kinetics

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 Kinematics Class Reference

Public Member Functions

- **Kinematics** (bool state=true)
- Eigen::MatrixXd ForwardKin (const Eigen::MatrixXd &M, const Eigen::MatrixXd &Slist, const Eigen::VectorXd &thetaList)

  Compute end effector frame.
- bool InverseKin (const Eigen::MatrixXd &Slist, const Eigen::MatrixXd &M, const Eigen::MatrixXd &T, Eigen↩ ::VectorXd &thetalist, double eomg, double ev)

  Compute joints thetas given the end-effector configuration.
- Eigen::MatrixXd Jacobian (const Eigen::MatrixXd &Slist, const Eigen::MatrixXd &thetaList)

  Gives the Jacobian.

Public Attributes

- bool **space_state**

2.1.1 Member Function Documentation

2.1.1.1 ForwardKin()

```
Eigen::MatrixXd Kinematics::ForwardKin (
            const Eigen::MatrixXd & M,
            const Eigen::MatrixXd & Slist,
            const Eigen::VectorXd & thetaList )
```

Compute end effector frame.

Parameters

M,Home	configuration of end-effector
Slist	joint screw axis in (space_state by default, unless space_state is false for body state), when the manipulator is at home position.
thetaList	A list of joint coordinates.

Returns

T transformation matrix representing the end-effector frame when the joints are at the specified coordinates

2.1.1.2 InverseKin()

```
bool Kinematics::InverseKin (
            const Eigen::MatrixXd & Slist,
            const Eigen::MatrixXd & M,
            const Eigen::MatrixXd & T,
            Eigen::VectorXd & thetalist,
            double eomg,
            double ev )
```

Compute joints thetas given the end-effector configuration.

Parameters

Slist	screw axis (space_state by default, unless space_state is false for body state),
M	is home configuration of end-effector
T	is the target configuration
thetalist	the the desired joint angles to calculate
eomg,ev	are error checkers

Returns

boolean success flag

2.1.1.3 Jacobian()

```
Eigen::MatrixXd Kinematics::Jacobian (
            const Eigen::MatrixXd & Slist,
            const Eigen::MatrixXd & thetaList )
```

Gives the Jacobian.

Parameters

Slist	Screw axis
thetaList	joint configuration

Returns

6xn Spatial Jacobian

The documentation for this class was generated from the following files:

- /home/poetry/recovery/MR/kinetics/include/kinematics.hpp
- /home/poetry/recovery/MR/kinetics/src/kinematics.cpp

2.2 Kinetics Class Reference

Public Member Functions

- Eigen::VectorXd InverseDynamics (const Eigen::VectorXd &thetalist, const Eigen::VectorXd &dthetalist, const Eigen::VectorXd &ddthetalist, const Eigen::VectorXd &g, const Eigen::VectorXd &Ftip, const std::vector< Eigen::MatrixXd > &Mlist, const std::vector< Eigen::MatrixXd > &Glist, const Eigen::MatrixXd &Slist) const

  Uses forward-backward Newton-Euler iterations to solve the equation: taulist = Mlist(thetalist) ∗ ddthetalist + c(thetalist, dthetalist) + g(thetalist) + Jtr(thetalist) ∗ Ftip.
- Eigen::VectorXd ForwardDynamics (const Eigen::VectorXd &thetalist, const Eigen::VectorXd &dthetalist, const Eigen::VectorXd &taulist, const Eigen::VectorXd &g, const Eigen::VectorXd &Ftip, const std::vector< Eigen::MatrixXd > &Mlist, const std::vector< Eigen::MatrixXd > &Glist, const Eigen::MatrixXd &Slist) const

  Computes ddthetalist by solving: Mlist(thetalist) ∗ ddthetalist = taulist - c(thetalist,dthetalist)
- void **EulerStep** (Eigen::VectorXd &thetalist, Eigen::VectorXd &dthetalist, const Eigen::VectorXd &ddthetalist, double dt) const
- Eigen::VectorXd **ComputedTorque** (const Eigen::VectorXd &thetalist, const Eigen::VectorXd &dthetalist, const Eigen::VectorXd &eint, const Eigen::VectorXd &g, const std::vector< Eigen::MatrixXd > &Mlist, const std::vector< Eigen::MatrixXd > &Glist, const Eigen::MatrixXd &Slist, const Eigen::VectorXd &thetalistd, const Eigen::VectorXd &dthetalistd, const Eigen::VectorXd &ddthetalistd, double Kp, double Ki, double Kd)
- std::vector< Eigen::MatrixXd > **SimulateControl** (const Eigen::VectorXd &thetalist, const Eigen::VectorXd &dthetalist, const Eigen::VectorXd &g, const Eigen::MatrixXd &Ftipmat, const std::vector< Eigen::MatrixXd > &Mlist, const std::vector< Eigen::MatrixXd > &Glist, const Eigen::MatrixXd &Slist, const Eigen::Matrix↩ Xd &thetamatd, const Eigen::MatrixXd &dthetamatd, const Eigen::MatrixXd &ddthetamatd, const Eigen::↩ VectorXd &gtilde, const std::vector< Eigen::MatrixXd > &Mtildelist, const std::vector< Eigen::MatrixXd > &Gtildelist, double Kp, double Ki, double Kd, double dt, int intRes)

2.2.1 Member Function Documentation

2.2.1.1 ForwardDynamics()

```
Eigen::VectorXd Kinetics::ForwardDynamics (
             const Eigen::VectorXd & thetalist,
             const Eigen::VectorXd & dthetalist,
             const Eigen::VectorXd & taulist,
             const Eigen::VectorXd & g,
             const Eigen::VectorXd & Ftip,
             const std::vector< Eigen::MatrixXd > & Mlist,
             const std::vector< Eigen::MatrixXd > & Glist,
             const Eigen::MatrixXd & Slist ) const  [inline]
```

Computes ddthetalist by solving: Mlist(thetalist) ∗ ddthetalist = taulist - c(thetalist,dthetalist)

- g(thetalist) - Jtr(thetalist) ∗ Ftip

Parameters

thetalist	n-vector of joint variables
dthetalist	n-vector of joint rates
taulist	An n-vector of joint forces/torques
g	Gravity vector g
Ftip	Spatial force applied by the end-effector expressed in frame {n+1}
Mlist	List of link frames {i} relative to {i-1} at the home position
Glist	Spatial inertia matrices Gi of the links
Slist	Screw axes Si of the joints in a space frame, in the format of a matrix with the screw axes as the columns.

Returns

ddthetalist The resulting joint accelerations.

2.2.1.2 InverseDynamics()

```
Eigen::VectorXd Kinetics::InverseDynamics (
             const Eigen::VectorXd & thetalist,
             const Eigen::VectorXd & dthetalist,
             const Eigen::VectorXd & ddthetalist,
             const Eigen::VectorXd & g,
             const Eigen::VectorXd & Ftip,
             const std::vector< Eigen::MatrixXd > & Mlist,
             const std::vector< Eigen::MatrixXd > & Glist,
             const Eigen::MatrixXd & Slist ) const  [inline]
```

Uses forward-backward Newton-Euler iterations to solve the equation: taulist = Mlist(thetalist) ∗ ddthetalist + c(thetalist, dthetalist) + g(thetalist) + Jtr(thetalist) ∗ Ftip.

Parameters

thetalist	n-vector of joint variables
dthetalist	n-vector of joint rates
ddthetalist	n-vector of joint accelerations
g	Gravity vector g
Ftip	Spatial force applied by the end-effector expressed in frame {n+1}
Mlist	List of link frames {i} relative to {i-1} at the home position
Glist	Spatial inertia matrices Gi of the links
Slist	Screw axes Si of the joints in a space frame, in the format of a matrix with the screw axes as the columns.

Returns

taulist The n-vector of required joint forces/torques

The documentation for this class was generated from the following files:

- /home/poetry/recovery/MR/kinetics/include/kinetics.hpp
- /home/poetry/recovery/MR/kinetics/src/kinetics.cpp

2.3 Trajectory Class Reference

Collaboration diagram for Trajectory:

Index