# A New Approach for Robot Arm Manipulation Using Depth Cameras and Inverse Kinematics

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

## Introduction

- A new robotic arm manipulation technique for offshore environments such as ocean-floor exploration is presented.
- Existing manipulators such as Titan IV work on forward kinematics where the user needs to control each joint of the robotic arm manually using a joystick type controller, which requires a lot of training. And, as these robotic arms are often deployed in rough conditions, user training is really expensive.
- There are, however, simulators which train users on particular manipulator arms. For example, GRI Simulation's Manipulator trainer (see Figure 1a) [3] does this job for a variety of manipulators. Commercial simulators work using forward kinematics and require a high degree of skill from the operator and thus extensive training.

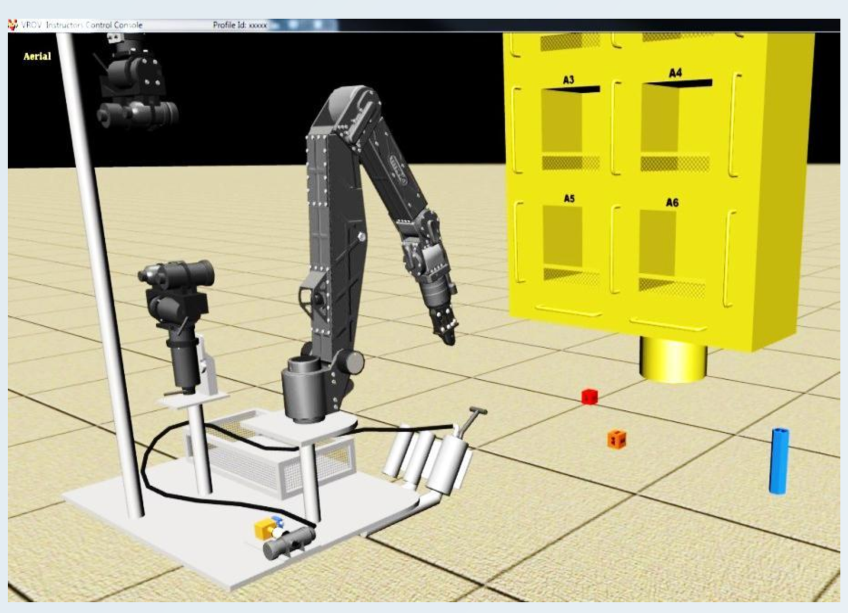

*Figure 1a: GRI Simulations Inc.'s Manipulator Trainer.*

