

移动机器人开发技术（激光SLAM版）配套教学PPT

序 章

机器人操作系统

机器人硬件平台

机器人核心技术

机器人应用实战

第1课 移动机器人的过去、现在及未来

第2课 初识ROS

第3课 ROS编程初步

第4课 机器人的坐标变换

第5课 机器人仿真环境

第6课 TurtleBot3仿真环境实战

第7课 自主搭建机器人小车

感知

第08课 环境感知基础
第09课 感知数据融合

建图与定位

第10课 机器人的移动控制
第11课 SLAM基础
第12课 SLAM实战

路径规划与导航

第13课 导航基础
第14课 ROS中的导航包
第15课 ROS导航实战

送餐

- 1 送餐机器人结构设计
- 2 送餐机器人环境搭建
- 3 送餐机器人建图
- 4 送餐机器人导航

物流（专题讲座）

- 1 物流机器人结构设计
- 2 物流机器人环境模拟
- 3 物流机器人关键技术
- 4 大规模多机器人调度

图书盘点（专题讲座）

- 1 图书盘点机器人结构
- 2 图书盘点机器人环境
- 3 图书盘点机器人工作模式
- 4 图书盘点中的视觉分析

移动机器人开发技术（激光SLAM版）配套教学PPT

第四课 机器人的坐标变换



北京邮电大学

Beijing University of Posts and Telecommunications

移动机器人与智能技术实验室编

宋桂岭 明安龙 2021.10

expsong@qq.com

第4课 机器人坐标变换

北邮移动机器人与智能技术实验室 编

1

理解机器人体系结构

2

坐标变换 (TF) 初步

3

ROS中的坐标变换TF



第4课 机器人的坐标变换

北邮移动机器人与智能技术实验室 编

1 理解机器人体系结构

1.1 工业机器人组成

文献：[工业机器人的基本组成及技术参数](#)

3大部分6个子系统组成：

机械部分

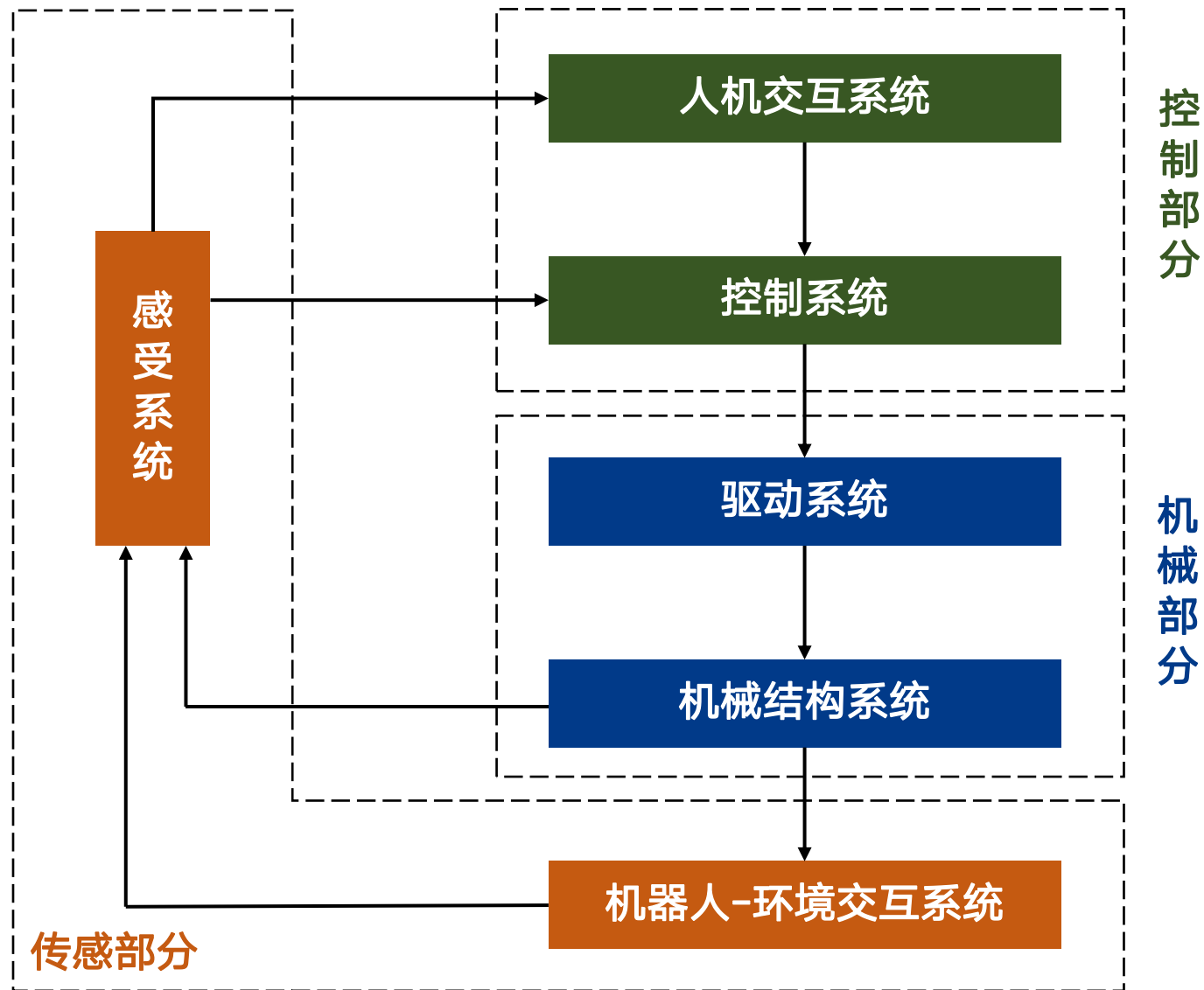
驱动系统、机械结构系统

传感部分

感受系统、机器人-环境交互系统

控制部分

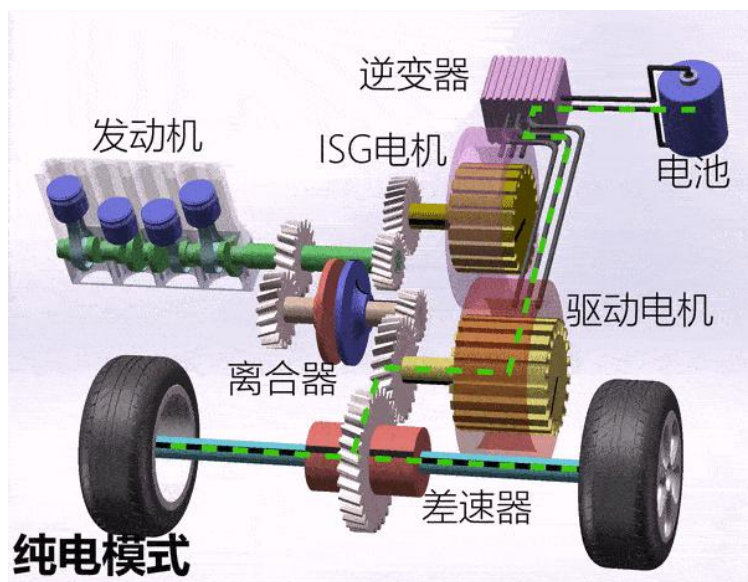
人机交互系统和控制系统



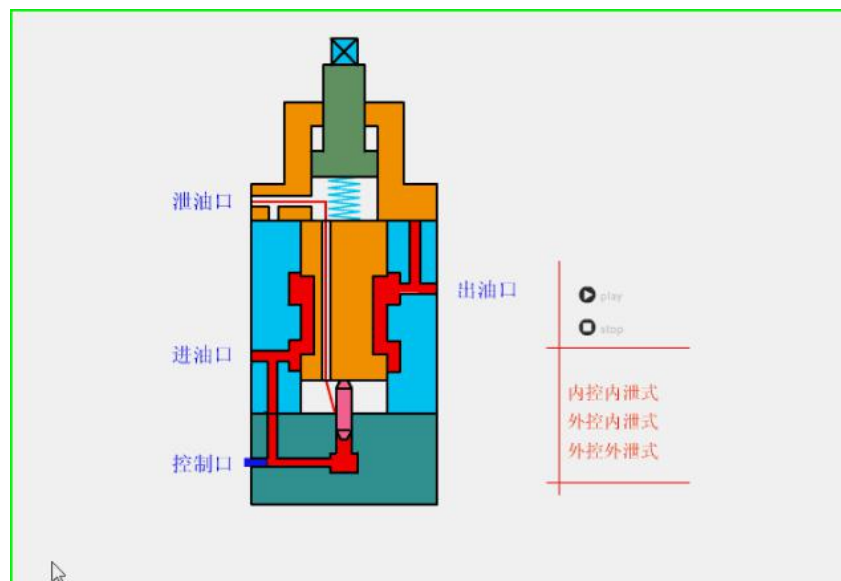
1.1 工业机器人组成

机械部分-驱动系统

要使机器人运行起来, 需给机器人各个关节即每个运动自由度安置传动装置, 这就是**驱动系统**。驱动系统可以是液压传动、气动传动、电动传动, 或者把它们结合起来应用的综合系统; 可以是直接驱动或者是通过同步带、链条、轮系、谐波齿轮等机械传动机构进行间接驱动



