

Pinocchio 速查表

Pinocchio Library

1 快速开始

功能	代码
简单安装	<code>conda install -c conda-forge pinocchio</code>
导入库	<code>import pinocchio as pin</code>
文档	<code>from pinocchio.utils import *</code> <code>pin.Model?</code>

2 空间量

2.1 变换 (SE3)

功能	代码
SE3 变换	<code>aMb = pin.SE3(aRb, apb)</code>
单位变换	<code>M = pin.SE3(1) or pin.SE3.Identity()</code>
随机变换	<code>pin.SE3.Random()</code>
旋转矩阵	<code>M.rotation</code>
平移向量	<code>M.translation</code>
SE3 逆变换	<code>bMa = aMb.inverse()</code>
SE3 作用	<code>aMc = aMb * bMc</code>
作用矩阵	<code>aXb = aMb.action</code>
齐次矩阵	<code>aHb = aMb.homogeneous</code>
对数运算 SE3 → 6D	<code>pin.log(M)</code>
指数运算	<code>pin.exp(M)</code>

2.2 空间速度 (Motion)

功能	代码
运动向量	<code>m = pin.Motion(v, w)</code>
线性加速度	<code>m.linear</code>
角加速度	<code>m.angular</code>
SE3 作用	<code>v_a = aMb * v_b</code>

2.3 空间加速度

功能	代码
算法中使用	<code>a = (\dot{\omega}, \dot{v}_O)</code>

功能	代码
获取经典加速度	$\mathbf{a}' = \mathbf{a} + (0, \omega \times \mathbf{v}_O)$ <code>pin.classicAcceleration(v, a, [aMb])</code>

2.4 空间力 (Force)

功能	代码
力向量	<code>f = pin.Force(l, n)</code>
线性力	<code>f.linear</code>
扭矩	<code>f.angular</code>
SE3 作用	<code>f_a = aMb * f_b</code>

2.5 空间惯性 (Inertia)

功能	代码
惯性	<code>Y = pin.Inertia(mass, com, I)</code>
质量	<code>Y.mass</code>
质心位置	<code>Y.lever</code>
转动惯量	<code>Y.inertia</code>

2.6 几何

功能	代码
四元数	<code>quat = pin.Quaternion(R)</code>
轴角	<code>aa = pin.AngleAxis(angle, axis)</code>

2.7 有用的转换器

功能	代码
$\text{SE3} \rightarrow (x,y,z,\text{quat})$	<code>pin.se3ToXYZQUAT(M)</code>
$(x,y,z,\text{quat}) \rightarrow \text{SE3}$	<code>pin.XYZQUATTose3(vec)</code>

3 数据 (Data)

功能	代码
与模型相关的数据	<code>data = pin.Data(model)</code> <code>data = model.createData()</code>
关节数据	<code>data.joints</code>
关节/框架位置	<code>data.oMi / [data.oMf]</code>
关节速度	<code>data.v</code>
关节加速度	<code>data.a</code>
关节力	<code>data.f</code>

功能	代码
质量矩阵	data.M
非线性效应	data.nle
质心动量	data.hg
质心矩阵	data.Ag
质心惯性	data.Ig

4 模型 (Model)

功能	代码
运动树模型	model = pin.Model()
模型名称	model.name
关节名称	model.names
关节模型	model.joints
关节位置	model.placements
连杆惯性	model.inertias
框架	model.frames
位置变量数量	model.nq
速度变量数量	model.nv
方法	使用 ? 获取文档和输入参数
添加关节	model.addJoint
附加物体	model.appendBodyToJoint
添加框架	model.addFrame
将子模型附加到父模型	model.appendModel
构建简化模型	model.buildReducedModel

5 解析器

功能	代码
加载 URDF 文件	pin.buildModelFromUrdf(filename, [root_joint])
加载 SDF 文件	pin.buildModelFromSdf(filename, [root_joint], root.link.name, parent.guidance)

6 参考坐标系

6.1 坐标系 (CS)

坐标系	描述
WORLD	世界坐标系
LOCAL	关节的局部坐标系
LOCAL_WORLD_ALIGNED	世界轴对齐的局部坐标系

7 配置 (Configuration)

