International Journal of Robotics and Automation
Vol. 1: Issue 1
www.journalspub.com

Simulation of the Inverse-Kinematics for JACO Manipulator Robot Arm

Salih Rashid Majeed, Klause D-Kuhnert*
Institute for Real Time Learning System, Siegen University, Germany

Abstract

The inverse kinematics is a very important issue in the automation of robotic manipulator. This paper deals with inverse kinematics (IK) in the simulation process, using "JACO" robotic arm and studies about the implementation of IK in a very simple manner, using BLENDER software. In this program, we simulated IK based on the relationship between the bones and the skin in natural life, making use of some options included in the program.

Keywords: inverse kinematics, BLENDER, JACO robot arm, Morse

***Corresponding Author**
E-mail: salih.majeed@uni-siegen.de

INTRODUCTION

The significance of inverse kinematics (KI) in the manipulator robotic arm due to direct effect in the movement of the arm, made the study of this relationship very important in the robot field. The use of robot arm in many applications persuades the necessity of studying the robot arm mechanism and discovers the easy methods to attain soft and real movement in the robot arm. Our algorithm subjects to one of the effective and well-known simulation programs for implementing this design and mechanism instead of the classical calculations applied before.

RELATED WORK

Monzani *et al*. represented a method to re target a motion from one character to a geometrically and topologically different one, and demonstrate how IK can conserve the motion-captured posture while enforcing Cartesian constraints. Their tool has been successfully applied to video-game projects. Extensions could be used to define more complex mapping functions, rather than one-to-one correspondences, to be able to handle coupled DOFs. [1]

Grochow *et al*. present an IK system based on a learned model of human poses given a set of constraints, our system can produce the most likely pose satisfying those constraints, in real time training, the model on different input data leads to different styles of IK. The model is represented as a probability distribution over the space of all possible poses which means that our IK system can generate any pose, but prefers to the poses that are most similar to the space of poses in the training data, they represent the probability with a novel model called a Scaled Gaussian Process Latent Variable Model and their style-based IK can replace conventional IK, wherever it is issued in computer animation and computer vision. [2]

Der *et al*. stated that their IK algorithm provides intuitive and direct control of non-rigid shapes similar to a conventional IK algorithm for jointed rigid skeletons. It manages the geometric complexity of detailed meshes by using a reduced deformable model that parameterizes shape deformation with a compact set of control parameters. [3]

In Sumner *et al*., skeleton-based articulation is often used in animation to approximate mesh kinematics compactly. However, skeletons cannot easily provide the rich class of deformations afforded by sculpting techniques and only allow indirect interaction with the mesh via the joint angles of the skeleton their method allows the user to directly position any subset of mesh vertices and produces a meaningful deformation automatically. Complex pose changes can be accomplished intuitively by manipulating only a few vertices. In analogy to traditional skeleton-based IK for posing skeletons, we call this general problem mesh based IK, and our example solution MESHIK. [4]

THEORETICAL WORK

IK is defined as the problem of determining a set of appropriate joint configurations for which the end effectors move to desired positions as smoothly, rapidly, and as accurately as possible. During the last decades, several methods and techniques, sophisticated or heuristic, have been presented to produce fast and realistic solutions to the IK problem. However, most of the currently available methods suffer from high computational cost and production of unrealistic poses.

This report reviews and compares the most popular IK methods regarding reliability, computational cost and conversion criteria.[5]

The method that used in this paper is implementation of the IK without any problems or any calculations by using simulation programs:

- MORSE: a new open-source robotics simulator. MORSE provides several features of interest to robotics projects: it relies on a component-based architecture to simulate sensors, actuators and robots; it is flexible, able to specify simulations at variable levels of abstraction according to the systems being tested; it is capable of representing a large variety of heterogeneous robots and full 3D environments (aerial, ground, maritime); and it is designed to allow simulations of multiple robots systems MORSE uses a “Software in- the-Loop” philosophy. [6] It focuses on realistic 3D simulation of small to large environments, indoor or outdoor, with one to tenths of autonomous robots MORSE can be entirely controlled from the command line simulation scenes are generated from simple Python scripts. [7] MORSE consists of many numbers of sensors, actuators, environments which deal with the real robot design with these components and we can implement the real environment with all the required details.
- Blender: it is the free and open source 3D creation suite. It supports the entirety of the 3D pipeline-modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. Advanced users employ Blender’s API for Python scripting to customize the application and write specialized tools; often these are included in Blender’s future releases. Blender is well suited to individuals and small studios that benefit from its unified pipeline and responsive development process. [8]
- JACO robot arm: this is developed by Kinova at its state-of-the-art R&D department, is a revolutionary device designed for multiple professional applications. It is a leading product in a new generation of lightweight portable robotic tools that enables users to interact with their environment with complete safety, freedom, and effectiveness. [9] JACO has 6 degree of freedom, smooth movement without dangerous effect when used in the human applications, the materials used to form this arm are suitable and risk-free so that robot arm can be easily

used in many human patients applications.

We can explain the idea of IK in a simple way- first we have an object and a robot arm. The robot arm should follow the object wherever the object goes; this would be done by calculating the coordinates of the object relative to each one of the robotic arm joints as shown in Figure 1.

The robot arm must know those coordinates of the ball relative, and then calculate the rotation angle required for each robot joint to reach the ball. But in the simulation, the object will be created from the same plane of the robot arm axis so the implementation of (IK) need to study mainly the (axis compatibility, bones relationships, IK options and use of each one of these options). This will be much easier than the classic algorithms.

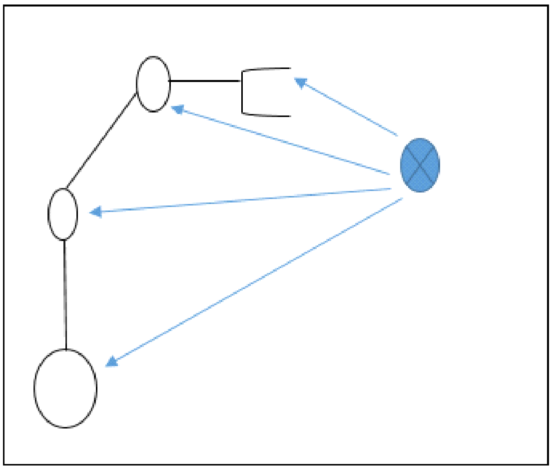

Fig. 1. *Simple Idea of Inverse Kinematics.*

RESULTS AND EXPERIMENTAL WORK

The JACO robot arm should be drawn in the simulation referred to the real design of this arm (the number of joints, links, and type of the material).

The arm consists of number of links and joints, each one of those links is controlled by one bone. When the bones rotate on the required axis this motion will be reflected on the body of the manipulator arm and this will lead to arm motion. Figure 2 shows the JACO arm with all the joints and links, and Figure 3 shows the joints of the hand with fingers.