纯电模式
四轮电驱动全向移动机器人底盘



液压驱动原理

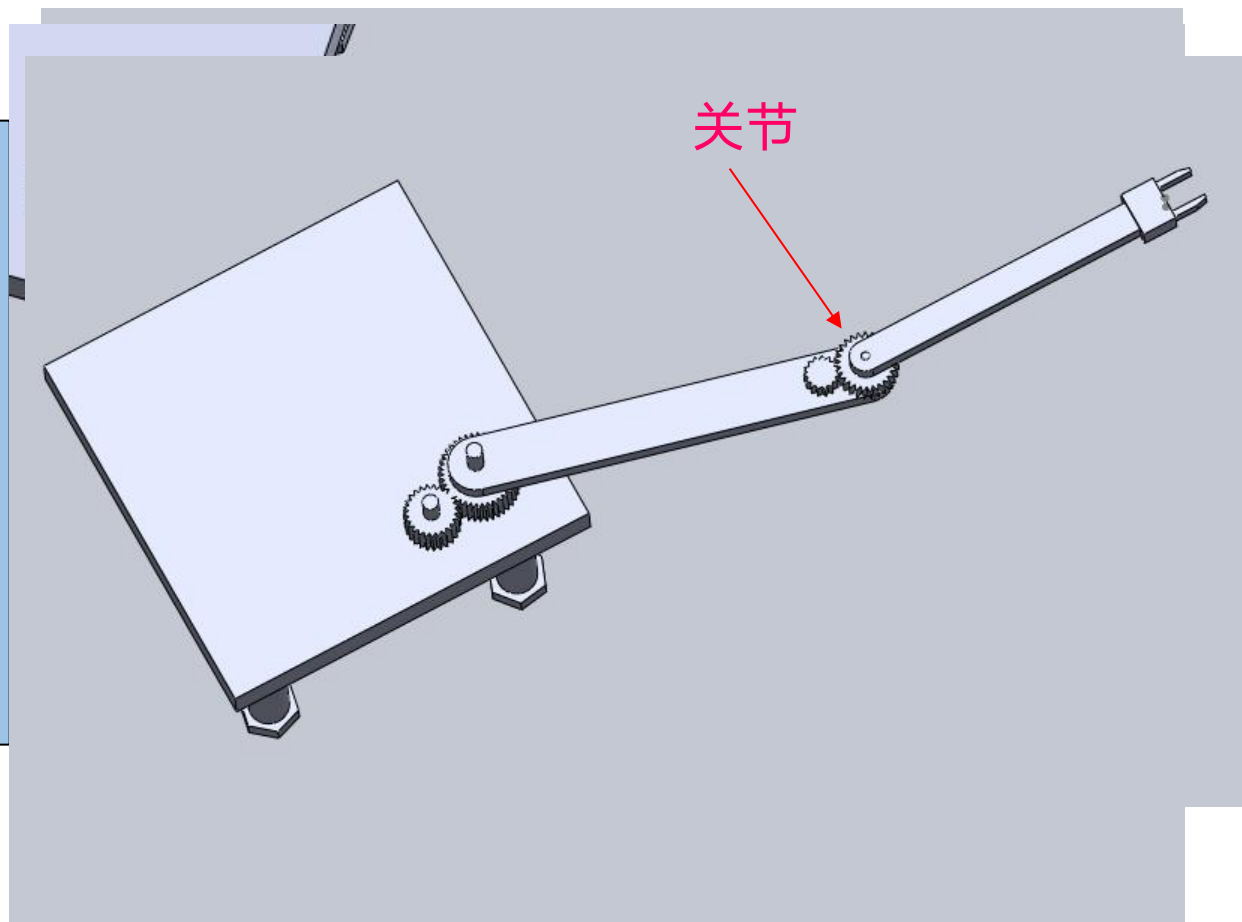


齿轮传动结构

1.1 工业机器人组成

机械部分-机械结构系统：由基座、手臂、末端操作器三大件组成

关节：分为滑动关节和转动关节。实现机身、手臂各部分、末端操作器之间的相对运动。

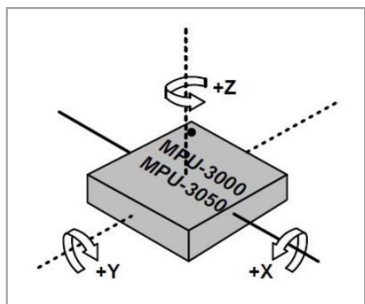


机器人每一大件都有若干自由度，构成一个多自由度的机械系统。若基座具备行走机构，则构成行走机器人；若基座不具备行走及腰转机构，则构成单机器人臂 (SingleRobotArm)。手臂一般由上臂、下臂和手腕组成。末端操作器是直接装在手腕上的一个重要部件，它可以是二手指或多手指的手爪，也可以是喷漆枪、焊具等作业工具。

1.1 工业机器人组成

传感部分-感受系统

- 1 感受系统由内部传感器模块和外部传感器模块组成, 用以获取内部和外部环境状态中有意义的信息
- 2 智能传感器的使用提高了机器人的机动性、适应性和智能化的水准
- 3 智能传感器的使用提高了机器人的机动性、适应性和智能化的水准
- 4 对于一些特殊的信息, 传感器比人类的感受系统更有效



