

Sampling of 3DOF Robot Manipulator Joint-Limits for Haptic Feedback

Kevin Huang¹, Yun-Hsuan Su², Mahmoud Khalil¹, Daniel Melesse¹, Rahul Mitra¹

Abstract—Teleoperation of robotic proxies can extend human control to spaces, tasks and constraints that would otherwise be intractable or unattainable. The robustness to harsh environments, scalability, precision and repeatability of robots make them ideal for dangerous or difficult missions, and their use oftentimes reduces monetary and human cost. Maneuverable robot devices can provide more flexibility and articulation, but come at the cost of kinematic complexity. The kinematics and workspace of the remote device is oftentimes dissimilar to the input device, leading to potential confusion and frustration of the human operator. One solution is to constrain the input device motion to a scaled version of remote device joint ranges. This paper presents a method for doing so with 3 degree of freedom (DOF) manipulators and input devices with kinematic dissimilarities. The approach utilizes a simple tree structure, whereby a local Cartesian workspace limit is sampled and indexed by joint. This generates a locally sampled joint limit surface, represented as a point cloud. This local point cloud is then used to provide 3DOF haptic feedback to the operator as an indication that a joint limit has been reached, and provides kinesthetic force feedback to efficiently remove the operator from that joint limit. This work can improve usability of human-robot interfaces for teleoperation by allowing users to naturally intuit remote device joint limits and avoid confusion.

Keywords: teleoperation, haptic rendering, joint limits, force feedback, point clouds

I. INTRODUCTION

Developments in the disciplines of robotics, artificial intelligence and software engineering, have led to human tasks increasingly becoming machine assisted [1]. However, these growths in machine assistance have their shortcomings. Particularly, difficulties remain in providing intuitive and seamless remote control or teleoperation of articulated robots [2]. In teleoperated systems, input and remote devices can have vastly different kinematics and workspace limits, which can lead to confusion and frustration for the operator [3]. Imagine maneuvering an input device to a reachable configuration, yet no action is observed on the remote device.

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1329751 and the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1256082. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

¹Kevin Huang, Mahmoud Khalil, Daniel Melesse and Rahul Mitra are with Trinity College, Dept. of Engineering, 300 Summit St, Hartford, CT 06106 USA {kevin.huang, mahmood.khalil, daniel.melesse, rahul.mitra}@trincoll.edu

²Yun-Hsuan Su is with the University of Washington Department of Electrical and Computer Engineering, 185 Stevens Way, Paul Allen Center - Room AE100R, Campus Box 352500, Seattle, WA 98195-2500, USA. yhsu83@uw.edu

It is not clear whether the remote device has reached a joint limit, broken communication channels, emergency stopped or failed otherwise, effectively reducing situational awareness. Haptic feedback may be a possible intuitive solution, as the human body itself leverages and provides proprioceptive haptic feedback at its own joint limits, indicating that a new approach or trajectory may be needed to complete a task [4].

This paper attempts to reconcile the difficulties of dissimilar input and remote device kinematics through a joint limit sampling haptic feedback technique – in particular, through the generation of a point cloud that represents the remote device Cartesian joint limit surface. Haptics is commonly defined as the real or simulated touch interactions between robots, humans, and simulated or real environments. Haptic feedback can play an important role in reinforcing presence and immersion of physical targets to the teleoperator [5]–[7]. There has been a growing interest in efficient haptic rendering techniques in order to provide an improved interactive experience from a distance. In fact, haptic feedback has already been implemented successfully in a number of teleoperated applications, including robot-assisted minimally invasive surgery (RMIS) [5], [8], micro assembly and rover control in space [9], remote welding [10], and obstacle avoidance [11] to name a few. While many of these applications efficiently address haptically rendering and reflecting realistic contact forces, restricting motion, or providing assistive guidance, few concern conflicts in reachability between input and remote devices.

A. Contribution

To the best of the authors' knowledge, the work presented here is the first to

- create and implement a synthetic sampled point cloud of joint-limits for haptic rendering purposes;
- develop a general approach to 3DOF joint limit haptic feedback using a tree structure approach.

The proposed approach is implemented on a teleoperated robot manipulator, the KUKA youBot, constrained to 3DOF motion, and has implications in development of other teleoperated systems. The method systematically samples the Cartesian joint limits indexed by joint; i.e. a rotational joint is placed at its limit as the remaining joints are revolved within the respective limit at a fixed granularity. The sampled point cloud is stored in a tree-like structure that is searched based on configuration in joint space, and is subsequently used for real-time haptic feedback.

