Autonomous Wall Cutting with an Atlas Humanoid Robot

Zheng-Hao Chong, Robert T.W. Hung, Kit-Hang Lee, Weijia Wang, Tony W.L. Ng, Wyatt Newman,
Senior Member IEEE

***Abstract*—Autonomous wall cutting is described using an Atlas humanoid robot. An integrated wall-cutting skill is presented, which only requires an operator to issue supervisory-level commands to prescribe a desired cutting path, leading to autonomous cutting.**

I. Introduction

Team HKU, based at University of Hong Kong, is competing in the DARPA Robotics Challenge (DRC) with a Boston Dynamics, Inc. "Atlas" humanoid robot [1]. DARPA challenge tasks are detailed in [2].

The Atlas robot, shown in Fig. 1, was unveiled by Boston Dynamics, Inc. in July, 2013. Only 7 Atlas robots exist: 6 owned by DARPA, plus the Atlas robot privately acquired by the University of Hong Kong. All Atlas robots were delivered equipped with a Carnegie Robotics sensor head. For hands, HKU chose the Sandia hands [3]. Atlas's 28 joints (not counting the Sandia hands nor the rotating LIDAR on the sensor head) are electrohydraulic.

Team HKU developed all of its robot control software in ROS (Robot Operating System) [4]. One of the challenges of this competition is to identify and design appropriate user interfaces for controlling a robot remotely under restrictions of limited communications.

The focus of this paper is on one of the DRC tasks: developing a wall-cutting skill that requires only simple and intuitive supervisory-level operator input. Wall-cutting is a significant challenge, particularly with respect to generating feasible inverse-kinematic path solutions. TeamHKU's approach is reviewed here, building on and extending earlier results in inverse kinematics [5], compliant motion control [6], design of operator interfaces [7], sensor-based task specification [8], and state-machine system integration [9].

To achieve skillful cutting, we addressed this challenge in four contexts: pre-planning, to establish optimal approach to a wall for a specific cutting task; means for an operator to visualize viable options and to enter a cutting-path specification naturally; on-line generation of 8-DOF trajectories to achieve a specified cutting path; and exploitation of compliant motion control and active balancing to interact robustly with cutting reaction forces. Innovations in these areas are presented separately in the following sections.

All authors are affiliated with the University of Hong Kong. The corresponding author is Wyatt Newman, EECS Dept, Case Western Reserve University, Cleveland, OH, wyatt.newman@case.edu., currently the Hung Hing Ying Distinguished Visiting Professor in Science and Engineering at HKU.

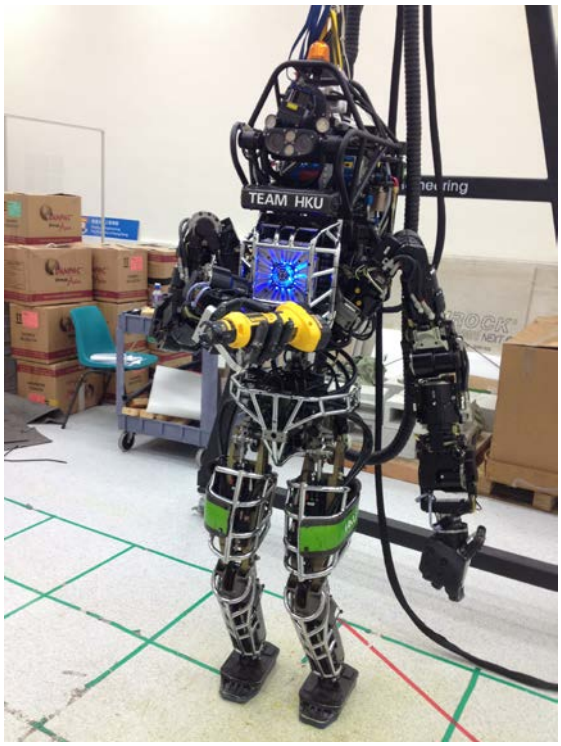

Figure 1: Atlas holding a cutting tool

II. Optimization of Wall Approach

In this section, we address the question: how should Atlas position his body relative to a wall to optimize wall-cutting performance?

