

学习pinocchio (三) multibody文件夹

知 zhuanlan.zhihu.com/p/698934448

©1993-2025 G000

尽心用心
一切都是命运石之门的选择

7 人赞同了该文章

☰
目录

该文件夹内的源码主要是多刚体动力学的结构、状态等运算操作。其包含三个子文件夹：joint、liegroup和visitor。

joint文件夹：集成不同关节类型的运动和算法。

liegroup文件夹：集成李群的各种运算和算法。

visitor文件夹：集成一些访问操作。

3.1 constraint-base.hpp

该源码主要建立约束模型的基类。

(1) ConstraintBase类

成员对象：

成员函数：

`l se3Action`：约束的位姿伴随变换，物体坐标系变向绝对坐标系。

`l se3ActionInverse`：约束的位姿逆伴随变换，绝对坐标系变向物体坐标系。

`l motionAction`：约束的运动伴随变换，六维运动矢量的6x6伴随运算。

`l rows`：约束空间为6。

`l nv`和`cols`：约束个数。

3.2 constraint-generic.hpp

该源码主要构件一般约束的类，继承于`ConstraintBase`。

(1) ConstraintTpl类

额外定义的常量和结构体：

`l NV`：约束的对偶空间+维度。

`l Transpose`：定义了约束矩阵与力的乘法准则：约束矩阵转置乘以六维力矢量。

成员对象：

`I S`：力的约束矩阵，维度为 $6 \times NV$ 。

成员函数：

`I se3Action`：约束的位姿伴随变换，物体坐标系变向绝对坐标系。

`I se3ActionInverse`：约束的位姿逆伴随变换，绝对坐标系变向物体坐标系。

`I motionAction`：约束的运动伴随变换，六维运动矢量的 6×6 伴随运算。

`I matrix_impl`：生成力的约束矩阵。

3.3 constraint.hpp

该源码主要对约束结构体的简化命名。

3.4 data.hpp/hxx

该源码主要是整个多刚体运动过程的状态数据变量和其操作。

(1) DataTpl类

成员对象：

`I joint`：所有关节状态数据，关节数据为JointDataTpl类。

`I a`：所有关节在关节坐标系下的空间加速度。

`I oa`：所有关节在世界坐标系下的空间加速度。

`I a_gf`：由重力加速度引起的所有关节在关节坐标系下的空间加速度。

`I oa_gf`：所有关节在世界坐标系下的空间角加速度（包含重力）。

`I v`：所有关节在关节坐标系下的空间速度。

`I ov`：所有关节在世界坐标系下的空间速度。

`I f`：所有关节在关节坐标系下的空间力（包括所有外力）。

`I of`：所有关节在世界坐标系下的空间速度（包括所有外力）。

`I h`：所有关节在关节坐标系下的空间动量。

`I oh`：所有关节在世界坐标系下的空间动量。

`I liMi`：所有关节相对于父类关节的位姿。

`I oMi`：所有关节在世界坐标系下的位姿。

`I tau`：所有关节力矩，维度为对应关节模型 nv 。

`I nle`：系统动力学的非线性项，包括科氏力 $\omega \times$ ，向心力和重力项。

I_g : 系统动力学+的重力项。

I_{oMf} : 系统动力学的所有关节重力项。

I_{Ycrb} : 所有关节包含其子树在其关节坐标系下的空间惯性矩阵。

I_{dYcrb} : 所有关节包含其子树在其关节坐标系下的空间惯性矩阵的一阶微分。

I_M : 系统动力学关节空间惯性矩阵, 模型 nv 大小的方阵。

I_{Minv} : 系统动力学关节空间惯性矩阵的逆, 模型 nv 大小的方阵。

I_C : 系统动力学科氏力, 模型 nv 大小的方阵。

I_{dHdq} : 系统空间动量相对于关节空间的微分, $6 \times nv$ 矩阵。

I_{dFdq} : 系统空间空间力相对于关节构型空间的微分, $6 \times nv$ 矩阵。

I_{dFdv} : 系统空间动量相对于关节速度空间的微分, $6 \times nv$ 矩阵。

I_{dFda} : 系统空间动量相对于关节加速度空间的微分, $6 \times nv$ 矩阵。

I_{SDinv} : 系统ABA算法S约束与D矩阵逆的乘积, $6 \times nv$ 矩阵。

I_{UDinv} : 系统ABA算法U与D矩阵逆的乘积, $6 \times nv$ 矩阵。

I_{IS} : 系统惯量I与ABA算法S约束的乘积, $6 \times nv$ 矩阵。

I_{vxl} : 所有关节惯量I的右微分, $\frac{dI}{dq}$ 。

I_{lvx} : 所有关节惯量I的左微分, $\frac{dI}{dq}$ 。

I_B : 所有关节惯量I的全微分, $\frac{dI}{dq}$ 。

$I_{oinertias}$: 在世界坐标系下所有关节的惯量。

I_{oYcrb} : 所有关节包含其子树在世界坐标系下的空间惯性矩阵。

I_{doYcrb} : 所有关节包含其子树在世界坐标系下的空间惯性矩阵的一阶微分。

I_{ltmp} : 系统的空间惯性矩阵 (微分算法临时变量)。

I_{M6tmp} : 系统的运动 (微分算法临时变量)。

I_{M6tmpR} : 系统的运动 (微分算法临时变量)。

$I_{M6tmpR2}$: 系统的运动 (微分算法临时变量)。

I_{ddq} : ABA算法的关节加速度。

