Robot Arm Manipulation Using Depth Cameras and Inverse Kinematics

Akhilesh Mishra[1] **Dr. Oscar Meruvia-Pastor[2]**
Department of Computer Science
Memorial University of Newfoundland.

1. Introduction

- New approach to manipulate a robotic arm, using a depth camera to capture user input and inverse kinematics to define the joint motion of the robotic arm, is presented.
- Exiting manipulators make use of a master controller (Fig 1b), which controls each joint of the arm separately, so It is harder to learn and requires a lot of training. There are simulators which help users train on a particular robotic arm, e.g. GRI Simulations Inc.'s manipulator trainer, fig 1a.[3]

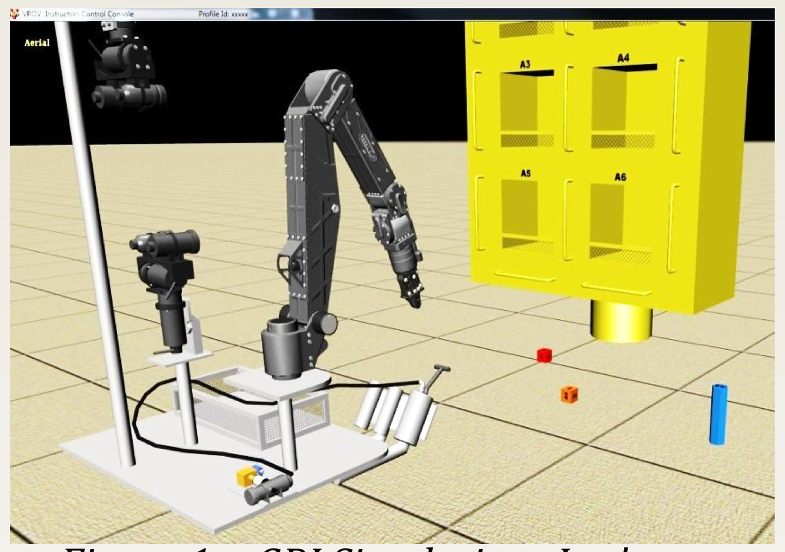

Figure 1a: GRI Simulations Inc.'s Manipulator Trainer.

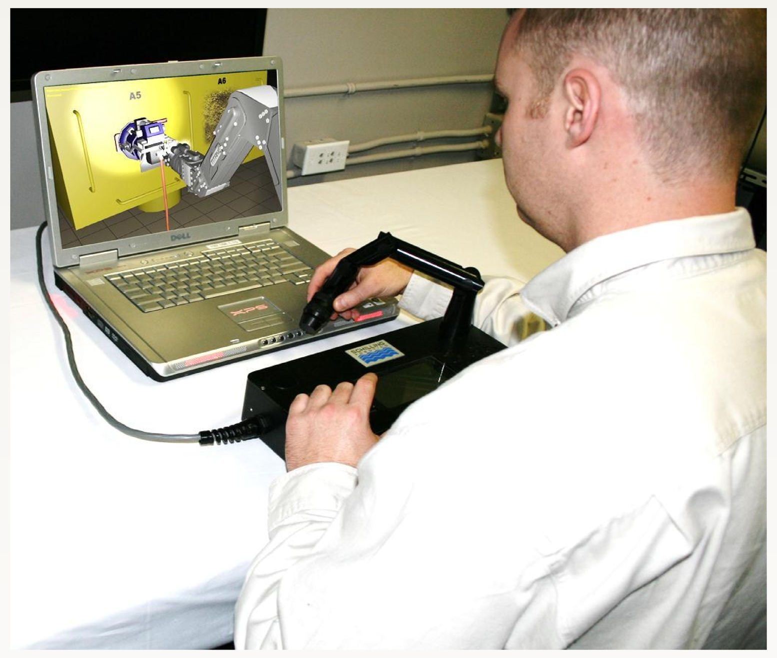

Figure 1b: Master controller interacting with GRI simulator

Image Courtesy: GRI Simulations Inc.

- Simulators are really helpful for training purposes. However, they can be quite expensive.
- With the presented approach the user just needs to point towards the target and the robotic arm reaches the target itself which makes the manipulation of the robotic arm easier.
- Simple speech or gesture commands could be added to open and close the end-effector, which makes the interaction with the robotic arm more intuitive.

