Pick and Place Bot using V-REP

Based on inverse kinematics and path planning

Project Report

EKLAVYA MENTORSHIP PROGRAMME

At

SOCIETY OF ROBOTICS AND AUTOMATION,
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
MUMBAI

JUNE 2020

ACKNOWLEDGEMENT

We thank the Society of Robotics and Automation (SRA), VJTI, for providing us with this opportunity and extending their support to us. We would also like to extend our gratitude to our mentors who helped us in making the project a success.

MENTORS

PRITI JAIN

MANAS THORAT

TEAM MEMBERS

TEJAL BEDMUTHA :
tezzb18@gmail.com
Contact no. : 9653341728

RESHMIKA SREENATH NAMBIAR :
reshmikasnambiar@gmail.com
Contact no. : 9967231124

TABLE OF CONTENTS

PROJECT OVERVIEW

This project is an Inverse Kinematics based simulation in V-REP software. Simulation is the process of mathematical modelling, performed on a computer, which is designed to predict the behaviour of or the outcome of a real-world or physical system.

They allow us to check the reliability of chosen mathematical models, hardware and working of the code. Simulation modeling solves real-world problems safely and efficiently. It provides an important method of analysis which is easily verified, communicated, and understood. Across industries and disciplines, simulation modeling provides valuable solutions by giving clear insights into complex systems. There are different softwares available for simulation. We have chosen V-REP developed by Coppelia Robotics for our project. The software already has some robots, robotic arms, conveyors, etc. inbuilt. This project uses KUKA LBR iiwa 14 R820 non-mobile robot. This paper covers the application of inverse kinematics and path-planning in simulating a robot. Language used for scripting is LUA.

The scene can be accessed from the following GitHub link :
https://github.com/Tejal-19/simbotix

To see the video of the final scene, go to
https://youtu.be/Pa8bjI16Gbc

1. INTRODUCTION

The ability to analyze the model as it runs sets simulation modeling apart from other methods, such as those using Excel or linear programming. By being able to inspect processes and interact with a simulation model in action, both understanding and trust are quickly built. The uses of simulation in business are varied and it is often utilized when conducting experiments on a real system is impossible or impractical, often because of cost or time. Therefore, we have used it for simulating our robot to pick and place white and black boxes to and from conveyors. We have set the joints to inverse kinematics mode in order to achieve the required motion.

1.1 Inverse Kinematics

Inverse kinematics is the mathematical process of calculating the variable joint parameters needed to place the end of a kinematic chain, such as a robot manipulator or animation character's skeleton, in a given position and orientation relative to the start of the chain. Given joint parameters, the position and orientation of the chain's end, e.g. the hand of the character or robot, can typically be calculated directly using multiple applications of trigonometric formulas, a process known as forward kinematics. However, the reverse operation is, in general, much more challenging.[1]

Inverse kinematics transforms the motion plan into joint actuator trajectories for the robot. For a 6-DoF robot (for example, 6 resolute joints) moving in 3D space (with 3 position degrees of freedom, and 3 rotational degrees of freedom). If the degrees of freedom of the robot exceeds the degrees of freedom of the end-effector, for example with a 7 DoF robot with 7 resolute joints, then there exist infinitely many solutions to the IK problem, and an analytical solution does not exist. Further extending this example, it is possible to fix one joint and analytically solve for the other joints, but perhaps a better solution is offered by numerical methods , which can instead optimize a solution given additional preferences (costs in an optimization problem).

An analytic solution to an inverse kinematics problem is a closed-form expression that takes the end-effector pose as input and gives joint positions as output,