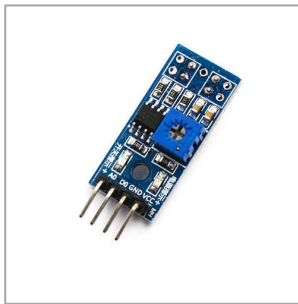
陀螺仪



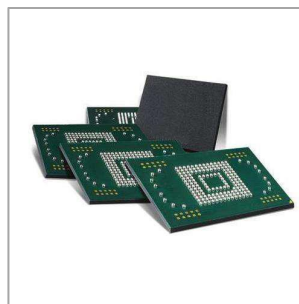
惯性测量单元IMU



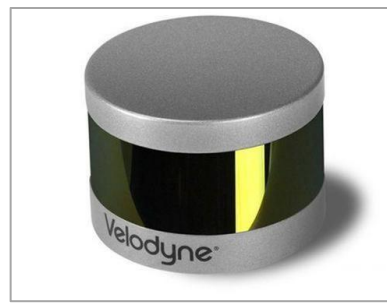
超声波传感器



红外传感器



毫米波雷达



激光雷达



双目深度视觉传感器

1.1 工业机器人组成

传感部分-环境交互系统

①

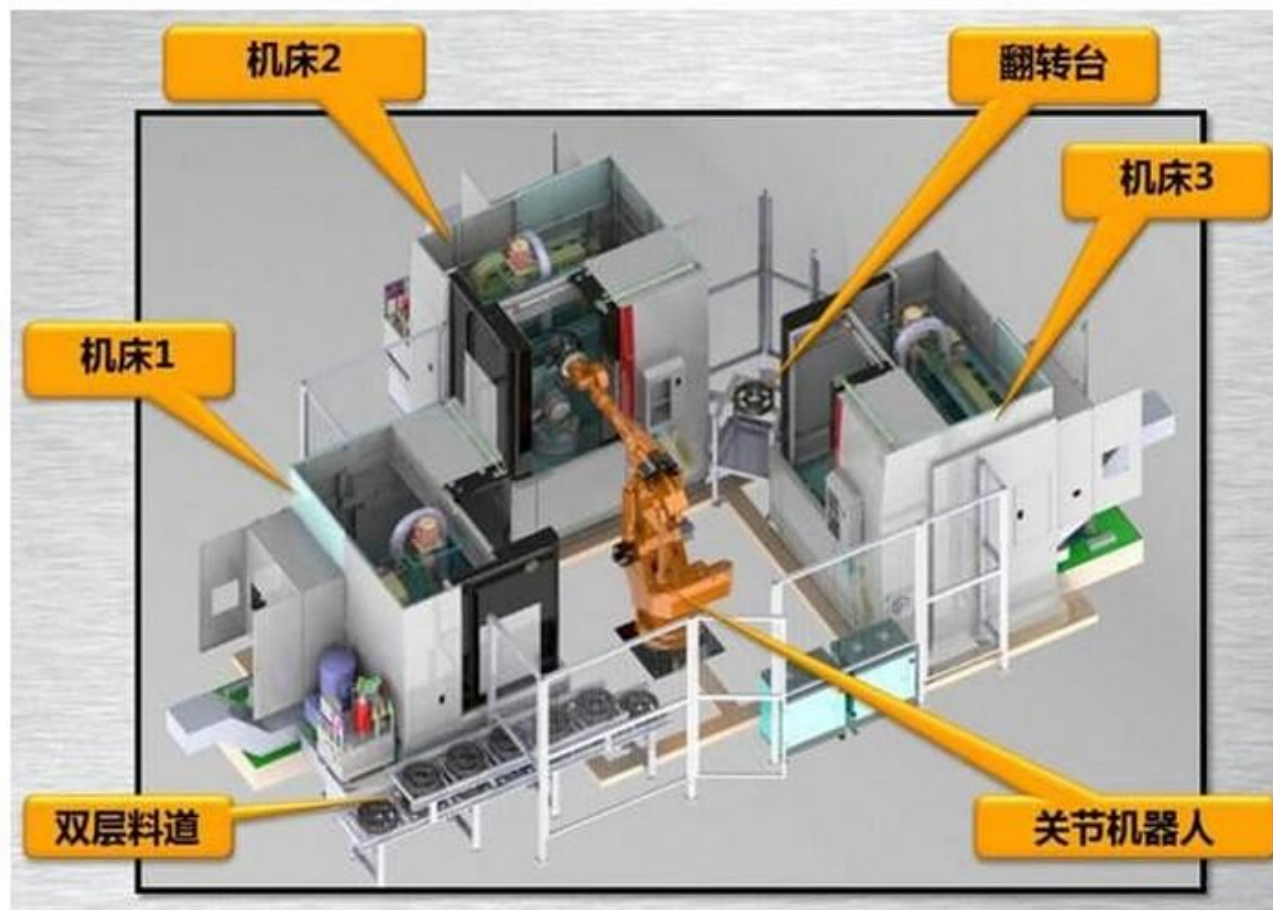
机器人-环境交互系统是实现工业机器人与外部环境中的设备相互联系和协调的系统

②

工业机器人与外部设备集成为一个功能单元，如加工制造单元、焊接单元、装配单元等

③

也可以是多台机器人、多台机床或设备、多个零件存储装置等集成为一个去执行复杂任务的功能单元

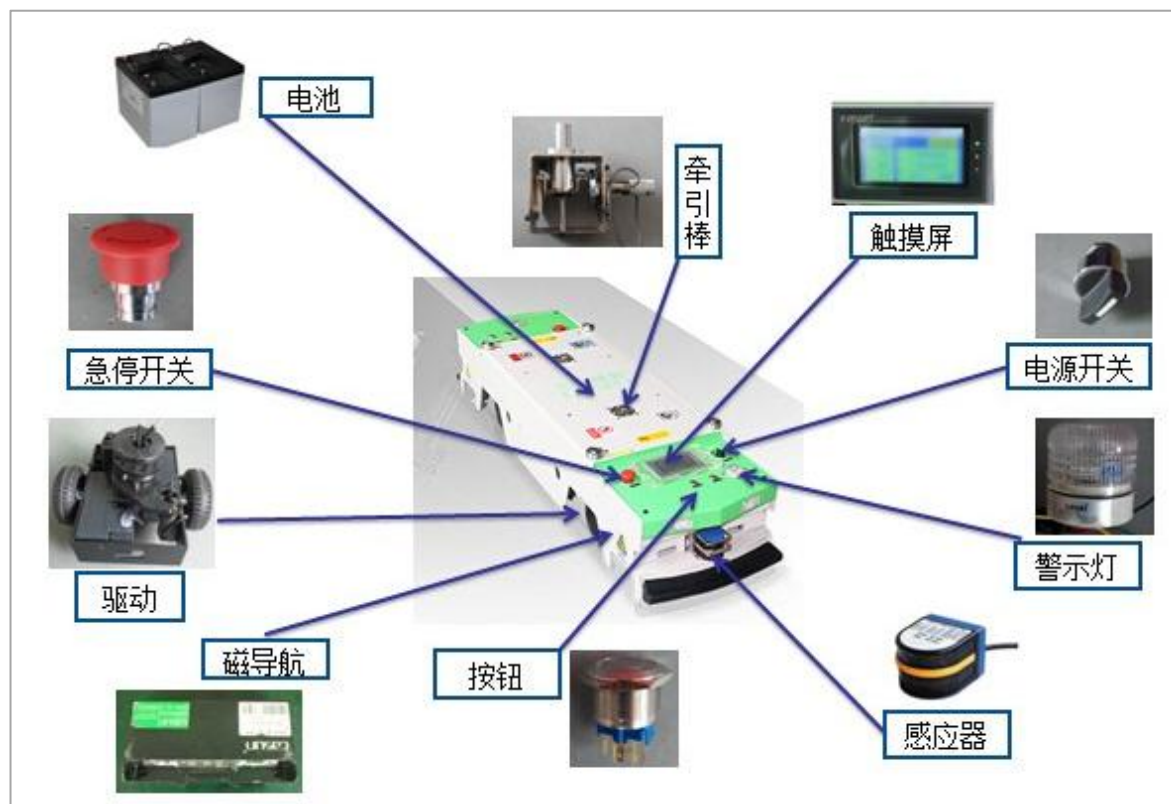


工业机器人与外部交互

1.1 工业机器人组成

控制部分-人机交互系统

人机交互系统是使操作人员参与机器人控制并与机器人进行联系的装置，例如，计算机的标准终端，指令控制台，信息显示板，危险信号报警器等。该系统归纳起来分为两大类：指令给定装置和信息显示装置。



指令给定装置

键盘、按钮、触摸屏、摇杆、舵机手柄等

信息显示装置

显示器、警示灯、LED矩阵等

自动搬运机器人（AGV）系统组成

1.1 工业机器人组成

控制部分-控制系统

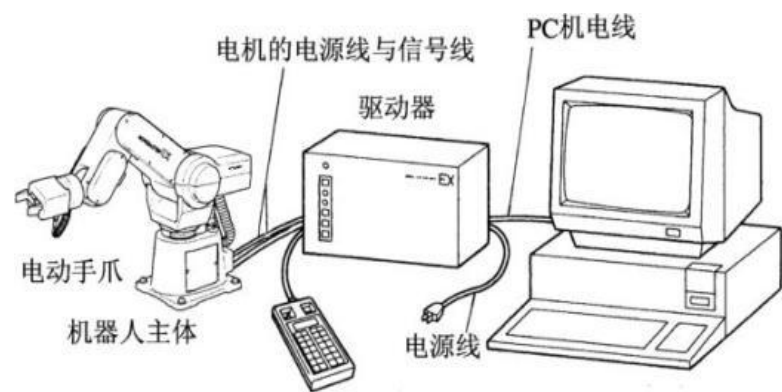
控制系统的任务是根据机器人的作业指令程序以及从传感器反馈回来的信号支配机器人的执行机构去完成规定的运动和功能。

控制系统分类

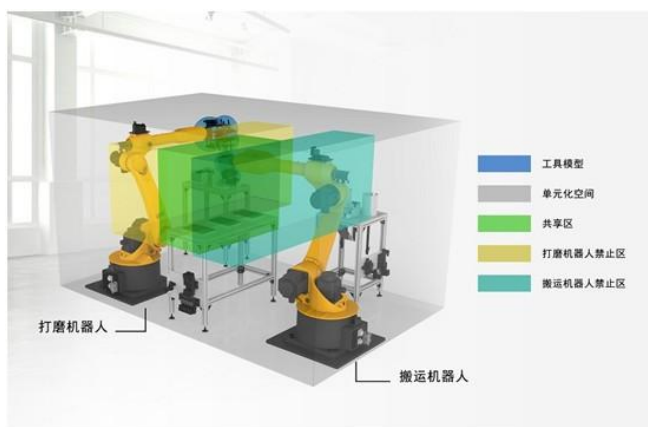
根据是否反馈信息，控制系统可分为开环控制系统和闭环控制系统

根据控制原理,控制系统可分为程序控制系统、适应性控制系统和人工智能控制系统

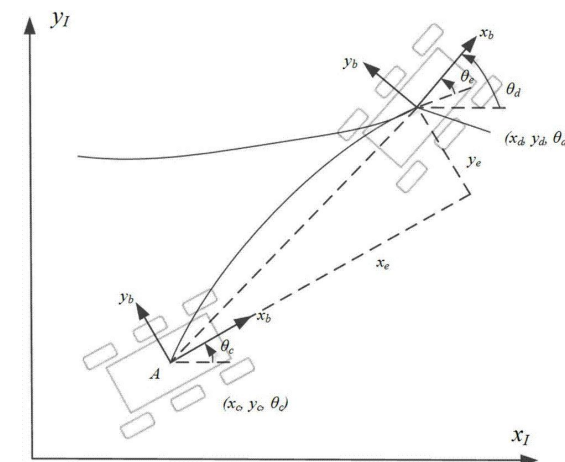
根据控制运动的形式,控制系统可分为点位控制和轨迹控制



三菱装配机器人系统



点位控制：机器人到达的点位



轨迹控制：机器人移动轨迹规划

1.1 工业机器人组成

控制部分-控制系统

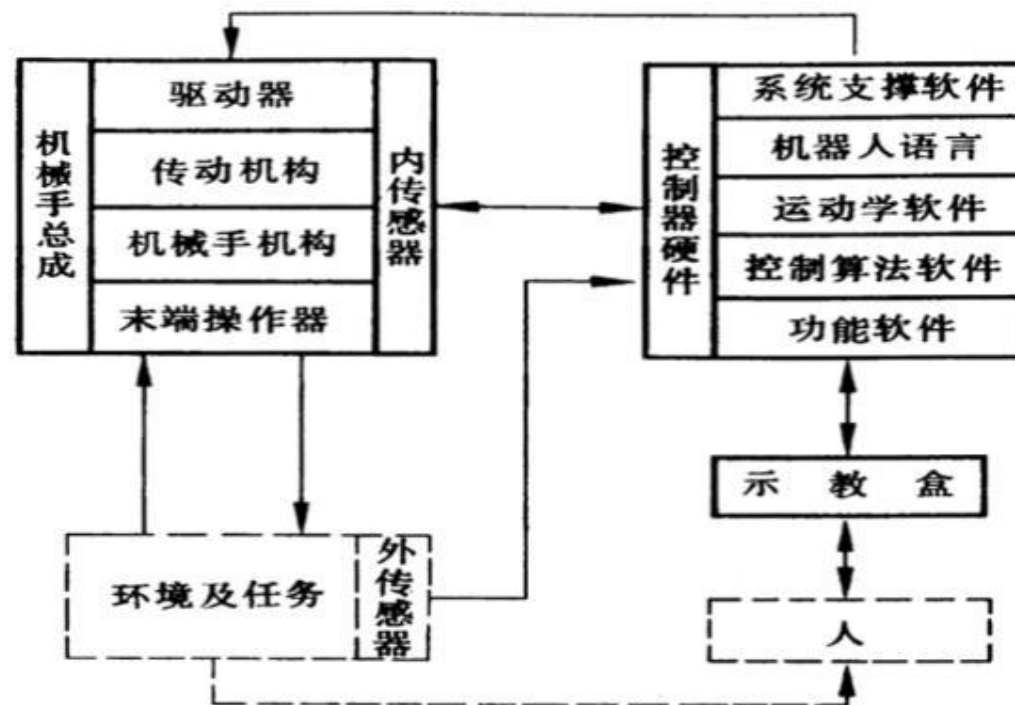
控制系统的任务是根据机器人的作业指令程序以及从传感器反馈回来的信号支配机器人的执行机构去完成规定的运动和功能。

系统运行-示教

机器人的基本工作原理是**示教再现**：

示教也称导引，即由用户导引机器人，一步步按实际任务操作一遍，机器人在导引过程中自动记忆示教的每个动作的位置、姿态、运动参数、工艺参数等，并自动生成一个连续执行全部操作的程序。

完成示教后，只需给机器人一个启动命令，机器人将精确地按示教动作，一步步完成全部操作。

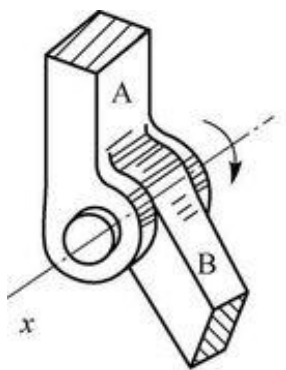


机器人的基本工作原理

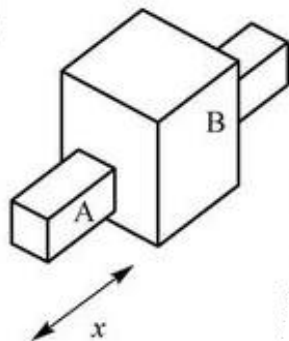
1.2 工业机器人的结构

机器人的基本结构由“**关节**”和“**连杆**”组成。

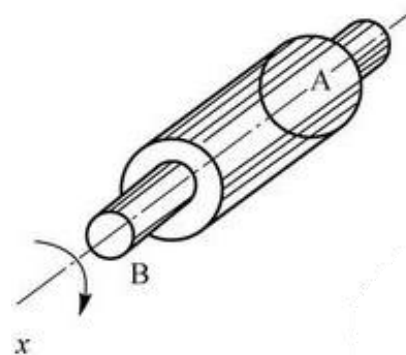
关节：关节（joint）即运动副(kinematic pairs)，是允许机器人手臂各零件之间发生相对运动的机构，是两构件直接接触并能产生相对运动的活动连接，A、B两部件可以做互动连接。



