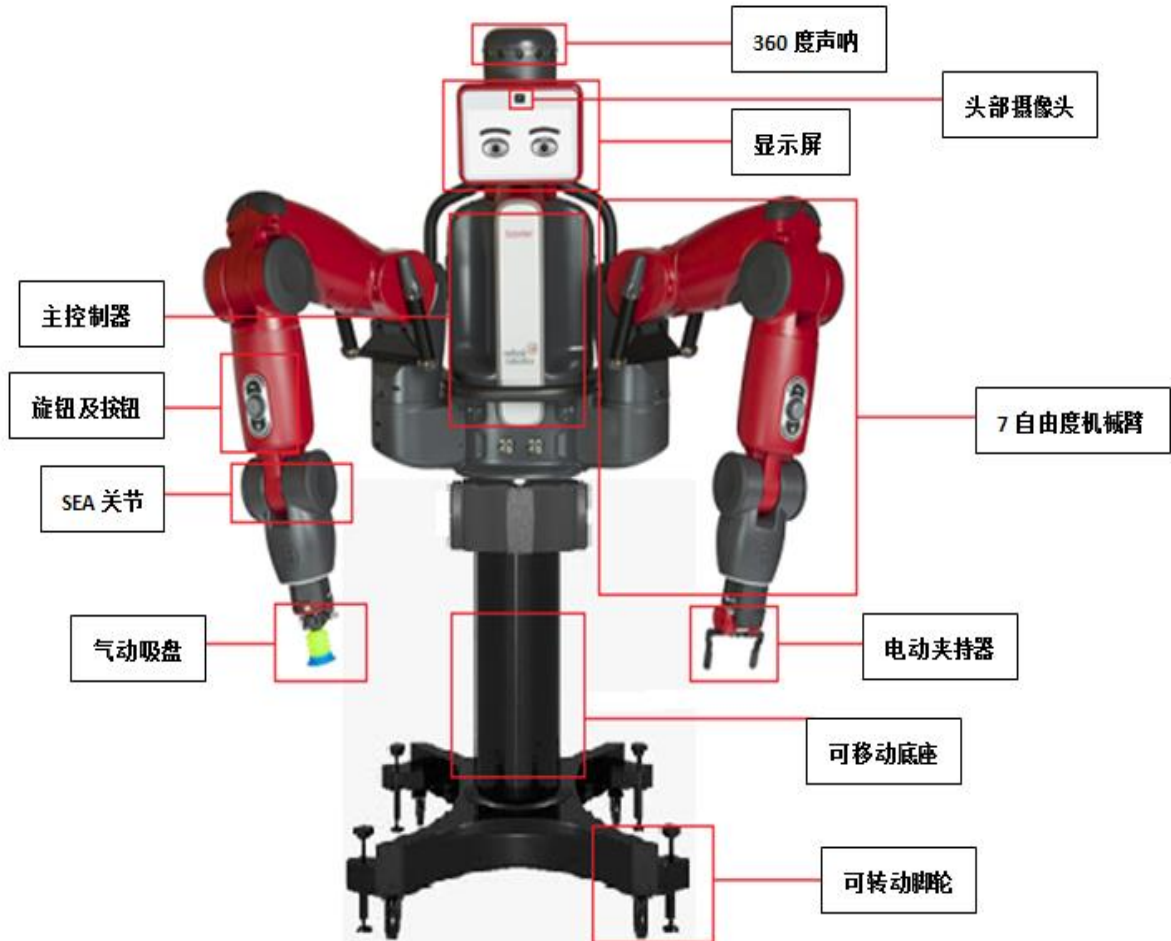


## Baxter 技术参数



## 1 Baxter 硬件参数

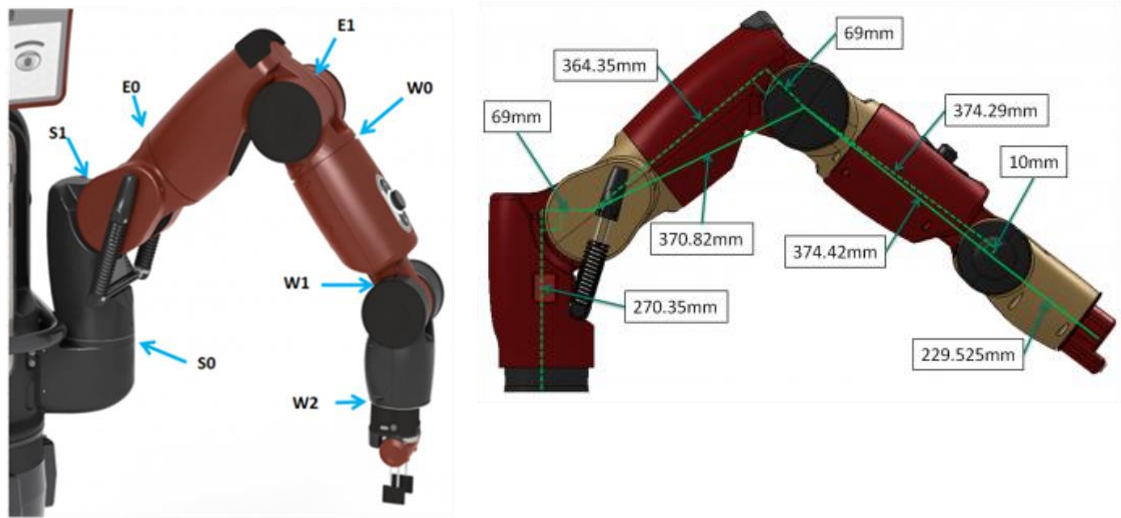


图 1 机械臂关节分布示意图

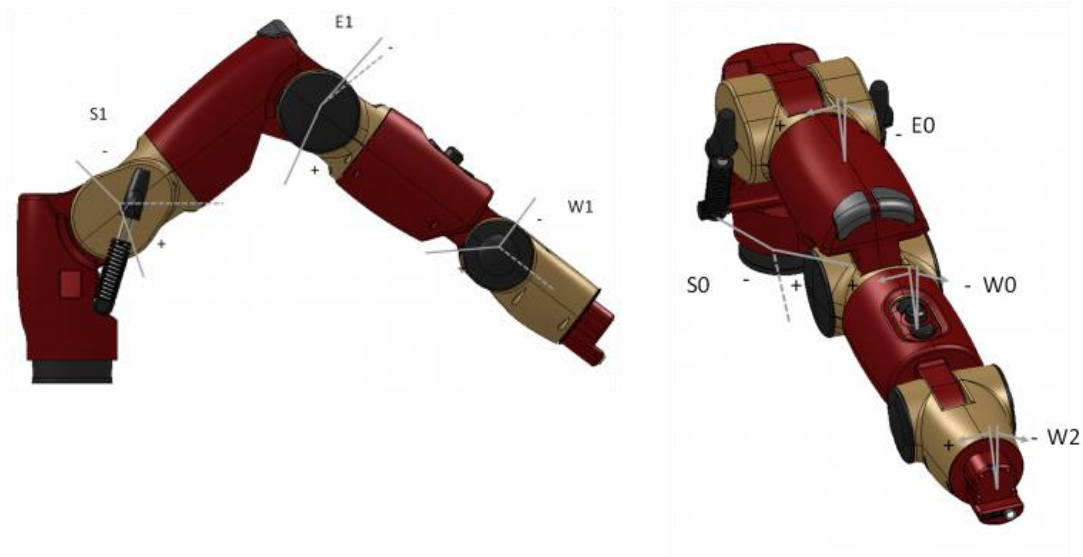


图 2 关节转动示意图

### 关节转动范围

关节	最小转角 (角度)	最大转角	转动范围	最小转角 (弧度)	最大转角	转动范围
S1	-123	+60	183	-2.147	+1.047	3.194

E1	-2.864	+150	153	-0.05	+2.618	2.67
W1	-90	+120	210	-1.5707	+2.094	3.6647
S0	-97.494	+97.494	194.998	-1.7016	+1.7016	3.4033
E0	-174.987	+174.987	349.979	-3.0541	+3.0541	6.1083
W0	-175.25	+175.25	350.5	-3.059	+3.059	6.117
W2	-175.25	+175.25	350.5	-3.059	+3.059	6.117

### 关节最大速度、弯曲刚度与峰值力矩

关节	最大速度 (弧度/秒)	弯曲刚度	峰值力矩
S0	2.0	50Nm 时 3.4 度 (约 843Nm/rad)	50Nm
S1	2.0	50Nm 时 3.4 度 (约 843Nm/rad)	50Nm
E0	2.0	50Nm 时 3.4 度 (约 843Nm/rad)	50Nm
E1	2.0	50Nm 时 3.4 度 (约 843Nm/rad)	50Nm
W0	4.0	15Nm 时 3.4 度 (约 250Nm/rad)	15Nm
W1	4.0	15Nm 时 3.4 度 (约 250Nm/rad)	15Nm
W2	4.0	15Nm 时 3.4 度 (约 250Nm/rad)	15Nm

