Supervisory Control of a DaVinci Surgical Robot

Der-Lin Chow, Peng Xu, Eser Tuna, Siqi Huang, M. Cenk Çavuşoğlu, and Wyatt Newman

***Abstract*— This paper presents an approach to supervisory control of a DaVinci surgical robot. At present, such robots are controlled by teleoperation, with dissimilar kinematics of the operator interface vs. the robot. As a result, it can be difficult for the operator to visualize the kinematic restrictions on the robot, particularly for desired extended, precise trajectories, such as circular needle driving. The interface presented here constitutes a means to elevate the operator from teleoperation mode to supervisory mode. The operator interacts directly with a point-cloud display, allowing selection of task specifications from which the system automatically computes and executes precise trajectories to achieve the task goals. The intent is to allow the operator to focus on task specifications and rely on automation to achieve faster and more precise execution.**

I. Introduction

Robotic Assisted Minimally Invasive Surgery (RAMIS) has the appeal of smaller incision wounds, faster recovery and reduced risk of infection. At present, however, using robots in surgery is slower and clumsier than conventional, more-invasive surgery. Loss of tactile feedback, poor endoscopic vision, and workspace constraints make teleoperated surgical procedures challenging to surgeons [1]. Notably, suturing and knot-tying are difficult, tedious and time consuming tasks to perform in minimally-invasive surgery.

Current RAMIS applications are still limited to direct teleoperation by surgeons. While fully autonomous robotic surgery will continue to be a challenge into the foreseeable future, some low-level tasks such as needle driving, suture pulling and knot-tying, could be performed by robots automatically, under the supervision of surgeons. Ideally, surgeons would select locations (e.g. needle entry and exit points), interacting with a visual display, implicitly commanding actions to be performed by the robot. Ideally, the robot would perform the implied tasks faster and better than a human surgeon.

Due to semi-circular needle shape, many researchers have proposed circular needle-driving path to limit the tissue deformation (tearing) [2–4]. Building on prior work, (notably [5]), this paper focuses on how to perform RAMIS circular needle-driving automatically. Our approach has three components: development of fast inverse kinematics for DaVinci needle driving; a human/machine interface that simplifies setting task specifications for needle driving; and automated planning and execution of needle driving. The present state of development is simulation within the Robot Operating System (ROS) [6], extending capabilities of the Davinci Research Kit (DVRK) [7].

The authors are with the EECS Department at Case Western Reserve University. The corresponding author, Professor Newman, can be reached at: Wyatt.Newman@case.edu.

II. System Modeling

The DaVinci robot in our lab is shown in Fig 1. A "Patient-Side Manipulator" (PSM) can be pre-positioned via large, proximal, non-servoed joints, establishing 3-D coordinates of an entry portal to the patient. Choosing where to place the entry portals can be a complex planning process in itself (see e.g. [8]). Once the portal is established, it is still possible to reposition the non-servoed joints, but subject to the constraint that the portal is fixed in space. During teleoperation, only the distal, servoed joints are moved.

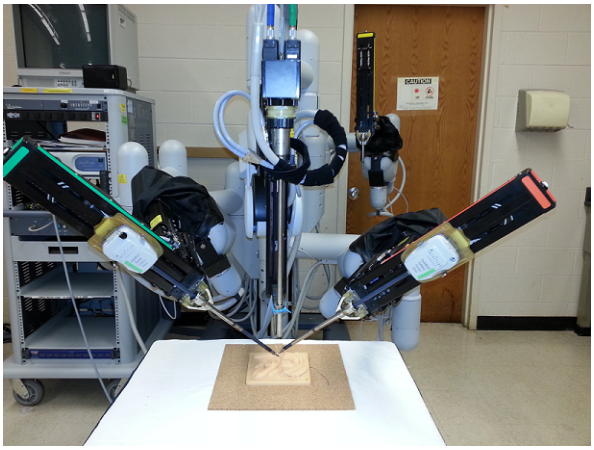

Fig. 1: View of a DaVinci surgical robot with a stereo endoscope and two patient-side manipulators

The servoed joints include 3 "base" joints plus up to 4 tool joints (with interchangeable tools). The base joints comprise a (R-R-P) mechanism that is kinematically constrained to pivot about a fixed point in space (which corresponds to the portal coordinates), plus a prismatic insertion degree of freedom through the portal. In the present work, we will focus on a needle-driver tool that has 4 degrees of freedom (see Fig 2). These are: a roll about the tool shaft, a pitch about the tool wrist, and rotation of two gripper fingers.