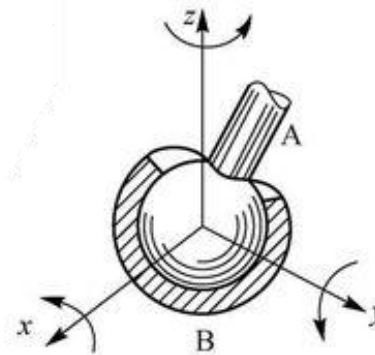
(a) 回转副



(b) 移动副



(c) 回转动副



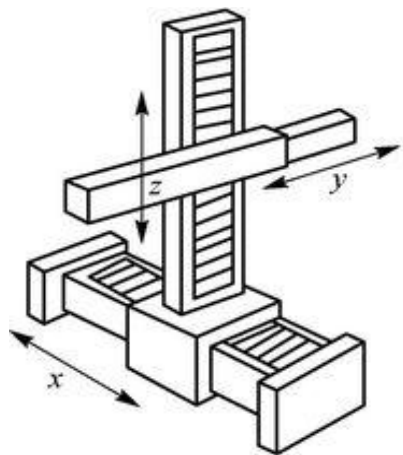
(d) 球面副

机器人主要关节形式

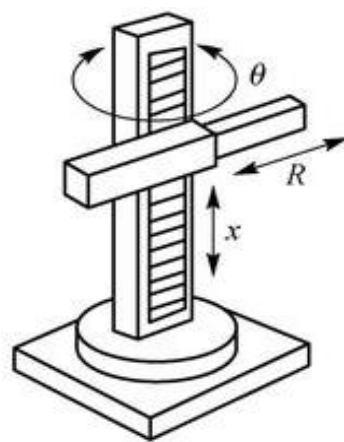
1.2 工业机器人的结构

机器人的基本结构由“**关节**”和“**连杆**”组成。

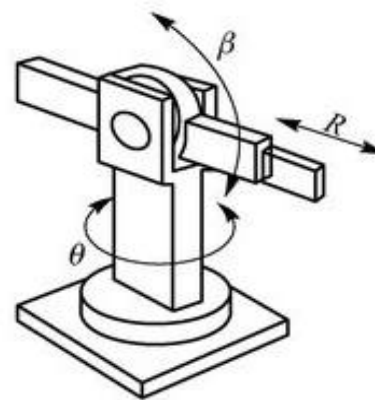
连杆：连杆（link）指机器人手臂上被相邻两关节分开的部分，是保持各关节间固定关系的刚体，是机械连杆机构中，两端分别与主动和从动构件铰接，以传递运动和力的杆件。连杆是机器人中的重要部件，它连接着关节，其作用是将一种运动形式转变为另一种运动形式，并把作用在主动构件上的力传给从动构件，以输出功率。



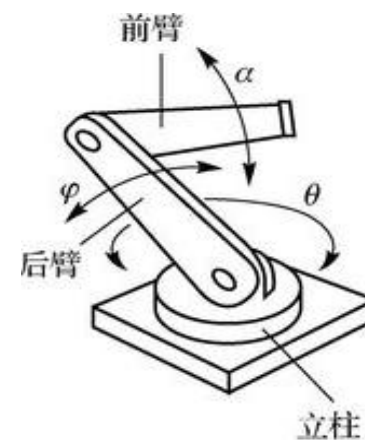
(a) 直角坐标型



(b) 圆柱坐标型



(c) 极坐标型

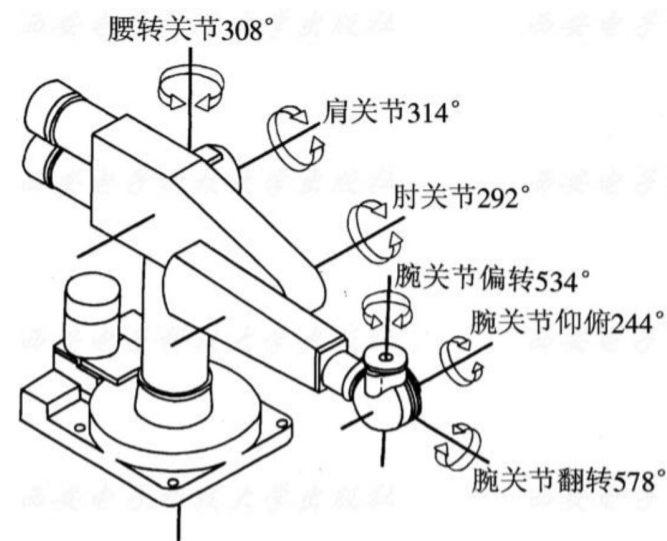
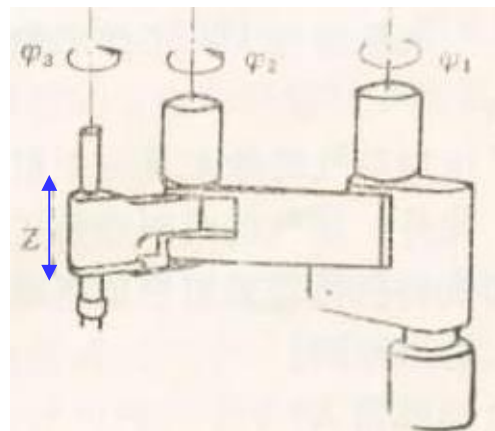
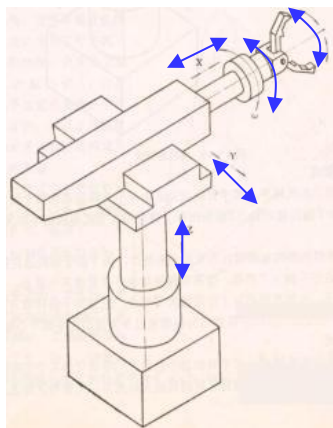
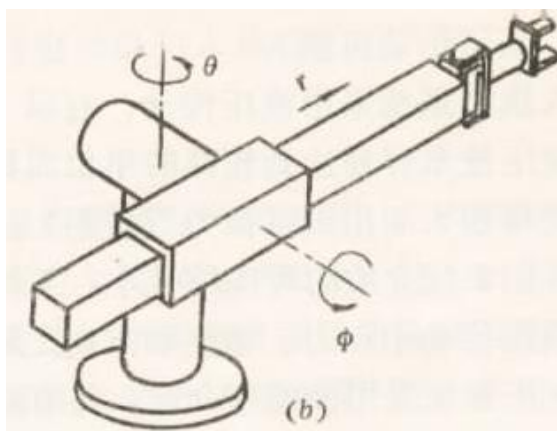


(d) 多关节坐标型

机器人主要结构形式

1.3 工业机器人的基本参数

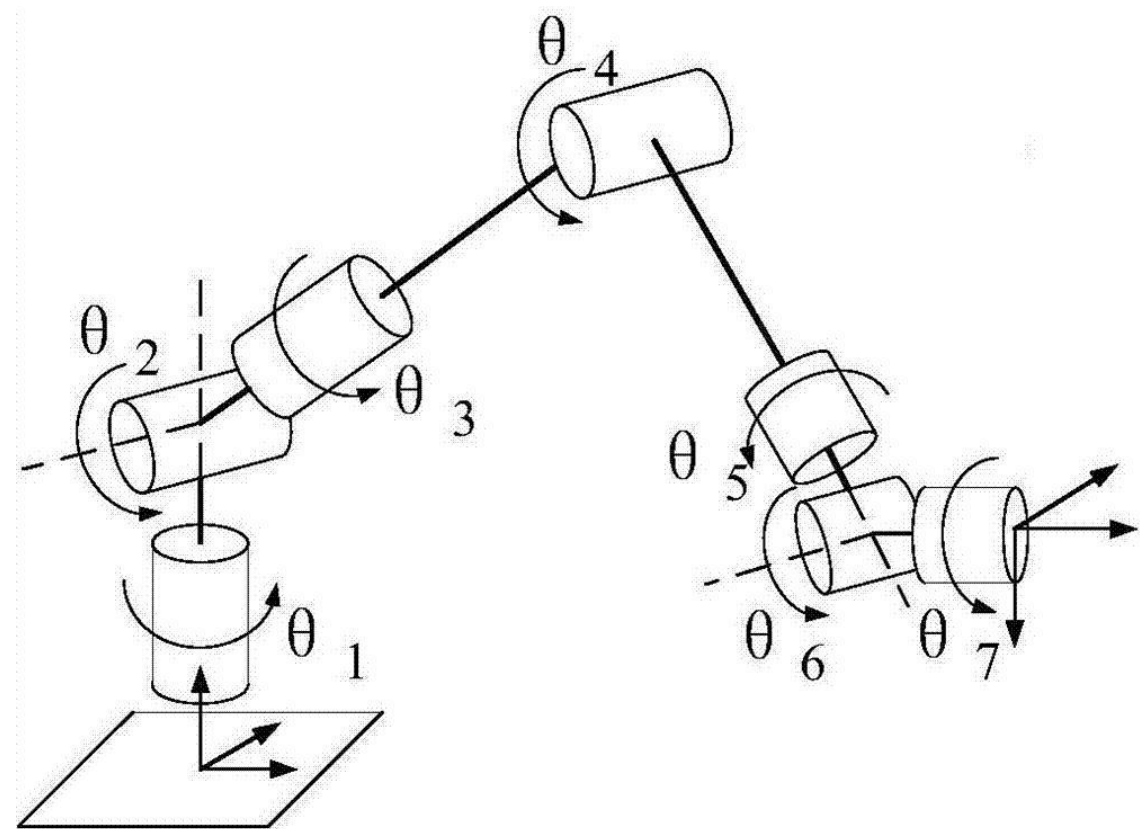
1.自由度 (Degree of freedom)：或者称坐标轴数，是指描述物体运动所需要的独立坐标数。手指的开、合，以及手指关的自由度一般不包括在内。



一般共有六种运动可能，称之为六个自由度，即沿直角坐标系X、Y、Z三轴方向的**移动自由度**和绕此三轴的**转动自由度**

1.3 工业机器人的基本参数

1.自由度(Degree of freedom): 或者称坐标轴数，是指描述物体运动所需要的独立坐标数。手指的开、合，以及手指关的自由度一般不包括在内。



7 自由度机械臂

Powered by DSP

整机性能（七自由度）		
额定负载	夹持	1kg
	末端	1kg
重复定位精度	$\pm 0.5\text{mm}$	
整体运动范围	1357*1069*1069mm	
本体重量（不含线缆）	23.68kg	
供电方式	24V 直流供电	
支持安装方式	台面安装、吊顶安装	
单模块工作环境	温度	0~55℃
	湿度	20~93%RH
控制单元数量	8个DSP（7自由度+1手爪）	