To help analyze this problem, we define "cuttability" of a "wall point." To be precise, we must define two reference frames: the tool frame and a local world frame. Target points to be evaluated for cuttability are expressed in a local world frame defined as follows. Once Atlas has planted his feet in preparation for wall cutting, the local world frame origin is defined to lie on the ground plane at the vertical projection of the pelvis-frame origin. The orientation of the local world frame has its z-axis pointing up, parallel to gravity, and its x-axis lying in the ground plane, oriented as close as possible to parallel with the pelvis x-axis.

Atlas's tool frame is defined with respect to the grasped tool (Fig 2). With reference to this figure, we define a tool frame as follows. The y-axis of the tool frame is coincident with the tool-bit spin axis. The origin of the tool frame lies on the y-axis in the plane of the face of the tool fixture (cutting guide). The z-axis of the tool frame is defined to be parallel to the hand's palm normal. With these definitions, we can define a (static) tool transform from the hand frame

to the tool frame. In terms of the defined tool frame, it is desirable that the cutting fixture be held in contact with the wall, coplanar with the wall (i.e., with the tool axis normal to the wall). A target point to be cut should be coincident with the tool-frame origin.

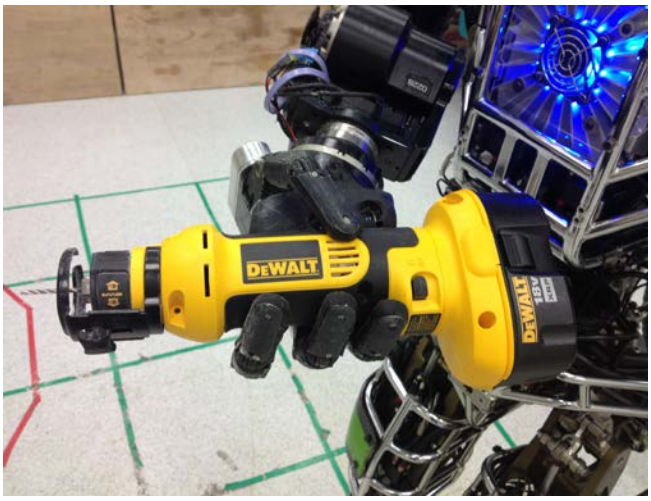

Figure 2: Atlas grasping the cutting tool in his right Sandia hand. A tool frame is defined with y-axis along the cutting bit, z-axis parallel to palm normal, and origin on the y axis at the intersection of the planar face of the cutting guide.

With respect to the defined local-world frame, a point to be evaluated--referred to here as a "wall point"--is specified in terms of 3 translational coordinates and one direction vector (the direction of the wall normal). We define a wall point as "cuttable" if there exists at least one inverse-kinematic (IK) solution within the domain of allowable robot poses such that the tool-frame origin is coincident with the wall point coordinates and the tool-frame y-axis is parallel to the wall normal. Note that determination of "cuttability" depends on how many degrees of freedom of the robot are to be exploited. For example, if only arm motions are considered (for the current 6-DOF arms), then the mobility domain of candidate IK solutions is limited to these 6 degrees of freedom.

For any point on the wall to be cut, there are 5 constraints on the tool frame. However, the tool may be rotated about its motor axis (y-axis) while still satisfying the position and wall-normal constraints, thus providing a 1-DOF null space. Pragmatically, it is preferred that the tool be held with the Sandia-hand palm normal pointing up, so the weight of the tool is carried by the palm instead of by the fingers. However, this is not a hard constraint; we allow for tilt of +/- 45deg relative to the ideal vertical palm normal, thus allowing for a range of possible solutions.

Desirably, the pelvis frame is oriented vertically with the pelvis origin approximately centered between the feet (as projected onto the ground plane). In a simplified solution, one may try to optimally pre-position Atlas relative to the target wall, freeze the legs (and thus the pelvis), freeze the torso degrees of freedom, and depend on the 6 degrees of freedom of the arm to achieve the desired cutting poses. To exploit more mobility, another option is to consider 8-DOF options including torso rotation about vertical and changes in pelvis elevation.