I_{Yaba} : ABA算法的所有子树的稠密矩阵+。

I_u : ABA算法的对应施加力矩的中间变量。

I_{Ag} : 质心动力学+的质心惯量矩阵,
$$h_g = A_g q \cdot$$

| dAg: 质心动力学的质心惯量矩阵的一阶微分,

$$h_g = A_g q + A_g q.$$

| hg: 质心动力学的系统动量（线动量和角动量），

$$h_g = \begin{pmatrix} mc & L_g \end{pmatrix}.$$

| dhg: 质心动力学的系统动量一阶微分,

$$h_g = \begin{pmatrix} mc & L_g \end{pmatrix}.$$

| Ig: 质心动力学的多刚体惯性矩阵,

$$h_g = I_g v_{mean}.$$

| Fcrb: 所有空间力的集合，用于CRBA和CCRBA。

| lastChild: 所有子类编号，用于CRBA。

| nvSubtree: 所有子类树的维度，用于CRBA。

| start_idx_v_fromRow: 所有关节运动在系统运动的开始序号。

| end_idx_v_fromRow: 所有关节运动在系统运动的结束序号。

| U: 关节空间惯性矩阵平方的Cholesky分解上三角矩阵。

| D: U矩阵的对角元素组成的对角矩阵。

| Dinv: D矩阵逆。

| tmp: Cholesky分解的临时变量。

| parents_fromRow: M矩阵的第一个非零元素（用于Cholesky分解）。

| supports_fromRow: 所有支撑树每一行对应的列表序列编号，有助于稀疏矩阵检索。

| nvSubtree_fromRow: 所有子类树的当前行的序号，用于Cholesky分解。

| J: 空间速度对关节的雅克比矩阵，每一列表示关节在基础空间速度的映射。若需要精准映射调用pinocchio::getJointJacobian。

| dJ: 空间速度对关节的雅克比矩阵关于时间的一阶微分。

| ddJ: 空间速度对关节的雅克比矩阵关于时间的二阶微分。

| psid: 空间角速度叉乘空间速度对关节的雅克比矩阵（伴随变换）。

| psidd: 空间加速度叉乘空间速度对关节的雅克比矩阵（伴随变换）。

| dVdq: 空间速度对关节空间的偏微分。

| dAdq: 空间加速度对关节空间的偏微分。

| dAdv: 空间速度对关节速度的偏微分。

| dtau_dq: 关节力对关节空间的偏微分。

| dtau_dv: 关节力对关节速度的偏微分。

| ddq_dq: 关节加速度对关节空间的偏微分。

| ddq_dv: 关节加速度对关节速度的偏微分。

| iMf: 所有关节位姿空间相对于算法的末端位姿。

| com: 所有关节作为根的树状多刚体的质心在各自关节的位置。

| vcom: 所有关节作为根的树状多刚体在各自关节的线速度。

| acom: 所有关节作为根的树状多刚体在各自关节的线加速度。

| mass: 所有关节作为根的树状多刚体的质量。

| Jcom: 质心（三维）对关节空间的雅可比矩阵, $v_{com} = J_{com} q$ 。

| kinetic_energy: 系统动能。

| potential_energy: 系统势能。

| JMinvJt: 操作空间惯性矩阵的逆, $J M^{-1} J$ 。

| llt_JMinvJt: JMinvJt的LLT分解。

| lambda_c: 相对于接触力的拉格朗日乘子, 参考pinocchio::forwardDynamics。

| sDUiJt: 临时变量, $\sqrt{D U^{-1} J^T}$ 。

| torque_residual: 临时变量, $\tau - b(q, q)$ 。

| dq_after: 冲击后的空间关节速度。

| impulse_c: 相对于接触冲击的拉格朗日乘子, 参考pinocchio::impulseDynamics。

| staticRegressor: 静态回归关联矩阵。

| bodyRegressor: 机身回归关联矩阵。

| jointTorqueRegressor: 关节力矩回归关联矩阵。

| kinematic_hessians: 包含所有关节运动学Hessian矩阵的张量。

| d2tau_dqdq: 关节力矢量对关节空间SO二次偏微分。

| d2tau_dvdv: 关节力矢量对关节速度SO二次偏微分。

| d2tau_dqdv: 关节力矢量对关节空间SO偏微分和关节速度SO偏微分。

| d2tau_dadq: 关节力矢量对关节加速度SO偏微分和关节空间SO偏微分。

3.5 fcl.hpp/hxx

该源码主要是构建fcl的碰撞模型描述。

(1) CollisionPair结构体

两个几何结构体的成对描述。

(2) FakeCollisionGeometry结构体

(3) AABB结构体

成员对象：

`I_min`：最小值0。

`I_max`：最大值1。

(4) GeometryType枚举

`I_VISUAL`：几何显示。

`I_COLLISION`：几何碰撞。

(5) GeometryNoMaterial结构体

(6) GeometryPhongMaterial结构体

成员对象：

`I_meshEmissionColor`：材料环境颜色RGBA。

`I_meshSpecularColor`：材料反射颜色RGBA。

`I_meshShininess`：反射亮度（范围0到1）。

(7) GeometryObject结构体

成员对象：

`I_name`：几何体名称。