	运动类型	最大运动范围	最大速度	额定力矩	重量	精度
关节1	Roll	$\pm 180^\circ$ （*）	25 deg/s	150 N.m	3.75 Kg	$\pm 0.05^\circ$
关节2	Pitch	$\pm 105^\circ$	25 deg/s	150 N.m	3.75 Kg	$\pm 0.05^\circ$
关节3	Roll	$\pm 180^\circ$ （*）	30 deg/s	50 N.m	2.76 Kg	$\pm 0.05^\circ$
关节4	Pitch	$\pm 118^\circ$	30 deg/s	50 N.m	2.76 Kg	$\pm 0.05^\circ$
关节5	Roll	$\pm 180^\circ$ （*）	30 deg/s	50 N.m	2.76 Kg	$\pm 0.05^\circ$
关节6	Pitch	$\pm 100^\circ$	30 deg/s	50 N.m	5.09 Kg	$\pm 0.05^\circ$
关节7	Roll	$\pm 180^\circ$ （*）	30 deg/s	25 N.m	2.76 Kg	$\pm 0.05^\circ$

关节7
云台旋转模块

关节6
云台摆动模块

关节5
85旋转模块

关节3
85旋转模块

关节4
85旋转模块

关节2
93旋转模块

关节1
93旋转模块

www.Cntrades.Com

冗余自由度机器人

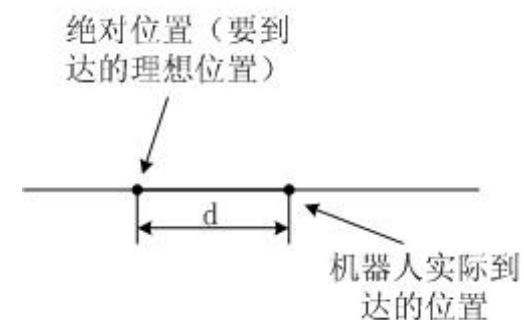
利用冗余自由度可以增加机器人的灵活性、躲避障碍物和改善动力性能。人的手臂(大臂、小臂、手腕)共有七个自由度, 所以工作起来很灵巧,手部可回避障碍而从不同方向到达同一个目的点。

1.3 工业机器人的基本参数

2.工作精度(Working Accuracy): 包括**定位精度**(Positioning Accuracy)和**重复定位精度**(Repeatability)

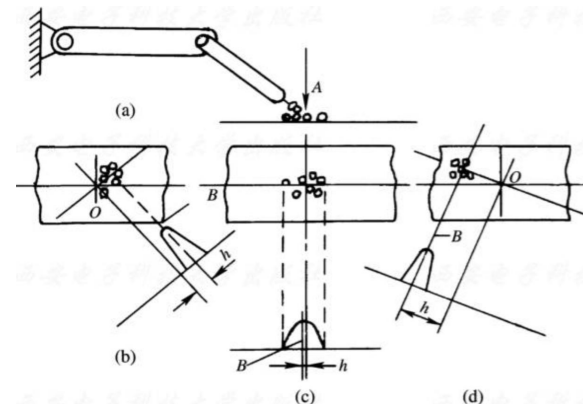
定位精度(Positioning Accuracy)：指机器人末端参考点实际到达的位置与所需要到达的理想位置之间的差距。

可以用**精密度**、**正确度**、和**准确度**三个参数来衡量



定位精度

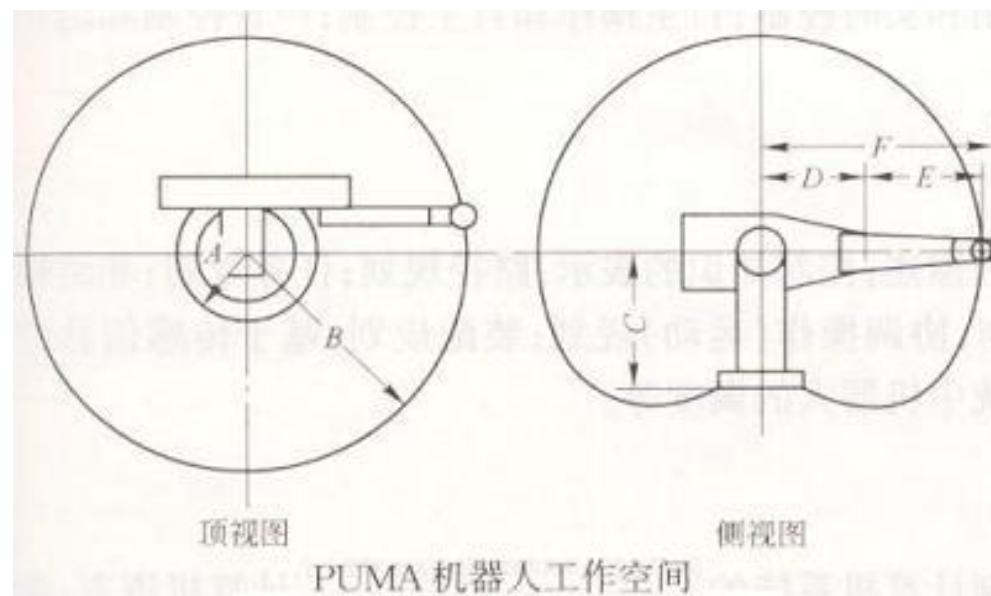
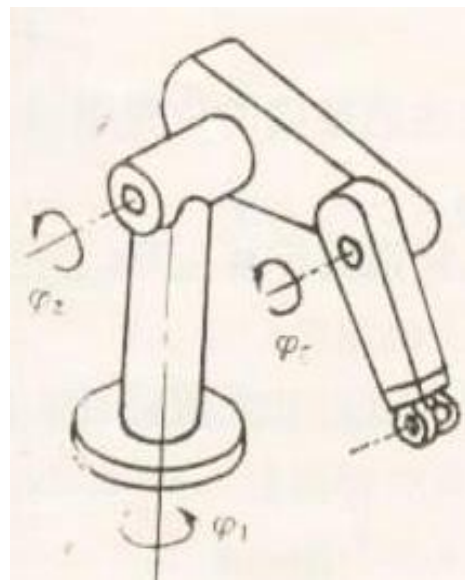
重复性（Repeatability）或**重复精度**：指机器人重复到达某一目标位置的差异程度。或在相同的位置指令下，机器人连续重复若干次其位置的分散情况。它是衡量一系列误差值的密集程度，即重复度。



重复性的离散程度

1.3 工业机器人的基本参数

3. 工作空间(Work Space): 工作空间是指机器人手臂末端或手腕中心所能到达的所有点的集合, 也叫工作区域或工作范围。因为末端操作器的尺寸和形状是多种多样的, 为了真实反映机器人的特征参数, 所以, 这里是指不安装末端操作器时的的工作空间。工作范围的形状和大小是十分重要的, 机器人在执行作业时可能会因为存在手部不能到达的作业死区(Dead Zone)而不能完成任务。



1.3 工业机器人的基本参数

4. 工作速度 (Working speed): 指机器人各个方向的移动速度或转动速度。这些速度可以相同, 可以不同。

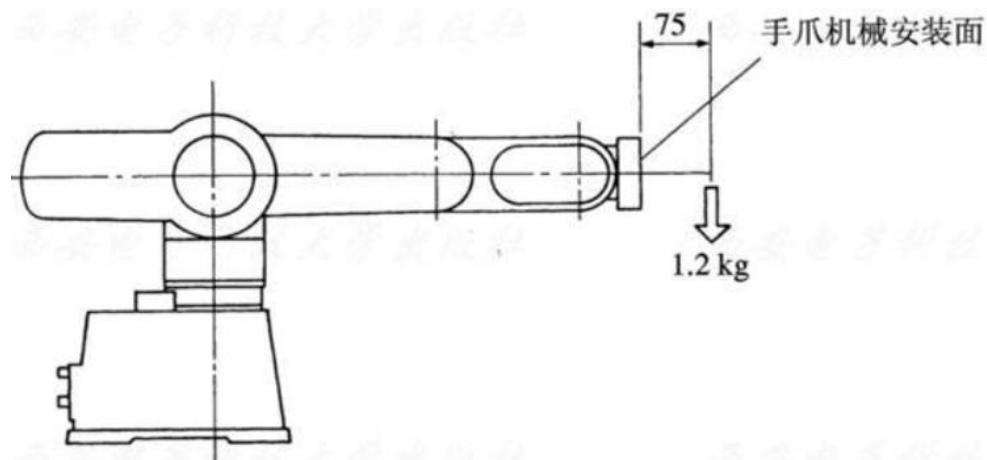
速度和加速度是表明机器人运动特性的主要指标。说明书中通常提供了主要运动自由度的最大稳定速度,但在实际应用中单纯考虑最大稳定速度是不够的。这是因为,由于驱动器输出功率的限制,从启动到达最大稳定速度或从最大稳定速度到停止,都需要一定时间。如果最大稳定速度高,允许的极限加速度小,则加减速的时间就会长一些,对应用而言的有效速度就要低一些;反之,如果最大稳定速度低,允许的极限加速度大,则加减速的时间就会短一些,这有利于有效速度的提高。但如果加速或减速过快,有可能引起定位时超调或振荡加剧,使得到达目标位置后需要等待振荡衰减的时间增加,则也可能使有效速度反而降低。所以,考虑机器人运动特性时,除注意最大稳定速度外,还应注意其最大允许的加减速速度。



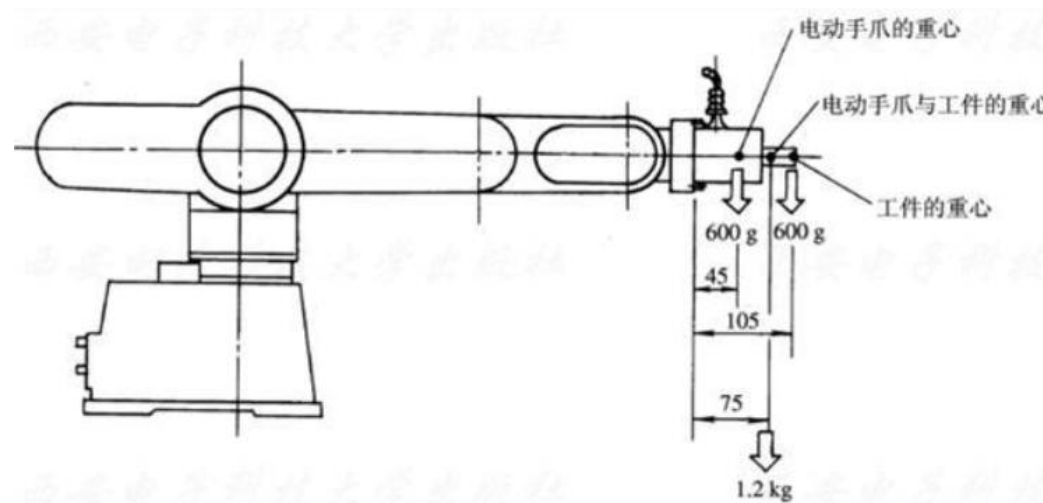
速度和加速度

1.3 工业机器人的基本参数

5.承载能力(Payload): 指机器人在工作范围内的**任何位姿(位置和姿态)**上所能承受的最大质量。承载能力不仅决定于负载的质量,而且还与机器人运行的速度和加速度的大小和方向有关。为了安全起见,承载能力这一技术指标是指高速运行时的承载能力。通常,承载能力不仅指负载,而且还包括了机器人末端操作器的质量。