B. Related Work

Constrained motion control in teleoperation is designed for robots to achieve desired behaviors including joint limits avoidance, prevention of self collisions and unintentional collision with the workspace.

1) *Medical Robotics*: While speed and accuracy are of paramount importance for robots deployed in manufacturing, medical robots must satisfy additional task-specific requirements. In RMIS, a fixed remote center-of motion (RCM) [8] is one major limitation to the robot. According to an in-depth survey by Bowyer et al., haptic feedback based on virtual fixtures for teleoperated robots has focused on impedance control on the master side integrated with proxy and linkage simulation [12]. A redundant slave robot can in theory have multiple configurations in order for the end effector pose to match the master side device with respect to each of their individual reference frames [13]. In order to both satisfy the RCM motion constraints and prevent undesired contacts, the surgical operator should have intuitive and seamless control. Joint limit haptic feedback becomes helpful to enable a smoother teleoperation experience by reducing the chances of interruption due to robot joint limits or kinematic singularities [5].

Some works generated workspace limit feedback by formulating quadratic constraint optimization problems [14]–[16]. In [17], three types of constraints were implemented using vector field inequalities to ensure prevention of collisions between tool shafts, collisions between the right tool and the peg transfer board, and joint limit constraint. A follow-up work by Li et al. demonstrated improvements in efficiency, consistency between workspace constraints, and cost functions [18]. Kwok et al. further extended the concept to snake robots in endoscopy [19]. In [20], a unified framework for general teleoperated robot control regardless of the robot geometry and under workspace constraints are proposed and tested on the da Vinci Surgical System [21] for laparoscopic surgeries. Most of these works deal with simple cone-shaped virtual fixtures for workspace feedback. This work provides a general approach for 3DOF workspaces.

C. System Components

The operator is provided with visual feedback via an LCD monitor and RViz, displaying the RGB-D captured geometry information and manipulator configuration. A sample of the visual feedback is shown in Fig. 1.

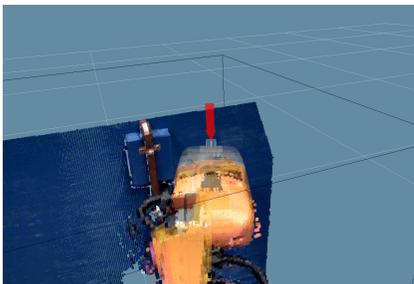


Fig. 1. Operator-side visual feedback

Simultaneously, the operator provides motion commands and receives 3DOF haptic force feedback via the Sensable PHANToM Omni haptic device.

The remotely operated device for this project is a KUKA youBot manipulator with mobile base. The remote environment is sensed via an RGB-Depth (RGB-D) camera, while communication is facilitated via an AC router. Finally, real-time robot actuation and state monitoring is achieved with the National Instruments Compact RIO controller. These components are shown below in Fig. 1.

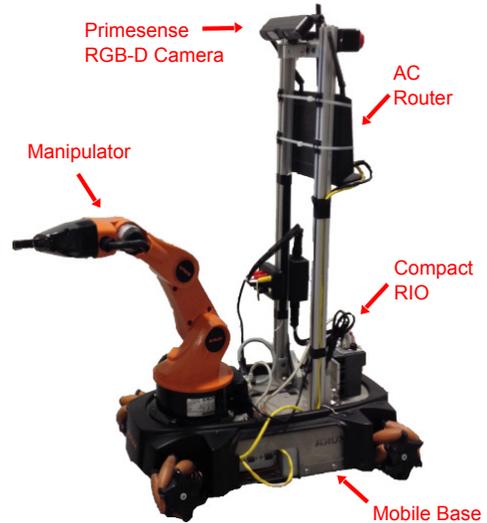


Fig. 2. KUKA youBot remote robotic platform

II. METHODS

The proposed approach is comprised of three distinct components:

- 1) Surface sampling, i.e. forward kinematics and point cloud generation
- 2) Indexing and retrieving local joint limit point cloud
- 3) Haptic rendering from point cloud

Each of the three components will be explored separately in the following subsections.

A. Surface Sampling

In order to generate a Cartesian point cloud sampling of the joint-limit surface, both forward kinematics of the robot manipulator and systematic sampling are required.

1) *Kinematic Analysis*: The KUKA youBot manipulator is a 5DOF device with gripper. For this work, the device is restrained to 3DOF in a controllable and predictable manner. To achieve this, the gripper is fixed in its closed position, and the terminal fifth joint, A5, is fixed at the center of its dynamic range, 0° . Furthermore, the fourth joint, A4, is restricted to maintain the final link level with the youBot base. With these constraints, the KUKA youBot manipulator's 3DOF constrained workspace can be determined by link geometries and joint limits (for remaining joints A1, A2 and A3), as shown below in Fig. 3.