A model of a DaVinci surgical robot has been designed in prior research at Johns Hopkins and at Worcester Polytechnic [7], and we are building on this work. The existing model allows display of the servoed joints in "Rviz" ([9]), a visualization tool in ROS. An example of this display with two arms is shown in Fig 3. Within ROS, one can command joint angles, resulting in animated display of the arms. Such visualization can be highly useful for developing and previewing motion plans. However, Rviz is limited to visualization–it does not consider robot dynamics, physical interactions between bodies (e.g. picking up a needle), nor

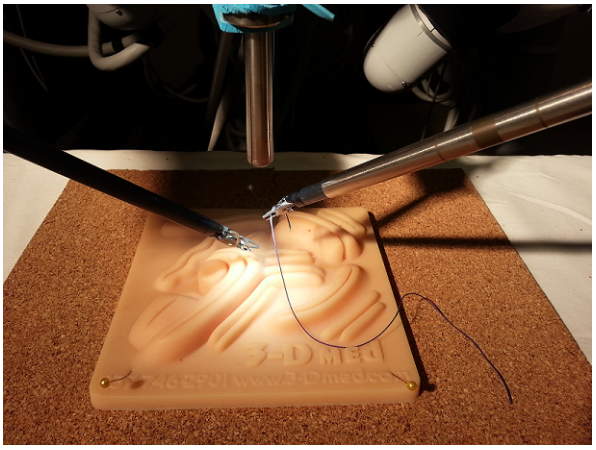

Fig. 2: View of DaVinci manipulators with needle-driver grasping a needle

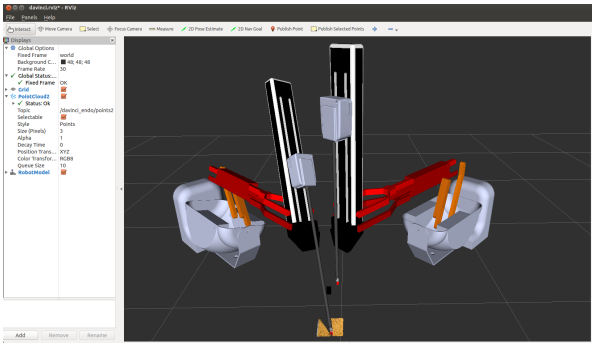

Fig. 3: View of DaVinci model within Rviz

does it emulate sensors.

Additional capability can be obtained through the use of "Gazebo" for simulation [10]. Gazebo also offers animated visualization, but it additionally incorporates a physics engine for dynamic simulation. With such simulation, one can change a modeled environment (e.g., pick up a needle and hand it off to another gripper), and one can simulate sensors. In particular, in our extensions to Gazebo simulation of the DaVinci robot, we exploit emulation of stereo cameras, which offers the opportunity to develop and simulate sensory-guided behaviors.

To obtain realistic physics simulation of the DaVinci endoscope, we first performed calibration steps on a physical DaVinci system. The relevant parameters include: 640x480 image capture from the two cameras; a horizontal field-of-view of 0.70 rad, and a baseline (inter-ocular distance) of approximately 5.8mm. These properties were then input into our Gazebo model, so that emulation of stereo vision would be representative of the physical system.

Additionally, to get a realistic simulation of camera properties, we took a snapshot of a textured surface with the DaVinci endoscope from a distance of approximately 80mm–and this snapshot from the physical system is introduced in our simulation as a textured model (see Fig 4). We currently use this image as a stand-in for tissue visualization. In fact, tissue scenes that are bland (have insufficient texture) are unsuitable for stero vision, as there are too few distinctive features to address the left-camera/right-camera pixel correspondence problem. (In ongoing research, we are exploring structured lighting to address this issue.).

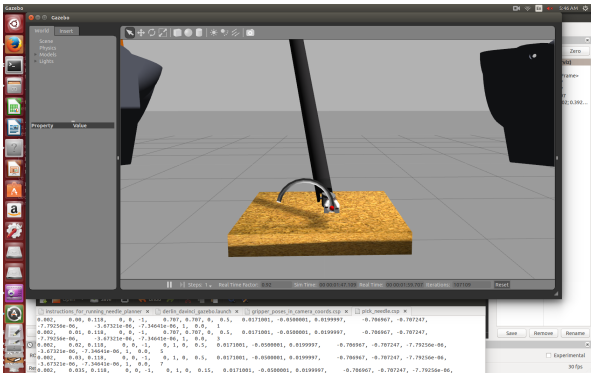

Fig. 4: View of DaVinci model and textured surface within Gazebo

With these additions, Rviz's capabilities are enhanced, allowing for 3-D display of point-cloud values computed from stereo cameras. Incorporation of point-cloud data within the Rviz display offers two important capabilities. First, the operator can rotate and zoom on the view, which helps to establish 3-D visualization better than can be achieved with the default stereoscopic view. Second, the point-cloud data in Rviz offers opportunity for a natural user interface. One can, for example, interactively "grab" points of interest via a click-drag operation via a mouse to imply 3-D points of interest. This capability constitutes our proposed graphical user interface, which will be described further in Section IV.