功能	代码
随机配置	pin.randomConfiguration(model, [lower_bound, upper_bound])
中性配置	pin.neutral(model)
归一化配置	pin.normalize(model, q)
配置差	pin.difference(model, q1, q2)
配置距离	pin.distance(model, q1, q2)
配置距离平方	pin.squareDistance(model, q1, q2)
插值配置	pin.interpolate(model, q1, q2, alpha)
积分配置	pin.integrate(model, q, v)
差分的偏导数	pin.dDifference(model, q1, q2, [arg_pos])
积分的偏导数	pin.dIntegrate(model, q, v, [arg_pos])

8 框架 (Frames)

功能	代码
所有操作框架的位置	pin.updateFramePlacements(model, data)
当前框架相对于原点的位置	data.oMf
框架速度	pin.getFrameVelocity(model, data, frame_id, ref_frame)
框架加速度	pin.getFrameAcceleration(model, data, frame_id, ref_frame)
框架经典加速度	pin.getFrameClassicalAcceleration(model, data, frame_id, ref_frame)
框架位置	pin.framesForwardKinematics(model, data, q)
框架雅可比	pin.computeFrameJacobian(model, data, q, frame_id, ref_frame)
框架雅可比时间变化	pin.frameJacobianTimeVariation(model, data, q, v, frame_id, ref_frame)
空间速度的偏导数	pin.getFrameVelocityDerivatives(model, data, frame_id, ref_frame)
空间速度的偏导数 (关节)	pin.getFrameVelocityDerivatives(model, data, joint_id, placement, ref_frame)
空间加速度的偏导数	pin.getFrameAccelerationDerivatives(model, data, frame_id, ref_frame)
空间加速度的偏导数 (关节)	pin.getFrameAccelerationDerivatives(model, data, joint_id, placement, ref_frame)

9 碰撞 (Collision)

功能	代码
更新碰撞对象位置	pin.updateGeometryPlacements(model, data, geometry_model, geometry_data, [q])
所有对的碰撞检测	pin.computeCollisions(model, data, geometry_model, geometry_data, q)
一对的碰撞检测	pin.computeCollisions(geometry_model, geometry_data, pair_index)
碰撞距离	pin.computeDistance(geometry_model, geometry_data, [pair_index])
每对的碰撞距离	pin.computeDistances([model, data], geometry_model, geometry_data, [q])
几何体积半径	pin.computeBodyRadius(model, geometry_model, geometry_data)
BroadPhase	pin.computeCollisions(broadphase_manager, callback) pin.computeCollisions(broadphase_manager, stop_at_first_collision)
+ 前向运动学更新几何位 置	pin.computeCollisions(model, data, broadphase_manager, q, stop_at_first_collision)

10 质心 (Center of Mass)

功能	代码
模型总质量	pin.computeTotalMass(model, [data])
每个子树的质量	pin.computeSubtreeMasses(model, data)
质心 (COM)	pin.centerOfMass(model, data, q, [v, a], [compute_subtree_com])
质心雅可比	pin.jacobianCenterOfMass(model, data, [q], [compute_subtree_com])

11 能量 (Energy)

功能	代码
前向运动学和动能	pin.computeKineticEnergy(model, data, [q, v])
前向运动学和势能	pin.computePotentialEnergy(model, data, [q, v])
前向运动学和机械能	pin.computeMechanicalEnergy(model, data, [q, v])

12 运动学 (Kinematics)

功能	代码
前向运动学 (FK)	pin.forwardKinematics(model, data, q, [v, [a]])
FK 导数	pin.computeForwardKinematicsDerivatives(model, data, q, v, a) WORLD> pin.getJointVelocityDerivatives(model, data, joint_id, pin.ReferenceFrame.WORLD) LOCAL> pin.getJointAccelerationDerivatives(model, data, joint_id, pin.ReferenceFrame.LOCAL)

13 雅可比 (Jacobian)

功能	代码
完整模型雅可比 → data.J	pin.computeJointJacobians(model, data, [q])
关节雅可比	pin.getJointJacobian(model, data, joint_id, ref_frame)
完整模型 dJ/dt	pin.computeJointJacobiansTimeVariation(model, data, q, v)
关节 dJ/dt	pin.getJointJacobianTimeVariation(model, data, joint_id, ref_frame)

14 正向动力学 (Forward Dynamics)