2. System Overview

The system consists of the 5 modules as shown below

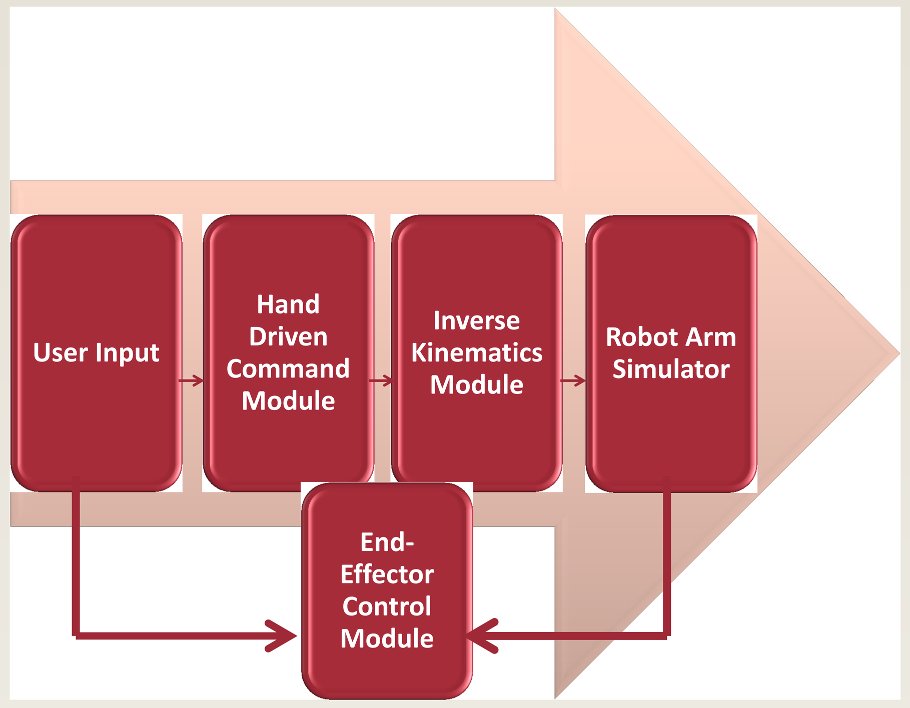

Fig 2: System Overview.

- *User Input Module:* This module collects the user input, the user input could be a hand movement or a speech command.
- *Hand Driven command Module:* This module converts the user hand movement into valid target points for the next module.
- *Inverse Kinematics Module:* This module calculates the angles with which each joint should rotate to reach the target point.
- *Robot Arm simulator Module:* This module applies the calculated angles to the robotic arm simulator.
- *End-effector Control Module:* The user can control the end-effector by a speech or a gesture command.

2.1. Hand Driven Command Module

- Intel's Creative Depth Camera [4] is used to obtain the user's hand position. The hand position is used to control a target ball which can move in 3D space.
- The input space is divided into grids as shown in figure 3. The user can move the ball left, right, near, far, up and down. Placing the hand at the central grid prevents the motion.

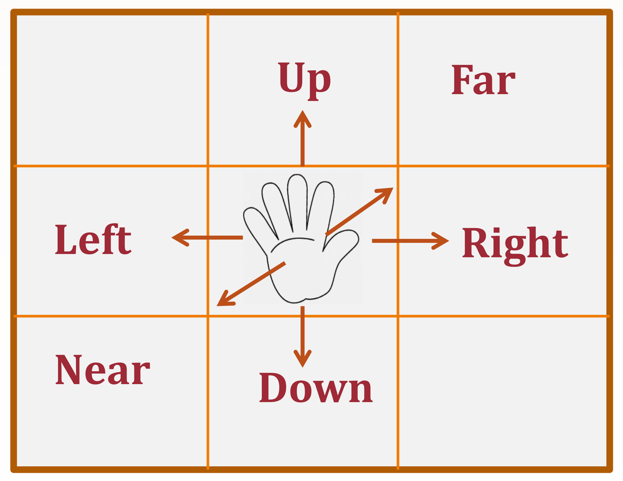

Figure 3: Grids showing the 3D input space

2.2 Inverse Kinematics module

- Inverse Kinematics is a technique in which the user specifies the target position and the joint angles are computed by the algorithm. Figure 4a shows a simple 2D 3 Link Manipulator. The user specifies the target position (x, y) and the joint angles Θ_1, Θ_2 and Θ_3 are calculated using the IK algorithm.

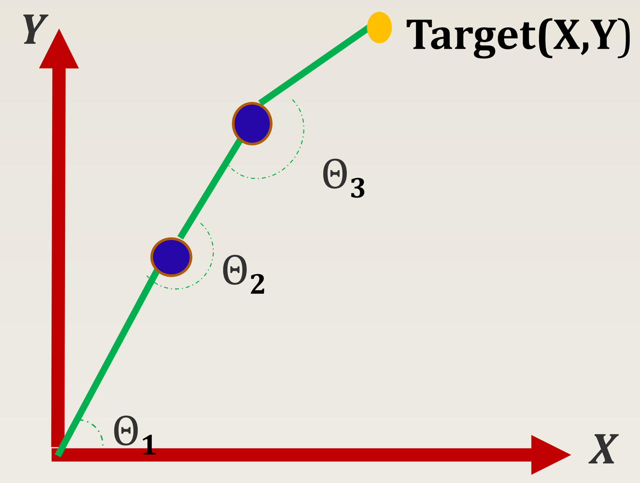

Figure 4a: 3 Link Manipulator

- Figure 4b shows the simulator developed in OpenGL for this research. CCD algorithm was used for solving IK.[1,2]