$$y = f(x)$$

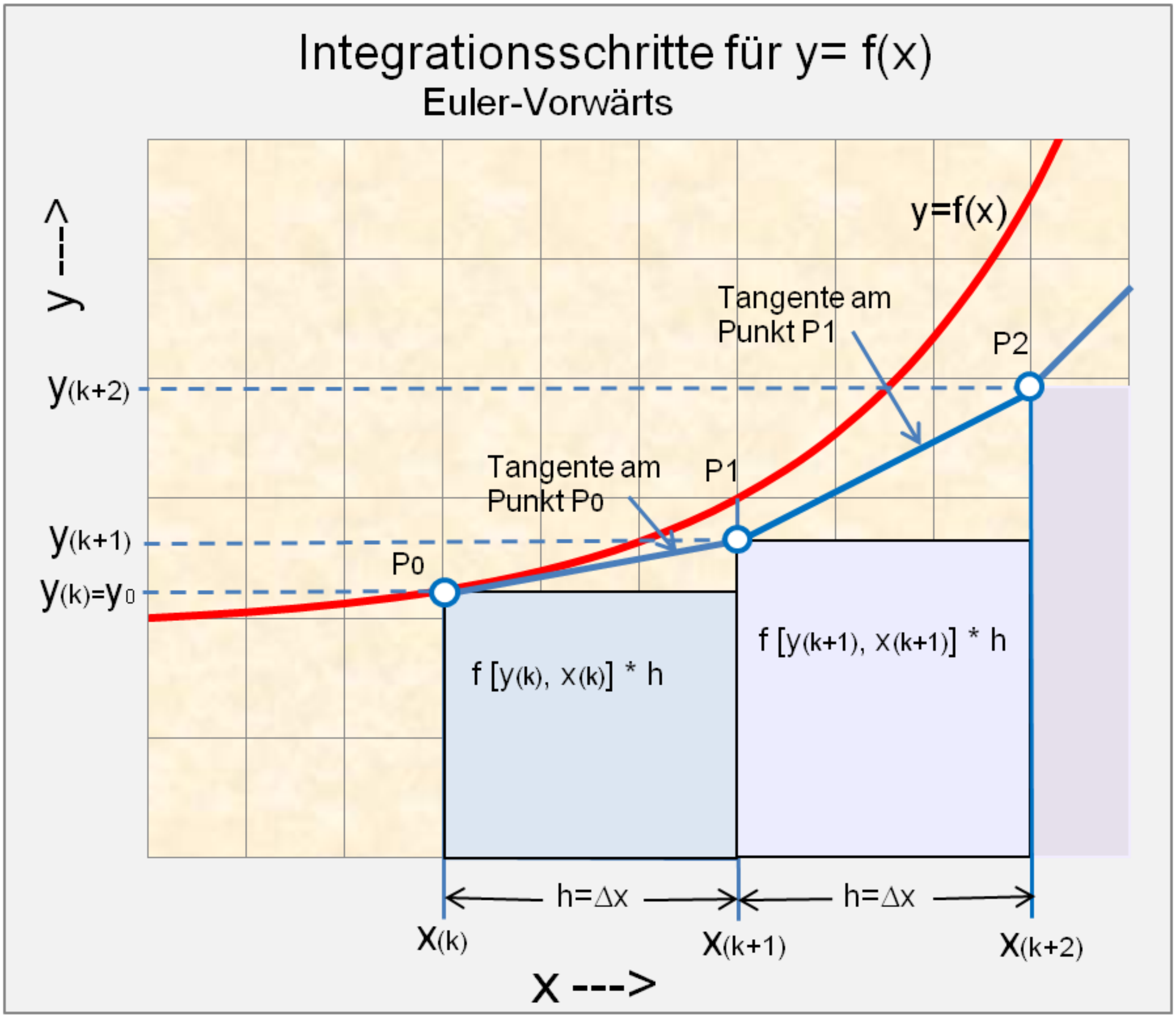

Figure 1 : Integration Steps for Euler-Vorwärts showing tangents at points P_0 and P_1

Analytical inverse kinematics solvers can be significantly faster than numerical solvers and provide more than one solution, but only a finite number of solutions, for a given end-effector pose.