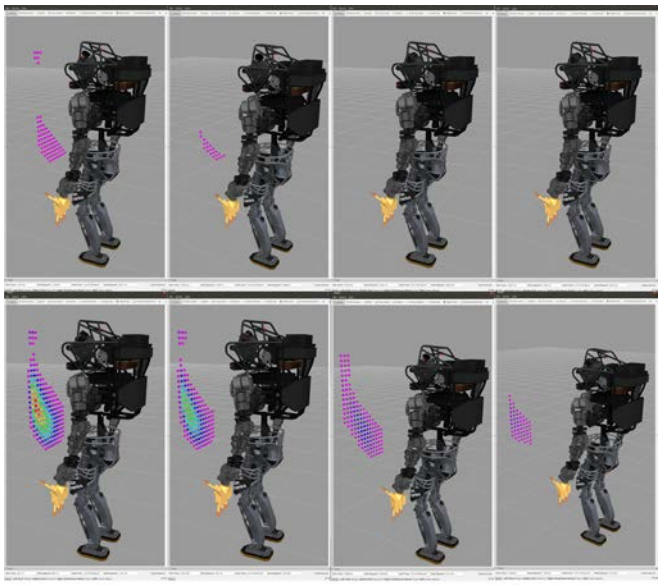

Figure 3: Wall cuttability regions. Top row: 6-DOF cuttability options. Bottom row: 8-DOF cuttability options. Wall distances, left to right: 0.5, 0.6, 0.7 and 0.8m from pelvis origin to wall. Marker color indicates number of kinematic solutions: red highest to violet lowest.

Figure 3 shows the results of pre-processing cuttabillity over alternative vertical wall poses with respect to the robot. Each colored dot is a wall point that was found to be cuttable by at least one IK solution. The top row of frames shows the cuttability assuming use of the right-arm 6-DOF joints only. The bottom row of frames shows the cuttability for 8 degrees of freedom, including squatting and torso rotation about vertical.

In Fig 3, from left to right, analyses are shown for wall distances from 0.5m (leftmost) to 0.8m (rightmost), with the robot assumed facing the wall head-on. (Alternative wall angles were considered, but these approach variations did not improve cuttability). Wall points were sampled at 4cm intervals. The cuttable points were found to lie within cuttable regions. The edges of these regions offer few kinematic solutions—as indicated by violet-colored points, whereas the interiors of the regions offered a larger variety of IK options.

The top row of Fig 3 shows that the cuttable regions using only the 6-DOF right-arm joints are impractically small. In fact, at wall distances of 0.7 and 0.8m, no cuttable points were found at all.

Eight-DOF solutions were explored by sampling at: pelvis heights from 0.5m to 0.8m at 2cm intervals; torso rotations from -0.6 to 0.6rad at 0.025rad intervals, and tool rotations about y-axis from -45 deg to 45deg in 0.05rad intervals. These chosen sampling resolutions resulted in each wall point being analyzed with roughly 30,000 tool-pose options. (For the illustrated red regions of Fig 3, as many as 4,000 IK

solutions were found). Exploiting the algorithm of [5], computation of 30,000 IK solutions (for each wall point considered) consumed approximately 150msec on one core of a 3.4GHz Intel i-7.

The results of Fig 3 reveal that greater wall distances result in smaller cuttable regions. On the other hand, we found that wall approaches closer than approximately 0.6m presented interference problems between the cutting tool and Atlas' body.

As a result of this analysis, we conclude the following. Using the chosen cutting tool, it is optimal for Atlas to approach a wall head-on with a distance of approximately 0.6m from wall to pelvis origin. This pose optimizes the cuttable area. Further regarding planned approach, the operator can use the results of Fig 3, lower row, second panel from the left, as a template for planning cutting. We have created a template from this analysis, which the operator can overlay on a scene in Rviz to attempt to optimize cutting-contour design to satisfy the cut-out requirements subject to inverse-kinematic feasible options. This approach is consistent with the philosophy of visualizing "affordances", as described in [10].

Once the operator locates the cuttability template on the target wall, the corresponding robot pose (i.e., foot placements) follows. This leads to automatic generation of a stepping plan, as shown in Fig 4, to arrive at the optimal pose (following the process described in [9]).

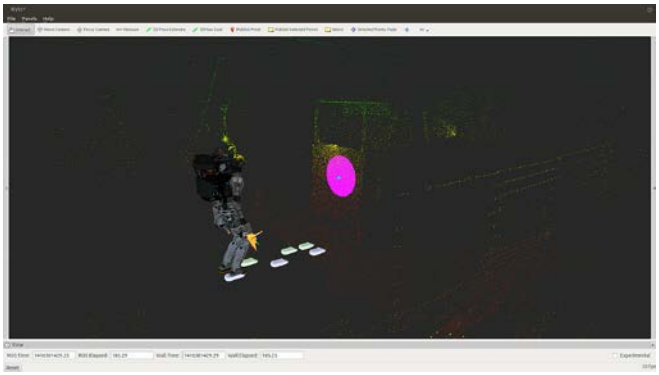

Figure 4: Operator's view of auto-generated stepping plan, based on operator's selection of target wall region and pre-optimized wall-approach pose.