### 其他硬件参数

相机	
最大分辨率	1280x800 像素
有效分辨率	640x480 像素
帧速	30 帧/秒
焦距	1.2mm

内置主控	
处理器	第三代 Intel Core i7-3770 (8MB, 3.4GHz)
内存	4GB, 1600MHZ, DDR3
硬盘	128GB 固态硬盘

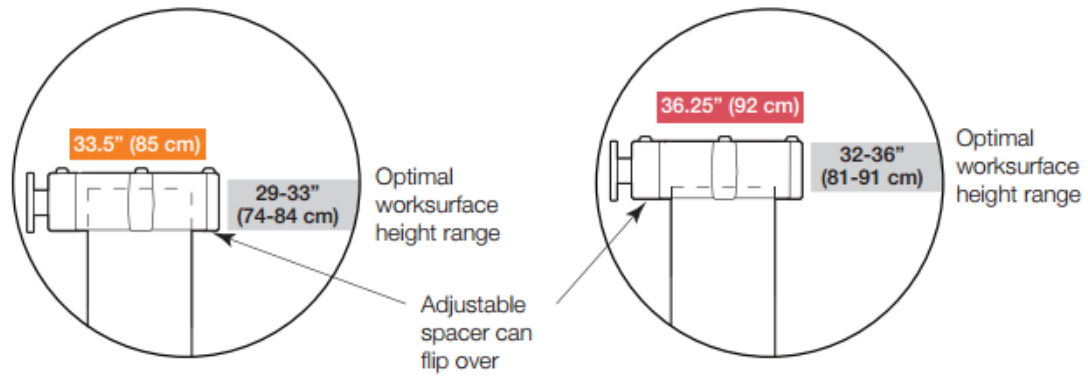


图 3 底座安装示意图

重量	
总重 (带底座)	135.2kg
单臂	21.3kg
躯干	31.8kg
底座	60.8kg

电气参数	
电池供电时	需要 DC-120VAC 逆变器 (Baxter 内部电脑不能通过 24VDC 直接供电)
接口	标准 120VAC 电源, Baxter 内部电源系统和电脑也支持 90-264VAC (47-63Hz)
最大功耗	120VAC, 6A, 约 720W
电气效率	87%-92%
内部电源	医用级直流开关电源
电压骤降	电压骤降至 90V 时, 持续中断需要手动启动
电压闪变	滞留时间 20ms
电压不平衡	只需单相电供电