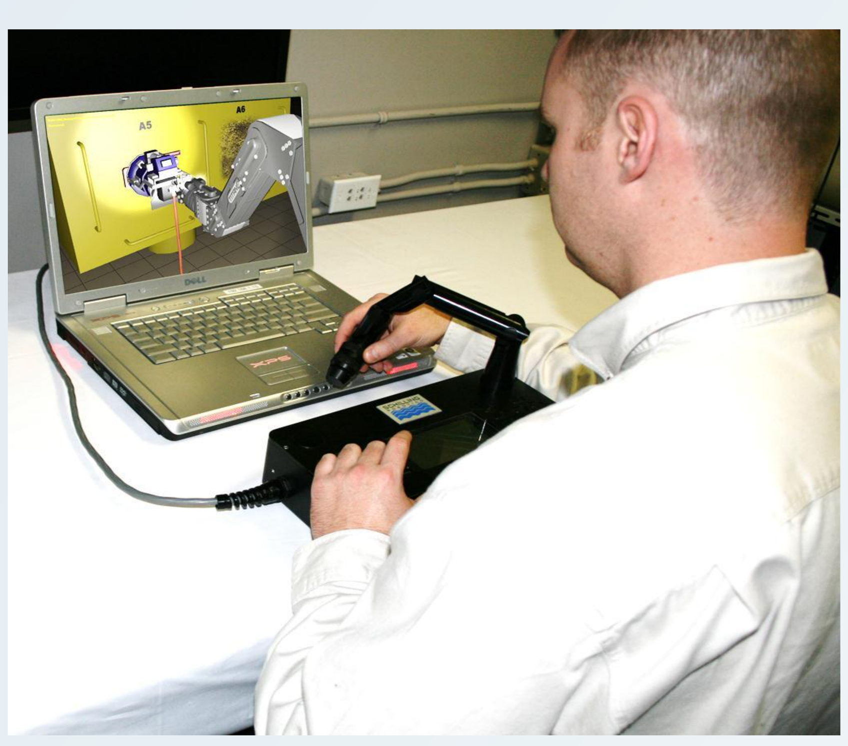

*Figure 1b: Master controller interacting with GRI simulator*

*Image Courtesy: GRI Simulations Inc.*

- This poster presents an approach in which the user just needs to point to the target and the robotic arm reaches the target on its own using Inverse kinematics.
- The user input is captured using a depth camera and passed to the controller module which calculates the joint angles or controls the end-effector.
- Speech and gestures commands were added so that the user can control the end-effector to pick, drop and rotate the objects, more intuitively and easily.

## System Overview

The system consists of the 5 modules as shown below

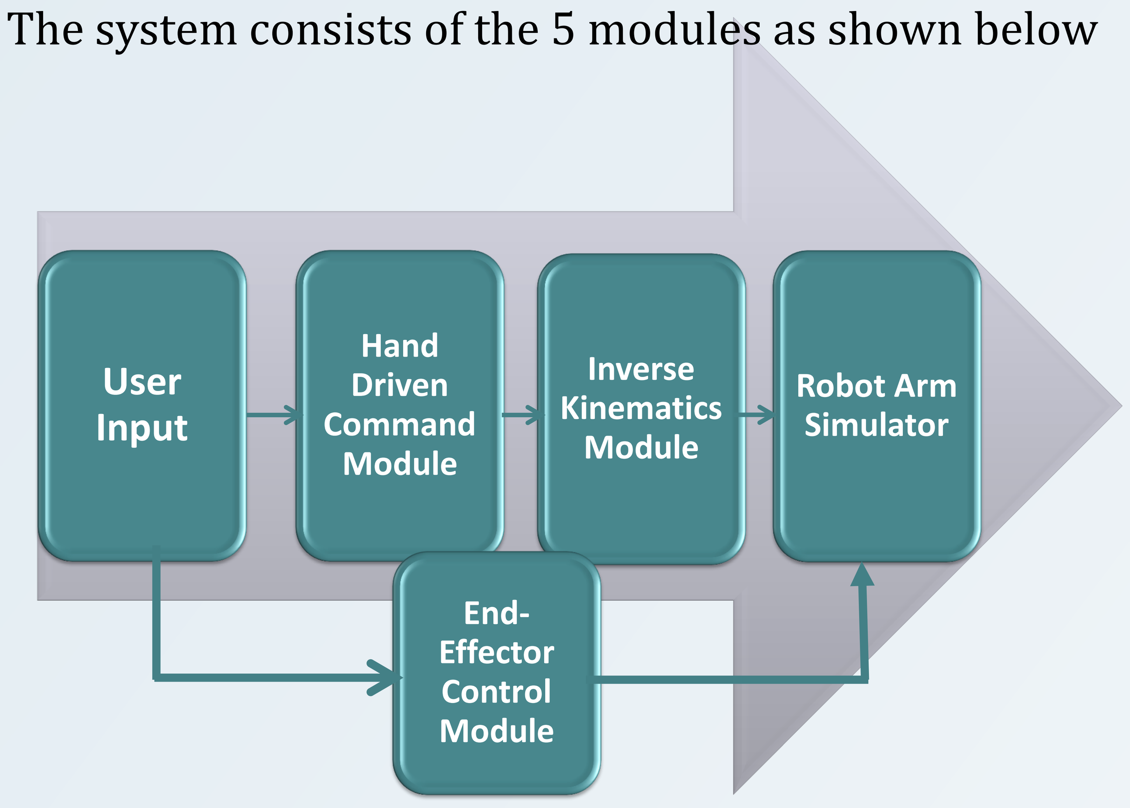


*Figure 2: System Overview.*

- *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

## Hand Driven Module

- Intel's Creative Depth Camera [4] (Fig-3a) 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 as shown in Figure 3b.