Having established an approach plan, the next consideration is how to specify a specific contour to cut, once the robot is in position in front of the wall.

III. Cutting Contour Specification

From an Rviz display of the robots's sensors at the remote Operator Console Station (OCS), the operator can perceive the target wall to cut and consider viable cutting contours to perform a given task. Choosing a specific contour among alternatives can be difficult, since the specified contour should be restricted to achievable (cuttable) points.

To address this problem, we invoke the results of our cuttability analysis. Once the robot has approached the wall per the approach plan, the resulting pose may be expected to be close to, but not precisely at the target pose. When the robot comes to a halt, the wall is scanned by Atlas's 3-D LIDAR in his sensor head. From this scan, the location of the wall (distance and normal) is easily obtained, as described in [7]. From this pose, the cuttability regions are recomputed for this specific case, and the result is graphically overlayed on the operator's view. The graphical overlay shows the operator the regions in which a cutting contour may be defined (Fig 5).

In our operator interface, the operator defines a cutting contour as a polyline. This is performed by selecting patches of points from the point cloud in Rviz [11]. Using only a mouse and a click-drag operation, small regions (patches) of 3-D points are selected, from which a 3-D centroid is computed, and this result defines a vertex on the wall. Vertices are specified sequentially in this manner, allowing the operator to specify an arbitrary cutting path. Since the operator is provided with a graphical overlay of cuttable regions, the path can be designed intelligently to lie within these regions, thus prescribing a task that is kinematically achievable.

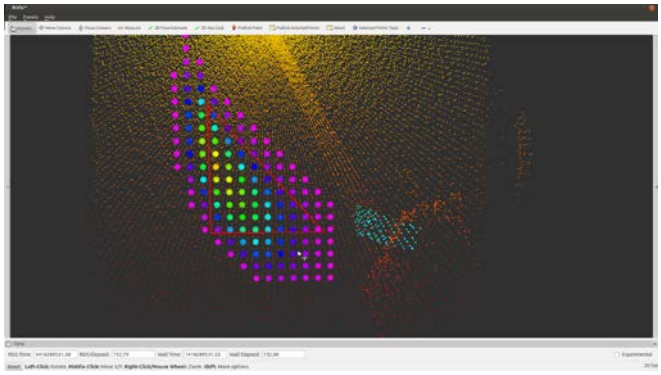

Figure 5: Rviz view of point-cloud data on a wall, 0.6m in front of Atlas, with a graphical overlay indicating the cuttable regions. The operator has selected points to define 3 vertices implying a triangle to be cut (shown in red). By restricting path definition to the cuttable region, existence of an inverse-kinematic path solution is assured.

A view of our operator interface in Rviz is shown in Fig 5. The scene is a screen capture of the operator defining a cutting contour. The view contains LIDAR points from Atlas facing a wall, 0.6m from the wall (relative to the pelvis origin). The cyan points resulted from the operator choosing a patch of points on the wall, helping to establish the relative wall offset and angle. A colored, graphical overlay indicates the cuttable region. The operator selects patches of points to imply desired vertices. In Fig 5, the operator has defined 3 vertices comprising a triangle (shown in red) to be cut. The operator chooses a path that lies entirely within cuttable space, thus assuring existence of an inverse-kinematic solution for the cutting plan.

One limitation of the Rviz point-cloud view is that the LIDAR-based points do not include visualization of texture on the wall, including key markings that may indicate visually where a cut must be performed. An alternative

operator view is shown in Figs 6 and 7.

In Fig 6, a poster on the wall indicates a region in which to cut. The flat green color of the interior of the large triangle offers little contrast, and thus stereo vision has difficulty triangulating 3-D points. Figure 6 includes a blue-tint overlay, indicating regions in which 3-D point-cloud points are available. Since points derived from stereo images constitute an organized point cloud, there is a computable mapping from 3-D onto 2-D overlay, as well as an inverse mapping that allows inference of 3-D points from selection of 2-D image points—provided corresponding 3-D points were identifiable. As shown by the red rectangular outline, the operator has selected a region of interest by interacting directly with the 2-D image. The operator can see that 3-D points are available in this region (per the blue tint). From selection of the 2-D region, we find the corresponding 3-D points, compute a centroid, and assign the result as a vertex of the desired cutting polyline.