其他参数	
屏幕分辨率	1024x600 像素
位置精度	+/-5mm
最大负载 (含夹持器)	2.2kg
最大夹持力	35N
红外探测距离	4-40cm

### Baxter 电动夹持器技术参数

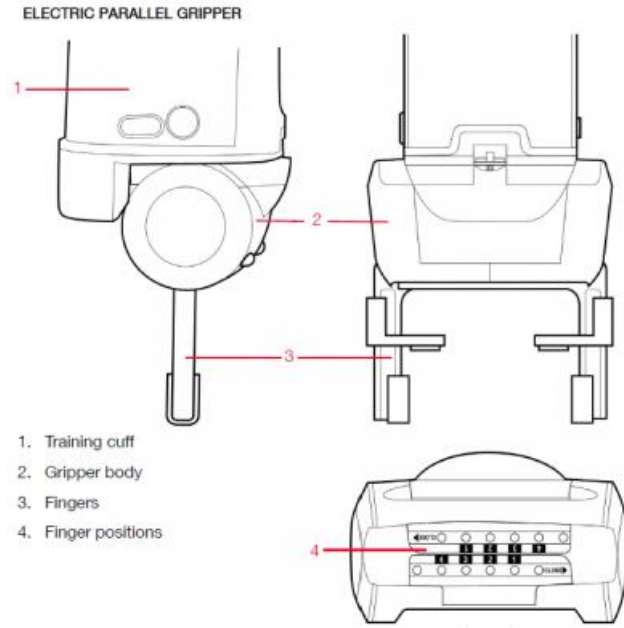


图 4 夹持器示意图

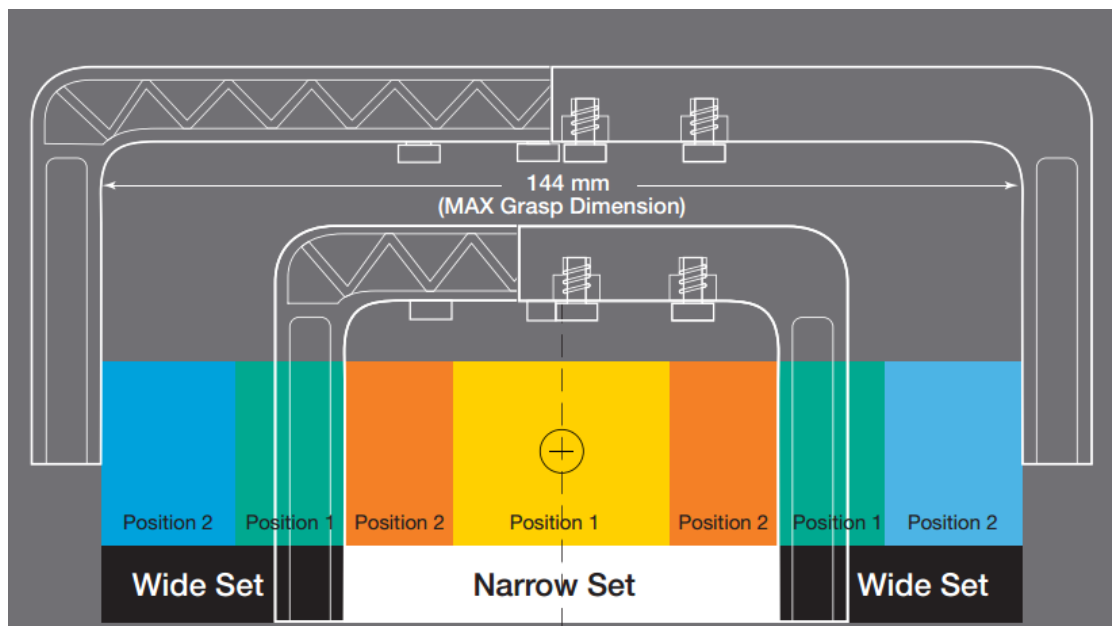


图 5 夹爪不同安装位置位置

Object Grasp Width	Finger	Position
0-34 mm	Narrow	1
34-68 mm	Narrow	2
68-102 mm	Wide	1
102-144 mm	Wide	2

