# Kinematic Libraries

OpenHRP uses the Body Library (Body.h) to compute forward and inverse kinematics based on the joint structure of the robot.

The joint structure of the robot is defined by the VRML (.wrl) files. Both work as part of the hiroArm class implementation.

The Body lib can instantiate different types of objects: BodyPtr's and JointPathPtr's.

**The BodyPtr Object**

* Captures the location of all joints in the robot. For the HIRO humanoid robot that would be 15 joints (CHEST,HEAD\_TILT,HEAD\_PAN,RARM0-5,LARM0-5). Usually created in forceSensorPlugin.
* The BodyPtr object is retrieved by using the function “ loadBodyFromModelLoader” and passing the path of the robot vrml simulation file ending in extension “.wrl.”  
    
  BodyPtr body;  
  body = OpenHRP::loadBodyFromModelLoader((manager->url).c\_str(),manager->orb);
* **Joint Angles**  
  One can retrieve the joint position of the robot by calling: body->joint(i)->q
* **Link Position and Orientation**  
  One can retrieve the position and orientation of a link attached to a joint by calling:  
    
  body->link(i)->p  
  body->link(i)->attitude
* **Forward Kinematics**  
  One can compute the forward kinematics (from angles to cartesian coordinates and rpy coordinates) by calling:   
    
  body->calkForwardKinematics()  
    
  When doing so, all positions, joint angles, and related parameters are automatically updated. Similary with inverse kinematics.

**The JointPathPtr Object**

* This object represents a segment of the whole body of the robot. For example, it can be used to contain the path from the base of the robot to the end-effector of the right robot arm. This is useful if we want to compute the Inverse Kinematics.  
    
  The path is returned by a BodyPtr object when the function “getJointPath” is called. “getJointPath” takes two arguments of type Link\*. For example:  
    
  Link\* S0 = body->joint(CHEST\_JOINT0);  
  Link\*W6 = body->joint(RARM\_JOINT5);  
  JointPathPtr m\_path;  
  m\_path = body->getJointPath(S0,W6);
* **Inverse Kinematics**  
  One can compute the joint angles of a robot for a give end-effector position and orientation by calling the following:  
    
  m\_path->calcInverseKinematics(PositionVector,RotationMatrix)  
    
  Gains can also be set by:  
    
  m\_path->setIKGain(\_Arm)->IkGain[]
* **Number of Joints**  
  One can find out the number of joints in the path by calling:  
    
  m\_path->numJoints()