Our simulation of a DaVinci surgical robot also includes knowledge of the joint limits of the servoed joints. Consideration of joint limits is important for automated trajectory planning and for provided graphical assistance to the operator regarding reachability of poses of interest. Such graphical assistance can be overlayed on the sensor data, equivalent to a heads-up display (as described further in section IV).

A key component of our supervisory control interface is the ability to plan candidate trajectories and inform the operator of viable options. This process requires fast and dependable inverse kinematics, which is discussed next.

III. Analytic Inverse Kinematics for the DaVinci Robot

In prior work, inverse-kinematic (IK) software for the DaVinci robot has been developed [7]. This prior work used Jacobians and numerical iterations to derive IK solutions. In general with numerical inverse-kinematic approaches, performance suffers from cases of failure to converge, as well as inability to find multiple solutions. In our evaluation of the existing numerical IK code, we found instances of inability to find known, valid solutions; instances of failure to converge; and instances of convergence to illegal solutions.

An analytic solution, introduced here, does not suffer from these conditions.

A. Inverse Kinematics Approach

The needle-driver tool we consider here is an Endowrist[1]. The tool plus DaVinci joints together offer a total of six degrees of freedom of motion of the tool tip (plus grip actuation) [11]. An analytic inverse-kinematic (IK) solution is desirable for its speed and reliability. Pieper [12] showed that one could always derive an analytic IK solution for a 6-DOF manipulator with a spherical wrist. The Endowrist, however, does not correspond to a spherical wrist, and thus an analytic IK solution is not immediately obvious.

When an analytic IK solution is not known, iterative, numerical techniques based on calculation of the Jacobian inverse [13],[14] are typically used instead. The inverse kinematics implementation provided by the DaVinci Research Kit [7] is based on the numerical Newton algorithm, one of the popular methods studied in the literature [15], which (ideally) iteratively converges to a single solution based on an initial guess.

However, numerical IK solutions have several drawbacks [16], including: can be slow to converge; can fail to converge; or can converge to illegal solutions (e.g. violating joint limits). To address these concerns, we present an analytic IK solution for the DaVinci with an Endowrist tool.

The kinematics of this system can be described in Denavit-Hartenberg representation [17], as shown in Table I. (For convenience, we additionally define static transforms from a convenient PSM base frame to the D-H frame 0; a static transform from D-H frame 6 to a convenient gripper-tip frame; and a static transforms from the camera frame to each PSM base frame).

Joint	Joint Type	θ_i [rad]	α_i [rad]	a_i [m]	d_i [m]
1	R	θ_1	0.5*π	0	0
2	R	0.5*π + θ_2	0.5*π	0	0
3	P	0	0	0	d_3
4	R	π + θ_4	0.5*π	0	0
5	R	0.5*π +θ_5	-0.5*π	a_5	0
6	R	0.5*π + θ_6	0.5*π	a_6	0

TABLE I: D-H parameters. Here a_6 is the gripper jaw length, $a_6 = 0.0102\ m$ and a_5 is the distance from wrist bend axis to gripper jaw rotation axis, $a_5 = 0.0091\ m$.

The limits of the described joints are as follows: -1 $\leq \theta_1 \leq$ 1, -0.7 $\leq \theta_2 \leq$ 0.7, 0.01 $\leq d_3 \leq$ 0.23, -2.25 $\leq \theta_4 \leq$ 2.25, -1.57 $\leq \theta_5 \leq$ 1.57, -1.39 $\leq \theta_6 \leq$ 1.39.

Following the approach of [12], the geometric solution to the IK problem is decomposed into two parts. Although a spherical "wrist point" does not exist for this robot, we refer to a useful "wrist point" as the origin of D-H frame 4, ${}^{base}\mathbf{o}_4$. It will be shown that the coordinates of this wrist point can be computed based on a given desired gripper pose. From this wrist point, solutions for the first three joint displacements θ_1, θ_2, and d_3 can be calculated. Once the first three joints are computed, the remaining three joint values (wrist rotations) are readily solved.

To help visualize our approach, consider Fig 5. This figure illustrates three of the Denavit-Hartenberg coordinate frames, including the base frame (with origin $\mathbf{o}_{base}$ at the portal coordinates), frame 4, with z-axis $\mathbf{z}_4$ through the wrist-bend rotation axis, and frame 5, with z-axis $\mathbf{z}_5$ through the last (gripper-jaw rotation) degree of freedom. The $\mathbf{x}_5$ axis points in the direction from origin $\mathbf{o}_4$ to origin $\mathbf{o}_5$ with corresponding DH offset parameter a_5. The fact that a_5 is nonzero is what makes this wrist non-spherical and thus more difficult to solve.

