Analytical Inverse Kinematics of a 6-DOF Mantis Robot Leg and Modeling for Minimum Joint Load

Feliphe Gonçalves Galiza, Dave Touretzky

Robotics Institute & Computer Science Department, Carnegie Mellon University

MOTIVATION

The Chiara Mantis is a robot based on the praying mantis insect, it can use its front legs both to walk and handle objects. Therefore computing the joint angles, which enables the robot to perform accurate trajectories is unavoidable.

Generally, the solution for the 6-DOF inverse kinematics problem is obtained by several trigonometric substitutions that are specific to one robot configuration, or by numerical methods which just provide one answer out of a bigger range of possible solutions.

The goal of this work is to describe the application of the OpenRAVE ikFast module, which analytically solves inverse kinematics equations for a variety of robot configurations and generates optimized C++ code files.

As an addition, we also discuss about a method for classifying the multiple inverse kinematics solutions based on choosing the posture which enables the Chiara Mantis robot to support its own weight conveniently.

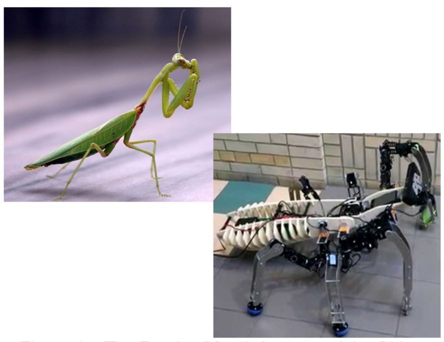

Figure 1 – The Praying Mantis Insect and the Chiara Mantis robot.

METHODOLOGY

■ ***Developing the IkFast robot model :***

The OpenRAVE ikFast module requires a XML file format containing information about the robot joint types and relative position between the robot links, which can be represented by using the Denavit-Hartenberg convention, also called D-H parameters.

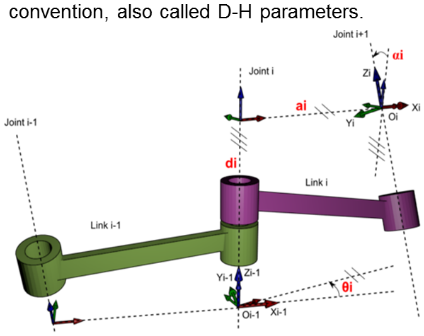

Figure 2 – D-H parameters convention.

The transformation from link i-1 to link i is defined by the matrix *Ai*.

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cdot\cos(\alpha_i) & \sin(\theta_i)\cdot\sin(\alpha_i) & a_i\cdot\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cdot\cos(\alpha_i) & -\cos(\theta_i)\cdot\sin(\alpha_i) & a_i\cdot\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

After defining the D-H parameters, each transformation matrix *Ai* can be used to build the robot model, and the OpenRAVE ikFast module is ready to generate a C++ file with the analytical inverse kinematics solution.

■ ***Modeling for Minimum Joints Static Torque:***

The solution of a 6-DOF analytical inverse kinematics problem leads to a set of multiple robot postures. In this case the target is constrained to maintain the foot frame perpendicular to the floor. If we make an analogy with a common robot manipulator, the foot frame is being considered as the base frame. The thorax-head assembly is treated as the end-effector, and its Center of Mass is the point at which forces are applied.

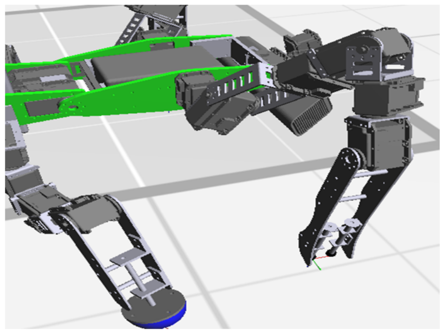

Figure 3 – Chiara Mantis Right Front Foot Frame.

The movement of the Chiara Mantis joints can be represented by the rotation matrix *Ri,* which represents a rotation about the local joint z-axis by an angle *qi*.

$$R_i = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 & 0 \\ \sin(q_i) & \cos(q_i) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation matrix from the 6th joint frame to the foot frame is given by T_{foot}. From the 1st joint to the thorax-plus-head Center of Mass frame, the transformation matrix is given by T_{COM}. Therefore the full transformation is represented by T^{foot}_{COM}.

$$T^{foot}_{COM} = \left[\left(\prod_{i=1}^{i=6} A_i \cdot R_i\right) \cdot T_{foot}\right]^{-1} \cdot T_{COM}$$

The Jacobian matrix can be obtained from T^{foot}_{COM}, by using the following definition:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} J_p \\ J_o \end{bmatrix} \dot{q}$$