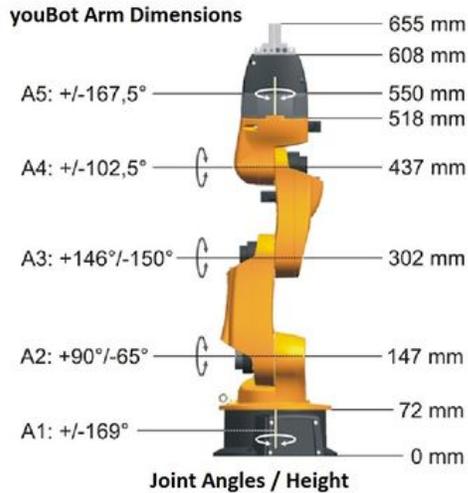


Fig. 3. The KUKA youBot link geometries, joint indices and joint limits. The joint configuration shown is the zero configuration. Note that in this work, joints A4 and A5 are restricted in order to constrain the manipulator to 3DOF [22].

Joints A1 - A5, are rotary joints. The axis of rotation for A1 is orthogonal to the base of the robot, and determines to the yaw of the manipulator end effector with respect to the base. Limits are best viewed along the axis of rotation, and so are best visualized from a “top” view, as shown below in Fig. 4.

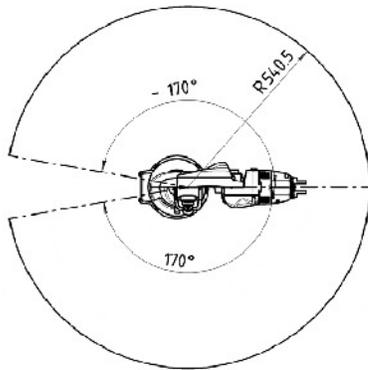


Fig. 4. KUKA youBot joint A1 joint limits, 340° range [22]. The zero configuration is shown.

The remaining joints of interest (A2-A4), have parallel axes of rotation, and together determine or define the pitch of the end effector. Note that joint A5 is fixed to maintain a fixed end effector roll. These rotational axes are all parallel to the base plane of the robot. Thus, the reachable space and limits for joints A2, A3 are best viewed from a “side” view, as shown in Fig. 5.

When viewed in 3D Cartesian space, these workspace limits are composed of intersecting, locally differentiable surfaces. In contrast, in n -D joint space where n is the number of joints, the surface does not intersect itself and is homeomorphic to the n -sphere. Beyond joint limits, workspace limits also define the entire reachable workspace.

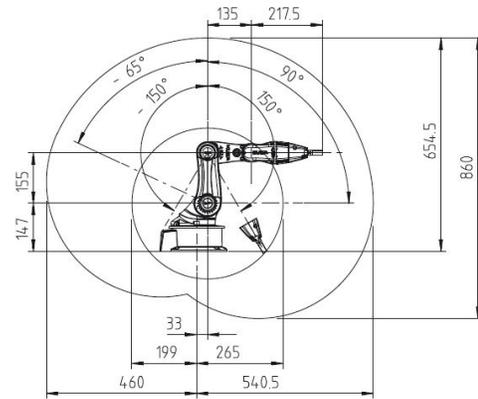


Fig. 5. KUKA youBot joints A2, A3 joint limits [22].

When implementing position control of a robot manipulator in a teleoperated architecture, reflecting or indicating a kinematic or joint limit haptically is a challenge. The joint limits as viewed in joint space are intuitive. However, the user commanded input, desired kinesthetic haptic feedback and workspace limits are better understood in Cartesian coordinates. Thus, the force vector that would most efficiently guide an operator away from a joint or workspace limit can be effectively realized so long as a Cartesian analog of the joint-space limits can be computed.

Basic inverse kinematic analyses allow for Cartesian position control of the manipulator, while forward kinematics determine workspace limits and were used to systematically sample the joint space limits as a Cartesian point cloud.

2) *Inverse Kinematics*: Inverse kinematics of the 3DOF manipulator are needed in order to realize user provided motion commands as end effector motion [23]. Rotational joints A4, A5 are constrained, effectively reducing the articulation of the KUKA youBot to 3DOF. A5 was fixed to remain at the neutral configuration within its dynamic range, while A4 was to be constrained to keep the end effector level with the robot base. Consider the base frame of the robot, as shown below in Fig. 6.

