

# 学习pinocchio (四) algorithm文件夹

知 zhuanlan.zhihu.com/p/698934195

©19299CC000000

尽心用心  
一切都是命运石之门的选择

14 人赞同了该文章

目录

该文件夹内的源码主要是运动学和动力学的核心算法实现。

## 4.1 aba-derivatives.hpp/hxx

该源码主要实现ABA算法的正动力学的微分，利用关节力矩求解关节加速度，以函数为主。

### (1) computeABADerivatives函数

采用ABA算法，计算模型的正动力学的微分，从关节力矩求解关节加速度，同名函数四个。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v，关节力矩tau，关节额外输入力fext。

输出：ABA相对于关节角度的微分aba\_partial\_dq (data.ddq\_dq)，ABA相对于关节速度的微分aba\_partial\_dv (data.ddq\_dv)，ABA相对于关节力矩的微分aba\_partial\_dg (data.Minv)。

实现过程中构造了四个结构体实现迭代：

| ComputeABADerivativesForwardStep1: 继承于fusion::JointUnaryVisitorBase，递推计算动力学正向参数第一步，具体算法在algo函数实现；

| ComputeABADerivativesBackwardStep1: 继承于fusion::JointUnaryVisitorBase，递推计算动力学逆向参数第一步，具体算法在algo函数实现；

| ComputeABADerivativesForwardStep2: 继承于fusion::JointUnaryVisitorBase，递推计算动力学正向参数第二步，具体算法在algo函数实现；

| ComputeABADerivativesBackwardStep2: 继承于fusion::JointUnaryVisitorBase，递推计算动力学逆向参数第二步，具体算法在algo函数实现；

## 4.2 aba.hpp/hxx

该源码主要实现ABA算法的正动力学，利用关节力矩求解关节加速度，以函数为主。

### (1) aba函数

采用ABA算法，计算模型的正动力学，从关节力矩求解关节加速度，同名函数两个。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v，关节力矩tau，关节额外输入力fext。

输出：关节加速度data.ddq。

实现过程中构造了三个结构体实现迭代：

| AbaForwardStep1: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节的相对位姿*liMi*, 速度*v*, 重力产生的加速度*a\_gf*, 惯性矩阵*Yaba*, 等效外力*f*, 具体算法在algo函数实现;

| AbaBackwardStep1: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节惯性矩阵*Yaba*, 等效外力*f*, 具体算法在algo函数实现;

| AbaForwardStep2: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节加速度ddq, 具体算法在algo函数实现;

## (2) computeMinverse函数

采用ABA+算法，计算关节空间惯性矩阵的逆。

输入：刚体模型model, 刚体数据data, 关节角度q。

输出：关节空间惯性矩阵的逆data.Minv。

实现过程中构造了三个结构体实现迭代：

| computeMinverseForwardStep1: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节的绝对位姿*oMi*, 雅克比矩阵*J*, 惯性矩阵*Yaba*具体算法在algo函数实现;

| computeMinverseBackStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节惯性矩阵*Yaba*, 关节空间力*Fcrb*, 具体算法在algo函数实现;

| computeMinverseForwardStep2: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节关节空间力*Fcrb*, 和关节空间矩阵的逆Minv, 具体算法在algo函数实现;

## (3) checkModel\_impl函数

检查ABA算法模型的正确性。

## 4.3 center-of-mass-derivatives.hpp/hxx

该源码主要用于计算质心运动学微分相关操作，以函数为主。

### (1) getCenterOfMassVelocityDerivatives函数

计算质心速度相对于关节空间的微分，需要先运算computeAllTerms或computeCenterOfMass。

输入：刚体模型model, 刚体数据data。

输出：质心速度相对于关节空间的微分vcom\_partial\_dq。

实现过程中构造了一个结构体实现迭代：

| CoMVelocityDerivativesForwardStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节的质心速度相对于关节空间的微分vcom\_partial\_dq, 具体算法在algo函数实现;

## 4.4 center-of-mass.hpp/hxx

该源码主要用于计算质心运动学相关操作，以函数为主。

## (1) computeTotalMass函数

计算整个多刚体的质量，同名函数两个（变量存于data.mass[0]）。

## (2) computeSubtreeMasses函数

计算所有关节作为根关节的多刚体质量（变量存于data.mass）。

## (3) centerOfMass函数

计算质心状态（位置、速度和加速度），重载了多个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节加速度a, 质心运算等级kinematic level, 是否计算所有关节子树的质心状态computeSubtreeComs。

输出：质心位置 data.com, 质心速度data.vcom, 质心加速度data.acom。

## (4) jacobianCenterOfMass函数

计算质心状态雅克比矩阵，重载了两个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 是否计算所有关节子树的质心状态computeSubtreeComs。