功能	代码
关节体算法 \ddot{q}	pin.aba(model, data, q, v, tau, [f_ext])
关节空间惯性矩阵逆	pin.computeMinverse(model, data, [q])
复合刚体算法	pin.crba(model, data, q)

15 逆向动力学 (Inverse Dynamics)

功能	代码
递归牛顿-欧拉算法	pin.rnea(model, data, q, v, a, [f_ext])
广义重力	pin.computeGeneralizedGravity(model, data, q)
dtau_dq, dtau_dv, dtau_da	pin.computeRNEDerivatives(model, data, q, v, a, [f_ext])

16 质心动力学 (Centroidal)

功能	代码
质心动量	pin.computeCentroidalMomentum(model, data, [q, v])
质心动量 + 时间导数	pin.computeCentroidalMomentumTimeVariation(model, data, [q, v, a])

17 通用 (General)

功能	代码
所有项 (查看文档)	pin.computeAllTerms(model, data, q, v)

18 回归器 (Regressor)

18.1 运动学回归器

功能	代码
运动学回归器 (关节)	pin.computeJointKinematicRegressor(model, data, joint_id, ref_frame, [placement])
运动学回归器 (框架)	pin.computeFrameKinematicRegressor(model, data, frame_id, ref_frame)

18.2 回归器

功能	代码
静态回归器	pin.computeStaticRegressor(model, data, q)
物体回归器	pin.bodyRegressor(velocity, acceleration)
附加到关节的物体回归器	pin.jointBodyRegressor(model, data, joint_id)
附加到框架的物体回归器	pin.frameBodyRegressor(model, data, frame_id)
关节扭矩回归器	pin.computeJointTorqueRegressor(model, data, q, v, a)

19 接触雅可比 (Contact Jacobian)

功能	代码
约束模型的运动学雅可比	pin.getConstraintJacobian(model, data, contact_model, contact_data)
约束模型集的运动学雅可比	pin.getConstraintJacobian(model, data, contact_models, contact_datas)

20 接触动力学 (Contact Dynamics)

功能	代码
带接触的约束动力学	pin.forwardDynamics(model, data, [q, v], tau, constraint_jacobian, constraint_drift, damping)
带接触的冲击动力学	pin.impulseDynamics(model, data, [q], v_before, constraint_jacobian, restitution_coefficient, damping)
约束矩阵的逆	pin.computeKKTContactDynamicMatrixInverse(model, data, q, constraint_jac, damping)

21 约束动力学 (Constraint Dynamics)

功能	代码
分配内存	pin.initConstraintDynamics(model, data, contact_models)
带接触约束的正向动力学	pin.constraintDynamics(model, data, q, v, tau, contact_models, contact_datas, [prox_settings])
带运动学约束的正向动力学导数	pin.computeConstraintDynamicsDerivatives(model, data, contact_models, contact_datas, prox_settings)

22 冲击动力学 (Impulse Dynamics)

功能	代码
带接触约束的冲击动力学	pin.impulseDynamics(model, data, q, v, contact_models, contact_datas, r_coeff, mu)
冲击动力学导数	pin.computeImpulseDynamicsDerivatives(model, data, contact_models, contact_datas, r_coeff, prox_settings)

23 Cholesky 分解

功能	代码
关节空间惯性矩阵的 Cholesky 分解	pin.cholesky.decompose(model, data)
求解 $Mx = y$ 中的 x	pin.cholesky.solve(model, data, v)
关节空间惯性矩阵的逆	pin.cholesky.computeMinv(model, data)

24 可视化器 (Viewer)

功能	代码
创建可视化器	<code>mv = pin.visualize.MeshcatVisualizer</code>
加载模型	<code>viz = mv(model, collision_model,</code> <code>visual_model)</code>
初始化	<code>viz.initViewer(loadModel=True)</code>
显示	<code>viz.display(q)</code>
添加基本形状	
球体	<code>viz.viewer[name].set_object(meshcat.geometry.Sphere(size),</code> <code>material)</code>
盒子	<code>viz.viewer[name].set_object(meshcat.geometry.Box([sizex,</code> <code>sizey, sizez]), material)</code>
更改几何体 [name] 的位置	<code>viz.viewer[name].set_transform(meshcat.transform(xyzquat_placement))</code>