三菱装配机器人不带电动手爪时的承载能力



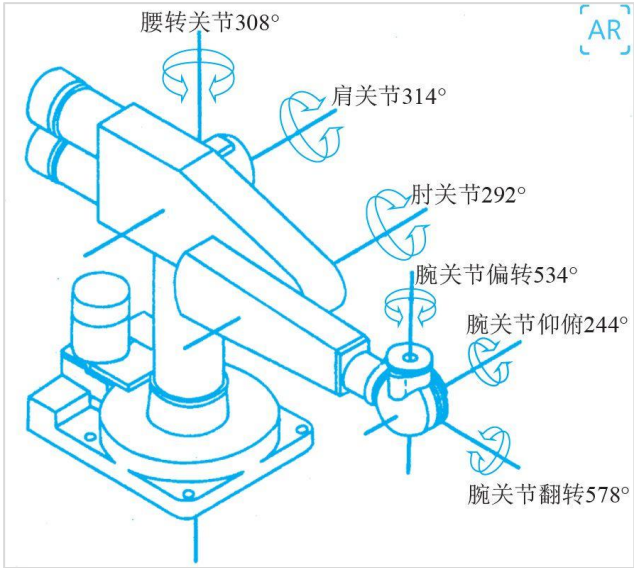
三菱装配机器人带电动手爪时的承载能力

1.3 工业机器人的基本参数

6. 示例：

项 目	技术参数
自由度	6
驱动	直流伺服电机
手爪控制	气动
控制器	系统机
重复定位精度	$\pm 0.1\text{ mm}$
承载能力	4.0 kg
手腕中心最大距离	866 mm
直线最大速度	0.5 m/s
功率要求	1150 W
重量	182 kg

PUMA 562机器人的主要技术参数



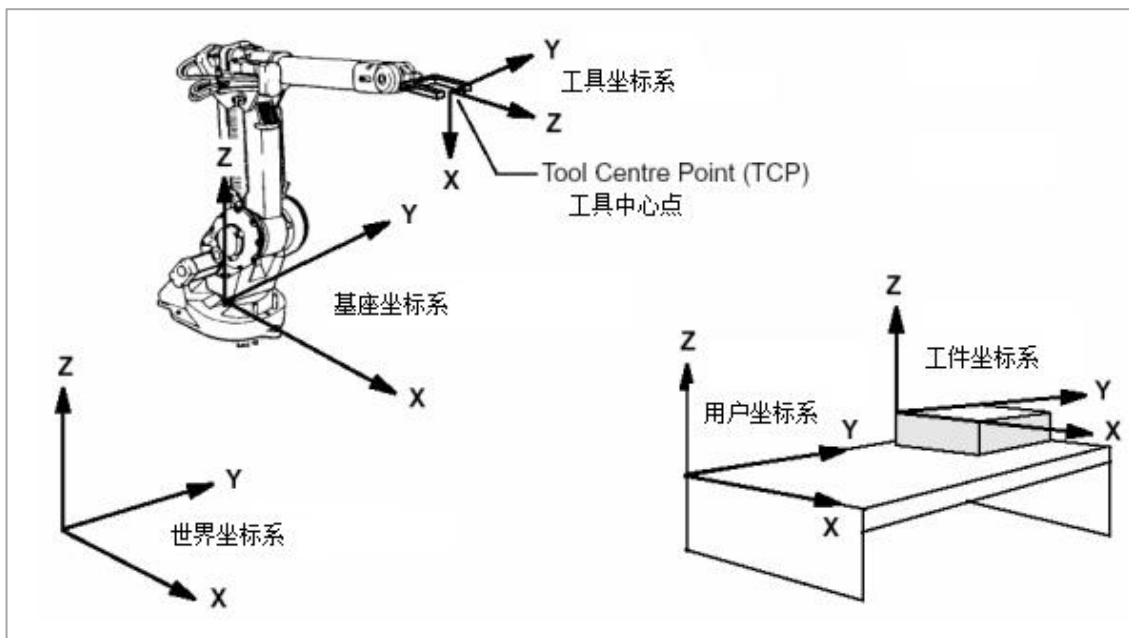
第4课 机器人的坐标变换

北邮移动机器人与智能技术实验室 编

2 坐标变换(TF)初步

ROS提供了tf库实现了坐标变换的封装

2.1 机器人的坐标系



机器人的常用坐标系

在分析机器人时会牵涉诸多坐标系，常见的坐标系有：

(1) 关节坐标系：主要描述各关节相对于标定零点的绝对位置，旋转轴常用 $^{\circ}$ 表示，线性轴的常用mm描述。关节坐标系下的坐标值均为机器人关节的绝对位置，方便用户调试点位时观察机器人的绝对位置，避免机器人出现极限位置或奇异位置。

(2) 基座坐标系：基座坐标系为直角坐标系，Z轴即第一轴的Z轴，X轴为回零后的正前方，Y轴由右手定则确定。直角坐标系下，用户可控制机器人末端沿坐标系任一方向移动或旋转，常用于现场点位示教。

(3) 工具坐标系：即安装在机器人末端的工具坐标系，原点及方向都是随着末端位置与角度不断变化的，该坐标系实际是将基座坐标系通过旋转及位移变化而来的，用户可以根据工具的外形、尺寸等建立与工具相对应的工具坐标系。

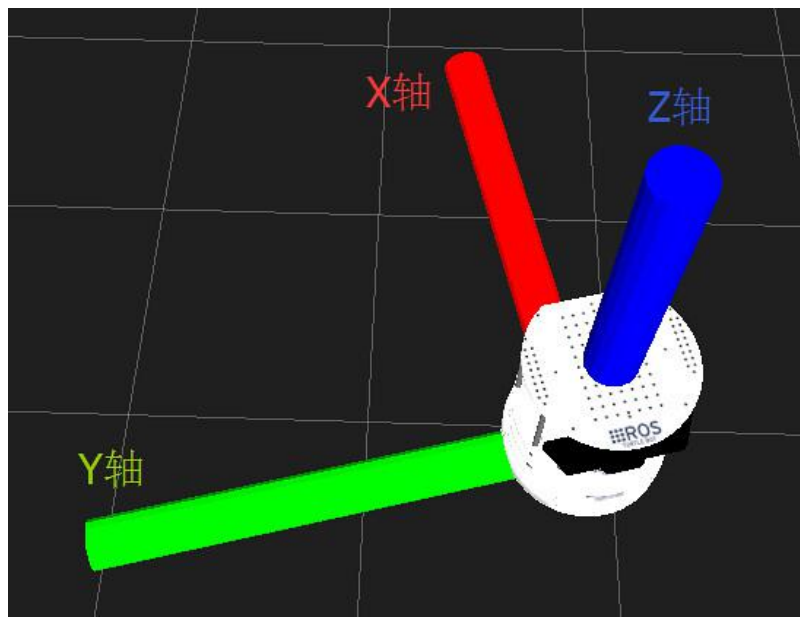
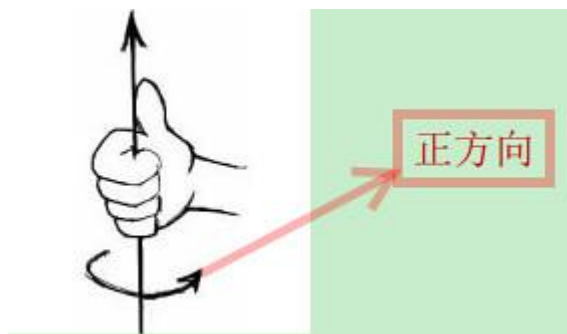
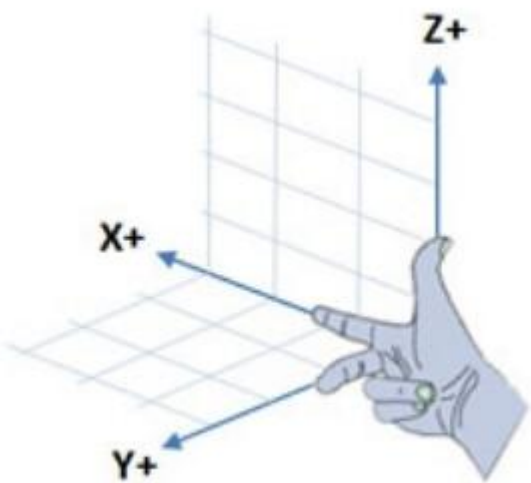
(4) 用户坐标系：用于描述各个物体或工位的方位的需要，用户常常在自己关心的平面建立自己的坐标系，以方便示教与生产线复制，减少调试工作量。离线仿真软件提取的基于定义坐标系轨迹控制点，可直接用于实际程序中，只需定义匹配的用户坐标系。

2.2 ROS中的坐标系

在 ROS 中坐标系总是 3 维的，而且遵循右手法则，Z 轴用拇指表示，X 轴用食指表示，Y 轴用中指。X 轴指向前方，Y 轴指向左方，Z 轴指向上方。

研究坐标系绕某轴旋转时，也是用右手法则，右手握住坐标轴，大拇指的方向朝着坐标轴朝向的正方向，四指环绕的方向定义沿着这个坐标轴旋转的正方向。

ROS提供了tf库实现了机器人坐标变换的封装



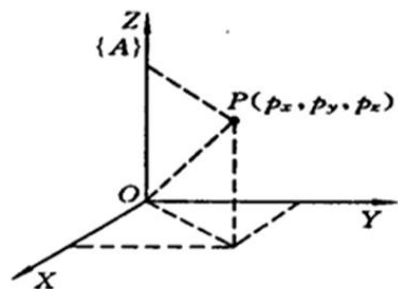
可视化工具中：默认 X 轴是红色、Y 轴时绿色、Z 轴是蓝色，也就是 XYZ 对应 RGB