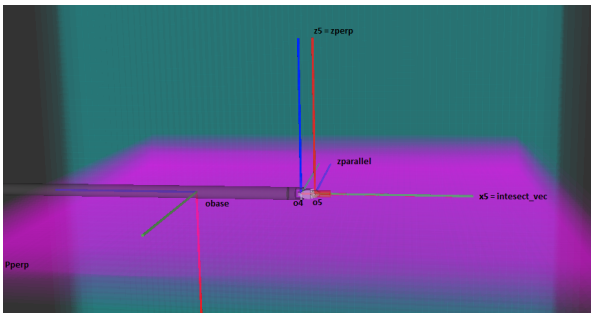

Fig. 5: Definition of D-H frames for the Endowrist. Consideration of planes parallel to and perpendicular to the z5 joint axis helps to visualize how to deduce origin-4 from specification of a desired gripper pose.

Assume a desired gripper configuration, ${}^{base}T_{tip}$, is the gripper-tip frame with respect to the base frame. (The origin of the base frame is located at the trocar point). Specification of ${}^{base}T_{tip}$ fully defines the 3-D components of ${}^{base}\mathbf{p}_{tip}$ (coordinates of the gripper tip with respect to the base frame) and the 3-D components of ${}^{base}\mathbf{z}_{tip}$ (the direction of the z-axis of the tip frame, which points from $\mathbf{o}_5$ towards the gripper tip). We can immediately infer the corresponding frame-5 origin and joint axis:

$$\mathbf{z}_5 = \text{-}\mathbf{x}_{tip} \tag{1}$$

$$\mathbf{o}_5 = \mathbf{p}_{tip} - a_6\mathbf{z}_{tip} \tag{2}$$

In order to calculate the wrist point, the following geometrical approach is employed. Consider two planes, $P_{parallel}$ (the green plane in Fig 5) and P_{perp} (the purple plane in Fig 5). These are constructed as follows. Plane P_{perp} is perpendicular to axis $\mathbf{z}_5$ and passes through point $\mathbf{o}_5$. Plane $P_{parallel}$ contains $\mathbf{z}_5$ (and thus also contains $\mathbf{o}_5$), and this plane also passes through the trocar point, $\mathbf{o}_0$ (by construction). We claim that both of these planes contain the wrist point, $\mathbf{o}_4$. Based on this assumption, the axis $\mathbf{x}_5$ lies along the intersection of planes $P_{parallel}$ and P_{perp}.

Equivalently,

$$\mathbf{x}_5 = (\mathbf{z}_5 \times (\mathbf{z}_5 \times \mathbf{o}_5))/||\mathbf{z}_5 \times (\mathbf{z}_5 \times \mathbf{o}_5)|| \tag{3}$$

(The denominator $||\mathbf{z}_5 \times (\mathbf{z}_5 \times \mathbf{o}_5)||$ cannot be zero, since the $\mathbf{z}_5$ vector cannot be parallel to the $\mathbf{o}_5$ vector, as this would

[1]Intuitive, EndoWrist and DaVinci are trademarks of Intuitive Surgical, Inc., Mountain View, CA.

require a wrist bend in excess of $\pi/2$, which exceeds the wrist-bend joint limits).

Although motivated by geometric visualization, Eqn 3 can be proven algebraically. For each Denavit-Hartenberg frame, there is a corresponding 4x4 homogeneous transformation matrix, ${}^{(i\ \ 1)}\mathbf{A}_{(i)}(q_i)$ for which the components are functions of the Denavit-Hartenberg variables (from Table I). Each of these transforms contains trigonometric functions of one joint displacement. Multiplying the first five transforms together yields ${}^{0}\mathbf{A}_5(\theta_1, \theta_2, d_3, \theta_4, \theta_5)$. Conceptually, if these 5 matrices are multiplied together, the resulting 4x4 contains trigonometric expressions for $\mathbf{x}_5$ (from the first column), $\mathbf{z}_5$ (from the third column) and $\mathbf{o}_5$ (from the fourth column). Given these expressions, one can compute $\mathbf{z}_5 \times (\mathbf{z}_5 \times \mathbf{o}_5)$ symbolically, as well as a symbolic expression for $\mathbf{x}_5 \times ((\mathbf{z}_5 \times (\mathbf{z}_5 \times \mathbf{o}_5))$. This latter expression should evaluate to a 3x1 zero vector ($[0; 0; 0]$) if our proposition of Eqn 3 for computation of $\mathbf{x}_5$ is true (i.e., that $\mathbf{x}_5$ lies along the intersection of P_{perp} and $P_{parallel}$).