Figure 7 shows the operator's view incorporating both LIDAR point-cloud and 2-D image interfaces. Selection of vertices is illustrated in the display, allowing the operator to confirm the specification before execution.

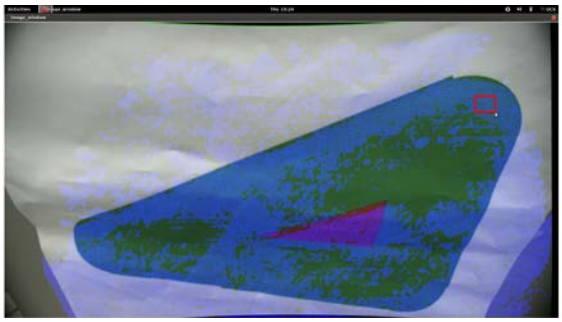

Figure 6: 2-D image with overlay of viable, stereo-based 3-D points. The red rectangle is the operator's selection of a region. Since this region is shown to have corresponding 3-D points available, the selection in 2-D yields 3-D coordinates of an implied vertex.

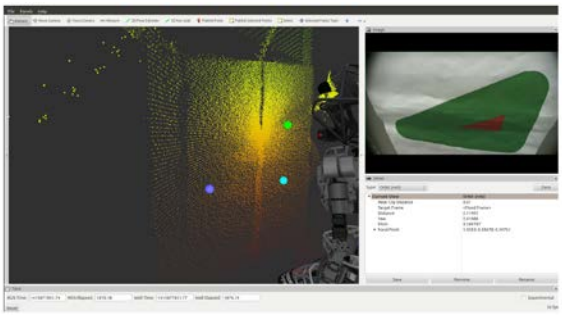

Figure 7: Operator's view including both LIDAR point-cloud display and 2-D image display. Chosen vertices of a triangle to be cut are illustrated for the operator to review prior to execution.

Once the operator has specified a polyline, a desired cutting path is implied. This path is subsequently sampled along the contour from initial to final vertex, defining a sequence of wall points to be cut by the robot. The next task is to convert the path from task space to joint space for robot execution. To simplify this, we consider the legs as a separate subsystem. Details on this subsystem are given in Section IV. Assuming pelvis height is a controllable variable, it is treated as one of the 8 DOFs for path optimization. For sample points along the task-space path, each such point is analyzed for cuttability, which can result in a large number of options per task-space point. An optimal path in joint space is found through the available options using linear programming. In contrast to methods using RRT planning, e.g. [12], the current plan optimization is simpler. Since the planning problem to be solved conforms to the "Stagecoach Problem", an optimal path can be found efficiently using the algorithm described in [13].

IV. Pelvis Subsystem Control

The chosen 8 degrees of freedom for path-cutting kinematics includes control of pelvis elevation. In fact, control of this degree of freedom requires coordination of 12 leg joints which, with both feet on the ground, comprise a closed-chain mechanism. We control these 12 joints as a subsystem, providing a simplified interface to perform motion control of pelvis height, which we call "hku_manipulate."

Although the lower body has 12 controllable joints, there are only 6 degrees of freedom of mobility, taking into account maintaining no-slip conditions of the feet. With respect to the right foot (which is to be kept stationary on the ground during manipulation), the pelvis frame twist (6-DOF velocity) follows from the right-leg joint velocities through a 6x6 Jacobian. For non-singular poses, right-leg joint velocities can be computed corresponding to a desired pelvis twist. Left-leg joint velocities follow similarly from the left-leg Jacobian.

Since the lower-body subsystem comprises a closed-chain mechanism, internal stresses can build up from accumulation of Jacobian-based motion errors. This is addressed, in part, by responding to sensed ankle torques. The ankle joint commands are continuously corrected to keep the sensed center of pressure as close as possible to the middle of each foot, which results in keeping the feet flat on the ground.