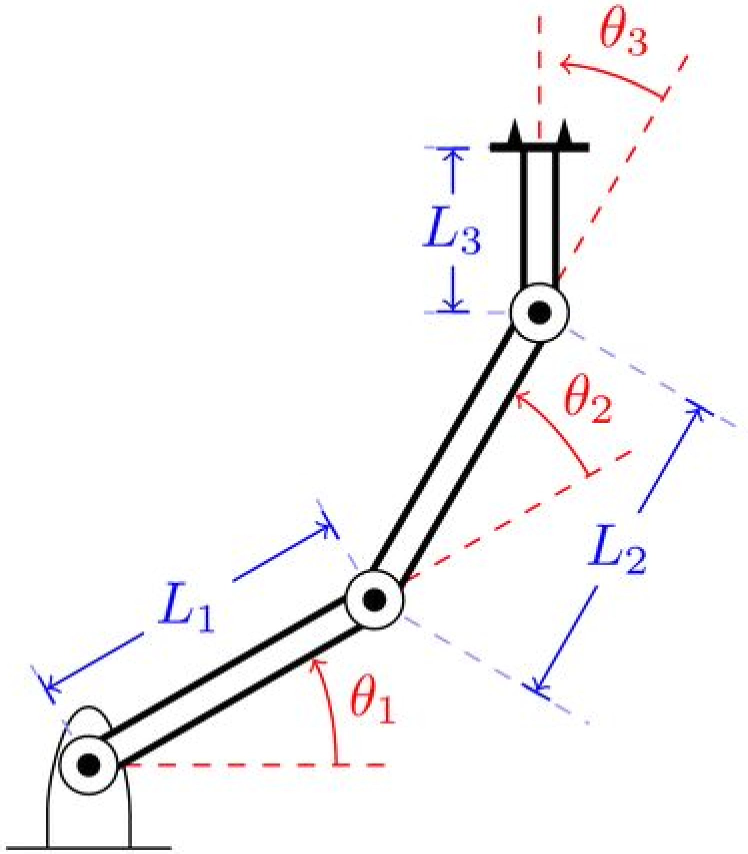

Figure 2 : A three-link robot arm with a desired end effector location and the angles θ_1, θ_2 and θ_3

Working of IK group : The Inverse Kinematics function/algorithm takes a target position as the input, and calculates the pose required for the end effector to reach the target position — the pose is the output. The input and output are switched between FK and IK.[2]

1.2 Motion Planning

Determining the movement of a robot so that its end-effectors move from an initial configuration to a desired configuration is known as motion planning. The goals of motion-planning are – to follow collision-free trajectories, and to make the robot reach the goal location as fast as possible . The problem of motion planning can be stated as follows:

- Given –
    1. A start pose of the robot
    2. A desired goal pose
    3. A geometric description of the robot .
    4. A geometric description of the world.
- Find a path that moves the robot gradually from start to goal while never touching any obstacle.[3]

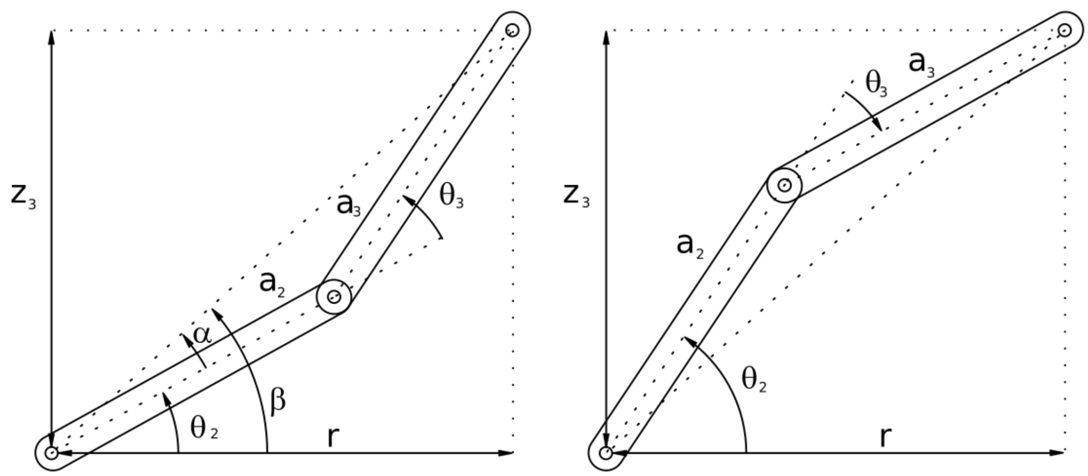

Figure 3 : An example of an Articulated Robotic arm using motion-planning

2. METHODS

Different methods can be used to do the task.

2.1 Options Available

I. Any other software can be used. For example Gazebo, MATLAB, etc.
II. Any other robot can be used instead of the one suggested.
III. Similarly, any other gripper can also be used.
IV. Vision sensors can be used to detect the color of the boxes.
V. Different types of proximity sensors can be used.
VI. Multi-coloured boxes can be generated in place of the white and black ones.
VII. Path can be defined exclusively with the path function or using OMPL.

2.2 Methods Applied

I. We used V-REP as it is highly user friendly, has many features, is free (some versions are paid), has inbuilt robots, grippers and environment set up.
II. We used the KUKA LBR iiwa 14 R820 non-mobile robot as it was giving appropriate motion and movement of joints (amongst the ones we sampled).
III. ROBOTIQ_85 gripper is used as it is simpler and efficient for the task (multiple fingers is not optimal for this task).
IV. Proximity sensors are used to detect the box and boolList to identify the color.
V. Black and white coloured is chosen to make the set up more industry-specific.
VI. We used the IK method for the kinematics module and defined the path in the code.