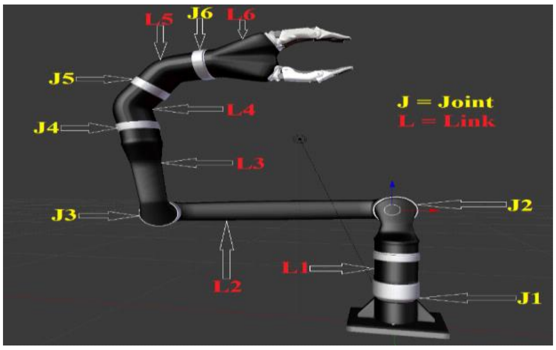

Fig. 2. *JACO Robot Arm Joints and Links.*

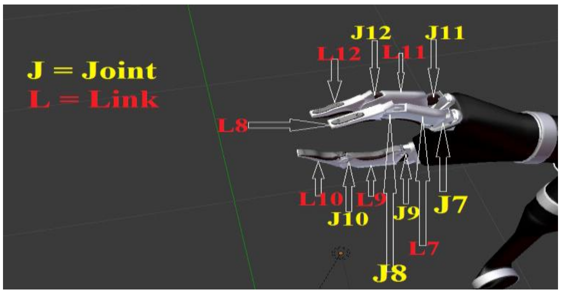

Fig. 3. *JACO Hand with Fingers Joints and Link.*

After the drawing of the arm in blender, we put the actuators which are responsible directly for the arm movement. The bones are moved according to the joint motion and distributed along each link. The bones have the ability to move vertically and horizontally and rotate as shown in Figure 4.

The Skeleton of the robot arm consists of a group from the connected bones. The IK is represented as the armature in the Blender program before being executed and the bones should be added to the skin of the arm as shown in the Figure 5. Each one of these bones should represent the link that will move according to the bone motion. The axis of the rotation for each one of the armature bones should be specified carefully, each joint must specify its type (rotation, translation) also the value of its maximum movement should be specified i.e. the maximum angel of the rotation in rotational joint and the maximum placement in the translation joint. The relationships between the bones should also be changed according to the required movement, the motion that will use its (parents–child) relationship, will be implemented to the bones and the mesh.