The ankle force-torque sensors are also summed to compute the net moment on the robot with respect to ground. This signal identifies the zero-moment point (ZMP [14]). For balancing, pelvis x-y motion is commanded to keep the ZMP inside the support polygon of the feet. Optionally, active balancing is suppressed during manipulation, allowing the ZMP to move around within the support polygon. However, if the ZMP approaches dangerously close to the boundary of the support polygon, the balance system intervenes to prevent falling.

The range of motion of the pelvis in the z-direction ("squatting") is limited by both joint range-of-motion limits and joint torque limits. Experimentally, we are able to

achieve a stable range of motion of pelvis height from 0.5m to 0.85m above the floor. This range of motion is imposed on the inverse-kinematic computations for computing cutting paths.

Although the pelvis can be controlled in 6-DOF, rotating the pelvis does not seem advantageous. Pelvis tilt forces the hip joints closer to their joint limits, and the equivalent mobility may be achieved using torso joints. Therefore, a simplification is to maintain pelvis orientation fixed, which reduces the remaining mobility to 3 translational degrees of freedom. At present, we only utilize pelvis vertical motion in cutting-path solutions. Motion in the x direction (front/back) is largely constrained by maintaining the ZMP within the support polygon to maintain balance. In ongoing work, we are incorporating use of lateral pelvis motion, since the ZMP can be shifted laterally by approximately +/-15cm within the support polygon, depending on the width of stance.

Figure 8 shows the ZMP of Atlas during a cutting operation. This data was obtained from the foot force/torque sensors. The reference frame is the support polygon with center defined at (0,0). The support polygon ranges +/-0.13m in the x direction, and +/-0.17m in the y direction. After walking towards the wall, Atlas establishes his stance for cutting, and hku_manipulate causes the pelvis to translate horizontally to place the ZMP at the center of the support polygon, as validated by Fig 8. Although the cutting forces did induce some motion of the ZMP, these forces were relatively low, and the ZMP remained close to the center of the support polygon, far enough from the safety zone that no active pelvis corrections were required during cutting. Squatting motions during cutting also resulted in some center-of-mass motion in the x direction, but this was a relatively small effect. Since active balancing was not required during cutting, pelvis x-y displacements were small, and thus pelvis lateral dynamics was conveniently decoupled from arm and torso motions.

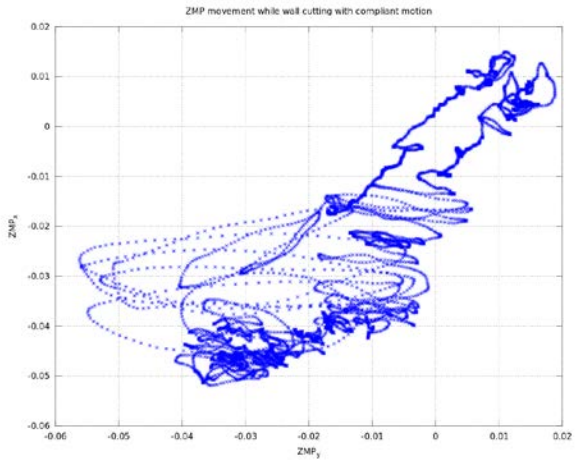

Figure 8: Zero Moment Point during cutting operation. The ZMP remained within a few cm of the center of the support polygon, and thus active balancing was not necessary during cutting.

V. Compliant Motion Control for Wall Cutting

Our wall-cutting solution needs to be robust with respect to a variety of uncertainties. There will be some error in measuring wall distance and angle, and further the wall is assumed to be vertical and flat, which may not be true. There will be servo-following errors in executing the planned path, notably from Coulomb friction and varying gravity loading. Kinematic calibration errors also result in distortion of the intended task-space path. Cutting forces will result in path-following errors, both due to arm deflections and whole-body deflections. Collectively, these errors can result in failure. The imprecise path of the tool might miss the wall altogether or can result in excessive forces against the wall, causing the tool to jam against the wall, or causing Atlas to drop the tool, or even induce falling.

Figure 9 shows a recording of a cutting path with errors deliberately introduced. (The wall was offset and rotated relative to the planned motion). In this recording, the tool bit was removed as the robot attempted to slide the tool along the wall in the desired trajectory, maintaining contact with the tool guide. As shown, the normal forces against the wall (y-direction, green trace) varied from a peak of nearly 40N to zero (at which point the tool had lost contact with the wall). As the normal forces increased, the tangential forces (red and blue traces) also increased due to friction between the wall and tool.