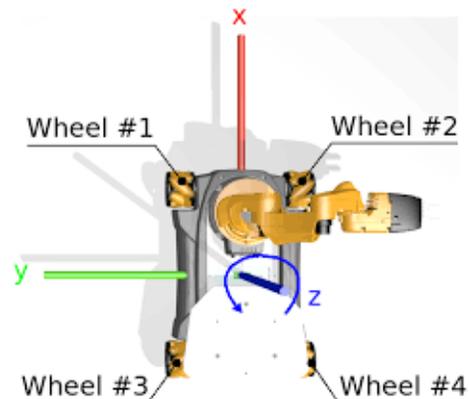


Fig. 6. KUKA youBot baseframe [22].

By the constraints on A4, A5, the operator input $I = [I_x \ I_y \ I_z]^T$ defines the position of the origin of coordinate frame A4 relative to the robot base frame. Then, given user input position command $I \in \mathbb{R}^3$, the task is to determine the joint configuration $\theta \in \mathbb{R}^5$

$$\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]^T$$

that satisfies all constraints, where θ_i is the joint configuration of A_i . Then θ_1 is easily determined as

$$\theta_1 = \text{atan2} \left(\frac{I_y}{I_x} \right)$$

Now assigning link lengths of L_1, L_2 as the length between joints A2–A3 and A3–A4, θ_2, θ_3 are determined with basic geometry. Consider Fig. 7 below.

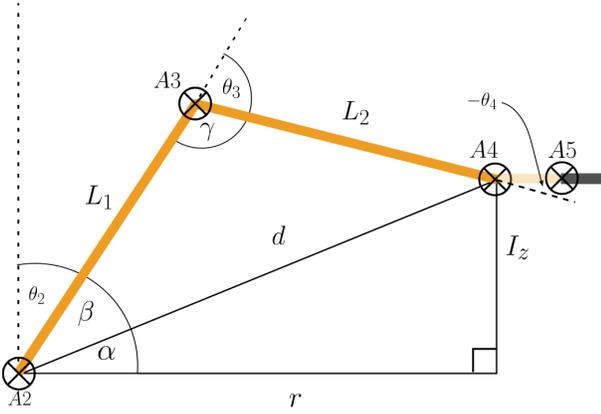


Fig. 7. Side-view for calculating θ_2, θ_3 .

Where $r = \sqrt{I_x^2 + I_y^2}$ is the projection length on the XY horizontal plane of I , and $d = \sqrt{I_x^2 + I_y^2 + I_z^2}$ is the L2 norm of the commanded input I . Then clearly:

$$\begin{aligned} \theta_2 &= \frac{\pi}{2} - (\beta + \alpha) \\ &= \frac{\pi}{2} - \left(\cos^{-1} \left(\frac{L_2^2 - L_1^2 - d^2}{-2dL_1} \right) + \text{atan2} \left(\frac{I_z}{r} \right) \right) \end{aligned}$$

and for joint A3

$$\begin{aligned} \theta_3 &= \pi - \gamma \\ &= \pi - \text{atan2} \left(\frac{-2dL_1L_2 \sin \beta}{L_2(d^2 - L_1^2 - L_2^2)} \right) \end{aligned}$$

and to keep joint A4 such that the end effector is level with the base frame XY plane

$$\theta_4 = \frac{\pi}{2} - (\theta_2 + \theta_3)$$

and θ_5 is left in the zero configuration.

3) *Cartesian Joint Limit Generation*: This work samples the joint limit surfaces as a point cloud to render haptic feedback at said limits. Recall that the user commands the locations of joint A4, and joint A5 is constrained to its zero configuration. Then the actuated joints and the joint limits of interest correspond to joints A1, A2, A3 and A4, whose joint angles are $\theta_1, \theta_2, \theta_3, \theta_4$ respectively, with limits and ranges of

$$\begin{aligned} \theta_1 &\in (-170^\circ, 170^\circ) \\ \theta_2 &\in (-65^\circ, 90^\circ) \\ \theta_3 &\in (-150^\circ, 150^\circ) \\ \theta_4 &\in (-102.5^\circ, 102.5^\circ) \end{aligned}$$

The goal is to create and store in a systematic manner positions of joint A4 for which at least one of the joints A1, A2, A3 are at its limit. This is achieved while conforming to A4 constraints as outlined previously. Sampling increments ϕ_i within each joint A_i are dependent on a predetermined approximate maximum euclidean interpoint distance, call it w . Then the increments for joint sampling are approximated by

$$\begin{aligned} \phi_1 &= \phi_2 = 2 \sin^{-1} \left(\frac{w}{2(L_1 + L_2)} \right) \\ \phi_3 &= 2 \sin^{-1} \left(\frac{w}{2L_2} \right) \end{aligned}$$

The granularity w should be chosen to accommodate the desired proxy diameter for point cloud based haptic rendering as described in [24]. The joint limits are then systematically sampled via Algorithm 1.

