SAMPLING OF 3DOF ROBOT MANIPULATOR JOINT-LIMITS FOR HAPTIC FEEDBACK

KEVIN HUANG, YUN-HSUAN SU, MAHMOUD KHALIL, DANIEL MELESSE, RAHUL MITRA

OUTLINE

- Introduction
- System Components
- Methods
  - Algorithms
  - Results
- Conclusion and future work

TELEOPERATION

Teleoperated robots used in surgery

- Remote control of robot by a human operator
- Teleoperated robots have numerous applications – assisting in medical procedure, rover control in space, micro-assembly etc.

INTRODUCTION

- Teleoperation - robot proxies can extend human-control to uncertain and dangerous task environments.
- Key is to build seamless and intuitive interfaces for remote control of sophisticated proxies.
- Controlling complex teleoperated robots can be confusing
    - Slave vs. Master kinematics

INTRODUCTION

Input device

Remote device

- The kinematic complexity may result in situations that are frustrating and confusing for the human master.
- **Joint limits might be inconsistent**
- The master/input device may freely command configurations that violate joint limit constraints of the slave/remotely operated proxy
  - Perceived failure mode is not clear; joint limit reached, communication failure, software e-stop etc.

INTRODUCTION

- Potential Solutions
  - Kinematically identical/scaled master and slave
    - Non-modular
    - Requires slave specific masters
  - Constrain master kinematics to that of slave via haptic feedback
    - Human sensomotoric pathway uses proprioception to indicate joint limits
    - Haptic feedback is an intuitive and efficient feedback channel

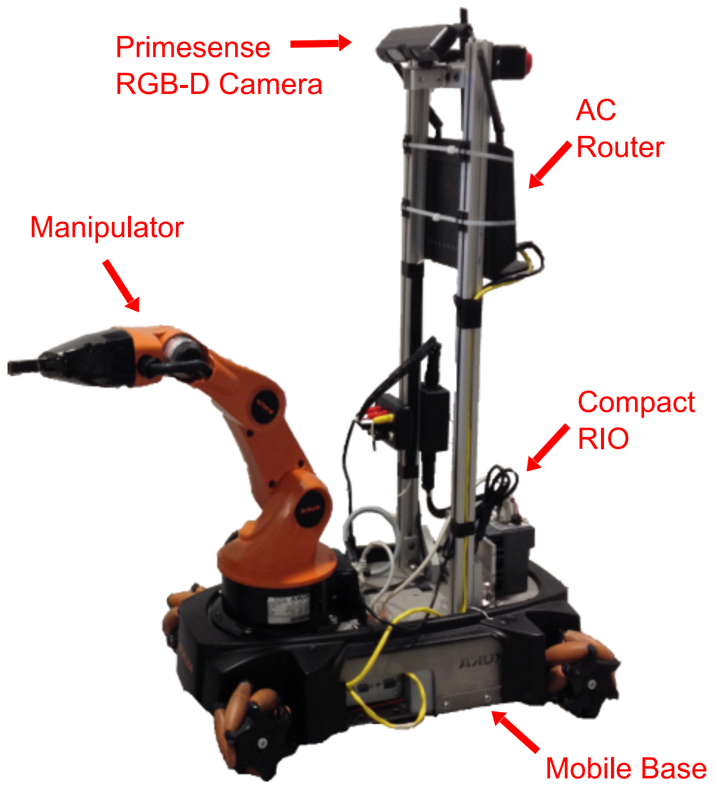
Input device and remote device with similar configuration

MOTIVATION

- Kinematic dissimilarities were addressed using haptic feedback – an ideal solution because the human body itself leverages proprioceptive haptic feedback at its own joint limits.
- Haptic feedback has proven to benefit telerobotic tasks – robot-assisted minimally invasive surgery(RMIS), micro assembly and remote welding etc.
  - Task or environmental cues
  - We introduce feedback about the slave device state to enhance operator awareness and reduce confusion