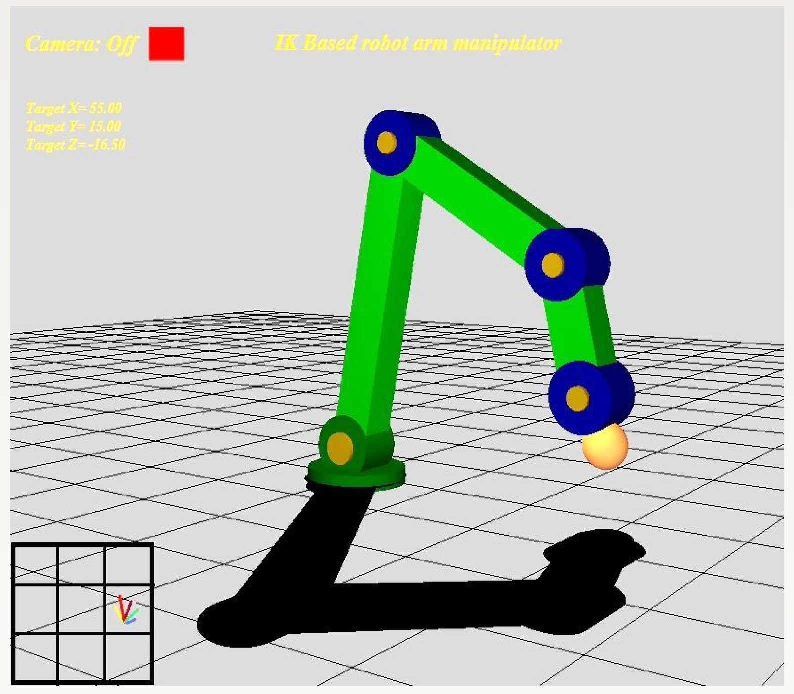

Figure 4b: IK based robot arm simulator

2.2.1 Cyclic Coordinate Descent

CCD is an iterative Inverse Kinematics algorithm. It loops through the joints from end to root, each joint gets rotated so as to bring the end effector as close to the target as possible .

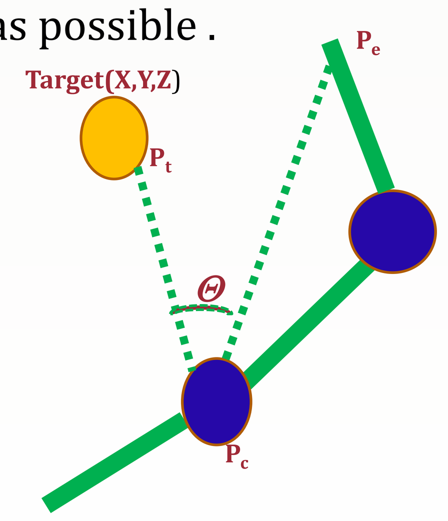

Figure 5: CCD algorithm, the end-effector rotates to make Θ zero.

$$\cos(\Theta) = \frac{P_e - P_c}{||P_e - P_c||} \circ \frac{P_t - P_c}{||P_t - P_c||} \quad (1)$$

$$\overrightarrow{dir} = \frac{P_e - P_c}{||P_e - P_c||} \text{ X } \frac{P_t - P_c}{||P_t - P_c||} \quad (2)$$

Equation (1) returns the angle of rotation and (2) returns the direction of rotation. As shown in Figure 5, To reach the target the equations (1) and (2) are solved for each joint until the difference between the end-effector and target is zero or the number of iterations has reached its limit

3. Evaluation

User study will be conducted to compare the existing approach and the presented approach. Users will be asked to perform a simple pick and drop tasks, using both a controller and a depth camera.

4. Acknowledgements

Research funded by the Research & Development Corporation (RDC) of Newfoundland & Labrador

5. References

1. Wang, L. C., & Chen, C. C. (1991). A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. Robotics and Automation, IEEE Transactions on, 7(4), 489-499.
2. Mukundan, R, & Member, S. (2008). A Fast Inverse Kinematics Solution for an n -link Joint Chain, (Icita), 349–354.
3. GRI Simulations Inc. In Manipulator Trainer. Retrieved April 2014, from http://www.grisim.com/products.html#ManipTrainer
4. Intel. In Intel® Perceptual Computing SDK 2013, Retrieved April 2014, from http://software.intel.com/en-us/vcsource/tools/perceptual-computing-sdk

Contact:
[1]akm565@mun.ca, [2]oscar@mun.ca