Algorithm 1 Joint Limit Sampling, Point Cloud Generation

- 1: **define** PC as point cloud to store joint limits
 - 2: **define** ϕ_i as sampling granularity for each joint A_i
 - 3: **define** θ_{mi}, θ_{Mi} as min and max limits for joint A_i
 - 4: **for** each joint, A_i **do**
 - 5: **let** $\theta_i = \theta_{mi}$
 - 6: **for** remaining two joints, A_j, A_k **do**
 - 7: **for** ($\theta_j = \theta_{mj}; \theta_j \leq \theta_{Mj}; \theta_j += \phi_j$) **do**
 - 8: **for** ($\theta_k = \theta_{mk}; \theta_k \leq \theta_{Mk}; \theta_k += \phi_k$) **do**
 - 9: **if** $|\frac{\pi}{2} - (\theta_2 + \theta_3)| \leq \theta_{M4}$ **then**
 - 10: $r = L_1 \sin(\theta_2) + L_2 \cos(\theta_2 + \theta_3)$
 - 11: $z = L_1 \cos(\theta_2) + L_2 \cos(\theta_2 + \theta_3)$
 - 12: $x = r \sin \theta_1$
 - 13: $y = r \cos \theta_1$
 - 14: add $[x, y, z]^T$ to PC
 - 15: **end if**
 - 16: **end for**
 - 17: **end for**
 - 18: **end for**
 - 19: **let** $\theta_i = \theta_{Mi}$
 - 20: **repeat** steps 5-18
 - 21: **end for**
-

B. Local Joint Limit Fetch, Tree Search

With the joint limits systemically sampled and stored in a tree structure, the goal is then to retrieve a point cloud of the local joint limit given a user input position command, I . The entire joint limit surface may overlap itself several times in cartesian space, however these overlapping sections are distinct and disconnected in joint space. Haptic rendering based on the entire point cloud PC would thus render undesired results; only the point cloud local to the current joint configuration should be rendered. The method for retrieving the appropriate sub point cloud for haptic rendering is described below in Algorithm 2.

Algorithm 2 Local Point Cloud Retrieval

- 1: **define** LPC as local point cloud
 - 2: **define** R_i as local joint range for each ϕ_i
 - 3: **if** current joint configuration at a joint limit **then**
 - 4: **for** joint limit reached at joint A_i **do**
 - 5: **calculate** indices of remaining joints, θ_j, θ_k , from
 - 6: point cloud generated at θ_i limit, call them
 - 7: q_j and q_k respectively
 - 8: **for** (int $x = q_j - R_j; x \leq q_j + R_j; x++$) **do**
 - 9: add PC[x] to LPC
 - 10: **end for**
 - 11: **repeat** steps 8,9 for q_k
 - 12: **end for**
 - 13: **else** proceed with inverse kinematics based on I
 - 14: **end if**
-

C. Point Cloud Haptic Feedback

Once a local point cloud is fetched, real-time haptic feedback is implemented via a proxy-based method for rendering with streaming point clouds [24]. This algorithm was demonstrated successfully with synthetically generated point clouds, real-time RGB-D point cloud data, as well as pre-touch sensed points [25], [26]. The reader is directed to [27], [28] for technical details and limitations for haptic rendering from point clouds and their use as virtual fixtures.

III. RESULTS

A. Joint Limit Cartesian Sampling

The method described in Algorithm 1 was performed for each joint of the KUKA youBot. A sampled set for joint A1 is shown below in Fig. 8.

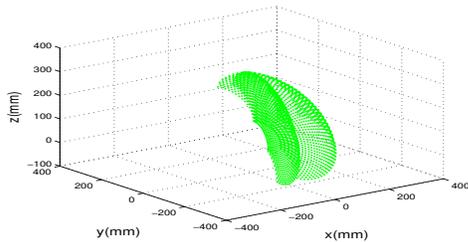


Fig. 8. Joint 1 Limit Sampled Surface

Similarly, for joint A2 a sampled surface is created as shown in Fig. 9.