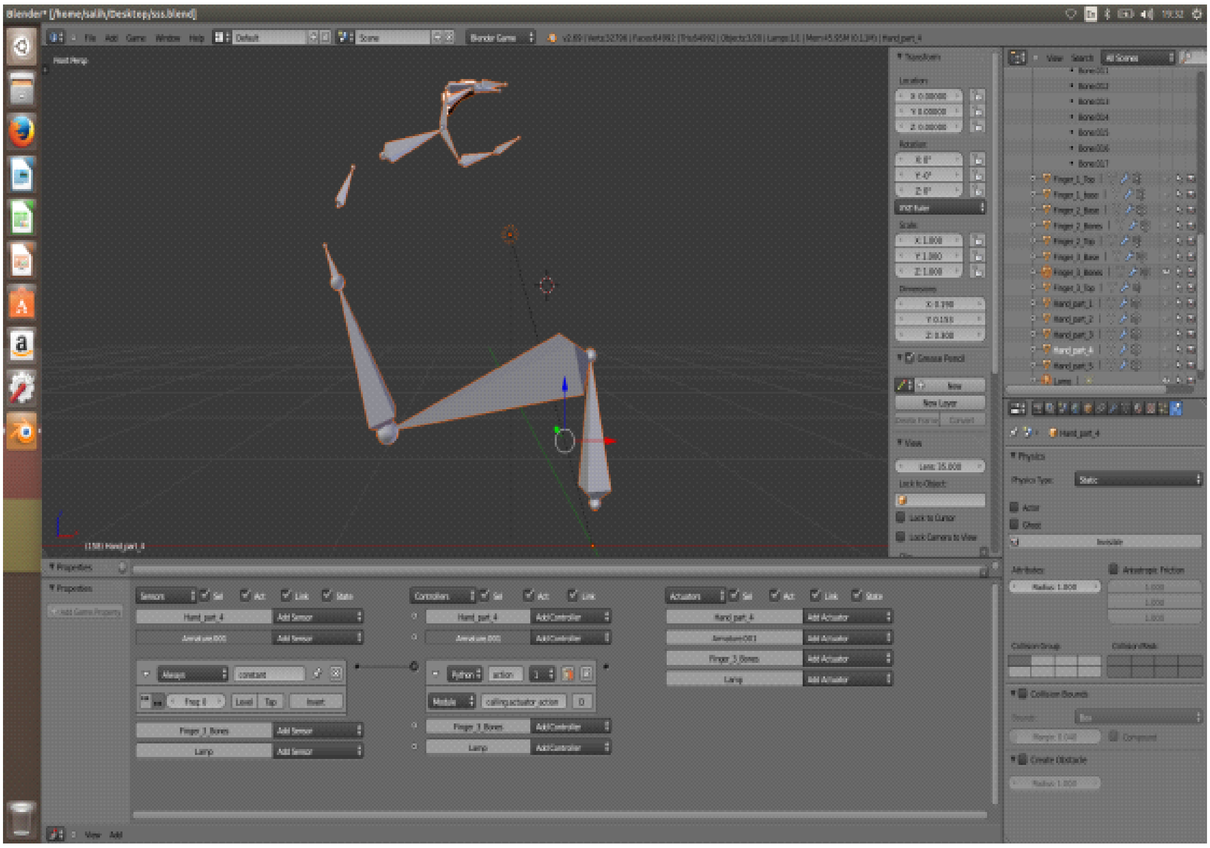

Fig. 4. *The Skeleton That Used for Moving the Arm.*

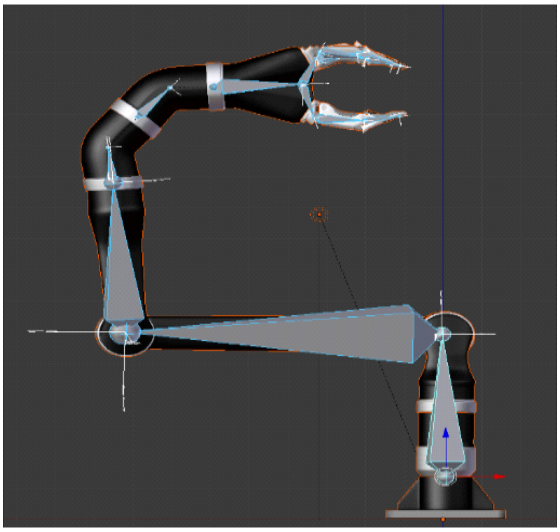

Fig. 5. *Bones Added to the Skin of JACO Arm.*

The number of bones that should be added to the arm depends on three factors:

- Number of joints and links which are used in the bones for direct motion.
- Number of axis rotation for each one of the joints.
- The kind of the joint motion (prismatic or rotation).

To implement IK, first step will start with the mesh. A cube in the center of the armature origin has been added as shown in Figure 6. This cube represented the target, according to whose movement, armature would move.

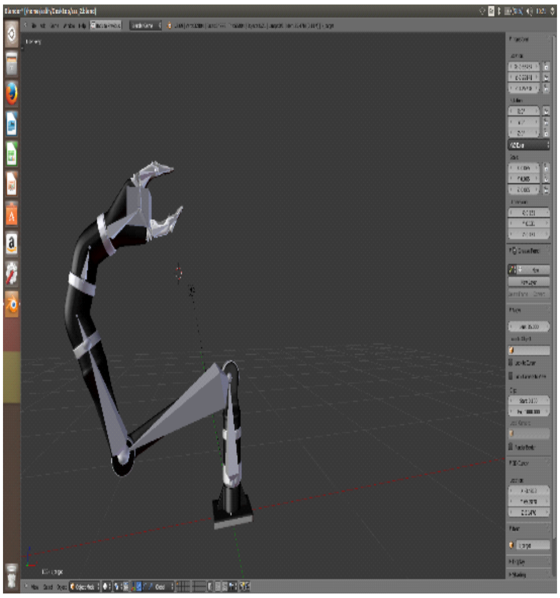

Fig. 6. *Cube in the Origin Axis of the Armature.*

The mesh (cube) should be in the same plane of axis with the armature bone which will be fixed and will work as reference for the other bones as shown in Figure 7. This step for making sure that the motion will become soft without the deformation in the arm links.