3. EXPERIMENTS AND RESULTS

3.1 Procedure Followed

This project is based on simulating the inbuilt KUKA robot. It has an attached ROBOTIQ_85 gripper aimed to pick and place the colored boxes from the conveyor belt to the respective customizable conveyors. The joints of the robot are set to the IK mode to achieve the required motion and values of pos. angles are set accordingly. Three dummies are introduced and named as Tip, Target, and Connector. All the three are positioned at the gripper, the tip is shifted above the gripper, the connector's position is retained and the target is placed near the base.

The Tip and the Target are linked and added as an IK group. Tip is made the child of the gripper, connector of the link below it and the target of the base. In order to reduce the distance between the Tip and Target, pos. angles of joints are set. In this way the Bot is set.

Environment is set up which includes a conveyor belt and two customizable conveyors. White and black boxes are generated randomly on the conveyor belt (50% chance of generation each).[4] The job of the robot is to detect the box with the help of the proximity sensors, identify the color and place them accordingly. For identification, boolList is used. BoolList is the list which saves boolean values. We have set white as the good color. So if a white box is detected, true will be saved in boolList and a specific condition will be applied. If black, false will be saved and the other condition will be applied. The entire scene is scripted in LUA. For the path planning, dummies are inserted and placed near the customizable conveyors and the path is created from the target to the dummies.[5] The next job was to add and function the gripper and with this all the elements are set up.[6]

3.2 Trials and Errors

This majorly involved the scripting part as LUA is used. Others are as follows :

- Choosing the ideal robot and the gripper as there was a lot redundant motion generated by other robots
- Setting up the environment
- Setting the pos. angles
- Functioning of the gripper