图 6 根据不同宽度的物体选择夹爪类型及安装位置

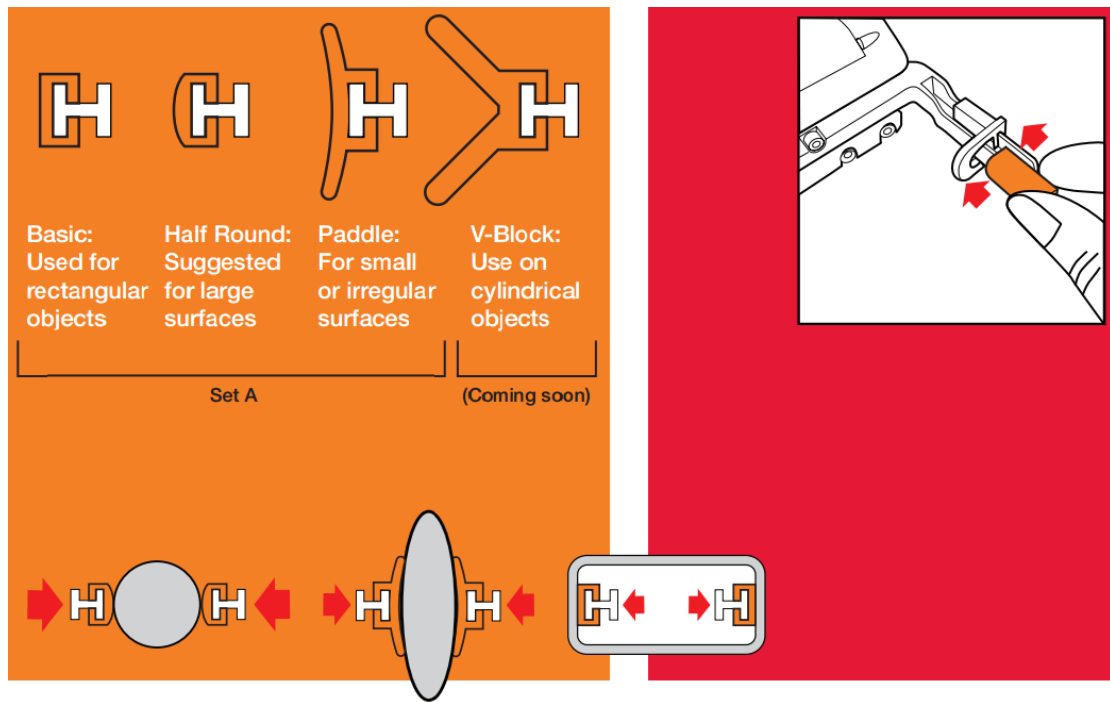


图 7 夹爪不同类型的指端

# Baxter 电动夹持器最大夹持力约 4.4kg

Baxter 气动吸盘图示

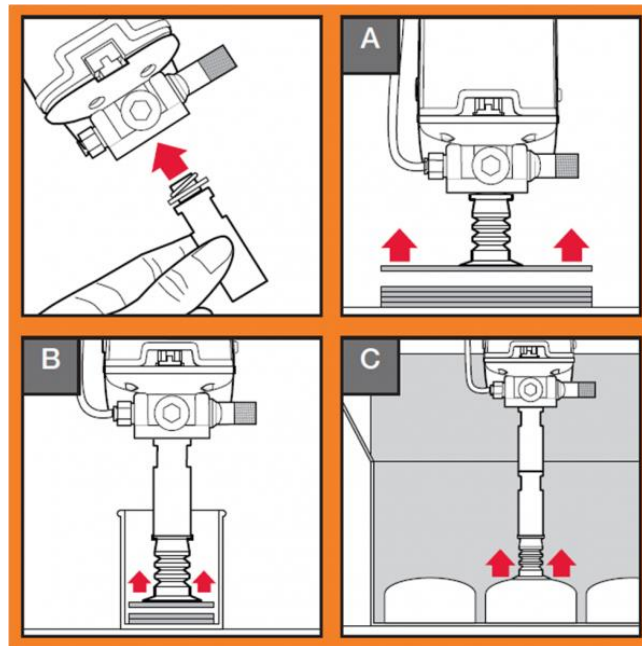


图 8 气动吸盘及其安装方式

## 2 Baxter 软件接口

### Baxter 应用程序接口

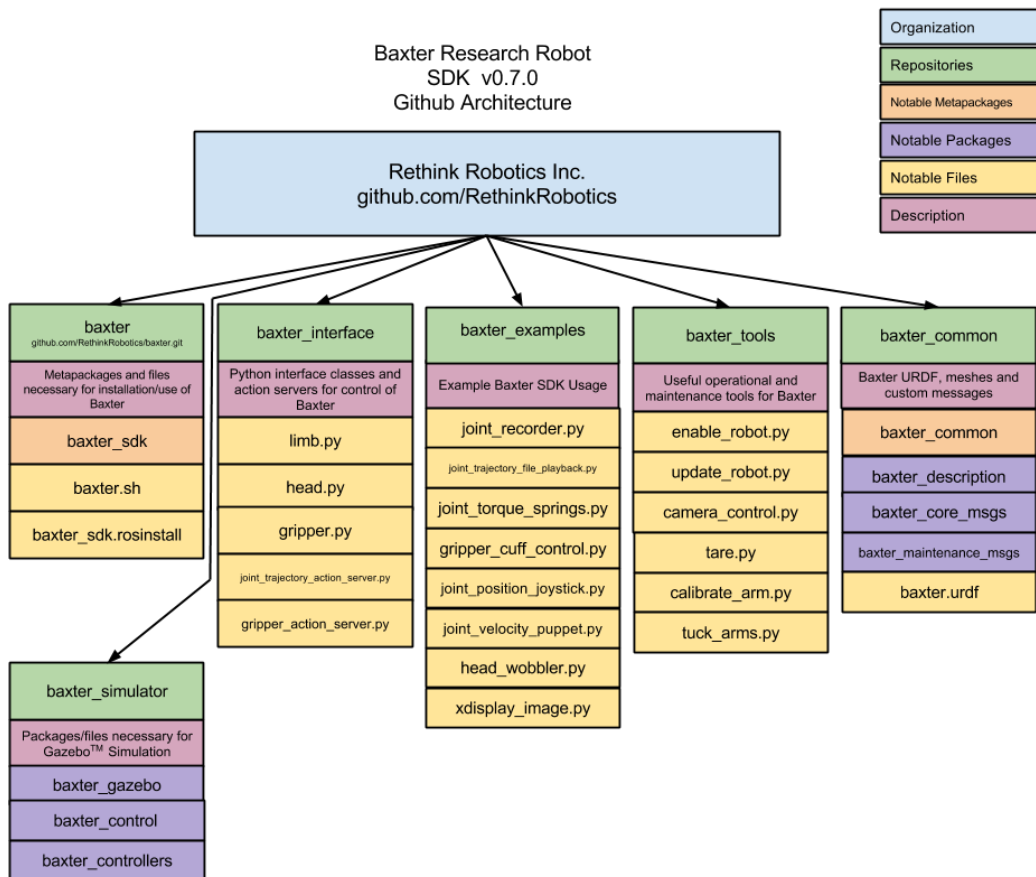


图 9 Baxter 应用程序接口布局图

#### baxter

该软件包包含用于安装 `baxtersdk` 所需的元数据包和文件，包括 `baxter_sdk` 元数据包及 SDK 包。也包括创建这个软件包所需要的 `rosinstall` 文件，`baxter.sh` 及用于设置 ROS 环境变量的 `bash` 脚本，旨在帮助用户轻松地设置自己的 ROS 环境并尽快开始使用 `baxter`。

#### baxter\_interface

用于控制 `baxter` 的 Python 接口类和 Action 服务器。这些 Python 模块直接封装了与 `baxter` 通讯所需要的 ROS 通讯功能，使用户能够简单的对 `baxter` 进行编程而不必事先熟悉 ROS。此软件包也包括用于控制机械臂运动轨迹和夹持器动作的 Action 服务器。这些标准的 Action 服务器可以和其他的 ROS 工具，比如和 MoveIt! 协同工作。

#### baxter\_tools

此软件包提供了一系列有用的用于的操作和维护 `baxter` 的工具软件。这些脚本可用于日常应用（如机器人开启/关闭，摄像头控制）和维护任务（去皮、校准、更新机器人软件）。



模块层级:

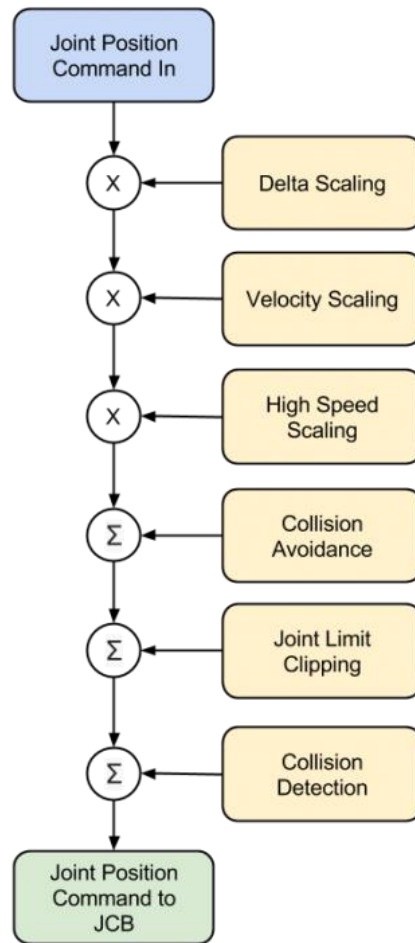
- [baxter\\_control](#)
  - [baxter\\_control.pid](#)
- [baxter\\_dataflow](#)
  - [baxter\\_dataflow.signals](#)
  - [baxter\\_dataflow.wait\\_for](#)
  - [baxter\\_dataflow.wait\\_for'](#)
  - [baxter\\_dataflow.weakrefset](#)
- [baxter\\_interface](#)
  - [baxter\\_interface.analog\\_io](#)
  - [baxter\\_interface.camera](#)
  - [baxter\\_interface.cfg](#)
    - [baxter\\_interface.cfg.GripperActionServerConfig](#)
    - [baxter\\_interface.cfg.HeadActionServerConfig](#)
    - [baxter\\_interface.cfg.PositionFFJointTrajectoryActionServerConfig](#)
    - [baxter\\_interface.cfg.PositionJointTrajectoryActionServerConfig](#)
    - [baxter\\_interface.cfg.VelocityJointTrajectoryActionServerConfig](#)
  - [baxter\\_interface.digital\\_io](#)
  - [baxter\\_interface.gripper](#)
  - [baxter\\_interface.head](#)
  - [baxter\\_interface.limb](#)
  - [baxter\\_interface.navigator](#)
  - [baxter\\_interface.robot\\_enable](#)
  - [baxter\\_interface.robust\\_controller](#)
  - [baxter\\_interface.settings](#)