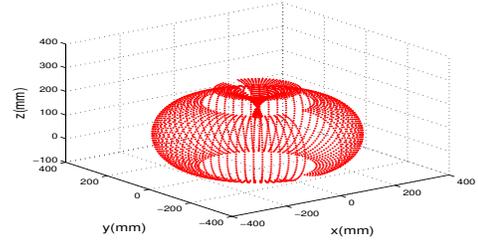


Fig. 9. Joint 2 Limit Sampled Surface

And the result for joint A3 is shown in Fig. 10

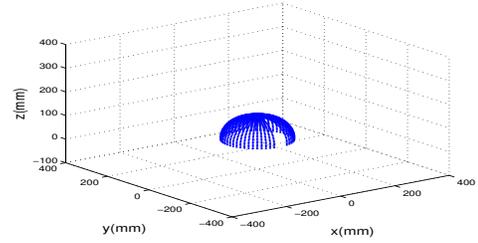


Fig. 10. Joint 3 Limit Sampled Surface

Joint A4 is constrained to keep the end effector level to the robot base, and is limited by A1, A2, and A3 when at a joint limit. A resultant joint limit for A4 is shown in Fig. 11.

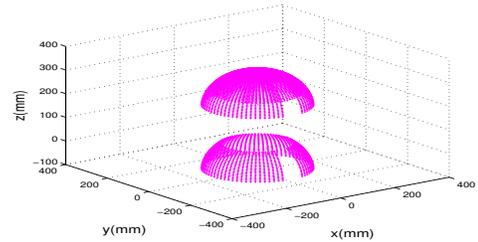


Fig. 11. Joint 4 Limit Sampled Surface

In addition to the joint limits for each of joints A1, A2, A3, and A4, general workspace limits rendered in tandem and are also displayed using haptic rendering methods for point clouds. This sampling is shown in Fig. 12.

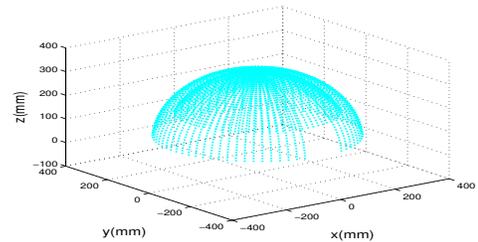


Fig. 12. Workspace Limits

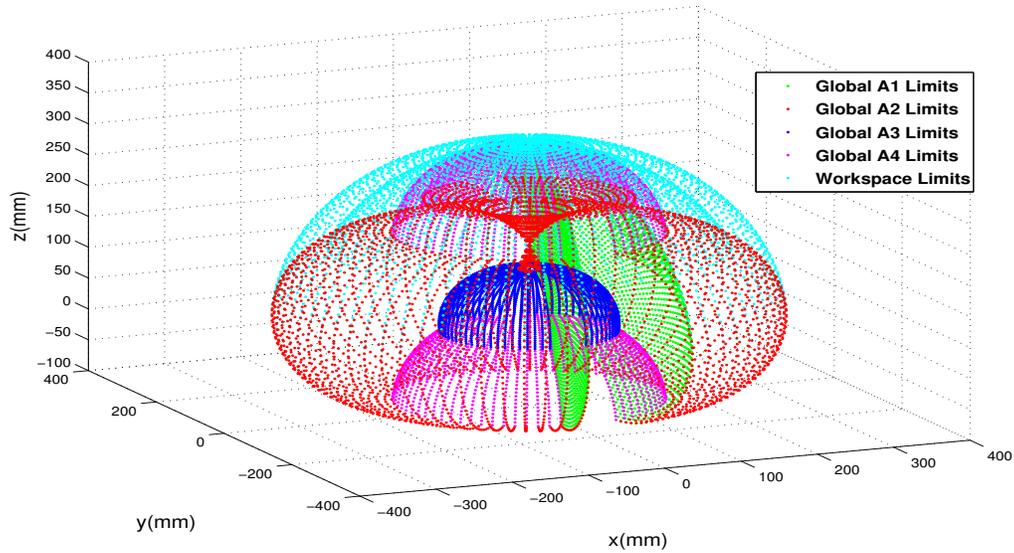


Fig. 13. Union of the point clouds representing sampled limits for rotary joints (A1-A4) along with the workspace limit. Observe that the sampled surfaces overlap one another in Cartesian space; these overlaps are disconnected in joint space. Algorithms 1 and 2 can be modified via parameter ϕ_i to generate either denser or sparser sampling.

The union of the sampled point clouds for each joint limit and workspace limits are the general constraints for which haptic feedback is rendered in this work. This is shown in Fig. 13. The surfaces overlap in Cartesian space and thus should not be rendered simultaneously; instead a locally non-overlapping surface is retrieved based on the current configuration of the manipulator.