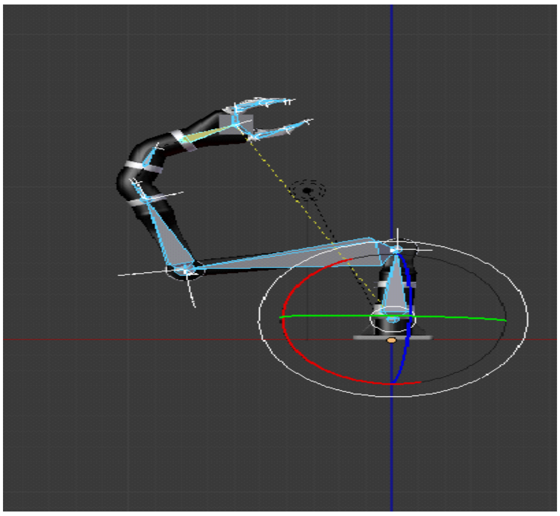

Fig. 7. *Plane Axis of the Cube Compatible with the Axis of Base Bone.*

We can of course, see the execution of IK when the cube moves in the direction as parent, so the armature should follow the cube as child, this happened when the whole system in pose mode. We can see this in Figure 8 which illustrates the results of this implementation that the bones will certainly move into "pose mode."

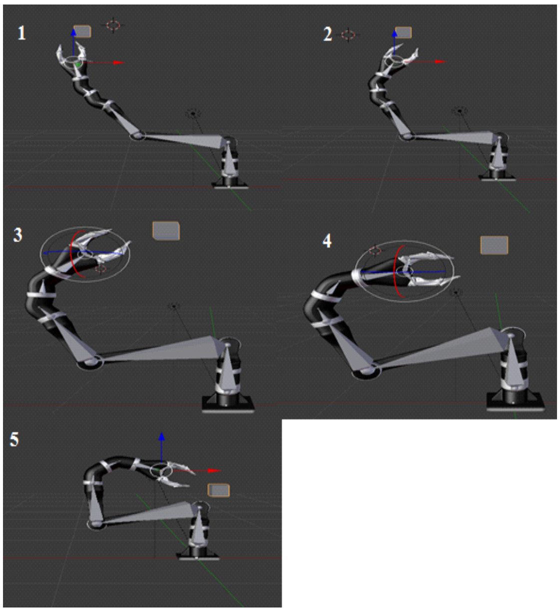

Fig. 8. *The Arm Follow the Created Object.*

CONCLUSION

In this paper, we used more efficient and easy way rather than the idea of classical calculations, for implementing IK, a time consuming and laborious process. Our methodology used in the simulation, is now highly preferred in designing operation over the old prototype building and testing approach. Along with this, the simulation of the JACO robotic arm was drawn and simulated in a very effective way that showed the real environments with the required results.

REFERENCES

1. Monzani J.-S., Baerlocher P., Boulic R., *et al.* Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargeting, *Eurographics.* 2000; 19(3).
2. Grochow K., Martin S.L., Hertzmann A., *et al.* Style-based inverse kinematics ACM transactions on graphics (TOG), *Proceedings of ACM SIGGRAPH 2004*. Volume 23 Issue 3.

3. Der K.G., Sumner R.W., Popovíc R.W. Inverse kinematics for reduced deformable models, *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2006*. Volume 25 Issue 3, July 2006. Computer Science and Artificial Intelligence Laboratory ETH Zurich, Massachusetts Institute of Technology.
4. Sumner R.W., Zwicker M., Gotsman C., *et al*. Mesh-Based Inverse Kinematics, appear in *SIGGRAPH 2005*. Computer Science and Articial Intelligence Laboratory Massachusetts Institute of Technology, Harvard University.
5. Aristidou A., Lasenby J. Technical Report Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver September 2009, Available on: http://www.researchgate.net/publication/273166356.
6. Echeverria G., Lassabe N., Degroote A., *et al*."Modular Open Robots Simulation Engine: MORSE Robotics and Automation (ICRA), *2011 IEEE International Conference*. 2011.
7. https://www.openrobots.org/morse/doc/latest/what_is_morse.html.
8. http://www.blender.org.
9. "JACO Robot Arm" By admin on June 1, 2011, Robotic Magazine Available on http://www.roboticmagazine.com/robot-review/jaco-robot-arm-2.