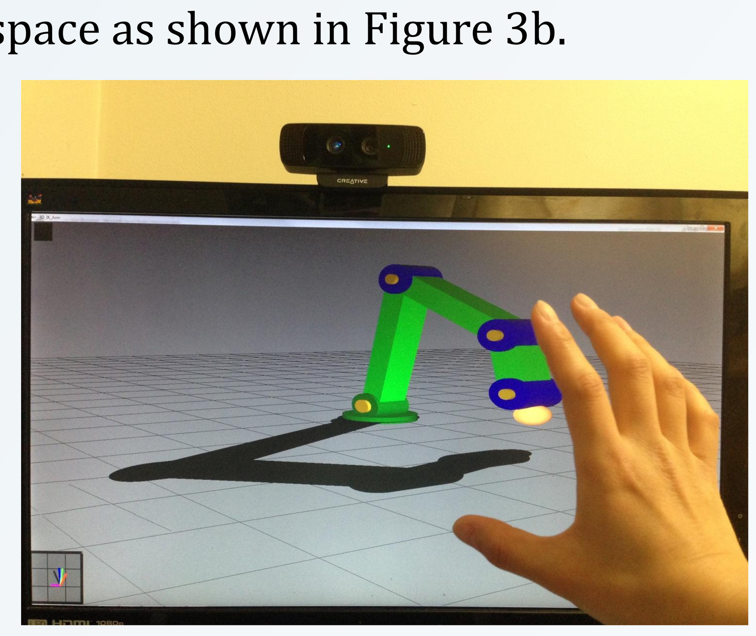

*Figure 3a: Depth camera used for this research*

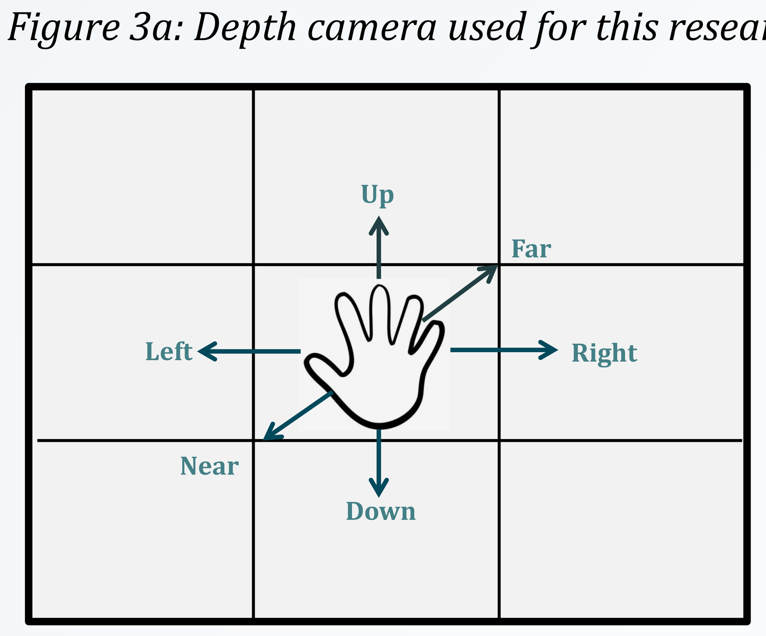


*Figure 3b: Grids showing the 3D input space*

- The input space is divided into grids as shown in Figure 3b. The user can move the ball left, right, near, far, up and down. Placing the hand at the central grid prevents the motion.

## Inverse Kinematics Module

- The output of Hand driven module is the input for Inverse kinematics module. As shown in Figure 4a, the user specifies the target position and the joint angles $\Theta_1$, $\Theta_2$ and $\Theta_3$ are calculated using the IK algorithm.

