

生成规则:

$$G = (\overset{\text{non-terminal}}{\uparrow} N, T, A, R, \underset{\text{terminal}}{\downarrow} S)$$

SRL基座:

$$r_1 \textcircled{S} \Rightarrow \textcircled{B} \rightarrow \textcircled{P}$$

延长:

$$r_2 -\textcircled{B} \Rightarrow -\textcircled{B} \rightarrow \textcircled{C} \rightarrow \textcircled{L}$$

$$r_3 -\textcircled{L} \Rightarrow -\textcircled{L} \rightarrow \textcircled{C} \rightarrow \textcircled{L}$$

收尾:

$$r_4 -\textcircled{L} \Rightarrow -\textcircled{L} \rightarrow \textcircled{E}$$

\textcircled{S} start symbol

\textcircled{B} base

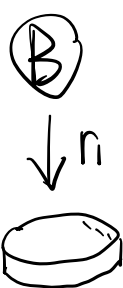
\textcircled{P} pinned

\textcircled{L} link

\textcircled{C} connector

\textcircled{E} end part $\in T$

替换规则(组件库)



Params: 尺寸
(直径, 厚度)

Fixed-P: P



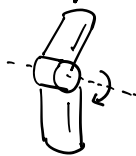
Params: /

F-P: M



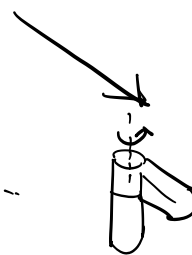
P: 长度
直径
旋转角度

F-P: P



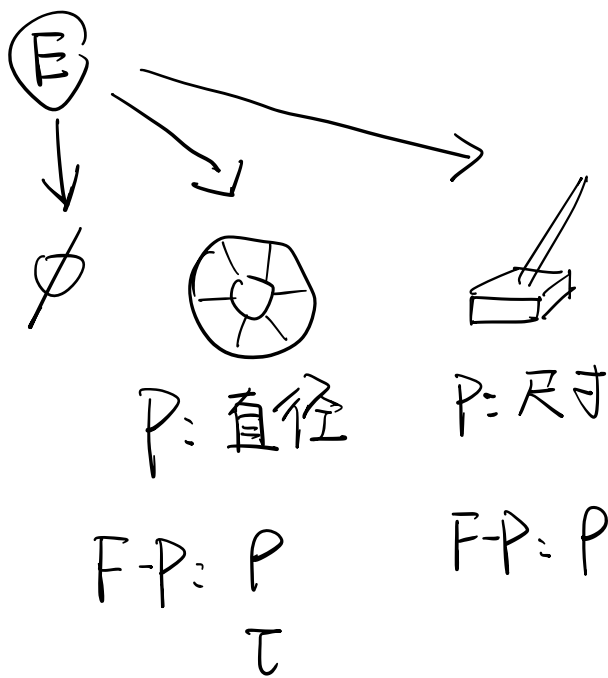
P: /

F-P: M, T



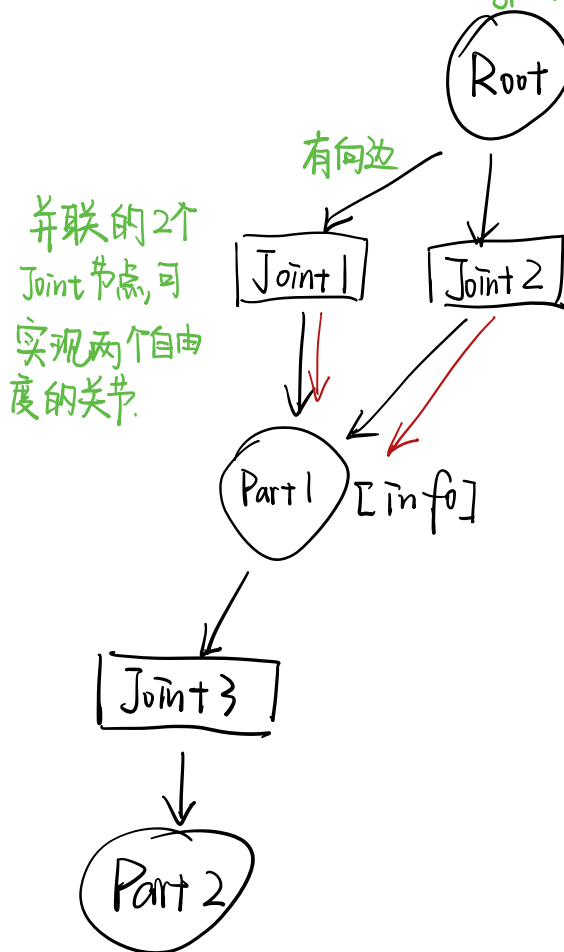
P: /

F-P: M, T



树结构表示机器人.