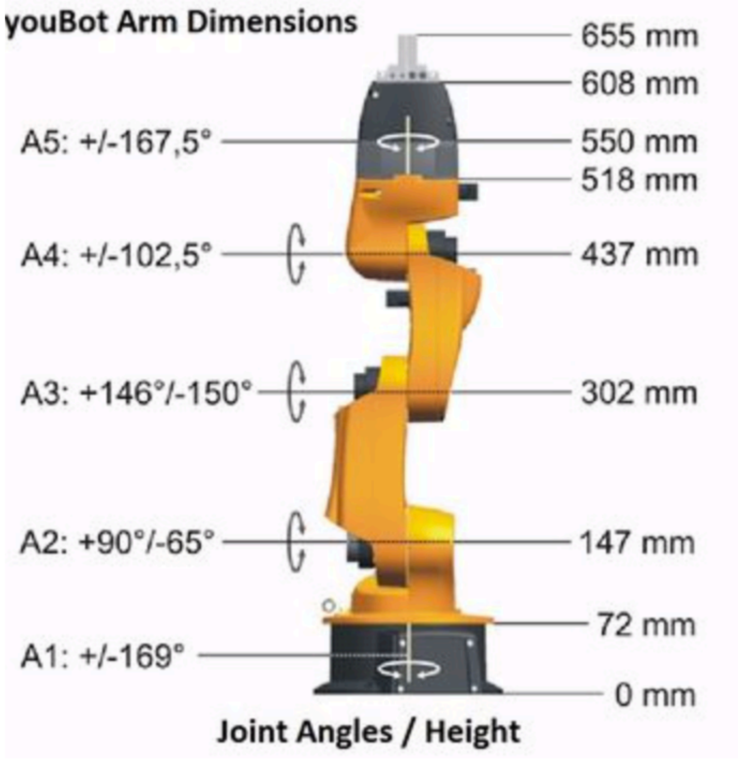
SYSTEM COMPONENTS

- KUKA youBot constrained to 3DOF motion
- Visual feedback provided via standard LCD monitor
- Bilateral teleop: Sensable PHANToM Omni
  - 3DOF haptic feedback
  - 3DOF motion commands
- Communication facilitated via AC router.
- National Instruments Compact RIO controller.



KUKA youBot

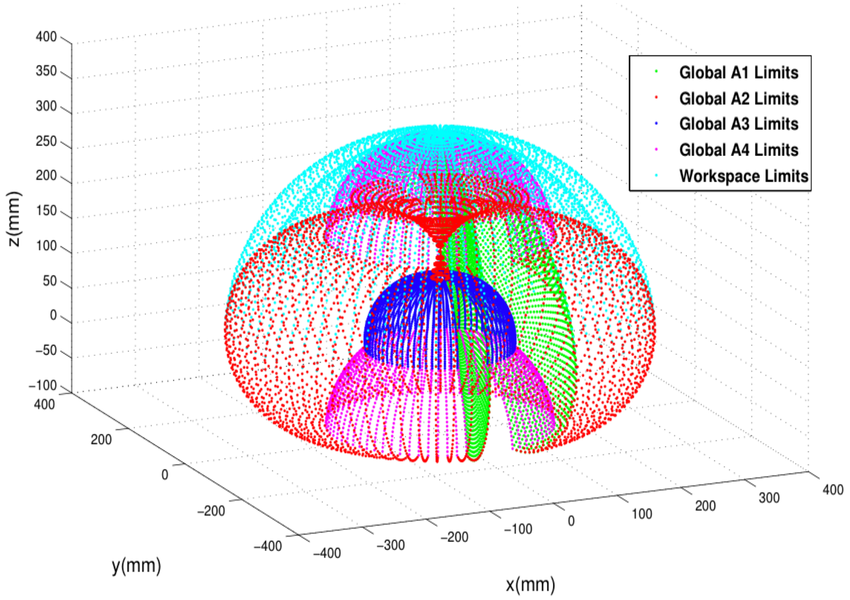
METHOD OVERVIEW



youBot - joints

- Surface sampling i.e. forward kinematics at joint limits systematically sample end effector location - at least one joint limit reached
- Cartesian points stored in simple tree like structure.
- The tree structure facilitated indexing and retrieval of local joint-limit point clouds.
- Efficient point-cloud based haptic rendering techniques employed using local point-clouds fetched at joint limits
  - Provides indication of translational motion to remove device from joint limit

POINT CLOUD HAPTIC FEEDBACK



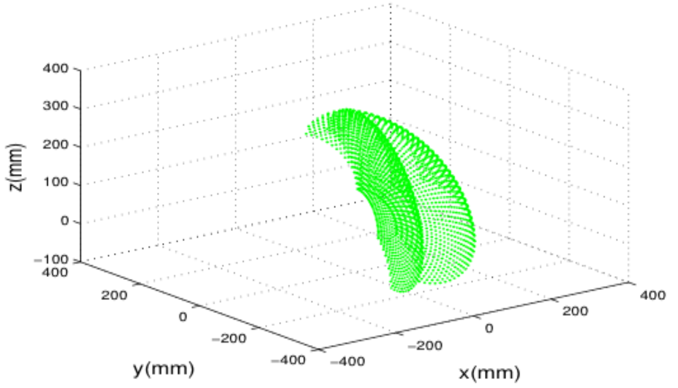
Union of Point Clouds for rotary joints A1-A4