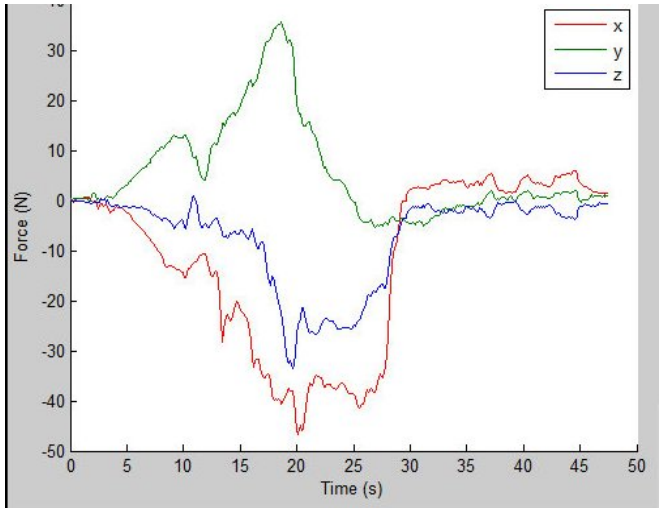

Figure 9: Wall-following forces without compliance. Wall position and angle errors were deliberately introduced to observe the resulting contact forces as the robot was commanded to move the tool along the wall in a prescribed cutting path.

Figure 10 shows the same situation as Fig 9, but with compliant-motion control invoked. As shown, in spite of errors between the computed trajectory and the wall, normal forces were maintained at roughly 10 to 20N, with corresponding reduction in tangential forces. Further, the tool never lost contact with the wall. Regulation of contact forces helps to accommodate uncertainty, supporting skillful cutting while avoiding excessive forces.

At present, our exploitation of compliant motion is rudimentary, depending on only one compliantly-controlled joint. Since our example paths resulted in the forearm staying approximately parallel to the wall and with the elbow bent, it was adequate to use compliance exclusively on the humerus joint rotation to provide a compliant pre-load force against the wall. In ongoing work, this is being generalized to higher-dimensional, constrained manipulation control (see, e.g., [15,16]).

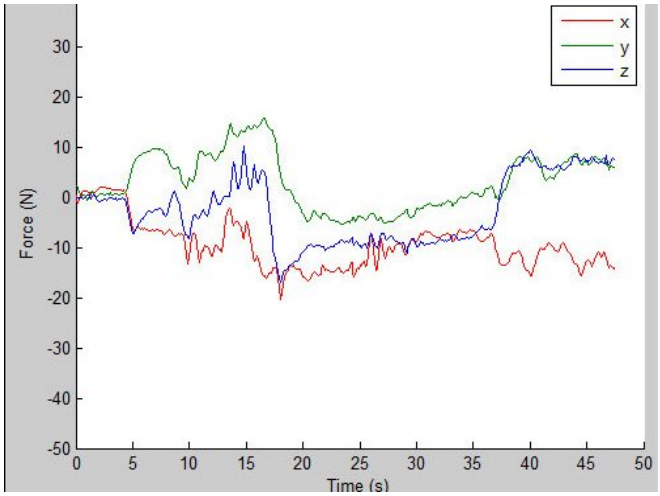

Figure 10: Wall-following forces with compliance. Normal forces against the wall are reduced while contact with the wall is maintained throughout the trajectory.

VI. Results

Our optimization analysis and integrated system have yielded a fast, simple and robust means for wall cutting. The operator can visualize both the environment and an overlay of cuttability. Using only a mouse, the operator can indicate a target wall, which automatically generates a stepping plan to approach the wall optimally for cutting. Once the robot is in position, the wall is re-scanned, presenting the operator with a visualization of the work surface overlayed with a cuttability template. From the interface, patches of point clouds are easily selected, implying vertices of a contour to be cut. This leads to automatic generation of an optimized inverse-kinematic path in 8-DOF. Execution of this plan is performed with compliant-motion control, which can accommodate significant uncertainty and help assure a successful cut while regulating contact forces to avoid jamming, dropping the tool, or falling.

Figure 11 shows an example of wall-board that was cut by Atlas in our lab, based on a specification of triangular contour using our described operator interface. The cutting speed was set to 2cm/sec (which is not yet optimized). As shown, the resulting cut is relatively clean, although not precise. In ongoing work, we are addressing the sources of error to improve cutting precision.