Actually performing the algebraic computation of the symbolic expression for $\mathbf{x}_5 \times (\mathbf{z}_5 \times (\mathbf{z}_5 \times \mathbf{o}_5))$ is extraordinarily tedious, lengthy and prone to human error. However, this expression is computable in Mathematica[2]. Using Mathematica's proof capabilities, the expression $\mathbf{x}_5 \times (\mathbf{z}_5 \times (\mathbf{z}_5 \times \mathbf{o}_5))$ evaluates identically to a zero vector, thus proving our assertion in general.

Having derived a means to compute $\mathbf{x}_5$ from a desired ${}^{base}T_{tip}$, the wrist point $\mathbf{o}_4$ follows as:

$$\mathbf{o}_4 = \mathbf{o}_5 - a_5\mathbf{x}_5 \tag{4}$$

which yields numerical values for this frame origin. Given coordinate values for this wrist point, the corresponding first three joint displacements follow.

The tool-insertion distance, d_3, is simply $d_3 = ||\mathbf{o}_4||$.

The wrist point $\mathbf{w} = \mathbf{o}_4$ can be computed symbolically as the fourth column of the product of 4x4 matrices ${}^{0}\mathbf{A}_4(\theta_1, \theta_2, d_3, \theta_4) = {}^{0}\mathbf{A}_1(\theta_1){}^{1}\mathbf{A}_2(\theta_2){}^{2}\mathbf{A}_3(d_3){}^{3}\mathbf{A}_4(\theta_4)$. Multiplying these out yields:

$$\mathbf{w} = \mathbf{o}_4 = [\cos(\theta_1)\sin(\theta_2), \sin(\theta_1)\sin(\theta_2), -\cos(\theta_2)]^T d_3 \tag{5}$$

Correspondingly, one can solve for θ_1 and θ_2 as: Thus,

$$\theta_2 = \cos\ {}^{1}(w_z) \tag{6a}$$

$$\theta_1 = \text{atan2}(w_y, w_x) \tag{6b}$$

Within the Endowrist joint limits, these solutions are unique.

With the first three joints displacements known, the next three joint values can be computed using the strategy described in Pieper [12]. The solved values for θ_1, θ_2 and d_3 are used within the computation of ${}^{0}\mathbf{A}_3(\theta_1, \theta_2, d_3)$. Inverting this matrix yields ${}^{3}\mathbf{A}_6(\theta_4, \theta_5, \theta_6) = ({}^{0}\mathbf{A}_3{}^{1}){}^{0}\mathbf{T}_6$. In this expression, all values on the right-hand side are known, and these can be matched to trigonometric expressions on the left-hand side to solve for θ_4, θ_5, and θ_6. (The solution, omitted here, is straightforward).

[2]Mathematica is a registered trademark of Wolfram Research, Inc., Champaign, IL

An unusual but useful property of the IK solution for the DaVinci arm plus Endowrist is: if a valid solution exists (within the joint limits), then this solution is unique. More generally, a 6-DOF robot may have, e.g., 8 IK solutions (although only a subset of these may be reachable). The uniqueness of valid IK solutions simplifies our trajectory-planning computations. Our analytic IK solution approach correctly identifies whether an IK solution exists, and if the solution exists, the solver yields the unique solution quickly and precisely. We subsequently rely on this capability for our automated planning.

B. Performance Evaluation

To validate our inverse-kinematics solution, a numerical test was performed. A routine looped through generation of random joint values, restricted to legal joint-value ranges. From these values, a corresponding (unique, unambiguous) forward-kinematic solution was computed. This forward-kinematic solution (a 6-DOF gripper pose) was then used as input to the IK solver to compute the joint angles corresponding to this gripper pose. In this evaluation, it was thus known (by construction) that every IK request had a valid solution, and it was further known precisely what that joint-space solution was. This loop was run for hundreds of thousands of iterations, and each solution was confirmed to match the known solution. If a solution existed, it was always discovered by our IK code. This test demonstrated that our IK solution was reliable, and that IK solutions–when they exist within joint constraints–are unique. Our IK solver also reliably recognized when no valid IK solution existed.

This same test was run using the previously-existing numerical IK solver. The numerical approach revealed the expected difficulties. It sometimes failed to converge, yielding either wrong solutions or illegal (out-of-range) alternative solutions. In addition to providing reliability, our analytic solution was also approximately 8x faster than the numerical approach, which is important for our real-time planning use.

IV. A Human/Machine Interface for Supervisory Control of Needle Driving

Using our Gazebo model of a DaVinci, including emulation of stereo vision, we can exploit the capabilities of Rviz to develop a natural graphical user interface for supervisory control.