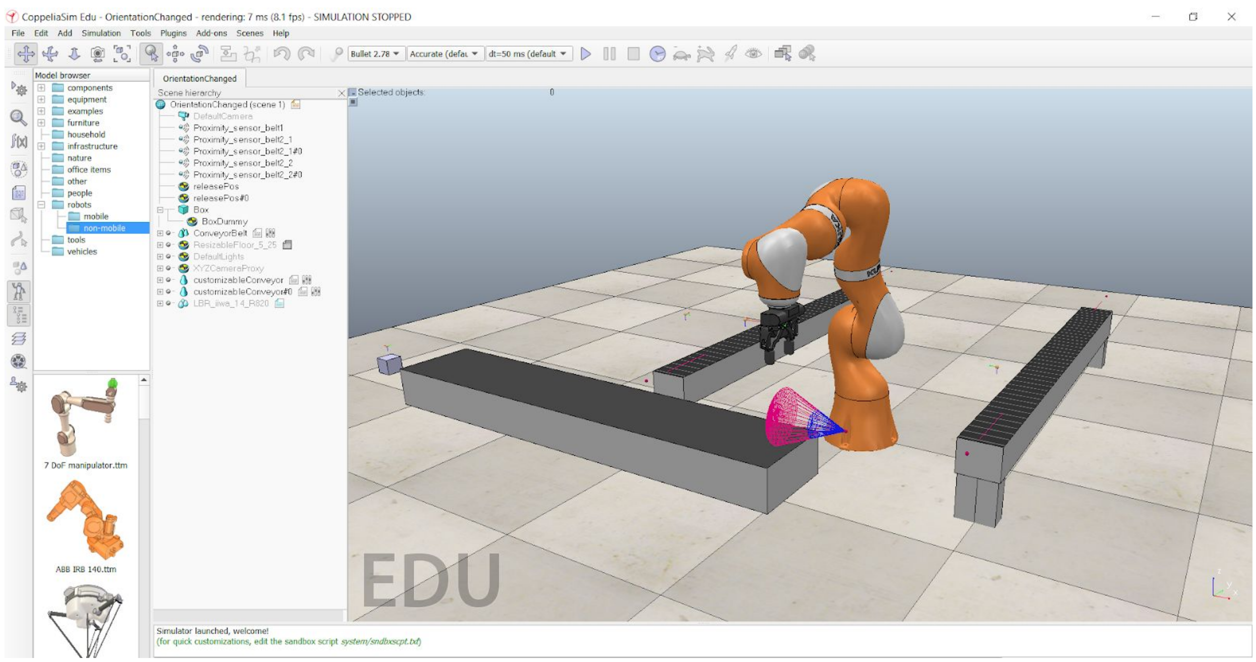

Figure 4 : The scene before starting the simulation

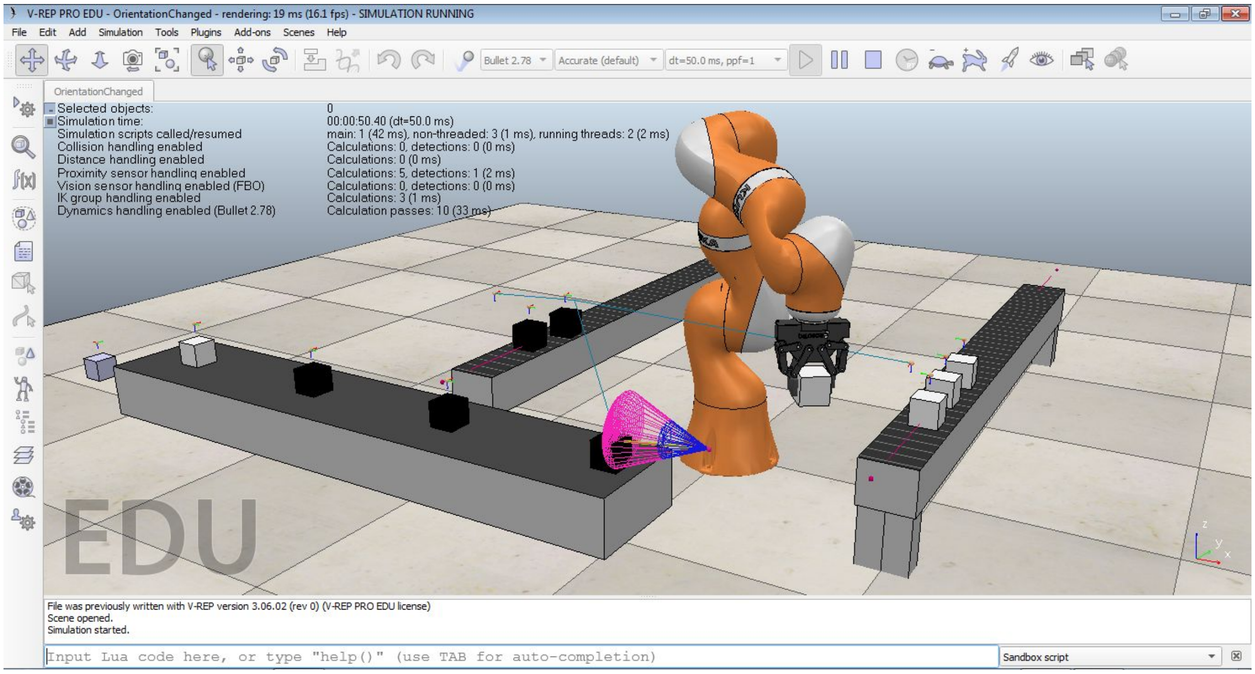

Figure 5 : The scene during the simulation; black boxes are placed on the left conveyor and the white boxes are placed on the right one.

4. CONCLUSIONS AND FUTURE WORK

We are able to achieve the goals we defined for this project. The robot performs the task pretty well without any redundant motion and avoids the obstacles. It follows the defined path and the gripper also functions properly.

4.1 Future Aspects of the Project

The simulation can be further developed by adding more conveyor belt systems using multiple robotic arms. Ragnar robots can also be used. We can add objects of different colours instead of just black and white. Objects can also be segregated based on shapes as well instead of just colours. Essentially, we can make a fully integrated complex conveyor belt system consisting of many different types of conveyor belts and robots. Vision sensors can be used to detect the boxes and extract the coordinates and find the depth as well. Environment can also be set up differently and obstacles can also be introduced.

4.2 Real Life Application

Conveyor belt systems have an extensive industrial application, especially in manufacturing industries. A scaled up version of this project can be used to segregate items, like waste in waste management systems, or different products in packaging industries.

REFERENCES

1. Donald L. Pieper, “The Kinematics of Manipulators Under Computer Control”, PhD thesis, Stanford University, Department of Mechanical Engineering, October 24, 1968
2. Overview of Inverse Kinematics - Unity3DAnimation - Medium, https://medium.com/unity3danimation/overview-of-inverse-kinematics-9769a43ba956
3. Robot Motion Planning, http://ais.informatik.uni-freiburg.de/teaching/ss11/robotics/slides/18-robot-motion-planning.pdf
4. V-REP Tutorial: Inverse Kinematics | Part 1/2, https://www.youtube.com/watch?v=JUiSZinyH1c
5. V-REP Tutorial: Inverse Kinematics | Part 2/2, https://www.youtube.com/watch?v=Sf8E16EK6BQ

6. Pick and Place Application with KUKA KR16 robot using V-REP (CoppeliaSim) and MATLAB, https://www.youtube.com/watch?v=CVoV08T0Aqo