`I_parentFrame`：父类索引序号。

`I_parentJoint`：父类关节序号。

`I_geometry`：有关FCL的碰撞几何体。

`I_CollisionGeometryPtr`：FCL的碰撞指针。

`I_placement`：几何体相对于父类关节坐标系位姿。

`I_meshPath`：几何体mesh的绝对路径。

`I_meshScale`：几何体mesh放缩比例。

`I_overrideMaterial`：几何体是否覆盖原有材料。

| meshColor: 几何体RCBA颜色。

| mechMaterial: 几何体材料, GemetryMaterial结构体。

| meshTexturePath: 几何体mesh texture文件路径。

| disableCollision: 几何体碰撞是否开启。

成员函数:

3.6 force-set.hpp

该源码主要构件空间力的集合, 内部有block的优化运算。

(1) ForceSetTpl类

成员对象:

| size: 空间力个数

| m_f: 空间力的3xsize力矩阵

| m_f: 空间力的3xsize力矩阵

成员函数:

| se3Action: 六维空间力从物体坐标系转换成绝对坐标系, m为位姿矩阵,

| se3ActionInverse: 六维运动矢量从绝对坐标系转换成物体坐标系, m为位姿矩阵,

3.7 frame.hpp

该源码主要定义导入多刚体数据类型的基本信息。

(1) FrameType枚举

OP_FRAME: 可操作数据类型

JOINT: 关节数据类型

FIXED_JOINT: 固定关节数据类型

BODY: 机身数据类型

SENSOR: 传感器数据类型

(2) FrameTpl类

该类主要是多刚体机器人结构树状图内表示父子关系的位姿坐标刚体和关节。

成员对象:

① name: 数据名称

- ② parent : 父类关节数据编号
- ③ previousFrame : 上一组数据编号
- ④ placement : 相对与父类关节的位姿状态
- ⑤ type : 数据类型

成员函数:

3.8 fwd.hpp

该源码主要是对整个文件夹包含文件的命名简化。

3.9 geometry.hpp/hxx

主要用于描述刚体成对碰撞的几何特征描述。

(1) GeometryModel结构体

成员对象:

l_ngeoms: 几何体的个数

l_geometryObjects: 用于碰撞计算的几何体矢量

l_collisionPairs: 成对碰撞的矢量

成员函数:

l_addGeometryObject: 将一个几何体添加到geometryObjects中, 设置其父类关节。

l_removeGeometryObject: 移除几何体 (根据名称)。

l_getGeometryId: 根据几何名称, 获取其序号。

l_existGeometryName: 检查是否有该名称的几何体。

l_addCollisionPair: 添加碰撞成对。

l_addAllCollisionPair: 设置所有的刚体成对碰撞 (除了相同父类关节)。

l_setCollisionPairs: 设置碰撞成对。

l_removeCollisionPairs: 删除碰撞成对。

l_removeAllCollisionPairs: 删除所有碰撞成对。

l_existCollisionPairs: 检查是否存在碰撞成对。

l_findCollisionPairs: 查询碰撞成对。

(2) GeometryData

成员对象:

`l oMg`: 所有碰撞体相对于世界的位姿。

`l activeCollisionPairs`: 碰撞对是否使用。

`l innerObjects`: 几何体和关节序列关联列表。

`l outerObejcts`: 几何体碰撞和关节序列关联列表。

成员函数:

`l fillInnerOuterObejctsMaps`: 填充所有关联列表。

`l activateCollisionPair`: 激活对应碰撞。

`l activateAllCollisionPair`: 激活所有碰撞。

`l setActivecollisionPairs`: 设置碰撞成对。

`l setGeometryCollisionStatus`: 设置碰撞成对激活状态。

`l deactivateCollisionPair`: 关闭对应碰撞。

`l deactivateAllCollisionPair`: 关闭所有碰撞。

3.10 model.hpp/hxx

该源码主要构件多刚体动力学的模型参数。

(1) ModelTpl类

成员变量

`l ng`: 系统的全自由度维度

`l nv`: 系统的全自由度速度矢量维度

`l njoints`: 系统的关节个数 (第一个关节为默认universe, 类型为FIXED_JOINT)

`l nbodies`: 系统的刚体个数

`l nframes`: 可操作数据个数

`l inertias`: 所有关节的所有刚体在关节坐标系下的空间惯性矩阵之和 (不变量), 大小为njoints (第一个默认为零)

`l jointPlacements`: 所有关节相对于父类关节的初始位姿状态, 大小为njoints (第一个关节默认位姿为单位初始值)

`l joints`: 所有关节模型的关节特性 (每个关节为JointModelTpl类), 大小为njoints (第一个关节比较特殊)

`l idx_gx`: 所有关节所对应自由度的序号, 大小为njoints (第一个为零)

