

FINAL PROJECT REPORT

IMPLEMENTATION OF FABRIK

INTRODUCTION

This project tries to implement a simple solution proposed for the inverse kinematics problem – Forward and Backward Reaching Inverse Kinematics. For the purpose of not complicating the implementation an unconstrained solution is implemented wherein the DOFs are not restricted by any constraints. The implementation involves two sub-chains with four joints and five links connected to one main chain at a common joint. The two sub-chains have two end-effectors that are used to move around the chain. The main chain is fixed at the origin and has the same four joints with five links.

MODULES

Methods for building the scene

The following are the methods that are used to create the chain with the links and joints along with adding an end effector at the end of the two sub chains.

`build_scene`

The whole chain along with the end effectors are added here. This method reads the chain data generated during the initialization of the object and calls `add_fingers` method to generate the chain with the links and joints. The number of joints, links and their length are stored in the data of the object. Finally, two `SnManipulators` are added to the end effectors in order to provide the user a way to move the chain.

`add_fingers`

This method is used to add the links between each joint. Two points are passed to this method within which a cylinder primitive is added. At the point of the joint a sphere is added to signify that the joint is unconstrained and can move around in any direction.

`change_end_effector`

This is the callback method for the end-effector manipulators. When the end effectors are moved around by the user, this method determines which end-effector is moved and calls the `update()` method to adjust the links accordingly and following which the scene is rendered again with the new chain configurations.

Methods to perform FABRIK

The following methods implement the complete FABRIK algorithm to solve the IK of the two sub chains and the main chain.

`update`

The end effector which is changed and the point to which it was changed are passed to this method. This method in turn first calls the `fabrik_backward` to determine the new configuration of the sub-chain to which this end-effector belongs to. With the new points of the sub-chain it then calls `fabrik` method

to update the configuration of the main chain. Finally, it calls `fabrik_forward` for both the sub chains to get the new points of the links and the joints. Following which the scene is rendered again.

`fabrik`

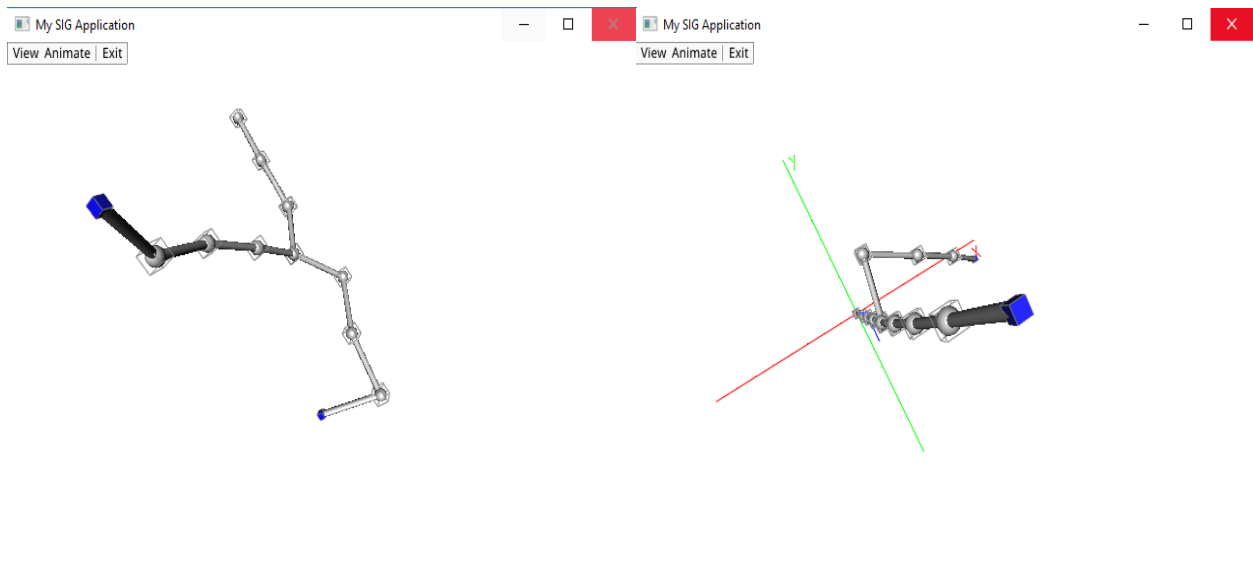
This method performs first backward reaching and then forward reaching for the specified chain. This is called only for the main chain to which both forward reaching and backward reaching is necessary.

`fabrik_forward` and `fabrik_backward`

As the name indicates these two methods perform forward reaching and backward reaching based on the target points and the origin points specified respectively. The part of the chain is also specified as well.

RESULTS

Following are some of the chain configurations for different position of the end effectors.



EVALUATION

For evaluation purposes two different chains were considered. One chain with no sub chains and having four links and one end effector. The other chain as described above. Following were the results:

	1 end-effector	2 end-effector
No of iterations	Varies from 1-88	Varies from 8-100
Time taken for the chain computation	Varies from 0.00004288 0.00124288	Varies from 0.00238862- 0.242239