Overall, the wall-cutting system implementation is successful. A remote operator can visualize a wall-cutting task via both 2-D color images and via 3-D point-cloud displays. User interaction only requires a mouse, and specification of a cutting contour requires mere seconds. Subsequently, wall cutting is robust and fully autonomous.

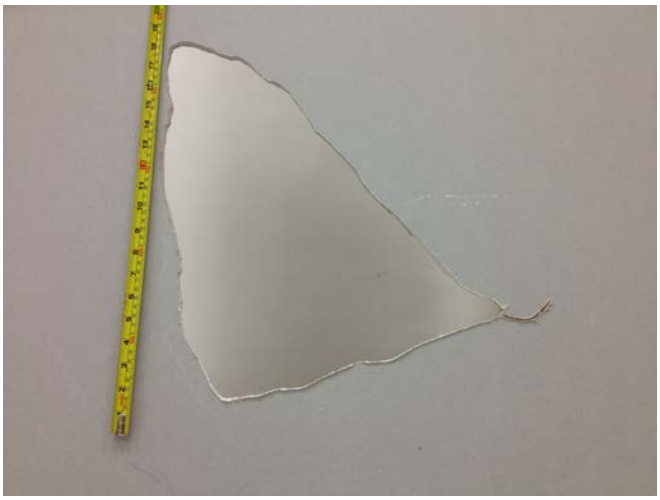

Figure 11: Wall-board cutting performed by Atlas at 2cm/sec, using a contour specified via our operator interface.

References

[1] Stephen Cass, "DARPA Unveils Atlas DRC Robot", IEEE Spectrum, Jul 2013.

[2] http://www.theroboticschallenge.org

[3] M. Quigley, C. Salisbury, A. Ng, J.K. Salisbury, "Mechatronic Design of an integrated Robotic Hand", Int J. Robotics Research, Feb, 2014.

[4] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2).

[5] Cong Du, Kit-Hang Lee and Wyatt Newman, "Manipulation Planning for the Atlas Humanoid Robot", to appear in IEEE Int. Conf. on Robotics and Biomimetics, 2014.

[6] Kit-Hang Lee and Wyatt Newman, "Natural Admittance Control of an Electro-Hydraulic Humanoid Robot", to appear in IEEE Int. Conf. on Robotics and Biomimetics, 2014.

[7] Li Ma, Ernest Cheung and Wyatt Newman, "Using Planar Features for Fast Localization in Indoor Environments," to appear in IEEE Int. Conf. on Robotics and Biomimetics, 2014.

[8] Ernest Cheung, Chao Cao and Wyatt Newman, "Initial Pose Estimation Using Cross-Section Contours" to appear in IEEE Int. Conf. on Robotics and Biomimetics, 2014.

[9] Zheng Hao Chong, et al., "Autonomous Valve Turning with an Atlas Humanoid Robot", to appear in 2014 IEEE-RAS International Conference on Humanoid Robots.

[10] Maurice Fallon, et al., "An Architecture for Online Affordance-Based Perception and Whole-body Planning," M.I.T. Comp. Sci. and A.I. Lab Technical Rpt, MIT-CSAIL-TR-2014-003, March, 2014.

[11] Rusu, R.B., Cousins, S., "3-D is Here: Point Cloud Library (PCL)", IEEE ICRA 2011.

[12] F.S. Hillier and G.J. Lieberman, Introduction to Operations Research, Holden-Day, 1967.

[13] F. Burget, A. Hornung and M. Bennewitz, "Whole Body Motion Planning for Manipulation of Articulated Objects", ICRA 2013.

[14] M. Vukobratovic and B. Borovac, "Zero-Moment Point: 35 Years of Its Life," Int. J. Humanoid Robotics, Vol 1, no1, 2004, pp 157-173.

[15] A. Dietrich, T. Wimbock, A. Albu-Schaffer, "Dynamic Whole-Body Mobile Manipulation with a Torque Controlled Humanoid Robot via Impedance Control Laws", IROS 2013.

[16] Deal, Aaron, Chow, Der-Lin, and Newman, Wyatt, "Hybrid Natural Admittance Control for Laparoscopic Surgery," Proceedings of the 2012 IEEE Intelligent Robots and Systems Conference, October, 2012.