Figure 6 illustrates the steps of our interface in sequential Rviz views. The initial scene, (a), shows the operator's view as a colored, 3-D point-cloud. The scene is created by (emulated) stereo vision, where the vision system is looking at our sample, textured image that has been additionally decorated with green fiducials (for the purpose of visualizing reference points). The operator has the opportunity to rotate and translate the scene to get a better sense of 3-D and a preferred viewpoint (subject to the limitation that only points within line-of-sight from the endoscope will be viewable).

Scene (b) of Fig 6 shows the user selecting a small patch of points on the rendered point-cloud. (The faint, yellow, rectangular patch of selected points is indicated by a

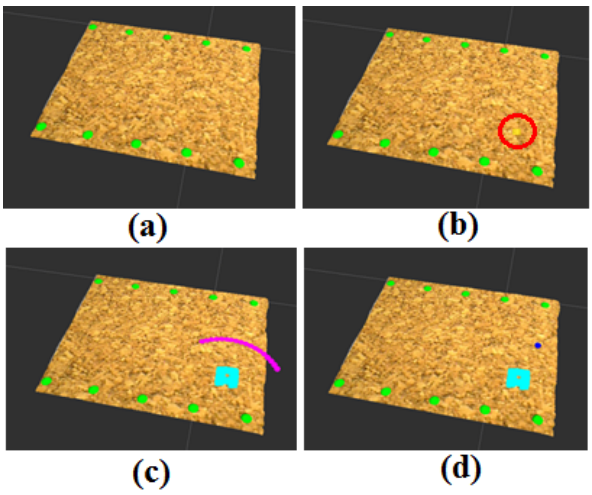

Fig. 6: Rviz views of the supervisory-control interface. Scene (a): point-cloud display of target tissue. Scene (b): selection of a desired needle entry point. Scene (c): display of possible exit points. Scene (d): selection of a valid exit point.

surrounding red circle). Point selection is performed in Rviz using the "Publish Selected Points" tool [18]. Selection of a patch of points invokes callback code that computes the centroid of the points (with outlier removal) and computes the surface normal of these points. The centroid of the selected-points region is published on a ROS topic as the desired entry point of the needle for a suturing task.

An independent planner node subscribes to publications of (centroids of) such selected points. When this planner node receives a selected-point publication, it interprets this information as the coordinates of a desired entry point for a needle-drive operation. A needle-drive trajectory is comprised of a sequence of desired time-stamped gripper poses. In our approach, the sequence of gripper poses is expressed in camera space. Each pose is a 6-DOF spatial specification, and a sequence of N poses thus specifies 6N values. Exploding 3 inputs (the entry-point 3-D coordinates) into 6N values requires imposing multiple assumptions on the task. One of the assumptions we impose is that the needle-drive path should be circular, as this minimizes trauma to the tissue [5]. Preferably, the circular-arc path swept by the gripper would lie in a plane that is perpendicular to the tissue surface. This is not a hard requirement, though, and exploration of candidate paths lying in tilted planes can be useful when joint-limit constraints are demanding.

A constraint on the circular-arc path is the needle-to-gripper transform–i.e., how the gripper is holding the needle. Conventionally, the needle would be held close to the suture end (to avoid interference between gripper and tissue during needle driving), and the needle would extend sideways from the gripper, e.g. as shown in Fig 4. In our current instantiation, we restrict consideration to needle-drive profiles with needle grasp as shown in Fig 4 (although consideration of alternative grasp transforms is a logical extension).

We further assume that the plane of the needle is perpendicular to the tissue surface. For circular needle-drive paths, a convenient reference frame for the needle consists of the center of the circular needle arc, and a vector normal to the needle plane (the "needle axis" for reference). The needle motion during circular driving maintains a constant needle center and a constant needle axis. With the needle plane perpendicular to the tissue surface, the needle axis is parallel to the tissue surface.

Our archetypical needle-drive strategy, then, can be implied succinctly in terms of four parameters: the 3-D coordinates of the proposed needle center and the "yaw" angle of the needle axis (a rotation about the tissue normal). Of these parameters, distance of the needle center from the tissue surface–i.e., the "needle height"– is chosen strategically. If this distance is greater than or equal to the needle radius, then the circular drive motion will not result in any needle penetration. At the other extreme, if the needle height is zero, then a circular needle-drive path would result in the needle fully embedded in the tissue–with no room for the driving gripper to hold the needle (without tissue interference), and no needle-tip emergence from the tissue (thus precluding grasp of the needle tip with the second gripper for completing the needle drive through the tissue). A needle height of approximately 50-70% of the needle radius is pragmatic.

