

<<typedef>>
JointAngles=std::array<double, 6>

<<typedef>>
DHTable=Eigen::Array<double, mNumDHRows, mNumDHCols>

struct Pose

+ Pose(const Matrix4d &T)
+ position : Eigen::Vector3d
+ orientation : Eigen::Quaterniond
+ friend std::ostream &operator<<(std::ostream &out, const Pose &pose)

struct DHParams

+ d1, d2, d3, d4, d5, d6, a2 : float
+ DHParams(float d1, float d2, float d3, float d4, float d5, float d6, float a2)

class ForwardKinematics

- mNumDHRows : constexpr static const size_t
- mNumDHCols : constexpr static const size_t
- alphaIndex : constexpr static const size_t
- aIndex : constexpr static const size_t
- thetaIndex : constexpr static const size_t
- dhTable : Eigen::Array<double, mNumDHRows, mNumDHCols>
- dhParams : const DHParams

+ fk(const JointAngles &ja) : Pose
+ ForwardKinematics()
+ getTransformationMatrix(const Eigen::Array<double, 1, mNumDHCols> &dhRow) : Matrix4d

<<typedef>>
JointAngles=std::array<double, 6>

class InverseKinematics

-deltaTimeSecs:unsigned float
-initialJointAngles:JointAngles

+ InverseKinematics(const JointAngles& inInitialJointAngles)
+ linearIK(const Pose& currentPose,const Pose& targetPose) : std::vector<JointAngles>
- getJacobian(const & JointAngles): MatrixXd