

```

#include <Eigen/Dense>
#include <math.h>
#include <stdexcept>

/*
   A mathematical model of a robot on the form
        $M(q) \cdot \ddot{q} + C(q, \dot{q}) \dot{q} + D(q, \dot{q}) \dot{q} + g(q) = Jf(q) B u(z)$ 
   z is a specified vector of inputs to the robot.

   M, C and D are n x n matrices
   J is a Nj x Nn matrix
   Jf is a Nn x Nb matrix
   B is a Nb x Nm matrix
   u is a Nm x 1 vector

   The values of nb, nj and m might be hard to calculate before the model is
   created.

   q = [xn, yn, zn, φ, θ, ψ]
   dq = [dxn, dyn, dzn, dφ, dθ, dψ]
   z = [f1, f2, τ1]
   u = [f1, f2, τ1]
*/

namespace Model {

constexpr int Nn = 6;
constexpr int Nj = 6;
constexpr int Nb = 7;
constexpr int Nm = 3;
constexpr int Nz = 3; // number of inputs
constexpr int Nt = 1; // number of transforms

Eigen::MatrixXd M(Eigen::VectorXd q);
Eigen::MatrixXd C(Eigen::VectorXd q, Eigen::VectorXd dq);
Eigen::MatrixXd D(Eigen::VectorXd q, Eigen::VectorXd dq);
Eigen::VectorXd g(Eigen::VectorXd q);
Eigen::MatrixXd J(Eigen::VectorXd q);
Eigen::MatrixXd Jf(Eigen::VectorXd q);
Eigen::MatrixXd B();
Eigen::VectorXd u(Eigen::VectorXd z);
Eigen::MatrixXd Ju(Eigen::VectorXd z);
Eigen::VectorXd T0_to_b(Eigen::VectorXd q, Eigen::VectorXd x_0);
Eigen::VectorXd ddq(Eigen::VectorXd q, Eigen::VectorXd dq, Eigen::VectorXd z);
Eigen::VectorXd u_to_z(Eigen::VectorXd u);

} // namespace

```