Author: Aiello Gregorio

Subject: Redundant Manipulators: Jacobian-Based Inverse Kinematics

Date: July 2017

For serial manipulators the map from the joint space to the cartesian space is given by the jacobian matrix:

$$\dot{x} = J(q)\dot{q} \tag{1}$$

and in order to find the inversion of the differential kinematics:

$$\dot{q} = J(q)^{-1}\dot{x} \tag{2}$$

The Jacobian matrix for redundant manipulators is not squared, thus its inversion needs special care, we can use an optimization problem to find the solution that fits the most the kinematic constraints set by the user. First we try to minimize the joint velocities:

$$f(\dot{q}) = \frac{1}{2}\dot{q}^T A\dot{q} \tag{3}$$

where A is a weight matrix, symmetric and positive definite. In order find the constrained minimum we can use the Lagrange's Multipliers method:

$$f(\dot{q}, \lambda) = \frac{1}{2}\dot{q}^T A\dot{q} + \lambda^T(\dot{x} - J(q)\dot{q}) \tag{4}$$

$$\left(\frac{\partial f}{\partial \dot{q}}\right)^T = 0 \rightarrow \dot{q} = A^{-1}J^T\lambda \tag{5}$$

$$\left(\frac{\partial f}{\partial \lambda}\right)^T = 0 \rightarrow \dot{x} = J\dot{q} \tag{6}$$

Combining (5) and (6) we obtain:

$$\dot{x} = (JA^{-1}J^T)\lambda \tag{7}$$

notice that $(JA^{-1}J^T)$ has full rank, thus:

$$\lambda = (JA^{-1}J^T)^{-1}\dot{x} \tag{8}$$

and back substituted into (5):

$$\dot{q} = A^{-1}J^T(JA^{-1}J^T)^{-1}\dot{x} \tag{9}$$

The latter is a working jacobian inversion since if we multiply by J each side we get equation (1). The matrix A may be equal to the identity, in such a way the algorithm turns into the right-pseudoinverse.

The values of $\dot{q}$ are the ones that minimize the joint velocities (since the first constraints is the kinetic energy).

If $\dot{q}$ is a solution hence $\dot{q} + N\dot{q}_{ns}$ is a solution as well, considering N as the matrix that projects joint velocities into the null space of the Jacobian. The solution found in (9) can be modified with the addition of an additional constraint $N\dot{q}_{ns}$:

$$g(\dot{q}) = \frac{1}{2}(\dot{q} - \dot{q}_{ns})^T(\dot{q} - \dot{q}_{ns}) \tag{10}$$

And following the procedure in (4):

$$g(\dot{q}, \lambda) = \frac{1}{2}(\dot{q} - \dot{q}_{ns})^T(\dot{q} - \dot{q}_{ns}) + \lambda^T(\dot{x} - J(q)\dot{q}) \tag{11}$$

and again:

$$\left(\frac{\partial g}{\partial \dot{q}}\right)^T = 0 \rightarrow \dot{q} = J^T\lambda + \dot{q}_{ns} \tag{12}$$

$$\left(\frac{\partial g}{\partial \lambda}\right)^T = 0 \rightarrow \dot{x} = J\dot{q} \tag{13}$$

finally after finding λ we obtain:

$$\dot{q} = J^T(JJ^T)^{-1}\dot{x} + (I - J^T(JJ^T)^{-1}J)\dot{q}_{ns} \tag{14}$$

Which minimize the joint velocities (homogeneous solution) and at the same time tries to minimize the second constraint, thus if $\dot{x} = 0$ the robot can still follow a trajectory in the joint space. In our case the function in the null space is:

$$\dot{q}_{ns} = \mu \left(\frac{\partial w(q)}{\partial q}\right)^T \tag{15}$$

with μ scalar coefficient > 0 and $w(q)$ function of the constraint in the joint space:

$$w(q) = (t(q) - o) \tag{16}$$

with $t(q)$ position of the sensor (world frame) and o position of the obstacle (world frame).

---

we need the position in the world frame of every sensor, moreover from the result of the sensor we detect the direction of the approaching obstacle.