- Joint limits easily visualized in joint space, but ideal cartesian translational haptic feedback not clear
  - If boundaries represented in cartesian space, haptic feedback is clearly defined via point-cloud rendering methods
- Unfortunately, joint limit surface may overlap in cartesian space – non overlapping point cloud local to the current joint configuration must be used

POINT CLOUD GENERATION ALGORITHM AND SAMPLING

- Joint space limits were systematically sampled as a Cartesian point cloud.
- Algorithm used to generate point clouds:

→ for minimum joint limit of each joint A_i , servo through all possible joint configurations for the remaining joints (forward kinematics determine servo step size to maintain minimum resolution)
→ repeat the above step for the maximum joint limit of A_i.
→ repeat above steps for all joints of interest



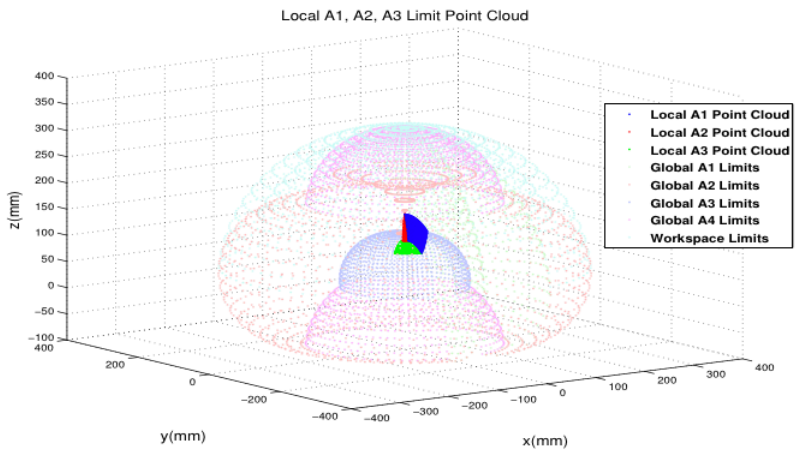
Joint 1 Limit Sampled Surface

- The joint limit surfaces now represented as a point clouds is sampled in a tree structure which is traversed via current configuration

LOCAL POINT CLOUD RETRIEVAL

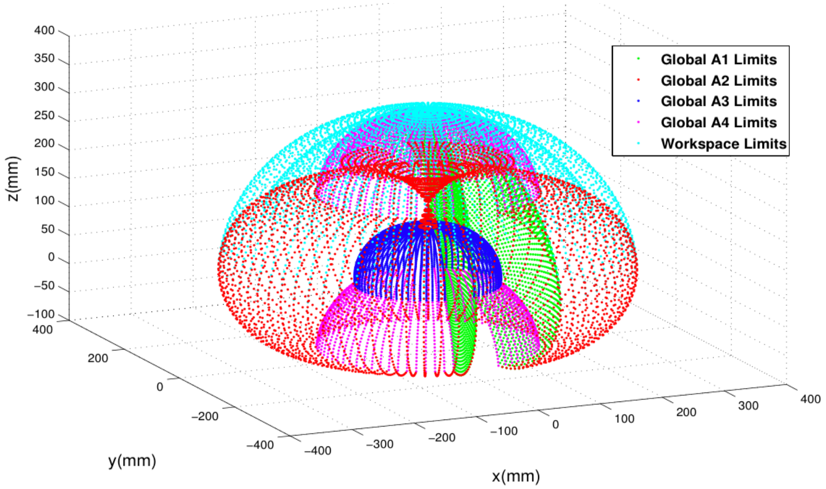
- Because of the tree-structure, locating the local non-overlapping point cloud is direct and trivial.
- Algorithm for point cloud retrieval:

→ If current joint configuration is at a limit, for each joint A_i, calculate indices of neighboring points from table (enabled by systematic sampling of points)
→ If not at joint limit, proceed with inverse kinematics based on user commanded input