B. Local Joint Limit Search

Given a commanded user input position I that is either at or beyond joint limits, a non-overlapping sub surface of the sampled constraints shown in Fig. 13 must be retrieved for haptic rendering. A local point cloud is retrieved for commanded input beyond limits for A1, A2 and is shown in Fig. 14.

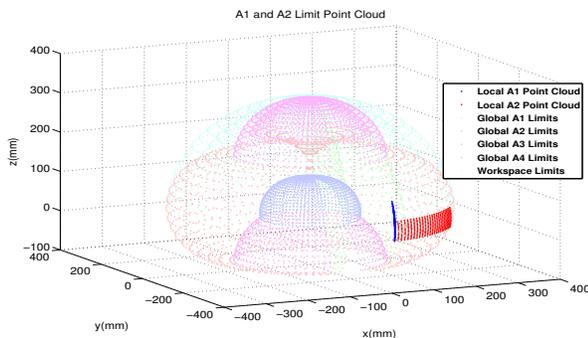


Fig. 14. Local joint limit point cloud when command input is beyond A1 and A2 limits.

When the input violates the A4 joint constraint, a local point cloud to prevent that violation is retrieved, as shown in Fig. 15.

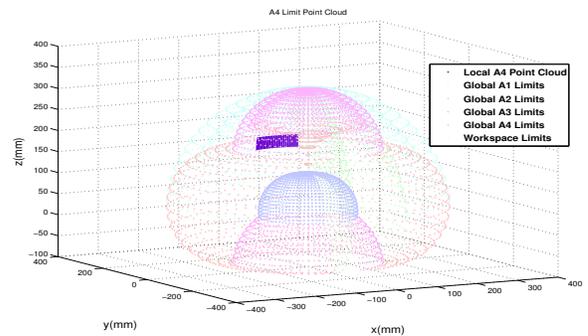


Fig. 15. Local joint limit point cloud when command input violated A4 limits.

Finally, when A1, A2, A3 limits are reached, the union of the retrieved sub point clouds represent a connected boundary as shown in Fig. 16.

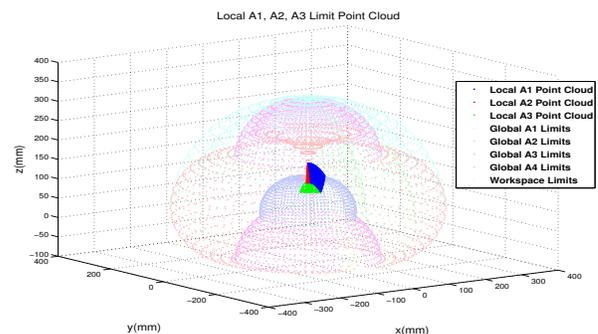


Fig. 16. Local joint limit point cloud when command input is at all of A1, A2, and A3 minimum joint limits. This point cloud will encourage the user to move away from the origin and thus away from inner joint limits.

IV. CONCLUSION

Robotic proxies extend human control to task spaces that would otherwise be unattainable by humans. However, kinematic difficulties associated with operating a remote maneuverable robotic device may arise due to dissimilarities between the reachable workspaces of the robot and the input device. This paper reconciled these kinematic complexities using a naive tree structure approach. A 3DOF robot was used to sample joint limits in Cartesian space. The sampled joint limits were then used to compile a synthetic surface point cloud representing the joint limits of the 3DOF end effector. Systematically storing point clouds in a tree data structure, a local point cloud at any given joint limit can be retrieved. Well established haptic rendering techniques can then be used for appropriate 3DOF feedback.

Results show that using this point cloud storage and retrieval method, the joint limits for a 3DOF robot can be better represented, understood and maneuvered in cartesian space. Using joint limit haptic rendering techniques can alert the human operator when a certain limit has been reached, and thus reduce frustration and confusion in teleoperation. There are still issues to be resolved in representing workspace kinematics accurately to an operator using a telerobot, however, techniques used in this paper raise the potential for using similar methods for telerobots having extremely complex task environments and numerous degrees of freedom. Direct next steps include algorithmic changes including the replacement of the tree data structure with a more efficient, constant look-up time mapping table.

ACKNOWLEDGEMENTS

The authors would like to thank National Instruments for generosity and support in contributing to this project, the Commercialization Gap Fund (CGF) for their aid in funding this research, as well as Fredrik Rydén and Howard Jay Chizeck for their support and feedback.