`l nqs`: 所有关节所对应自由度的个数, 大小为njoints (第一个为零)

`l idx_vx`: 所有关节所对应速度的序号, 大小为njoints (第一个为零)

`l parents`: 所有关节所对应父类关节的序号, 大小为njoints (第一个为零)

`I names`: 所有关节名称, 大小为`njoints` (第一个为universe)

`I referenceConfigurations`: 所有关节名称和序号对应列表, 大小为`njoints` (第一个为universe)

`I rotorInertia`: 所有自由度的传动惯性参数, 大小为`ng`

`I rotorGearRatio`: 所有自由度的传动系数, 大小为`ng`

`I friction`: 所有自由度的摩擦系数, 大小为`ng`

`I damping`: 所有自由度的阻尼系数, 大小为`ng`

`I effortLimit`: 所有自由度的力限制的大小值, 大小为`ng`

`I velocityLimit`: 所有自由度的速度限制的大小值, 大小为`ng`

`I lowerPositionLimit`: 所有自由度的下限值, 大小为`ng`

`I upperPositionLimit`: 所有自由度的上限值, 大小为`ng`

`I frames`: 可操作数据内容, 大小为`nframes`

`I supports`: 所有关节的所有上游关节编号, 大小为`njoints`

`I subtrees`: 所有关节的所有下游关节编号, 大小为`njoints`

`I gravity`: 重力加速度的六维运动形式

`I name`: 模型的名字

成员函数:

`I addJoint`: 将关节添加到具有无限边界的运动树当中。

`I addJointFrame`: 将关节添加到框架运动树当中。

`I appendBodyToJoint`: 将刚体的空间惯性矩阵增加到对应随动的关节上, 叠加形式, 坐标变换。

`I addBodyFrame`: 将刚体添加到框架运动树当中。

`I getBodyId`: 获取刚体在运动树当中的序号。

`I existBodyName`: 检查刚体名称在运动树中是否存在。

`I getJointId`: 获取关节在运动树当中的序号。

`I existJointName`: 检查关节名称在运动树中是否存在。

`I getFrameId`: 获取框架 (关节、刚体和感知等) 在运动树当中的序号。

`I existFrame`: 获取框架 (关节、刚体和感知等) 名称在运动树中是否存在。

`I addFrame`: 将框架 (关节、刚体和感知等) 添加到框架运动树当中。

`I check`: 检查模型是否有效, 由派生类`checkModel`实现。

`I hasConfigurationLimit`: 检查关节是否有约束。

I hasConfigurationLimitInTangent: 检查关节切空间是否有约束。

I addJointIndexToParentSubtrees: 将关节序号添加到其父类子树上。

(2) FilterFrame结构体

该结构体主要适配框架的类型和名称。

成员对象:

I name: 名称。

I typeMask: 类型, 枚举类型FrameType。

3.11 visitor.hpp

该源码主要为了包含visitor文件内容。

3.12 joint文件夹fwd.hpp

该源码主要是对整个文件夹包含文件的命名简化。

3.13 joint文件夹joint-base.hpp

该源码主要是一个文件包含作用。

3.14 joint文件夹joint-basic-visitor.hpp/hxx

该源码主要是提供关节模型和关节数据之间的操作内联函数, 一部分是关于关节模型的函数, 另一部分是关于关节数据的函数。

(1) createData函数

构建新的关节模型数据JointModelTpl。

(2) calc_zero_order函数

通过对应的关节模型JointModelTpl读取所要更新的零阶(位姿)数据JointDataTpl。

实现里里面构造了一个结构体JointCalcZeroOrderVisitor, 采用静态函数。

(3) calc_first_order函数

通过对应的关节模型JointModelTpl读取所要更新的一阶(位姿和速度)数据JointDataTpl。

实现里面构造了一个结构体JointCalcFirstOrderVisitor, 采用静态函数。

(4) calc_aba函数

通过对应的关节模型JointModelTpl计算ABA算法的U, D内置矩阵。

实现里面构造了一个结构体JointCalcAbaVisitor, 采用静态函数。

(5) nv函数

通过对应的关节模型`JointModelTpl`读取关节切空间维度（比如旋转副是1）。

实现里面构造了一个结构体`JointNvVisitor`，采用静态函数。

(6) nq函数

通过对应的关节模型`JointModelTpl`读取关节构型维度（比如旋转副是1）。

实现里面构造了一个结构体`JointNqVisitor`，采用静态函数。

(7) hasConfigurationLimit函数

通过对应的关节模型`JointModelTpl`读取关节构型空间是否有约束，`nq`维布尔类型数组。

实现里面构造了一个结构体`JointConfigurationLimitVisitor`，采用静态函数。

(8) hasConfigurationLimitInTangent函数

通过对应的关节模型`JointModelTpl`读取关节切空间是否有约束，`nv`维布尔类型数组。

实现里面构造了一个结构体`JointConfigurationLimitInTangentVisitor`，采用静态函数。

(9) idx_q函数

通过对应的关节模型`JointModelTpl`读取关节第一个自由度在全刚体构型空间的序号。

实现里面构造了一个结构体`JointIdxQVisitor`，采用静态函数。

(10) idx_v函数

通过对应的关节模型`JointModelTpl`读取关节第一个切空间在全刚体切空间的序号。

实现里面构造了一个结构体`JointIdxVVisitor`，采用静态函数。

(11) id函数

通过对应的关节模型`JointModelTpl`读取关节在多刚体运动学的编号。

实现里面构造了一个结构体`JointIdVisitor`，采用静态函数。

(12) setIndexes函数

设置关节模型`JointModelTpl`的编号、关节第一个自由度和切空间在多刚体运动学的序号。

实现里面构造了一个结构体`JointSetIndexesVisitor`，采用静态函数。

(13) shortname函数

设置关节模型`JointModelTpl`的短名称。

实现里面构造了一个结构体`JointShortnameVisitor`，采用静态函数。

(14) cast_joint函数

从已有的关节模型`JointModelTpl`的构造新的关节模型。

实现里面构造了一个结构体`JointCastVisitor`，采用静态函数。

(15) isEqual函数

对比两个关节模型`JointModelTpl`是否相同。

实现里面构造了一个结构体`JointModelComparisonOperatorVisitor`，采用静态函数。

(16) hasSameIndexes函数

对比两个关节模型`JointModelTpl`是否索引。

实现里面构造了一个结构体`JointModelHasSameIndexesVisitor`，采用静态函数。

(17) constraint_xd函数

通过关节数据`JointDataTpl`获取关约束。

实现里面构造了一个结构体`JointConstraintVisitor`，采用静态函数。

(18) joint_transform函数

通过关节数据`JointDataTpl`获取相对于父类的位姿。

实现里面构造了一个结构体`JointTransformVisitor`，采用静态函数。

(19) motion函数

通过关节数据`JointDataTpl`将其内部运动数据集成。

实现里面构造了一个结构体`JointMotionVisitor`，采用静态函数。

(20) bias函数

通过关节数据`JointDataTpl`获取其偏置。

实现里面构造了一个结构体`JointBiasVisitor`，采用静态函数。

(21) u_inertia函数

通过关节数据`JointDataTpl`获取其惯性矩阵的U矩阵。

实现里面构造了一个结构体`JointUInertiaVisitor`，采用静态函数。

(22) dinv_inertia函数

通过关节数据`JointDataTpl`获取其惯性矩阵的D矩阵的逆。

实现里面构造了一个结构体`JointDInvInertiaVisitor`，采用静态函数。

(23) udinv_inertia函数

通过关节数据`JointDataTpl`获取其惯性矩阵的U矩阵和D矩阵逆的乘积。

实现里面构造了一个结构体`JointUDInvInertiaVisitor`，采用静态函数。

(24) isEqual函数

对比两个关节数据`JointDataTpl`是否相同。

实现里面构造了一个结构体`JointDataComparisonOperatorVisitor`，采用静态函数。

3.15 joint文件夹joint-collection.hpp

该源码主要定义一个默认的关节集合默认模版。

(1) JointCollectionDefaultTpl类

该类主要是一个基类，简化了一些类内命名，无实际操作内容。

3.16 joint文件夹joint-common-operations.hpp

该源码主要是定义了一些关节共有操作。

(1) PerformStYsInversion结构体

匹配参数求解 线性方程组，采用Eigen的LLT函数（要求D为正定矩阵，惯性矩阵符合，变相快速求解矩阵D的逆）

(2) LinearAffineTransform结构体

配置矢量空间的标量线性仿射变换，。

3.17 joint文件夹joint-composite.hpp/hxx

该源码主要是关节集合的默认模板，没有实际操作。

(1) JointCollectionDefaultTpl类

关节集合的默认模板，简化类内的命名，无实际成员对象和函数

3.18 joint文件夹joint-data-base.hpp

该源码主要是构件关节数据的基类，包括关节的运动参数。

(1) JointDataBase结构体

没有成员对象，定义了一系列接口，接口函数有

`I S`：获取关节力约束矩阵。

`I M`：获取关节相对于父类的位姿。

`I_v`: 获取关节运动速度。

`I_q`: 获取关节的偏置。