Having prescribed parameters of a driving strategy (e.g., circular arc with needle axis parallel to tissue at chosen needle height), one can complete the specification of a needle-drive trajectory by specifying an entry point on the tissue and an exit point. In fact, selection of an exit point would over-define the problem, since this point must be consistent with the needle radius and the driving strategy. To address this constraint, we provide graphical assistance to the operator.

Scene (c) of Fig 6 shows an Rviz scene after the operator has selected a proposed entry point. Having received the selected entry point, the planner node computes viable exit points, which it publishes for use by the HMI. The HMI node subscribes to the exit-points topic and displays markers at the computed coordinates of the candidate exit points. In Scene (c) of Fig 6, the viable exit points are shown as magenta markers.

Restricting exit-point coordinates to geometrically-consistent candidates is actually a minor part of what the needle-planner node computes. More importantly (and more computationally intensive), the needle-planner node considers complete needle-drive paths based on hypothetical exit points. This node computes samples of candidate exit points, consistent with the needle radius and drive strategy, and uses this information to complete the specification of a needle-drive trajectory. The planner node then considers samples of gripper poses along the consistent needle-drive trajectory and evaluates the inverse-kinematic viability of these gripper poses. Ideally, a dense sampling of gripper poses along the circular drive path would have valid inverse kinematics over an arc of nearly 180 degrees, which would result in driving the needle fully through the tissue with the needle tip emerging through the exit point with enough

length to be grasped by the second gripper. This IK path computation is performed over many iterations to consider details of multiple trajectories corresponding to multiple candidate exit points. If a valid trajectory exists for a candidate exit point, then this exit point is published by the planner node and displayed by the HMI in Rviz.

As a result of the needle-planner computations, the operator can immediately see a visualization of all viable alternatives for needle exit points (subject to the various constraints, including joint limits, needle-grasp transform and choice of needle axis and needle height). This computation and display occurs in negligible time from the user's perspective. The operator may then choose one of the suggested exit points with an assurance that the corresponding circular needle drive is kinematically possible.

Scene (d) of Fig 6 shows the result of the operator's second input. Using a second tool option in Rviz, "Publish Point", the operator clicks on a point near one of the illustrated, viable exit points. The HMI then erases the exit-point options and illustrates the chosen exit point with a blue marker. The selected exit point is published to a separate (exit-point) topic, and this action induces replanning for the specific needle-drive task implied by the user's graphical selections.

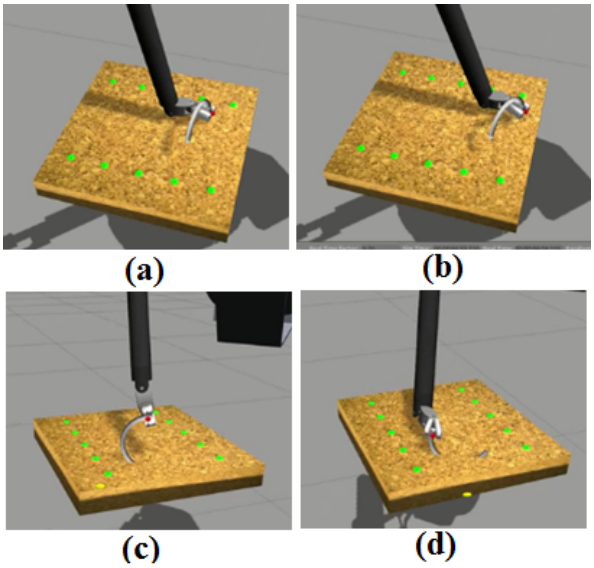

Fig. 7: Gazebo views of needle-drive operation. The resulting path is a precise circular arc for which the needle enters and exits at user-chosen points.

V. An Automated Needle-Driving System

The next step in our system is to invoke execution of a needle-drive trajectory based on the operator's input. As described above, the operator would view the scene in Rviz and would select a desired entry point for the needle. With this information, the system displays an array of viable exit points. If at least one of these points is suitable, in the operator's opinion, a desirable exit point may be selected. The selection of an exit point automatically launches needle-drive execution of the computed trajectory. Illustration of this process is shown in Fig 7 (as well as in this paper's accompanying video). The scenes in Fig 7 correspond to the entry and exit points selected in the user-interface illustration of Fig 6.

Execution of the computed trajectory achieves a precise circular-arc drive for which the needle enters the tissue at the prescribed entry point and exits the tissue at the prescribed exit point. This behavior illustrates our vision for supervisory control of robotic surgery. Ideally, execution would be faster, more precise and more reliable than humanly possible.