Link Type 节点

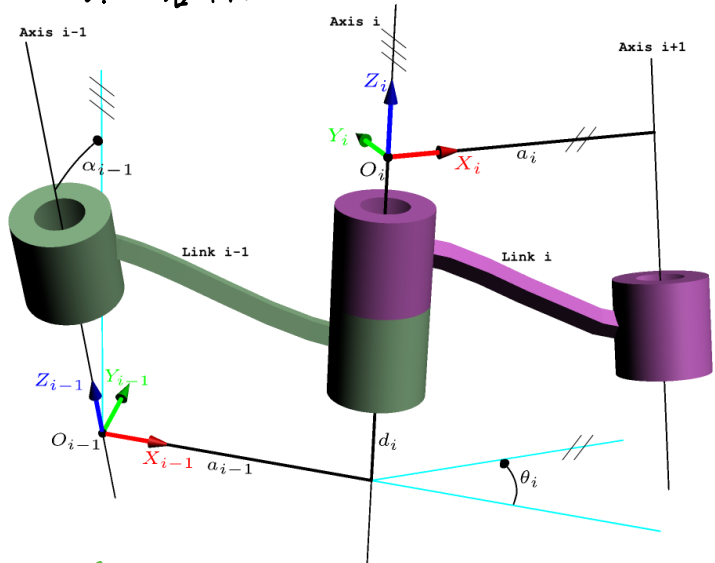


size
basic frame
length
....

一组 DH 参数表示一个自由度

Modified DH Method.

a_{i-1} : 沿 X_{i-1} 轴, 从 Z_{i-1} 移动到 Z_i 的距离
 α_{i-1} : 绕 X_{i-1} 轴, 从 Z_{i-1} 转动到 Z_i 的角度.
 d_i : 沿 Z_i 轴, 从 X_{i-1} 移动到 X_i 的距离
 θ_i : 沿 Z_i 轴, 从 X_{i-1} 转动到 X_i 的角度.



关节轴: Z_i

X_i 始终与 Z_i 垂直.

XML 格式:

`<body>1..... </>`

`<geom> Root info </>`

`<body>2... </>`

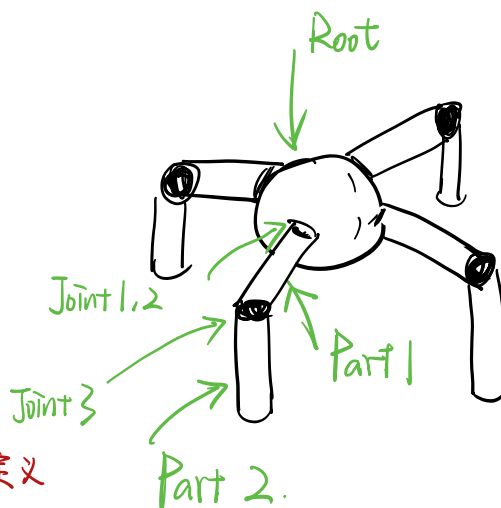
`<geom> Part 1 </>`

`<joint> Joint 1 </>`

`<joint> Joint 2 </>`

一个 Body 表示一个实体部分

这两个 Joint 定义了 body 1 到 body 2 之间的关节。

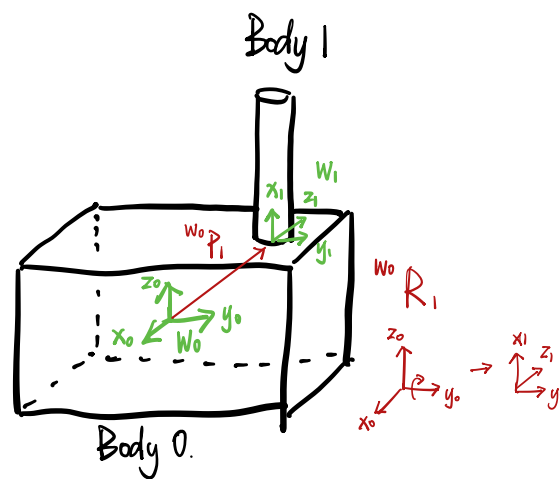


每一个 body 构成一个实际机器人事件。

body {
 geom 形状实体定义
 joint 与上一级 body 之间的关节。

构建机器人时, 要求:

- (1) (暂时) 只考虑正交直连杆 Linkage
- (2) 杆件方向永远朝向所属 Body 坐标系的 x 轴。
- (3) 关节永远在 Body 坐标系的原点上。
- (4) 一个 Joint 节点, 仅包含一个自由度, 如需多个自由度, 通过并联的多个 Joint 节点实现



对 D-H 表示法进一步简化:

`<body>` 定义中, 包含 Body 坐标系变换矩阵 (P, R)

`<geom>` 定义中, 从 Body 坐标系原点向 x 轴延伸

结构图中:

Robot Link 结点包括: Body 坐标系, 该 link 在 Body 坐标系中的位置

Robot Joint 结点包括: 关节轴。

Link Type 节点

生成模型时,

自 Root 向下遍历各个分支.

并联的2个
Joint 节点, 可
实现两个自由
度的关节.

遇到 Joint 节点, 标记,
暂不处理, 继续访问其子节点

遇到 Link 节点, 若标记有未添加的 Joint,
则遍历其父节点, 依次添加 Joint.