`I_U`: 获取关节的ABA算法惯性矩阵的U矩阵 $U = MS$ 。

`I_D_t`: 获取关节ABA算法惯性矩阵的D矩阵的逆 $D = S^T U$ 。

`I_UD_t`: 获取关节ABA算法惯性矩阵的U矩阵和D矩阵逆的乘积。

3.19 joint文件夹joint-generic.hpp

该源码主要构件关节模型和关节数据的通用基类，可以形成关节模型集合。

(1) JointDataTpl类

继承与JointDataTpl类和JointCollectionTpl，接口基本与JointDataBase一致。

(2) JointModelTpl类

继承于JointModeBase和JointCollectionTpl，接口基本与JointModeBase一致。

3.20 joint文件夹joint-model-base.hpp

该源码主要构件关节模型的基类，包括关节模型基本特征。

(1) JointModelBase类

关节模型的基类，实现均在派生类后缀_impl中。

额外定义的常量：

`I_NQ`: 关节构型空间的大小

`I_NV`: 关节速度空间的大小

成员对象：

`I_id`: 关节在多刚体系统的编号

`I_i_q`: 关节构型空间的第一个自由度在多刚体系统的构型空间序号

`I_i_v`: 关节速度空间的第一个速度在多刚体系统的速度空间序号

成员函数：

`I_createData`: 构建新的关节模型数据。

`I_hasConfigurationLimit`: 关节构型空间是否有约束，`NQ`维布尔类型数组。

`I_hasConfigurationLimitInTangent`: 关节切空间是否有约束，`NV`维布尔类型数组。

`I_calc`: 根据匹配参数计算关节的位姿和速度。

`I calc_aba`: 根据匹配参数计算ABA算法的U, D内置矩阵。

`I setIndexes`: 设置三个成员变量。

`I jointConfigSelector`: 从构型矩阵中提取对应构型空间矩阵

`I jointVelocitySelector`: 从速度矩阵中提取对应速度空间矩阵

`I jointCols`: 从矩阵中提取对应速度空间列

`I jointRows`: 从矩阵中提取对应速度空间行

`I jointBlock`: 从矩阵中提取对应速度空间块

3.21 joint文件夹各种具体关节hpp

根据关节不同, 设计针对的关节运动模板、约束模板、关节模型和关节数据:

`I joint-planar.hpp`: 平面运动副, 构型空间为4, 3个自由度。

`I joint-prismatic-unaligned.hpp`: 平动副 (空间), 构型空间为1, 1个自由度。

`I joint-prismatic.hpp`: 平动副 (XYZ其中一个轴), 构型空间为1, 1个自由度。

`I joint-revolute-unaligned.hpp`: 转动副 (空间, 有边界), 构型空间,1, 1个自由度。

`I joint-revolute.hpp`: 转动副 (XYZ其中一个轴, 有边界), 构型空间为1, 1个自由度。

`I joint-revolute-unbounded-unaligned.hpp`: 转动副 (空间, 无边界), 构型空间为2, 1个自由度。

`I joint-revolute-unbounded.hpp`: 转动副 (XYZ其中一个轴, 无边界), 构型空间为2, 1个自由度。

`I joint-spherical-ZYX.hpp`: 球副 (XYZ欧拉角描述), 构型空间为3, 3个自由度。

`I joint-spherical.hpp`: 球副 (四元数描述), 构型空间为4, 3个自由度。

`I joint-translation.hpp`: 三维移动副, 构型空间为3, 3个自由度。

`I joint-free-fly.hpp`: 空间自由运动, 构型空间为7 (四元数), 6个自由度。

由于代码内容类似, 简单介绍类似内容, 用&代替关节名称。

(1) Motion&Tpl结构体

继承于MotionBase

额外定义的常量和结构体:

`I NQ`: 关节构型空间。

`I NV`: 关节速度空间。

成员对象:

`I` 根据数据类型定义。

成员函数：

`se3Action_impl`：运动矢量从物体坐标系转换成绝对坐标系。

`se3ActionInverse_impl`：运动矢量从绝对坐标系转换成物体坐标系。

`motionAction`：运动矢量的运动伴随运算。

(2) Constraint&Tpl结构体

继承于ConstraintBase

额外定义的常量和结构体：

`NV`，关节速度空间。

`ConstraintTranspose`：定义了约束矩阵与力的乘法准则。

成员函数：

`se3Action_impl`：约束矢量从物体坐标系转换成绝对坐标系。

`se3ActionInverse_impl`：约束矢量从绝对坐标系转换成物体坐标系。

`motionAction`：约束矢量的运动伴随运算。

`matrix_impl`：生成约束力矩阵。

(3) JointData&Tpl结构体

继承于JointDataBase

成员对象：

`S`：关节的驱动力约束结构，Constraint&Tpl。

`M`：关节的位姿结构，SE3Tpl。

`v`：关节的运动结构，Motion&Tpl。

`q`：关节的运动偏置，MotionZeroTpl。

`U`：关节的惯性矩阵的U矩阵 $U = MS$ 。

`Dinv`：关节惯性矩阵的D矩阵的逆 $D = S^T U$ 。

`UDInv`：关节惯性矩阵的U矩阵和D矩阵逆的乘积。

`StU`：关节惯性矩阵的 $D = S^T U$ 。

(4) JointModel&Tpl结构体

继承于JointModelBase

成员对象：

成员函数：

| createData：构建新的关节模型数据。

| hasConfigurationLimit：关节构型空间是否有约束。

| hasConfigurationLimitInTangent：关节切空间是否有约束。

| calc：根据匹配参数计算关节的位姿和速度。

| calc_aba：根据匹配参数计算ABA算法的U，D内置矩阵。

| shortname和classname：名称为JointModel&。

3.22 joint文件夹joint-mimic.hpp

该源码主要是将各种具体关节hpp进行模仿克隆。

3.23 joint文件夹joints.hpp

该源码主要是集成joint文件夹的所有头文件➦。

3.24 liegroup文件夹cartesian-product-variant.hpp/hxx

该源码主要是建立李群的笛卡尔乘积➦运算。

(1) CartesianProductOperationVariantTpl结构体

继承于LieGroupBase结构体，各种后缀impl实现基类的操作，不再重复解释相同接口。

成员对象：

| liegroup：用于操作的李群集合，LieGroupGeneric结构体。

| m_nq：构型空间维度大小。

| m_nv：切空间的维度大小。

| lg_nqs：所有李群的构型空间维度大小。

| lg_nvs：所有李群的切空间维度大小。

成员函数：

3.25 liegroup文件夹cartesian-product.hpp

该源码主要是建立两个李群的笛卡尔乘积运算。

(1) CartesianProductOperation结构体

继承于LieGroupBase结构体，各种后缀impl实现基类的操作，不再重复解释相同接口。

成员对象：

I lg1和lg2: 两个操作李群。

成员函数:

I ng: 获取李群构型空间维度大小=两个李群构型空间求和。

I nv: 获取李群切空间维度大小=两个李群切空间求和。

I Q1和Q2: 提取相应李群构型空间的序列位置, 左值。

I V1和V2: 提取相应李群切空间的序列位置, 左值。

I Qo1和Qo2: 提取相应李群构型空间的序列位置, 右值。

I Vo1和Vo2: 提取相应李群切空间的序列位置, 右值。

I J11和J12和J21和J22: 提取雅可比矩阵+对应李群的矩阵块。

3.26 liegroup文件夹fwd.hpp

该源码主要是对整个文件夹包含文件的命名简化。

3.27 liegroup文件夹liegroup-algo.hpp/hxx

该源码主要是将李群的具体计算内容与算法对应接口, 以结构体为主, 名称对应LieGroupBase的函数, Step和StepAlgo后缀成对出现, Step对接StepAlgo, StepAlgo调用李群的对应函数, 主要有:

I IntegrateStep和IntegrateStepAlgo: 对应李群的integrate函数。

I dIntegrateStep和dIntegrateStepAlgo: 对应李群的dintegrate函数。

I dIntegrateTransportStep和dIntegrateTransportStepAlgo: 对应李群的dIntegrateTransport函数。

I dIntegrateTransportInPlaceStep和dIntegrateTransportInPlaceStepAlgo: 对应李群的dIntegrateTransport函数。

I dDifferenceStep和dDifferenceStepAlgo: 对应李群的dDifference函数。

I InterpolateStep和InterpolateStepAlgo: 对应李群的interpolate函数。

I DifferenceStep和DifferenceStepAlgo: 对应李群的difference函数。

I SquareDistanceStep和SquareDistanceStepAlgo: 对应李群的squareDistance函数。

I SquareDistanceSumStep和SquareDistanceSumStepAlgo: 对应李群的squareDistance函数, 所有关节距离平方和的和。

I RandomConfigurationStep和RandomConfigurationStepAlgo: 对应李群的randomConfiguration函数。

I NormalizeStep和NormalizeStepAlgo: 对应李群的Normalize函数。

I IsNormalizedStep和IsNormalizedStepAlgo: 对应李群的isNormalized函数。

I IsSameConfigurationStep和IsSameConfigurationStepAlgo: 对应李群的isSameConfiguration函数。

I NeutralStep和NeutralStepAlgo: 对应李群的neutral函数。

I IntergrateCoeffWiseJacobianStep和IntergrateCoeffWiseJacobianStepAlgo: 对应李群的intergrateCoeffWiseJacobian函数。