输出：质心位置 data.com, 质心雅克比矩阵data.Jcom。

实现过程中构造了一个结构体实现迭代：

| JacobianCenterOfMassBackwardStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节的质心位置 data.com, 质心雅克比矩阵data.Jcom, 具体算法在algo函数实现；

## (5) jacobianSubtreeCenterOfMass函数

计算指定关节作为根关节的质心状态雅克比矩阵，重载了两个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节IDrootSubtreeId。

输出：质心位置res。

实现过程中构造了一个结构体实现迭代：

| JacobianSubtreeCenterOfMassBackwardStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节的质心位置 data.com, 质心雅克比矩阵data.Jcom, 具体算法在algo函数实现；

## (6) getJacobianSubtreeCenterOfMass函数

获取指定关节作为根关节的质心状态雅克比矩阵。

## (7) getComFromCrba函数

若已经计算的CRBA算法，可以简单直接获取质心位置。

## (8) getJacobianComFromCrba函数

若已经计算的CRBA算法，可以简单直接获取质心状态雅克比矩阵。

## 4.5 centroidal-derivatives.hpp/hxx

该源码主要实现质心动力学+算法微分及其相关操作，以函数为主。

### (1) computeCentroidalDynamicsDerivatives 函数

计算质心动力学的微分，即动量和动量微分相对于关节的微分。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节角加速度a。

输出：动量相对于关节角度微分dh\_dq, 动量时间微分相对于关节角度微分dhdot\_dq, 动量时间微分相对于关节角速度微分dhdot\_dv, 动量时间微分相对于关节角加速度微分dhdot\_da。

实现过程中构造了两个结构体实现迭代：

| I\_CentroidalDynDerivativesForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节的动力学正向递推，具体算法在algo函数实现；

| I\_CentroidalDynDerivativesBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节的动力学反向递推，具体算法在algo函数实现；

### (2) getCentroidalDynamicsDerivatives 函数

获取质心动力学的微分，即动量和动量微分相对于关节的微分，该函数在RNEA微分算法computeRNEDerivatives之后。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节角加速度a。

输出：动量相对于关节角度微分dh\_dq, 动量时间微分相对于关节角度微分dhdot\_dq, 动量时间微分相对于关节角速度微分dhdot\_dv, 动量时间微分相对于关节角加速度微分dhdot\_da。

实现过程中构造了一个结构体实现迭代：

| I\_GetCentroidalDynDerivativesBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节的动力学反向递推，具体算法在algo函数实现；

## 4.6 centroidal.hpp/hxx

该源码主要实现质心动力学算法及其相关操作，以函数为主。

### (1) computeCentroidalMomentum 函数

计算多刚体的质心动量，重载了三个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v。

输出：质心位置 data.com, 整体动量data.hg。

### (2) computeCentroidalMomentumTimeVariation 函数

计算多刚体的质心动量的时间微分，重载了两个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节角加速度a。

输出：质心位置 data.com, 整体动量data.hg, 质心速度data.vcom, 整体动量时间微分data.dhg。

### (3) computeCentroidalDynamics函数

计算多刚体的质心动力学，即调用computeCentroidalMomentumTimeVariation函数。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v，关节角加速度v。

输出：质心位置 data.com，整体动量data.hg，质心速度data.vcom，整体动量时间微分data.dhg。

### (4) ccrba函数

计算多刚体的质心动量矩阵。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v。

输出：整体动量data.hg，质心惯性矩阵data.lg，质心关节空间矩阵data.Ag。

实现过程中构造了一个结构体实现迭代：

I CcrbaBackwardStep：继承于fusion::JointUnaryVisitorBase，递推计算每个关节的质心关节空间矩阵data.Ag，具体算法在algo函数实现；

### (5) computeCentroidalMap函数

计算多刚体的质心映射。

输入：刚体模型model，刚体数据data，关节角度q。

输出：质心关节映射data.Ag。

实现过程中构造了一个结构体实现迭代：

I CcrbaBackwardStep：继承于fusion::JointUnaryVisitorBase，递推计算每个关节的质心关节空间矩阵data.Ag，具体算法在algo函数实现；

### (6) dccrba函数

计算多刚体的质心动量矩阵的微分。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v。

输出：整体动量data.hg，质心惯性矩阵data.lg，质心关节空间矩阵data.Ag，质心关节空间矩阵微分data.dAg。

实现过程中构造了一个结构体实现迭代：

IDCcrbaBackwardStep：继承于fusion::JointUnaryVisitorBase，递推计算每个关节的质心关节空间矩阵data.dAg，具体算法在algo函数实现；

### (7) computeCentroidalMapTimeVariation函数

计算多刚体的质心映射的微分。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v。

输出：整体动量data.hg，质心惯性矩阵data.lg，质心关节空间矩阵data.Ag，质心关节空间矩阵微分data.dAg。

实现过程中构造了一个结构体实现迭代：

IDCcrbaBackwardStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节的质心关节空间矩阵data.dAg, 具体算法在algo函数实现；

## 4.7 check.hpp/hxx

该源码主要是对所有算法的各种数据进行检查，采用boost。

### (1) AlgorithmCheckerBase结构体

该结构体是算法检查的基类。

成员函数：

| checkModel: 检查模型数据类，具体有派生类checkModel\_impl实现。

### (2) AlgorithmCheckerList结构体

该结构体继承于AlgorithmCheckerBase, 是算法检查的列表。

成员对象：

| checkerList: 需要检查的列表，boost格式。

成员函数：

| checkModel\_impl: 检查模型数据类，调用boost。

### (3) checkData函数

该函数是检查写入模型的数据有效性，特别是模型已被修改的情况下。并关联成ModelTpI内置check函数。

## 4.8 cholesky.hpp/hxx

该源码主要是实现矩阵的cholesky分解和相关操作，利用关节空间惯性矩阵求解多刚体正动力学，主要是Cholesky命名空间的内联函数。

### (1) decompose函数

将关节模型数据中所有的关节空间惯性矩阵进行Cholesky分解，即  $M(q)=UDU^T$ ，其中D为对角矩阵，U对角线元素均为1的上三角矩阵。

### (2) solve函数

采用Cholesky分解 (Eigen::LLT)，求解 $M(q)x=y$ 方程的解 $\rightarrow x$ 。在正动力学方程 $\ddot{q} = M^{-1}(q)(\tau - b(q, \dot{q}))$ 中使用可以求解关节角加速度。

### (3) Mv函数

采用惯性矩阵的稀疏性，加快计算动量  $M(q)V$ 。

## (4) UDUtv函数

采用存储在关节空间惯性矩阵的Cholesky分解U和D矩阵，计算动量  $M(q)V$ 。

## (5) Uv函数

利用稀疏性，计算存储在关节空间惯性矩阵的Cholesky分解U和矢量乘法UV。

## (6) Utv函数

利用稀疏性，计算存储在关节空间惯性矩阵的Cholesky分解U和矢量乘法 $U^T V$ 。

## (7) Utiv函数

利用稀疏性，计算存储在关节空间惯性矩阵的Cholesky分解U和矢量乘法 $(U^T)^{-1} V$ 。

## (8) computeMinv函数

采用Cholesky分解，计算关节空间惯性矩阵的逆。

## 4.9 compute-all-terms.hpp

该源码主要用于有效的计算动力学所有项目，一个函数。

### (1) computeAllTerms函数

该函数等于forwardKinematics、crba、nonLinearEffects、computeJointJacobians、centerOfMass、jacobianCenterOfMass、ccrba、computeKineticEnergy、computePotentialEnergy和computeGeneralizedGravity函数。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v。

输出：刚体数据data里面对应变量。

实现过程中构造了两个结构体实现迭代：

| CATForwardStep: 继承于fusion::JointUnaryVisitorBase，计算每个关节正向递推的运算，具体算法在algo函数实现；

| CATBackStep: 继承于fusion::JointUnaryVisitorBase，计算每个关节反向递推的运算，具体算法在algo函数实现；

## 4.10 contact-dynamics.hpp/hxx

该源码主要用于计算带有接触约束的动力学，在使用该源码的函数需要先运行computeAllTerms函数。

### (1) forwardDynamics函数

计算带有接触约束的正动力学，重载三个同名函数。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v，关节力tau，约束的雅可比矩阵+J，约束的漂移量gamma，约束的阻尼系数+inv damping（JMinvJt变量的Cholesky分解），是否更新运动学updateKinematics。

输出：关节角加速度data.ddq。

## (2) computeKKTContactDynamicMatrixInverse函数

计算带有接触约束的KKT矩阵的逆。

输入：刚体模型model, 刚体数据data, 约束的雅可比矩阵J，。

输出：KKT矩阵的逆KKTMatrix\_inv。

## (3) getKKTContactDynamicMatrixInverse函数

获取带有接触约束的KKT矩阵的逆。

输入：刚体模型model, 刚体数据data, 关节角度q, 约束的雅可比矩阵J, 约束的阻尼系数inv\_damping。

输出：KKT矩阵的逆KKTMatrix\_inv。

## (4) impulseDynamics函数

计算带有接触约束的脉冲（切换）动力学，重载三个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节冲击前的角速度y\_before, 约束的雅可比矩阵J, 回归补偿系数（0到1）r\_coeff, 约束的阻尼系数inv\_damping, 是否更新运动学updateKinematics。

输出：冲击后的关节速度data.dq\_after。

## 4.11 copy.hpp

该源码主要是用于自定义拷贝数据。

### (1) copy函数

主要是将数据从原始拷贝到从属。

参数中有kinematic\_level: 0是拷贝所有关节在世界坐标系下的位姿oMi; 1是拷贝所有关节在关节坐标系下的空间速度v; 2是拷贝所有关节在关节坐标系下的空间加速度a, 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a\_af, 所有关节在关节坐标系下的空间力(包括所有外力) f。

## 4.12 crba.hpp/hxx

该源码主要用于计算关节空间惯性矩阵（算法参考书本第六章），以函数为主。

### (1) crba函数

计算关节空间惯性矩阵的上三角部分，采用简单的力递推方式。

输入：刚体模型model, 刚体数据data, 关节角度q。

输出：刚体数据data.M。

实现过程中构造了两个结构体实现迭代：

| CrbaForwardStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节相对于父类坐标系的位姿矩阵liMi和惯性矩阵Ycrb, 具体算法在algo函数实现；

**I CrbaBackStep**: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节作为根部多刚体的惯性矩阵Ycrb和空间力Fcrb, 具体算法在algo函数实现;

## (2) crbaMinimal函数

---

计算关节空间惯性矩阵的上三角部分, 采用的算法为惯性矩阵data.Ag和data.J。

输入: 刚体模型model, 刚体数据data, 关节角度q。

输出: 刚体数据data.M。

实现过程中构造了两个结构体实现迭代:

**I CrbaForwardStepMinimal**: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节相对于父类坐标系的位姿矩阵lMi和惯性矩阵Ycrb, 绝对坐标系位姿oMi和雅克比矩阵J, 具体算法在algo函数实现;

**I CrbaBackStepMinimal**: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节作为根部多刚体的惯性矩阵Ycrb和空间力Fcrb, 具体算法在algo函数实现;

## 4.13 default-check.hpp

---

该源码是这只检查函数的默认函数, 并关联成ModelTpI内置check函数。

## 4.14 dynamics.hpp

---

该源码将dynamics.hpp和contact-dynamics.hpp等价映射。

## 4.15 energy.hpp/hxx

---

该源码主要计算多刚体的能量, 以函数为主。

### (1) computeKineticEnergy函数和kineticEnergy函数

---

计算多刚体系统的动能。

输入: 刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 是否更新运动学update\_kinematics。

输出: 动能。

### (2) computePotentialEnergy函数和PotentialEnergy函数

---

计算多刚体系统的重力势能。

输入: 刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 是否更新运动学update\_kinematics。

输出: 重力势能。

## 4.16 frames-derivatives.hpp/hxx

---

该源码主要计算多刚体系统的坐标系相关的位姿等运动学微分, 以函数为主。

## (1) getFrameVelocityDerivatives函数

---

获取指定坐标系空间速度的微分。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 参考系rf (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：空间速度相对于关节角度的微分v\_partial\_dq, 空间速度相对于关节角速度的微分v\_partial\_dv。

## (2) getFrameAccelerationDerivatives函数

---

获取指定坐标系空间加速度的微分，重载两个同名函数。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 参考系rf (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：空间速度相对于关节角度的微分v\_partial\_dq, 空间速度相对于关节角速度的微分v\_partial\_dv, 空间加速度相对于关节角度的微分a\_partial\_dq, 空间加速度相对于关节角速度的微分a\_partial\_dv, 空间加速度相对于关节角加速度的微分a\_partial\_da。

## 4.17 frames.hpp/hxx

---

该源码主要计算多刚体系统的坐标系相关的位姿等运动学，以函数为主。

### (1) updateFramePlacements函数

---

更新坐标系位姿，重载同名函数两个（更新全部或指定）。

输入：刚体模型model, 刚体数据data, 指定坐标系frame\_id。

输出：坐标系位姿data.oMf。

### (2) framesForwardKinematics函数

---

更新所有坐标系位姿，重载同名函数两。

输入：刚体模型model, 刚体数据data, 关节角度q。

输出：坐标系位姿data.oMf。

### (3) getFrameVelocity函数

---

获取指定坐标系空间速度。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 参考系rf (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：空间速度。

### (4) getFrameAcceleration函数

---

获取指定坐标系空间加速度。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 参考系rf (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：空间加速度。

## (5) getFrameClassicalAcceleration函数

---

获取指定坐标系标准空间速度。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 参考系rf (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：标准空间速度。

## (6) getFrameJacobian函数

---

获取指定坐标系的空间速度与关节空间的雅可比矩阵。

输入：刚体模型model, 刚体数据data, 坐标系序号frameld, 参考系reference\_frame (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：雅可比矩阵J。

## (7) computeFrameJacobian函数

---

计算指定坐标系的空间速度与关节空间的雅可比矩阵，重载两个同名函数。

输入：刚体模型model, 刚体数据data, 坐标系序号frameld, 参考系rf (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：雅可比矩阵J。

## (8) frameJacobian函数

---

计算指定坐标系在局部坐标系的空间速度与关节空间雅可比矩阵，直接调用computeFrameJacobian。

## (9) computeFrameJacobianTimeVariation函数

---

计算指定坐标系的空间速度与关节空间的雅可比矩阵时间微分。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 参考系rf (定义了三个坐标系下的描述：WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：雅可比矩阵时间微分dJ。

## (10) computeSupportedInertiaByFrame函数

---

计算指定坐标系在局部坐标的惯性矩阵。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 是否包含子树的惯性矩阵。

输出：惯性矩阵。

## (11) computeSupportedInertiaByFrame函数

---

计算指定坐标系在局部坐标下受到的合力。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 是否包含子树的惯性矩阵。

输出：合力。

## 4.18 geometry.hpp/hxx

该源码主要是计算碰撞的几何体位姿，以函数为主。

### (1) updateGeometryPlacements函数

更新几何体位姿。

### (2) setGeometryMeshScales函数

设置几何材料的比例。

### (3) appendGeometryModel函数

增加几何模型。

## 4.19 jacobian.hpp/hxx

该源码主要用于计算多刚体动力学的雅克比矩阵，以函数为主。

### (1) computeJointJacobians函数

计算全模型的雅克比矩阵。实现过程中构造了两个结构体并对应构造了两个同名不同参数的函数用于分辨步骤：

| JointJacobiansForwardStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节相对于绝对坐标系的位姿矩阵oMi, 具体算法在algo函数实现；

| JointJacobiansForwardStep2: 继承于fusion::JointUnaryVisitorBase, 用每个关节绝对位姿矩阵与运动约束（旋量）相乘构成雅克比矩阵对应的列，存放在多刚体模型数据的J矩阵中，具体算法在algo函数实现。

### (2) getJointJacobian函数

计算指定关节在世界坐标系或局部坐标系的雅克比矩阵。实现过程中构建了details域名的同名函数：

| translateJointJacobian: 转换关节的雅克比矩阵（同名函数具有三个，迭代使用），getJointJacobian直接调用其中一个。定义了三个坐标系下的描述：WORLD世界坐标系；LOCAL WORLD ALIGNED局部位置+世界姿态坐标系；LOCAL局部坐标系。

### (3) computeJointJacobian函数

计算指定关节的所有父类关节在指定关节坐标系的雅克比矩阵。构建了一个结构体：

| JointJacobianForwardStep: 继承于fusion::JointUnaryVisitorBase, 采用algo函数具体实现算法。

### (4) computeJointJacobiansTimeVariation函数

计算全模型的雅克比矩阵关于时间的一阶微分。实现过程中构建了结构体：

I JointJacobiansTimeVariationForwardStep: 采用旋量的表达方式, 即 $\dot{J} = \text{ad}(V_i^S)J(i)$ , 其中  $J(i)$  表示雅可比矩阵的第*i*列, algo 函数具体实现算法。

## (5) getJointJacobianTimeVariation函数

计算指定关节在世界坐标系或局部坐标系的雅克比矩阵关于时间的一阶微分。直接调用translateJointJacobian函数即可。

## 4.20 joint-configuration.hpp/hxx

该源码主要是针对多刚体的关节空间运算, 同时为了API的存储优化, 基本以函数为主, 名称和李群操作的函数对应, 调用对应函数的后缀Step结构体。

### (1) integrate函数

该函数实现多刚体模型的所有关节独立李群空间的单位时间积分。

### (2) interpolate函数

该函数实现多刚体模型的所有关节独立李群空间的插值。

### (3) difference函数

该函数实现多刚体模型的所有关节独立李群空间的微分。

### (4) squaredDistance函数

该函数实现多刚体模型的所有关节独立李群空间的欧式空间平方和。

### (5) randomConfiguration函数

该函数实现多刚体模型的所有关节独立李群空间的随机产生数据。

### (6) neutral函数

该函数实现多刚体模型的所有关节独立李群空间的单位群。

### (7) dIntegrate函数

该函数实现多刚体模型的所有关节独立李群空间的: 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。

### (8) dIntegrateTransport函数

该函数实现多刚体模型的所有关节独立李群空间的: 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。

### (9) dDifference函数

该函数实现多刚体模型的所有关节独立李群空间的: 计算相对于初始或结束构型的雅克比矩阵。

### (10) squaredDistancesum函数

该函数实现多刚体模型的所有关节独立李群空间的欧式空间平方和的和。

## (11) distance函数

该函数实现多刚体模型的所有关节独立李群空间的欧式空间平方和的和，调用squaredDistancesum。

## (12) isNormalized函数

该函数实现多刚体模型的所有关节独立李群空间的：关节空间规范化检查。

## (13) isSameConfiguration函数

该函数实现多刚体模型的所有关节独立李群空间的：计算两个构型是否相同。

## (14) integrateCoeffWiseJacobian函数

该函数实现多刚体模型的所有关节独立李群空间的：位姿（姿态为四元数）对切空间一阶雅克比矩阵。

## 4.21 kinematics-derivatives.hpp/hxx

该源码主要是用于多刚体的运动学的微分，以函数为主。

### (1) computeForwardKinematicsDerivatives函数

计算模型的所有位姿，空间速度和加速度的一阶微分项（等价于先计算运动学，在计算雅克比矩阵，最后计算关节的空间速度和加速度）。

输入：刚体模型model，刚体数据data，关节角度q，关节角速度v，关节角角速度a。

输出：刚体数据data.oMi, data.v, data.a, data.ov, data.oa。

实现过程中构造了一个结构体实现迭代：

| ForwardKinematicDerivativesForwardStep: 继承于fusion::JointUnaryVisitorBase，递推计算每个关节相对于绝对坐标系的位姿矩阵oMi，关节空间速度v和ov，关节空间加速度a和oa，具体算法在algo函数实现；

### (2) getJointVelocityDerivatives函数

获取指定的关节的刚体空间速度及其一阶微分。

输入：刚体模型model，刚体数据data，关节序号jointId，参考系rf（定义了三个坐标系下的描述：WORLD世界坐标系；LOCAL WORLD ALIGNED局部位置+世界姿态坐标系；LOCAL局部坐标系）。

输出：空间速度对关节角度的微分v\_partial\_dq，空间速度对关节角速度的微分v\_partial\_dv。

实现过程中构造了一个结构体实现迭代：

| JointVelocityDerivativesBackwardStep: 继承于fusion::JointUnaryVisitorBase，递推计算每个关节速度对关节角度和角速度的微分，具体算法在algo函数实现；

### (3) getJointAccelerationDerivatives函数

获取指定的关节的刚体空间加速度及其一阶微分。

输入：刚体模型model，刚体数据data，关节序号jointId，参考系rf（定义了三个坐标系下的描述：WORLD世界坐标系；LOCAL WORLD ALIGNED局部位置+世界姿态坐标系；LOCAL局部坐标系）。

输出：空间速度对关节角度的微分 $v_{\text{partial}}_{dq}$ , 空间速度对关节角速度的微分 $v_{\text{partial}}_{dv}$ , 空间加速度对关节角度的微分 $a_{\text{partial}}_{dq}$ , 空间加速度对关节角速度的微分 $a_{\text{partial}}_{dv}$ , 空间加速度对关节角速度的微分 $a_{\text{partial}}_{da}$ 。

## (4) computeJointKinematicsHessians函数

计算模型的所有位姿，空间速度和加速度的二阶微分（Hessians矩阵）。

输入：刚体模型model, 刚体数据data。

输出：Hessians矩阵data.kinematic\_hessians。

## (5) getJointKinematicsHessians函数

获取指定关节状态的二阶微分（Hessians矩阵）。

输入：刚体模型model, 刚体数据data, 关节序号jointId, 参考系rf（定义了三个坐标系下的描述：WORLD世界坐标系；LOCAL WORLD ALIGNED局部位置+世界姿态坐标系；LOCAL局部坐标系）。

输出：Hessians矩阵。

## 4.22 kinematics.hpp/hxx

该源码主要是用于多刚体的运动学，以函数为主。

### (1) updateGlobalPlacements函数

更新数据中的全局关节位姿oMi。

输入：刚体模型model, 刚体数据data。

输出：刚体数据data。

### (2) forwardKinematics函数

计算模型的正运动学，即每个关节的刚体位姿、空间速度和空间加速度，同名函数三个。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节角角速度a。

输出：刚体数据data.oMi, data.v, data.a。

实现过程中构造了三个结构体实现迭代（对应计算位姿，位姿+速度，位姿+速度+加速度）：

| ForwardKinematicZeroStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节相对于绝对坐标系的位姿矩阵oMi, 具体算法在algo函数实现；

| ForwardKinematicFirstStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节相对于绝对坐标系的位姿矩阵oMi, 关节空间速度v, 具体算法在algo函数实现；

| ForwardKinematicSecondStep: 继承于fusion::JointUnaryVisitorBase, 递推计算每个关节相对于绝对坐标系的位姿矩阵oMi, 关节空间速度v, 关节空间加速度a, 具体算法在algo函数实现；

### (3) getVelocity函数

获取指定的关节的刚体空间速度。

输入：刚体模型model, 刚体数据data, 关节序号jointId, 参考系rf (定义了三个坐标系下的描述: WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：空间速度。

## (4) **getAcceleration**函数

---

获取指定的关节的空间加速度。

输入：刚体模型model, 刚体数据data, 关节序号jointId, 参考系rf (定义了三个坐标系下的描述: WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：空间加速度。

## (5) **getClassicalAcceleration**函数

---

获取指定的关节的刚体标准空间加速度。

输入：刚体模型model, 刚体数据data, 关节序号jointId, 参考系rf (定义了三个坐标系下的描述: WORLD世界坐标系; LOCAL WORLD ALIGNED局部位置+世界姿态坐标系; LOCAL局部坐标系)。

输出：空间加速度out=a+w\times v。

## 4.23 model.hpp/hxx

---

该源码主要是对多刚体模型的基本操作，基本是函数。

### (1) **appendModel**函数

---

该函数是将子模型添加到父类模型中，在父类索引后增加新的索引序号。

### (2) **buildRedecedModel**函数

---

该函数是根据给定的锁定关节，简化整个多刚体模型。

### (3) **details::getJointId**函数

---

获取关节序列号。

### (4) **details::getFrameId**函数

---

获取框架序列号。

### (5) **details::appendUniverseToModel**函数

---

向模型中添加整个的刚体，即多刚体如果是同一个父类关节，合并整合。

### (6) **details::AppendJointOfModelAlgoTpI**结构体

---

该结构体继承于fusion::JointUnaryVisitorBase基础结构体。

其中algo函数为将模型加入到父类关节中。

## 4.24 regressor.hpp/hxx

---

该源码主要是计算运动学关节的回归相关运算，以函数为主。

### (1) computeJointKinematicRegressor函数

---

计算指定关节（及其所有父类）在指定坐标系的运动回归矩阵，重载四个同名函数。

输入：刚体模型model, 刚体数据data, 关节序号joint\_id, 参考系rf（定义了三个坐标系下的描述：WORLD世界坐标系；LOCAL WORLD ALIGNED局部位置+世界姿态坐标系；LOCAL局部坐标系），指定坐标系在全局坐标系位姿placement,

输出：运动回归矩阵kinematic\_regressor.

### (2) computeFrameKinematicRegressor函数

---

计算指定坐标系（及其所有父类）在指定坐标系的运动回归矩阵，重载两个同名函数。

输入：刚体模型model, 刚体数据data, 坐标系序号frame\_id, 参考系rf（定义了三个坐标系下的描述：WORLD世界坐标系；LOCAL WORLD ALIGNED局部位置+世界姿态坐标系；LOCAL局部坐标系），指定坐标系在全局坐标系位姿placement,

输出：运动回归矩阵kinematic\_regressor.

### (3) computeStaticRegressor函数

---

计算所有关节的质心位置的静态回归器。

输入：刚体模型model, 刚体数据data, 关节角度q.

输出：静态回归器data.staticRegressor.

### (4) bodyRegressor函数

---

计算单刚体动力学参数回归器，重载了两个同名函数。

输入：关节角速度v。关节角加速度a。。

输出：回归器regressor.

### (5) jointBodyRegressor函数

---

计算指定关节的单刚体动力学参数回归器。

输入：刚体模型model, 刚体数据data, 指定关节jointId.

输出：回归器。

### (6) frameBodyRegressor函数

---

计算指定坐标系的单刚体动力学参数回归器。

输入：刚体模型model, 刚体数据data, 指定坐标系frameId.

输出：回归器。

## (7) computeJointTorqueRegressor函数

---

计算所有关节力的回归器。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节角加速度a。

输出：回归器data.jointTorqueRegressor。

实现过程中构造了两个结构体实现迭代：

| JointTorqueRegressorForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节力回归器的正向递推, 具体算法在algo函数实现;

| JointTorqueRegressorBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节力回归器的逆向递推, 具体算法在algo函数实现;

## 4.25 rnea-derivatives.hpp/hxx

---

该源码主要是采用RNEA (Recursive Newton-Euler algorithm) 计算逆动力学的微分, 以函数为主。

### (1) computeGeneralizedGravityDerivatives函数

---

采用RNEA动力学方法, 计算重力的微分。

输入：刚体模型model, 刚体数据data, 关节角度q。

输出：重力项相对于关节角度微分gravity\_partial\_dq。

实现过程中构造了两个结构体实现迭代：

| ComputeGeneralizedGravityDerivativeForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力微分的正向递推, 具体算法在algo函数实现;

| ComputeGeneralizedGravityDerivativeBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力微分的逆向递推, 具体算法在algo函数实现;

### (2) computeStaticTorqueDerivatives函数

---

采用RNEA动力学方法, 计算静态力 (包含重力和外力) 微分。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节额外输入力fext。

输出：静态力相对于关节角度微分static\_torque\_partial\_dq。

实现过程中构造了两个结构体实现迭代：

| ComputeGeneralizedGravityDerivativeForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力微分的正向递推, 具体算法在algo函数实现;

| ComputeGeneralizedGravityDerivativeBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力微分的逆向递推, 具体算法在algo函数实现。

### (3) computeRNEDerivatives函数

---

采用RNEA动力学方法, 计算动力学微分, 重载四个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节角加速度a, 关节额外输入力ext。

输出：关节力矩相对于关节角度的微分rena\_partial\_dq (data.dtau\_dq) , 关节力矩相对于关节角速度的微分rena\_partial\_dv (data.dtau\_dv) , 关节力矩相对于关节角加速度的微分rena\_partial\_da (data.M) 。

实现过程中构造了两个结构体实现迭代：

| IComputeRNEDerivativeForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节力的相关正向递推, 具体算法在algo函数实现;

| IComputeRNEDerivativeBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节力微分的逆向递推, 具体算法在algo函数实现。

## 4.26 rnea-second-order-derivatives.hpp/hxx

该源码主要是采用RNEA (Recursive Newton-Euler algorithm) 计算逆动力学的二阶微分, 以函数为主。

### (1) ComputeRNASEcondOrderDerivatives函数

采用RNEA动力学方法, 计算动力学二阶微分, 重载两个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节角加速度a。

输出：关节力矩相对于关节角度的二阶微分d2tau\_dqdq (data.d2tau\_dqdq) , 关节力矩相对于关节角速度的二阶微分d2tau\_dvdv (data.d2tau\_dvdv) , 关节力矩相对于关节角度和角速度的二阶微分d2tau\_dqdv (data.d2tau\_dqdv) , 关节力矩相对于关节角加速度和角度的二阶微分d2tau\_dadq (data.d2tau\_dadq) 。

实现过程中构造了两个结构体实现迭代：

| IComputeRNASEcondOrderDerivativeForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节力相关的正向递推, 具体算法在algo函数实现;

| IComputeRNASEcondOrderDerivativeBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节力微分的逆向递推, 具体算法在algo函数实现。

## 4.27 rnea.hpp/hxx

该源码主要是采用RNEA (Recursive Newton-Euler algorithm) 计算逆动力学, 以函数为主。

### (1) rnea函数

采用RNEA计算动力学, 已知关节加速度求关节驱动力矩, 重载两个同名函数。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v, 关节额外输入力ext。

输出：关节力矩data.tau。

实现过程中构造了两个结构体实现迭代：

| IRneaForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节动力学的力正向递推, 具体算法在algo函数实现;

| IRneaBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节动力学的力逆向递推, 具体算法在algo函数实现;

## (2) nonLinearEffects函数

---

采用RNEA动力学方法，计算非线性力。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v。

输出：动力学非线性项data.nle。

实现过程中构造了两个结构体实现迭代：

| NLEForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节非线性力正向递推, 具体算法在algo函数实现;

| NLEBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节非线性力逆向递推, 具体算法在algo函数实现;

## (3) computeGeneralizedGravity函数

---

采用RNEA动力学方法，计算重力项。

输入：刚体模型model, 刚体数据data, 关节角度q。

输出：动力学重力项data.g。

实现过程中构造了两个结构体实现迭代：

| ComputeGeneralizedGravityForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力正向递推, 具体算法在algo函数实现;

| ComputeGeneralizedGravityBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力逆向递推, 具体算法在algo函数实现;

## (4) computeStaticTorque函数

---

采用RNEA动力学方法，计算关节静态力（包含重力和外力）。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节额外输入力fext。

输出：关节静态力data.tau。

实现过程中构造了两个结构体实现迭代：

| ComputeGeneralizedGravityForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力正向递推, 具体算法在algo函数实现;

| ComputeGeneralizedGravityBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节重力逆向递推, 具体算法在algo函数实现;

## (5) computeCoriolisMatrix函数

---

采用RNEA动力学方法，计算关节科氏力矩阵。

输入：刚体模型model, 刚体数据data, 关节角度q, 关节角速度v。

输出：关节科氏力矩阵data.C。

实现过程中构造了两个结构体实现迭代：

| CoriolisMatrixForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节科氏力矩阵正向递推, 具体算法在algo函数实现;

| CoriolisMatrixBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节科氏力矩阵逆向递推, 具体算法在algo函数实现;

## (6) computeCoriolisMatrix函数

---

采用RNEA动力学方法, 计算关节科氏力矩阵, 在采用动力学微分运算后。

输入: 刚体模型model, 刚体数据data。

输出: 关节科氏力矩阵data.C。

实现过程中构造了两个结构体实现迭代:

| CoriolisMatrixForwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节科氏力矩阵正向递推, 具体算法在algo函数实现;

GetCoriolisMatrixBackwardStep: 继承于fusion::JointUnaryVisitorBase, 计算每个关节科氏力矩阵逆向递推, 具体算法在algo函数实现;