frac{y_3-y_2}{h_2}-\frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3}-\frac{y_3-y_2}{h_2} \\ \vdots \\ \frac{y_n-y_{n-1}}{h_{n-1}}-\frac{y_{n-1}-y_{n-2}}{h_{n-2}} \end{bmatrix}$$

6.2. 两点间五次多项式插值推导

1. 五次多项式总共有6个系数，两个点分别对位置、速度和加速度进行约束，总共有6个约束，正好满足需要。
2. 五次多项式可以表示为：

$$s(t)=a_0+a_1(t-t_0)+a_2(t-t_0)^2+a_3(t-t_0)^3+a_4(t-t_0)^4+a_5(t-t_0)^5$$

3. 已知的约束条件可以表示为：

$$\begin{cases} s(t_0)=s_0, s(t_1)=s_1 \\ \dot{s}(t_0)=v_0, \dot{s}(t_1)=v_1 \\ \ddot{s}(t_0)=a_0, \ddot{s}(t_1)=a_1 \end{cases}$$

4. 令 $T=t_1-t_0$, $h=s_1-s_0$ 计算可以得到五次多项式系数：

$$\begin{cases} a_0 = s_0 \\ a_1 = v_0 \\ a_2 = \frac{1}{2}a_0 \\ a_3 = \frac{1}{2T^3}\left[20h - (8v_1 + 12v_0)T - (3a_0 - a_1)T^2\right] \\ a_4 = \frac{1}{2T^4}\left[-30h + (14v_1 + 16v_0)T + (3a_0 - 2a_1)T^2\right] \\ a_5 = \frac{1}{2T^5}\left[12h - 6(v_1 + v_0)T + (a_1 - a_0)T^2\right] \end{cases}$$

6.3. 三点间四次多项式插值

1. 四次多项式总共有5个系数，三个点中的首尾点对位置、速度和加速度进行约束，中间点对加速度和速度保证连续且位置进行约束，总共有10个约束(中间点的位置约束对两段曲线都有所以有两个约束)，对两段曲线分别有5个约束，正好满足需要。
2. 五次多项式可以表示为：

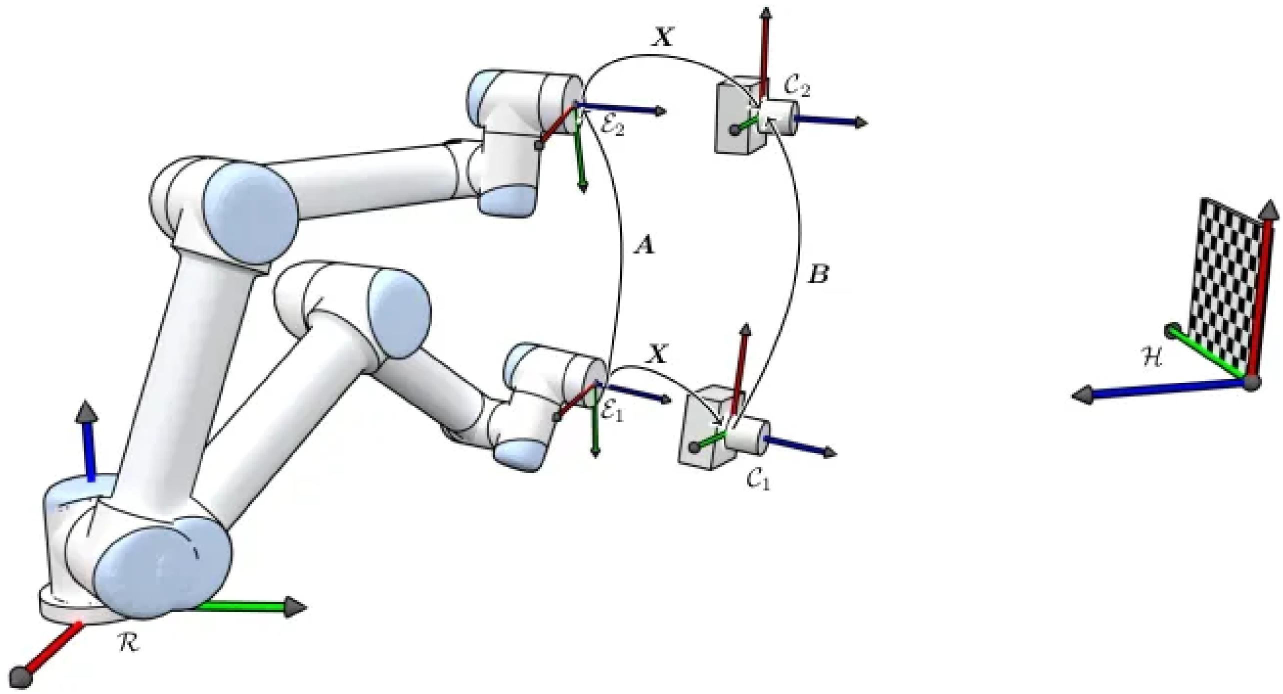
$$s(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3 + a_4(t - t_0)^4$$

3. 已知的约束条件可以表示为：

$$\begin{cases} s(t_0) = s_0, s(t_1) = s_1 \\ \dot{s}(t_0) = v_0, \dot{s}(t_1) = v_1 \\ \ddot{s}(t_0) = a_0, \ddot{s}(t_1) = a_1 \end{cases}$$

7. 手眼标定

7.1. 眼在手标定



1.由位姿关系可以得到: Tend_to_base1 * Tcam_to_hand * Tobj_to_cam1 = Tend_to_base2 * Tcam_to_hand * Tobj_to_cam2

2.由上式可以得到: (Tend_to_base2的逆) * Tend_to_base1 * Tcam_to_hand = Tcam_to_hand * Tobj_to_cam2 * (Tobj_to_cam1的逆)

3.令(Tend_to_base2的逆) * Tend_to_base1 = Ta, Tobj_to_cam2 * (Tobj_to_cam1的逆) = Tb, Tcam_to_hand = Tx, 可以得到方程:TaTx=TxTb

4.该方程可以把位置部分提取出来，另变换矩阵X、A、B的位置向量分别为Px、Pa、Pb，旋转矩阵分别为X、A、B，可以得到:A*Px+Pa = X*Pb+Px

5.可以得到X的位置部分为: (A−I)*Px = (X*Pb−Pa) ==> Px = ((A−I)的逆)*(X*Pb−Pa)

解算旋转矩阵部分的AX=XB方程步骤:

1.方程两边同乘旋转矩阵B的旋转向量Vb得到: AX*Vb=λ*X*Vb

2.由于X*Vb也是一个向量，并且旋转矩阵A也存在旋转轴，假设A的旋转向量为Va，可以得到: X*Vb=λ*Va

3.只要求出三个相互独立的Vb就可以得到: X*[Vb1、Vb2、Vb3] = λ*[Va1、Va2、Va3] ==> X = λ*[Va1、Va2、Va3]*（[Vb1、Vb2、Vb3]的逆）