REFERENCES

- [1] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.
- [2] J. Y. Chen, E. C. Haas, and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1231–1245, 2007.
- [3] T. L. Brooks, I. Ince, and G. Lee, "Operator vision aids for space teleoperation assembly and servicing," 1992.
- [4] N. Diolaiti and C. Melchiorri, "Teleoperation of a mobile robot through haptic feedback," in *Haptic Virtual Environments and Their Applications, IEEE International Workshop 2002 HAVE*. IEEE, 2002, pp. 67–72.
- [5] A. M. Okamura, "Haptic feedback in robot-assisted minimally invasive surgery," *Current opinion in urology*, vol. 19, no. 1, p. 102, 2009.
- [6] H. I. Son, J. Kim, L. Chuang, A. Franchi, P. R. Giordano, D. Lee, and H. H. Bühlhoff, "An evaluation of haptic cues on the teleoperator's perceptual awareness of multiple uavs' environments," in *World Haptics Conference (WHC), 2011 IEEE*. IEEE, 2011, pp. 149–154.
- [8] R. H. Taylor, J. Funda, D. D. Grossman, J. P. Karidis, and D. A. LaRose, "Remote center-of-motion robot for surgery," Mar. 14 1995, uS Patent 5,397,323.
- [7] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Computers & Graphics*, vol. 21, no. 4, pp. 393–404, 1997.
- [9] C. Preusche and G. Hirzinger, "Haptics in telerobotics: Current and future research and applications," *The Visual Computer*, vol. 23, no. 4, pp. 273–284, 2007.
- [10] D. Ni, A. Yew, S. Ong, and A. Nee, "Haptic and visual augmented reality interface for programming welding robots," *Advances in Manufacturing*, vol. 5, no. 3, pp. 191–198, 2017.
- [11] A. Leeper, K. Hsiao, M. Ciocarlie, I. Sucan, and K. Salisbury, "Arm teleoperation in clutter using virtual constraints from real sensor data," in *RSS workshop on robots in clutter: preparing robots for the real world*, 2013.
- [12] S. A. Bowyer, B. L. Davies, and F. R. y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2014.
- [13] S. Bruno, S. Lorenzo, V. Luigi, and O. Giuseppe, "Robotics: modelling, planning and control," *Advanced Textbooks in Control and Signal Processing Series, Spring Printed*, pp. 502–518, 2009.
- [14] M. Li, A. Kapoor, and R. H. Taylor, "Telerobotic control by virtual fixtures for surgical applications," in *Advances in Telerobotics*. Springer, 2007, pp. 381–401.
- [15] T. Xia, A. Kapoor, P. Kazanzides, and R. Taylor, "A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 639–644.
- [16] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 1996.
- [17] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic active constraints for surgical robots using vector field inequalities," *arXiv preprint arXiv:1804.11270*, 2018.
- [18] M. Li, M. Ishii, and R. H. Taylor, "Spatial motion constraints using virtual fixtures generated by anatomy," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 4–19, 2007.
- [19] K.-W. Kwok, K. H. Tsoi, V. Vitiello, J. Clark, G. C. Chow, W. Luk, and G.-Z. Yang, "Dimensionality reduction in controlling articulated snake robot for endoscopy under dynamic active constraints," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 15–31, 2013.
- [20] M. M. Marinho, B. V. Adorno, K. Harada, K. Deie, A. Deguet, P. Kazanzides, R. H. Taylor, and M. Mitsuishi, "A unified framework for the teleoperation of surgical robots in constrained workspaces," *arXiv preprint arXiv:1809.07907*, 2018.
- [21] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6434–6439.
- [22] "youBot Store - Hardware Specification," www.youbot-store.com/developers/hardware-specification, accessed: 2018-12-30.
- [23] Z. L. Richard Murray, in *A Mathematical Introduction to Robotic Manipulation*. California Institute of Technology, Hong Kong University of Science and Technology, 1994, pp. 97–104.
- [24] F. Ryden and H. Chizeck, "A proxy method for real-time 3-dof haptic rendering of streaming point cloud data," *Haptics, IEEE Transactions on*, vol. 6, no. 3, pp. 257–267, 2013.
- [25] K. Huang, L.-T. Jiang, J. R. Smith, and H. J. Chizeck, "Sensor-aided teleoperated grasping of transparent objects," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4953–4959.
- [26] K. Huang, P. Lancaster, J. R. Smith, and H. J. Chizeck, "Visionless tele-exploration of 3d moving objects," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 2238–2244.
- [27] F. Ryden and H. Chizeck, "Forbidden-region virtual fixtures from streaming point clouds: Remotely touching and protecting a beating heart," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 3308–3313.
- [28] F. Ryden, S. Kosari, and H. Chizeck, "Proxy method for fast haptic rendering from time varying point clouds," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 2614–2619.