2.3 机器人的位姿描述

参考阅读：机器人位姿描述与坐标变换

机器人的位置描述：在选定的直角坐标系 $\{A\}$,空间任一点的 P 位置可用的 3×1 位置矢量 ${}^A P$ 表示。

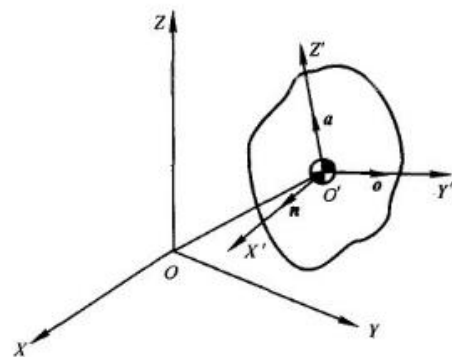
机器人的姿态描述：刚体与OX轴的夹角 r_x 、与OY轴的夹角 r_y 、与OZ轴的夹角 r_z



$${}^A p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

机器人的位置描述：X, Y, Z坐标

tf::Vector3(0.1, 0.0, 0.2)



$$R_j^i = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$$

机器人的姿态描述：与x,y,z轴夹角的余弦

$$R = \begin{bmatrix} \cos \angle X'X & \cos \angle Y'X & \cos \angle Z'X \\ \cos \angle X'Y & \cos \angle Y'Y & \cos \angle Z'Y \\ \cos \angle X'Z & \cos \angle Y'Z & \cos \angle Z'Z \end{bmatrix}$$

X' 方向 Y' 方向 Z' 方向

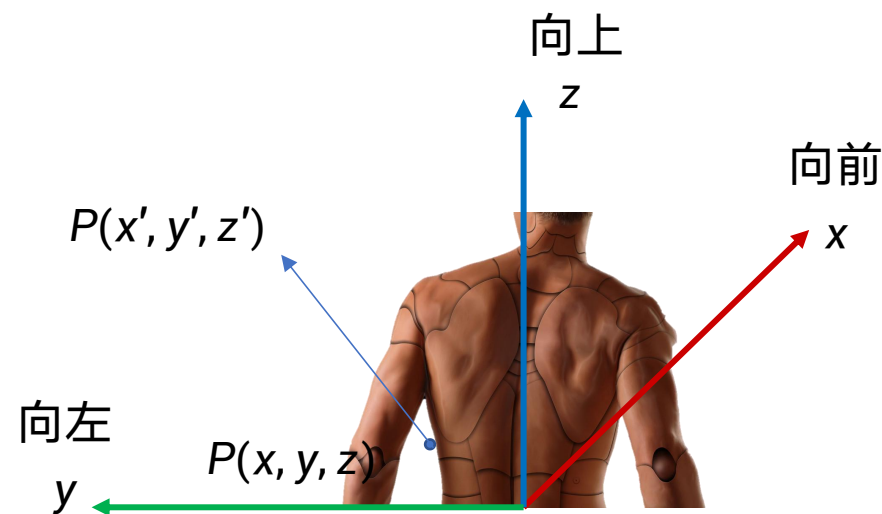
2.4 平移变换

[ROS规范阅读: REP 103 — Standard Units of Measure and Coordinate Conventions \(ROS.org\)](#)

为了变换方便，一般采用齐次坐标形式

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix}$$

平移变换矩阵



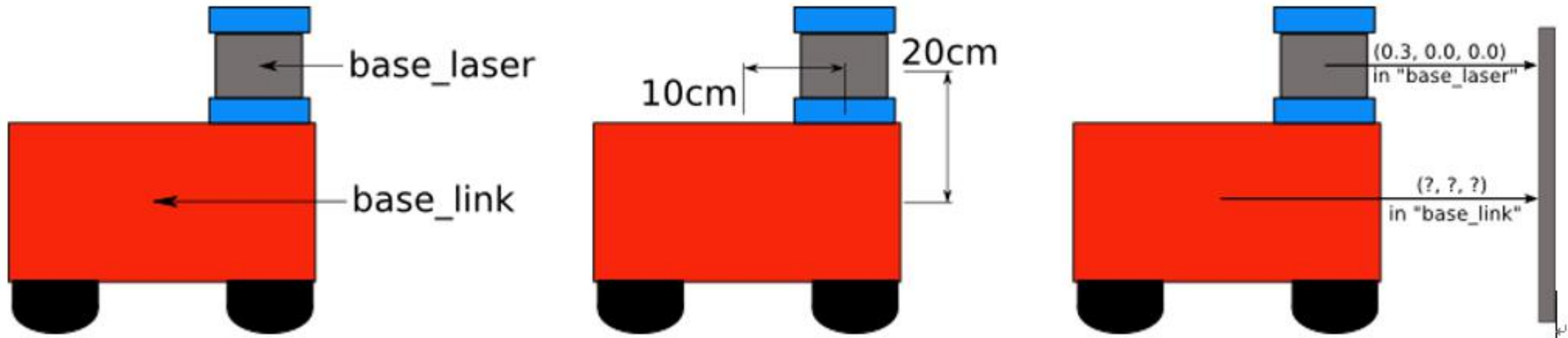
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2.4 平移变换

以移动机器人为例，定义两个坐标系：

base_link参考坐标系：以机器人移动平台的中心为原点

base_laser参考坐标系：以激光雷达的中心为原点



在ROS的tf变换树中定义了不同坐标系之间的平移与旋转变换关系，tf功能包提供了存储、计算不同数据在不同参考系之间变换的功能，因此只需要告诉tf树这些参考系之间的变换公式即可。

```
tf::Transform transform; // 初始化tf数据
transform.setOrigin( tf::Vector3(0.1, 0.0, 0.2) ) //定义了两个坐标系之间的变换关系
static tf::TransformBroadcaster br; // 广播坐标关系
br.sendTransform(tf::StampedTransform(transform, ros::Time::now(), "base_link", "base_laser"));
```

2.5 旋转变换—欧拉角和四元数

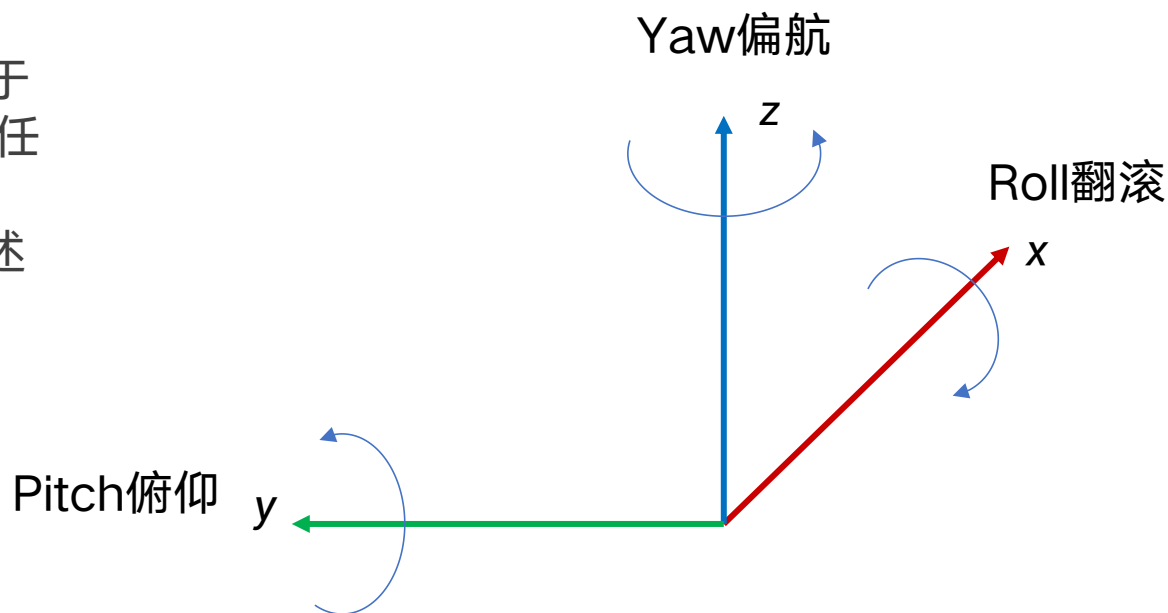
- **欧拉角**就是绕 Z 轴、Y 轴、X 轴的旋转角度，记做 ψ 、 θ 、 φ ，分别为 Yaw（偏航）、Pitch（俯仰）、Roll（翻滚）
- 欧拉角会出现万向节死锁，比如三维空间中有一个平行于 X 轴的向量，将它绕 Y 轴旋转直到它平行于 Z 轴，这时任何绕 Z 轴的旋转都改变不了该向量的方向
- **四元数**(w,x,y,z)是为解决以上问题引入的旋转另一种表述形式，它们的平方和为1，即 $w^2 + x^2 + y^2 + z^2 = 1$

欧拉角到四元数的转换：

$$q = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\varphi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\varphi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\varphi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\varphi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\varphi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\varphi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\varphi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix}$$

四元数到欧拉角的转换：

$$\begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(wx + yz)}{1 - 2(x^2 + y^2)}\right) \\ \arcsin(2(wy - zx)) \\ \arctan\left(\frac{2(wz + xy)}{1 - 2(y^2 + z^2)}\right) \end{bmatrix}$$



推导过程：四元数与欧拉角（Yaw、Pitch、Roll）的转换

2.5 旋转变换—欧拉角和四元数

```
#include "tf/transform_datatypes.h"//转换函数头文件
#include <nav_msgs/Odometry.h>//里程计信息格式

/*****四元数转RPY欧拉角，以odomsub的回调函数为例*****/

void odomCallback(const nav_msgs::Odometry &odom) {

    tf::Quaternion quat;
    tf::quaternionMsgToTF(odom.pose.pose.orientation, quat);

    double roll, pitch, yaw;//定义存储r\p\y的容器
    tf::Matrix3x3(quat).getRPY(roll, pitch, yaw);//进行转换

}

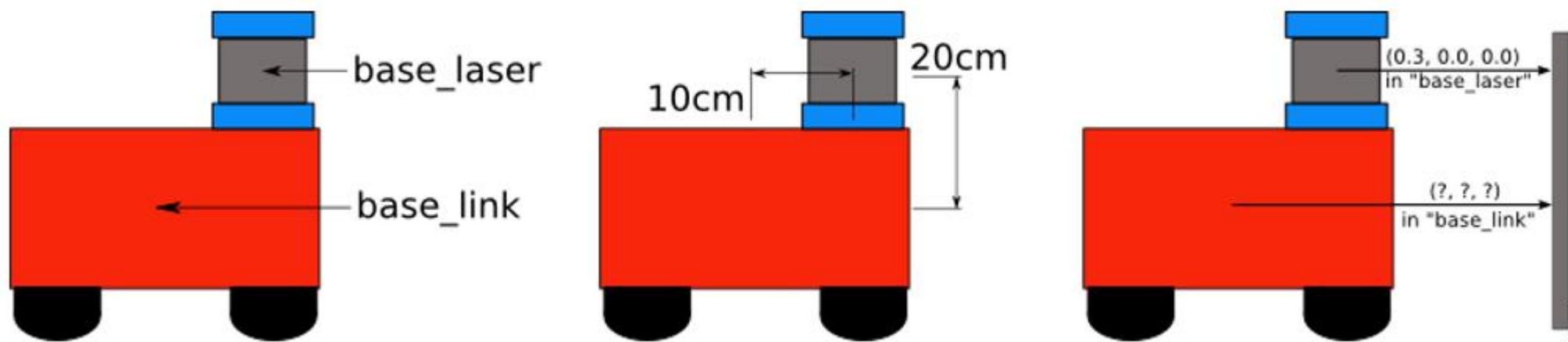
/*****RPY欧拉角转四元数*****/
tf::createQuaternionMsgFromRollPitchYaw(double r, double p, double y);//返回四元数
tf::createQuaternionMsgFromYaw(double y);//只通过y即绕z的旋转角度计算四元数，用于平面小车。返回四元数
```