LPC when all three joints are at limits

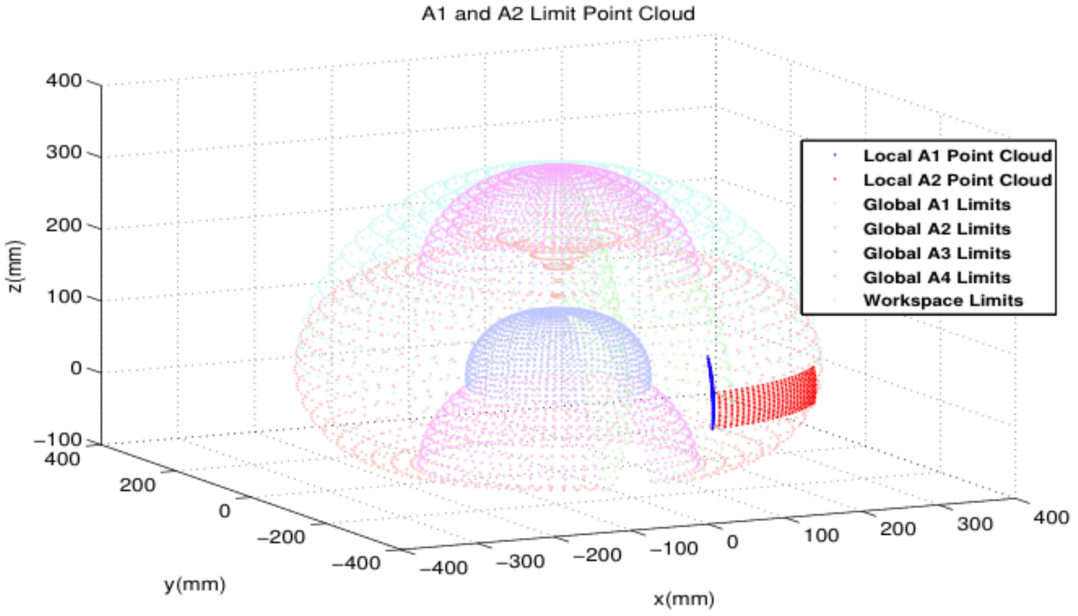
LOCAL JOINT LIMIT SEARCH



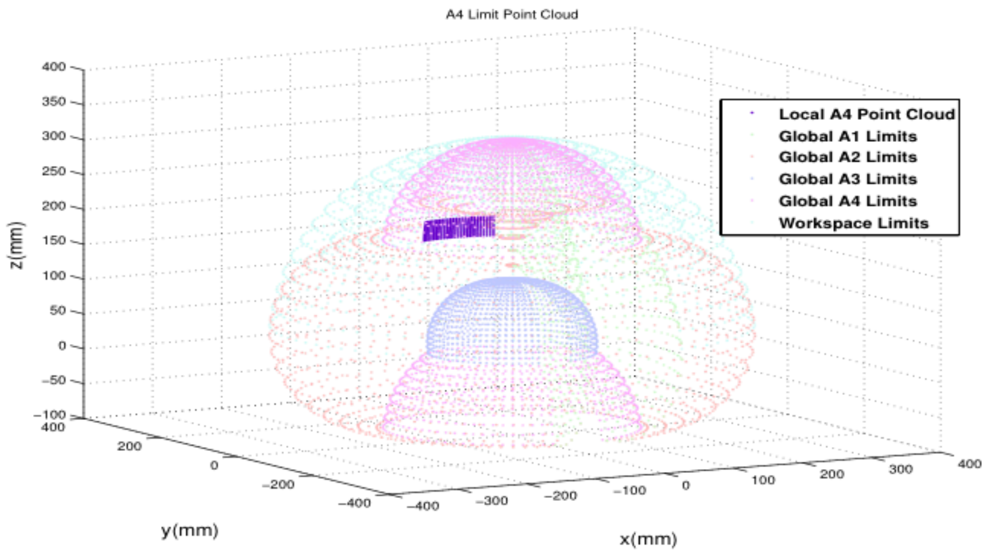
Union of Point Clouds for rotary joints A1-A4

- Workspace limits feedback can also be rendered in tandem.

LOCAL JOINT LIMIT SEARCH



Local point cloud at A1 and A2 limits



Local point cloud at A4 limits

CONCLUSION AND FUTURE WORK

- Results indicate that using this naïve tree structure approach for point cloud storage and retrieval, the joint limits for a 3DOF robot manipulator can be well represented and maneuvered in cartesian space – as the commanded position moves along a joint limit, the correct local point cloud is retrieved
- Techniques used in this paper raise the potential for using similar point-cloud based methods in higher DOFs (both input and slave device).
- Immediate next steps include algorithmic improvements - replacing the tree structure with a more efficient, constant look-up time mapping table.
- Extend research to user studies that include teleoperated robots in more sophisticated task environments

ACKNOWLEDGEMENT

The authors would like to extend their gratitude to

- National Instruments
- Commercialization Gap Fund (CGF)
- Fredrik Rydén
- Howard Jay Chizeck