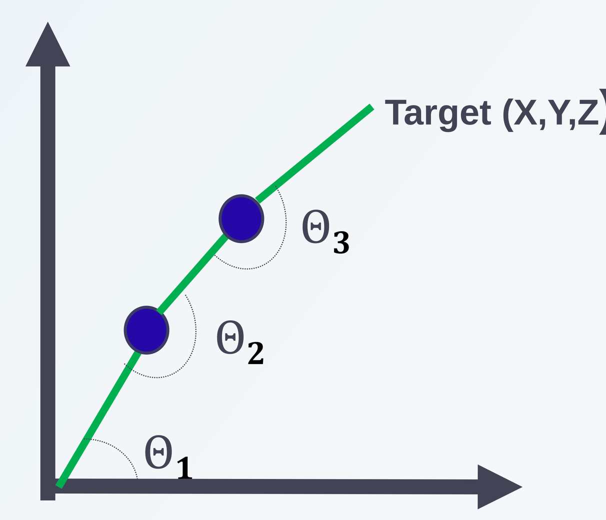


*Figure 4a: 3 Link Manipulator*

- Figure 4b shows the simulator developed in OpenGL for this research. The yellow ball is the hand driven ball that user controls to specify a target point.

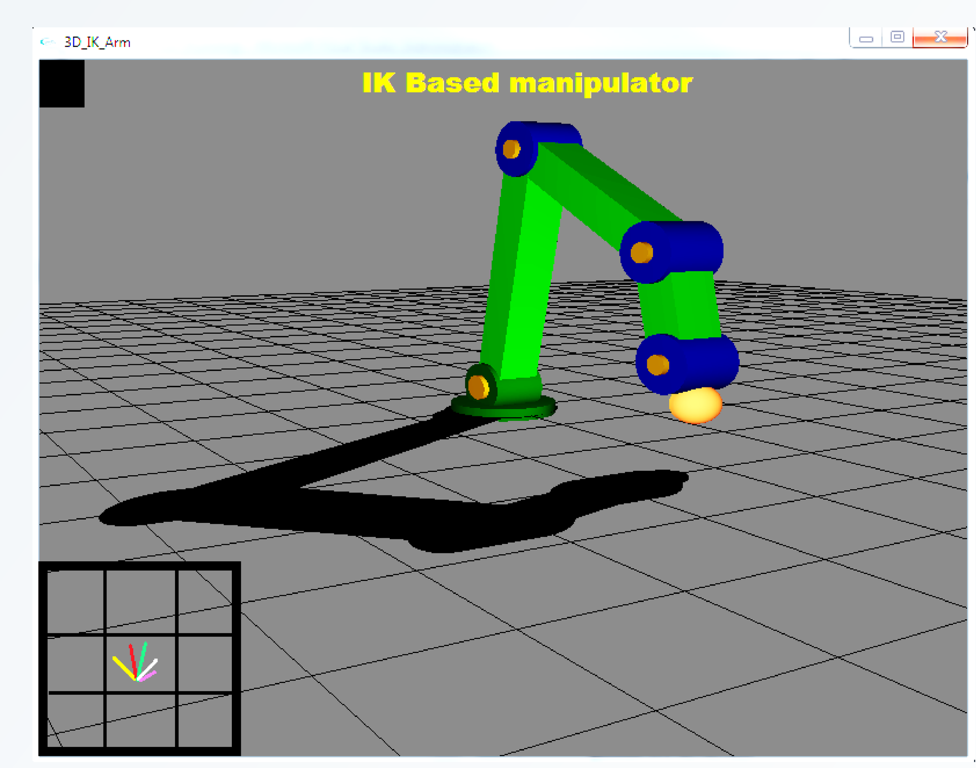


*Figure 4b: IK based robot arm simulator*

- Cyclic coordinate descent (CCD) algorithm was used for this research. [1,2]

## 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.

## Acknowledgements

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

## 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