详细函数和变量可参考: [http://api.rethinkrobotics.com/baxter\\_interface/html/frames.html](http://api.rethinkrobotics.com/baxter_interface/html/frames.html)。

### 3 关节控制方式

Baxter 机械臂采用 SEA (Series Elastic Actuator) 驱动, 每个关节都有专门的力矩传感器, 可以进行位置、速度、力矩控制。

## 位置控制模式



### Delta Scaling:

Scale setpoint based on which joint is going to take the longest to achieve. Allows all joints to arrive simultaneously.

### Velocity Scaling:

'Speed Ratio' describes the overall velocity scaling.

### High Speed Scaling:

High speed scaling reduces execution speed when commanded speed exceeds a high speed velocity threshold **and** the arm's high-speed collision links are in collision.

### Collision Avoidance:

Applies offsets to joint commands based on depth of intersection between arm collision geometries and the opposing arm or torso.

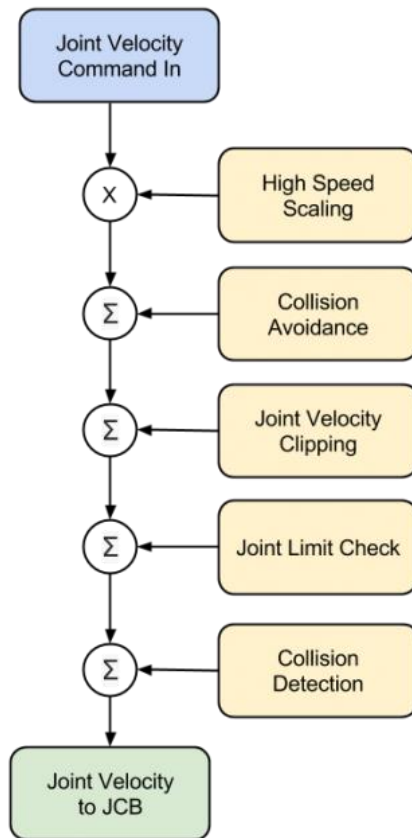
### Joint Limit Clipping:

If the joint command is beyond limits, clip the command to respect joint limits.

### Collision Detection:

If collision (impact) is detected, set position command to hold current compensating for the impact.

## 速度控制模式



**High Speed Scaling:**

High speed scaling reduces execution speed when commanded speed exceeds a high speed velocity threshold **and** the arm's high-speed collision links are in collision.

**Collision Avoidance:**

Applies offsets to joint commands based on depth of intersection between arm collision geometries and the opposing arm or torso.

**Joint Velocity Clipping:**

Limits joint velocity command to not exceed maximum joint velocities.

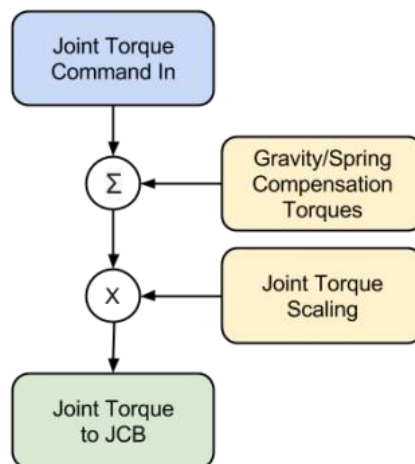
**Joint Limit Check:**

Validates that resulting joint position will be within joint limits. If not, no velocity will be commanded to any joint.

**Collision Detection:**

If collision (impact) is detected, set position command to hold current compensating for the impact.

力矩控制模式



**Gravity/Spring Compensation:**

The joint torque command is applied in addition to the gravity and S1 spring compensation torques.

**Joint Torque Scaling:**

Scales all joint torques if a torque command exceeds the maximum allowable torque for that joint. This scaling ratio is defined as  $\text{torque\_max} / \text{torque\_command}$ .