3.28 liegroup文件夹liegroup-base.hpp/hxx

该源码主要构件李群的基础和其响应操作。

(1) LieGroupBase结构体

该类是李群的一个基础类，主要定义接口，实现均在派生类后缀_impl中。

额外定义的常量和结构体：

I NQ：李群构型空间。

I NV：李子群空间。

成员对象：

成员函数：

I integrate：在一个单位时间内根据构型空间和速度切空间积分求出新的构型空间。

I intergrateCoeffWiseJacobian：计算积分算子在李群零空间附近的雅克比矩阵。

I dIntegrate：计算李群零空间处的微小变量（构型或速度）切空间的雅克比矩阵 $q \oplus v$ 。

I dIntegrate_dq：计算李群零空间处的微小构型切空间的雅克比矩阵 $q \oplus v$ 。

I dIntegrate_dv：计算李群零空间处的微小速度切空间的雅克比矩阵 $q \oplus v$ 。

I dIntegrateTransport：将 $q \oplus v$ 作用的微小变量（构型或速度）切空间转换成q空间的切向量。

I dIntegrateTransport_dq：将 $q \oplus v$ 作用的微小构型切空间转换成q空间的切向量。

I dIntegrateTransport_dv：将 $q \oplus v$ 作用的微小速度切空间转换成q空间的切向量。

I interpolate：关节空间插值。

I normalize：关节空间规范化，如四元数四个数需要满足的条件。

I isNormalized：关节空间规范化检查。

I random：产生随机的规范化关节空间。

I randomConfiguration：产生随机的规范化并且满足约束的关节空间。

I difference：计算一个单位时间的切向量。

I dDifference：计算相对于初始或结束构型的雅克比矩阵。

I squareDistance：计算相对于两个构型之间距离平方和。

I isSameConfiguration：计算两个构型是否相同。

I ng：获取李群构型空间维度大小。

I nv：获取李群切空间维度大小。

l neutral: 获取李群的单位元素。

3.29 liegroup文件夹liegroup-collection.hpp

该源码主要是构件默认李群集合的结构体。

(1) LieGroupCollectionDefaultTpl结构体

3.30 liegroup文件夹liegroup-generic.hpp

该源码主要是构建李群结构体的一般形式模板。

(1) LieGroupGenericTpl结构体

继承于LieGroupBase结构体。

3.31 liegroup文件夹liegroup-variant-visitors.hpp/hxx

该源码主要是构建李群变量接口，主要以内联函数为主，具体实现见算法。

(1) nq函数

获取李群构型空间维度大小。

(2) nv函数

获取李群切空间维度大小。

(3) neutral函数

获取李群的单位元素。

(4) integrate函数

在一个单位时间内根据构型空间和速度切空间积分求出新的构型空间。

(5) random函数

产生随机的规范化关节空间。

(6) normalize函数

关节空间规范化，如四元数四个数需要满足的条件。

(7) isNormalized函数

关节空间规范化检查。

(8) isSameConfiguration函数

计算两个构型是否相同。

(9) squareDistance函数

计算相对于两个构型之间距离平方和。

(10) distance函数

计算两个构型的欧式空间。

(11) difference函数

计算一个单位时间的切向量。

(12) randomConfiguration函数

产生随机的规范化并且满足约束的关节空间。

(13) interpolate函数

关节空间插值。

(14) dIntegrate函数

计算流行零空间处的微小变量（构型或速度）切空间的雅克比矩阵。

(15) dDifference函数

计算相对于初始或结束构型的雅克比矩阵。

(16) dIntegrateTransport函数

将作用的微小变量（构型或速度）切空间转换成空间的切向量。

3.32 liegroup文件夹liegroup.hpp

该源码主要是对整个文件夹包含文件的打包。

3.33 liegroup文件夹special-euclidean.hpp

该源码主要构建特殊欧式空间SE(2)和SE(3)的李群操作模版。

(1) SpecialOrthogonalOperationTpl结构体

继承于LieGroupBase结构体，各种后缀impl实现基类的操作，不再重复解释相同接口。

SE(2)的构型空间维度为4（旋转角度正弦和余弦），切空间维度为3。

SE(3)的构型空间维度为7（四元数），切空间维度为6。

成员对象：

成员函数：

`I difference_impl`：位姿误差的指数映射。

`I dDifference_impl`：位姿误差的指数映射对切空间一阶雅克比矩阵。

`I integrate_impl`：位姿乘以切空间指数映射姿态矩阵。

`I integrateCoeffWiseJacobian_impl`：位姿对切空间一阶雅克比矩阵。

`I dIntegrate_dq_impl`：切空间的指数映射成位姿。

`I dIntegrate_dv_impl`：位姿求其指数映射的雅克比矩阵。

`I interpolate_impl`：位姿空间插值。

3.34 liegroup文件夹special-orthogonal.hpp

该源码主要构建旋转空间 $SO(2)$ 和 $SO(3)$ 的李群操作模版。

(1) SpecialOrthogonalOperationTpl结构体

继承于LieGroupBase结构体，各种后缀impl实现基类的操作，不再重复解释相同接口。

$SO(2)$ 的构型空间维度为2（旋转角度正弦和余弦），切空间维度为1。

$SO(3)$ 的构型空间维度为4（四元数），切空间维度为3。

成员对象：

成员函数：

`I difference_impl`：姿态误差的指数映射。

`I dDifference_impl`：姿态误差的指数映射对切空间一阶雅克比矩阵。

`I integrate_impl`：姿态乘以切空间指数映射姿态矩阵。

`I integrateCoeffWiseJacobian_impl`：四元数的对切空间一阶雅克比矩阵。

`I dIntegrate_dq_impl`：切空间的指数映射成四元数。

`I dIntegrate_dv_impl`：四元数表示的姿态求其指数映射的雅克比矩阵。

`I interpolate_impl`：四元数空间插值。

3.35 liegroup文件夹vector-space.hpp

该源码主要构建矢量空间的李群操作模版。

(1) VectorSpaceOperationTpl结构体

继承于LieGroupBase结构体，各种后缀impl实现基类的操作，不再重复解释相同接口，矢量空间就是欧式空间的操作。

成员变量:

| size : 矢量大小。

成员函数:

3.36 pool文件夹geometry.hpp

3.37 pool文件夹model.hpp

3.38 visitor文件夹fusion.hpp

该源码主要是将数据迭代成boost类型，主要为一些append函数。

3.39 visitor文件夹joint-binary-visitor.hpp

该源码主要是构件二自由度关节的基类封装各种访问形式与boost接口。

(1) JointBinaryVisitorBase结构体

该结构体主要实现二自由度关节访问接口。

成员对象:

成员函数:

3.40 visitor文件夹joint-unary-visitor.hpp

该源码主要是构件单自由度关节的基类封装各种访问形式与boost接口。

(1) JointUnaryVisitorBase结构体

该结构体主要实现单自由度关节访问接口。

成员对象:

成员函数: