<<typedef>> JointAngles=std::array<double, 6> <<typedef>> DHTable=Eigen::Array<double, mNumDHRows, mNumDHCols> class ForwardKinematics struct Pose - mNumDHRows : constexpr static const size_t - mNumDHCols : constexpr static const size_t + Pose(const Matrix4d &T) - alphaIndex : constexpr static const size t + position : Eigen::Vector3d - alndex : constexpr static const size_t + orientation : Eigen::Quaterniond - thetaIndex : constexpr static const size t + friend std::ostream &operator<<(std::ostream - dhTable : Eigen::Array<double, mNumDHRows, mNumDHCols> &out, const Pose &pose) - dhParams : const DHParams + fk(const JointAngles &ja): Pose struct DHParams + ForwardKinematics() + getTransformationMatrix(const Eigen::Array<double, 1, mNumDHCols> + d1, d2, d3, d4, d5, d6, a2 : float &dhRow): Matrix4d + DHParams(float d1, float d2, float d3, float d4, float d5, float d6, float a2)

class InverseKinematics

- dhTable : Eigen::Array<double, mNumDHRows, mNumDHCols>
- $\hbox{-} Forward Transformation Matrix: Matrix 4d}\\$
- + InverseKinematics(const Eigen::Matrix4d &FM)
- + ik(const JointAngles& currentAngles, targetAngles) : std::vector<JointAngles>