2.4 平移变换

以移动机器人为例，定义两个坐标系：

base_link参考坐标系：以机器人移动平台的中心为原点

base_laser参考坐标系：以激光雷达的中心为原点



ROS的tf定义旋转：

```
// 初始化tf数据tf::Transform transform;  
transform.setOrigin( tf::Vector3(0.1, 0.0, 0.2) ) //定义了两个坐标系之间的变换关系  
transform.setRotation( tf::Quaternion(0,0,0,1) ); //定义了两个坐标系之间的旋转关系  
//广播关系  
...
```

第4课 机器人的坐标变换

北邮移动机器人与智能技术实验室 编

3 ROS中的TF

3.1 ROS TF概念的理解

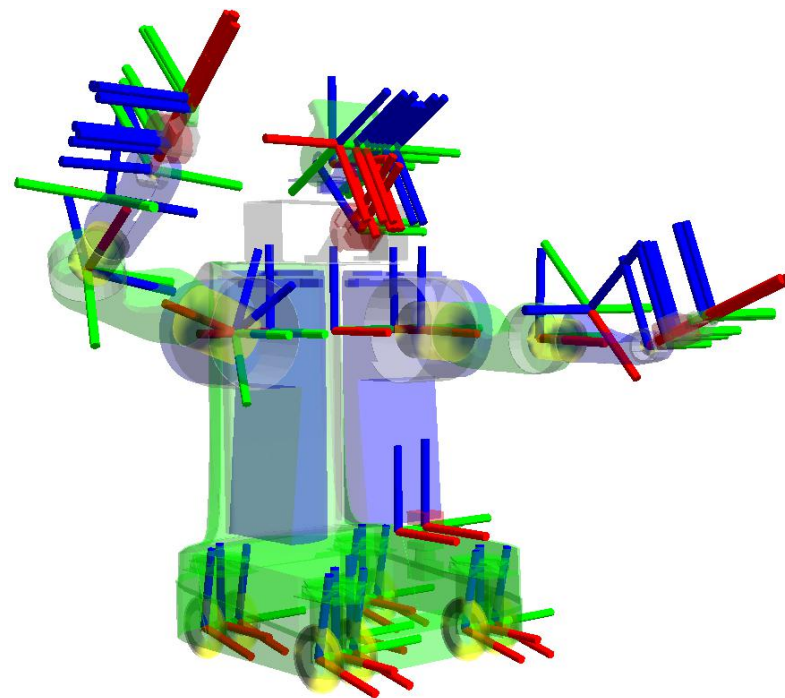
坐标变换（TransForm，简称TF）是ROS世界里的一个基本且重要的概念。坐标变换包括了位置和姿态两个方面的变换，ROS中机器人模型包含大量的部件，每一个部件称为Link，每一个Link部件拥有一个坐标系（Frame）。

TF通过树状结构维护坐标系之间的关系，依靠Topic话题通信机制来持续的发布不同Link部件之间的坐标关系。

需要保证父子坐标系都有某个节点在持续发布他们之间的位姿关系，才能保证树状结构的完整性，只有父子坐标系的位姿关系均被正确发布，才能保证任意两个Frame坐标系之间的连通性。

官网: [About tf2 — ROS 2 Documentation: Foxy documentation](#)
[tf - ROS Wiki](#)

论文: [Foote T . Tf: The transform library\[C\]// IEEE Conference on Technologies for Practical Robot Applications. IEEE, 2013.](#)

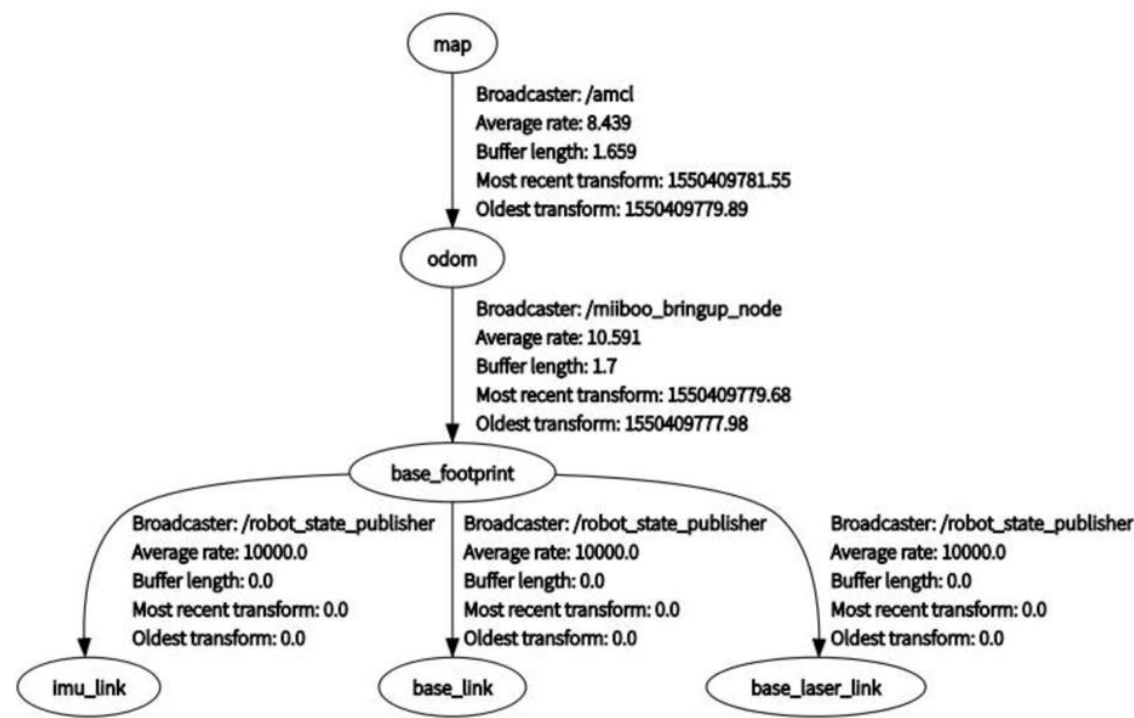


每一个Link部件拥有一个坐标系（Frame），构成了TF树

3.1 ROS TF概念的理解

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
  string child_frame_id
  geometry_msgs/Transform transform
    geometry_msgs/Vector3 translation
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion rotation
      float64 x
      float64 y
      float64 z
      float64 w
```

TF数据类型TransformStamped.msg



每一个Link部件拥有一个坐标系（Frame），构成了TF树

3.2 课下练习

ROS文档：tf2/Tutorials – ROS Wiki

tf2/ Tutorials

Many of the tf2 tutorials are available for both C++ and Python. The tutorials are streamlined to complete either the C++ track or the Python track. If you want to learn both C++ and Python, you should run through the tutorials once for C++ and once for Python.

目录

- 1. Workspace Setup
- 2. Learning tf2
- 3. Debugging tf2
- 4. Using sensor messages with tf2
- 5. Setting up your robot with tf2
- 6. Extending tf2
- 7. Migrating from tf1 to tf2

1. Workspace Setup

If you have not yet created a workspace in which to complete the tutorials, [click here for some brief instructions](#) .

2. Learning tf2

1. [tf2/Tutorials/Introduction to tf2](#)

This tutorial will give you a good idea of what tf2 can do for you. It shows off some of the tf2 power in a multi-robot example using turtlesim. This also introduces using tf2_echo, view_frames, and rviz.

C++	Python
1. Writing a tf2 static broadcaster (C++) This tutorial teaches you how to broadcast static coordinate frames to tf2	1. Writing a tf2 static broadcaster (Python) This tutorial teaches you how to broadcast static coordinate frames to tf2
2. Writing a tf2 broadcaster (C++) This tutorial teaches you how to broadcast coordinate frames of a robot to tf2.	2. Writing a tf2 broadcaster (Python) This tutorial teaches you how to broadcast the state of a robot to tf2.
3. Writing a tf2 listener (C++) This tutorial teaches you how to use tf2 to get access to frame transformations.	3. Writing a tf2 listener (Python) This tutorial teaches you how to use tf2 to get access to frame transformations.

ROS 2 文档：tf2 Tutorials — ROS 2 Documentation

tf2 Tutorials

- Introduction to tf2
- Writing a tf2 static broadcaster (Python)
- Writing a tf2 static broadcaster (C++)
- Writing a tf2 broadcaster (Python)
- Writing a tf2 broadcaster (C++)
- Writing a tf2 listener (Python)
- Writing a tf2 listener (C++)
- Adding a frame (Python)
- Adding a frame (C++)
- Learning about tf2 and time (Python)
- Learning about tf2 and time (C++)
- Time travel with tf2 (Python)
- Time travel with tf2 (C++)
- Debugging tf2 problems
- Quaternion fundamentals
- Using stamped datatypes with tf2_ros::MessageFilter

[Home](#) » [Tutorials](#) » tf2 Tutorials

[Edit on GitHub](#)

You're reading the documentation for an older, but still supported, version of ROS 2. For information on the latest version, please have a look at [Galactic](#).

tf2 Tutorials

Many of the tf2 tutorials are available for both C++ and Python. The tutorials are streamlined to complete either the C++ track or the Python track. If you want to learn both C++ and Python, you should go through the tutorials once for C++ and once for Python.

Contents

- [Workspace setup](#)
- [Learning tf2](#)
- [Debugging tf2](#)
- [Using sensor messages with tf2](#)

Workspace setup

移动机器人开发技术（激光SLAM版）配套教学PPT

谢 谢 观 看



北京邮电大学

Beijing University of Posts and Telecommunications

移动机器人与智能技术实验室编

宋桂岭 明安龙 2021.9

expsong@qq.com