J_p 3 x 6 Position Jacobian

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = J\dot{q} = \begin{bmatrix} J_p \\ J_o \end{bmatrix} \dot{q} = \begin{bmatrix} \frac{\partial p_x}{\partial q_1} & \frac{\partial p_x}{\partial q_2} & \frac{\partial p_x}{\partial q_3} & \frac{\partial p_x}{\partial q_4} & \frac{\partial p_x}{\partial q_5} & \frac{\partial p_x}{\partial q_6} \\ \frac{\partial p_y}{\partial q_1} & \frac{\partial p_y}{\partial q_2} & \frac{\partial p_y}{\partial q_3} & \frac{\partial p_y}{\partial q_4} & \frac{\partial p_y}{\partial q_5} & \frac{\partial p_y}{\partial q_6} \\ \frac{\partial p_z}{\partial q_1} & \frac{\partial p_z}{\partial q_2} & \frac{\partial p_z}{\partial q_3} & \frac{\partial p_z}{\partial q_4} & \frac{\partial p_z}{\partial q_5} & \frac{\partial p_z}{\partial q_6} \\ a_{x_0} & a_{x_1} & a_{x_2} & a_{x_3} & a_{x_4} & a_{x_5} \\ a_{y_0} & a_{y_1} & a_{y_2} & a_{y_3} & a_{y_4} & a_{y_5} \\ a_{z_0} & a_{z_1} & a_{z_2} & a_{z_3} & a_{z_4} & a_{z_5} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}$$

J_o 3 x 6 Orientation Jacobian

Finally we can define the torque required by each joint to keep the robot in static equilibrium as the vector τ.

Where: $$\tau = J^T \cdot F$$

- $\tau = [\tau_1\ \tau_2\ \tau_3\ \tau_4\ \tau_5\ \tau_6]^T$ and $F = [F_x\ F_y\ F_z\ T_x\ T_y\ T_z]^T$.

RESULTS

■ ***IkFast Analytical Inverse Kinematics Solution Analysis***

The OpenRave IkFast module was applied at the Chiara Mantis 6-DOF front legs. A sample of several points forming a cylinder was used as a trajectory the leg should be able to follow. At the end of the experiment, the OpenRAVE IkFast function returned from 4 to 8 solutions for each trajectory point. On each iteration the first solution was taken, and the angle vector $q = [q_1\ q_2\ q_3\ q_4\ q_5\ q_6]$ was updated at the forward kinematics equation. The result representing the foot frame positions after each forward kinematics iteration is shown in Figure 4.

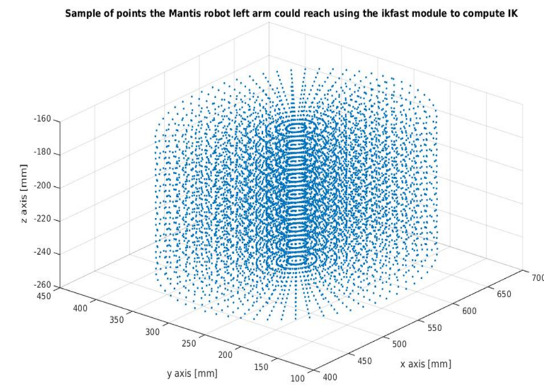

Figure 4 – Inverse Kinematics Experiment.

■ ***Joints Static Torque Analysis***

In this experiment the target point used for the robot foot frame was constrained to be perpendicular to the floor. The force input vector was composed just by the gravitational force acting at the thorax-plus-head COM. For each solution, the results of the computed absolute static torque at each joint are shown below.

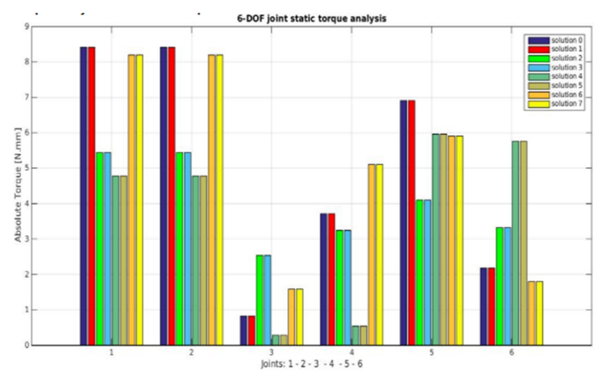

Figure 5 – Joints Static Torque Analysis

CONCLUSION

The results show the robot could perform accurate trajectories using the OpenRAVE ikFast module, therefore it is a valid approach to solve the 6-DOF Inverse Kinematics problem. In the second experiment, solutions 4 and 5 can be selected as the best solutions since they require the least torque at 4 of the 6 leg joints.

REFERENCES

[1] Rosen Diankov. ikfast The Analytical Inverse Kinematics Generator. http://openrave.programmingvision.com/en/main/openravepy/databases.inversekinematics.html.

[2] Robert L. Williams. Robot Mechanics. NotesBook Supplement for ME 4290/5290 Mechanics and Control of Robotic Manipulators. 2015 Dr. Bob Productions. http://www.ohio.edu/people/williar4/html/PDF/Supplement4290.pdf