VI. Conclusions and Future Work

The work presented here is an initial step towards supervisory control of robotic surgery. Computation of viable paths is made possible by our fast and reliable IK solver. Natural extensions of this initial work in supervisory control of needle driving would consider a larger space of options, including (as necessary): alternative needle-grasp transforms (including automation of achieving optimal grasps); alternative needle-drive parameters (e.g. tilted planes and alternative needle heights); and plans that involve needle re-grasp (if a single 180-deg drive is kinematically infeasible). Such requirements, constraints, options and strategies are not immediately obvious to the operator, since they involve joint-limit constraints that are not easily visualized by the operator. Automation of trajectory planning and execution thus offers a potentially significant productivity and quality improvement.

We are currently extending automation of needle driving to needle hand-off (to complete the needle drive) and suture knot tying (which we formerly explored with alternative robots and grippers [19]).

Supervisory control of needle driving has natural extensions to additional operations. Cauterizing, cutting, and stapling are example operations that could benefit from custom (task-specific) user interfaces and on-line planners that exploit 3-D point-cloud visualization and automated motion planning and execution.

Use of automated trajectory planning may be even more critical in operations where entry ports and free space are highly constrained, including operations within pelvic boundaries.

Our fast IK algorithm may be exploited further by considering portal placement and base-link repositioning. It is not immediately apparent to the operator how the arms should be pre-positioned. Further, once portal locations are established and the task site is observed endoscopically, the specific manipulation needs may be unachievable with the chosen robot-arm approach. Our fast planner may be useful in computing alternative robot-approach poses that are kinematically consistent with the portal constraints and with the desired manipulation operations.

The work presented here, developed in simulation, is migrating to a research interface on a physical DaVinci robot. By exploiting ROS, we anticipate that it should be possible to port this work to the physical robot quickly. Experience with the physical system will, in turn, feed back to our modeling,

thus making off-line software development more realistic and applicable.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grant CISE IIS-0905344 and IIS-1524363.

REFERENCES

[1] Adrian Park, Gyusung Lee, F Jacob Seagull, Nora Meenaghan and David Dexter, "Patients Benefit while Surgeons Suffer: An Impending Epidemic", Journal of the American College of Surgeons, Vol. 210, Issue 3, March 2010, pp 306-313.

[2] F. Nageotte, P. Zanne, M. de Mathelin and C. Doignon, "A Circular Needle Path Planning Method for Suturing in Laparoscopic Surgery", Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 514-519.

[3] F. Nageotte, P. Zanne, C.Doignon and M. de Mathelin, "Stitching planning in laparoscopic surgery: Towards robot-assisted suturing",The International Journal of Robotics Research. Vol 28, Issue 10, pp. 1303 - 1321, 2009.

[4] J. Ding and N. Simaan, Choice of handedness and automated suturing for anthropomorphic dual-arm surgical robots, Robotica, vol. 33, no. 8, pp. 17751793, 2015.

[5] R. Jackson and M. C. Cavusoglu, "Needle Path Planning for Autonomous Robotic Surgical Suturing", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2013), May, 2013.

[6] Morgan Quigley,Ken Conley,Brian P. Gerkey,Josh Faust,Tully Foote, Jeremy Leibs, Rob Wheeler and Andrew Y. Ng, "ROS: an open-source Robot Operating System", ICRA Workshop on Open Source Software, 2009.

[7] Z. Chen, A. Deguet, R. Taylor, S. DiMaio, G. Fischer and P. Kazanzides, "An Open-Source Hardware and Software Platform for Telesurgical Robotics Research", The Insight Journal - 2006 MICCAI Open Science Workshop, http://hdl.handle.net/10380/3419, August, 2013.

[8] Hayashibe, M., Suzuki, N., Hashizume, M., Kakeji, Y., Konishi, K., Suzuki, S. and Hattori, A., "Preoperative planning system for surgical robotics setup with kinematics and haptics", Int. J. Med. Robotics Comput. Assist. Surg., 1: 7685. doi: 10.1002/rcs.18

[9] Hyeong Ryeol Kam, H Lee, Sung-Ho, Taejung Park, and Chang-Hun Kim, "RViz: a toolkit for real domain data visualization", J. Telecommunication Systems, v60,no2,pp337-345, 2015.

[10] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator", Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, pp 2149-2154.

[11] G. S. Guthart and J. K. Salisbury, "The IntuitiveTM telesurgery system: overview and application", Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, pp 618-621.

[12] D. L. Pieper, "The Kinematics of Manipulators Under Computer Control", Ph.D. Dissertation, Stanford U., 1968.

[13] C. A. Klein and C. H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators" IEEE Transactions on Systems, Man, and Cybernetics, 1983, vol SMC-13, no 2, pp 245-250.

[14] A. D'Souza, S. Vijayakumar and S. Schaal, "Learning inverse kinematics", Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on,

[15] Deepak Tolani, Ambarish Goswami and Norman I. Badler, "Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs", Graphical Models, 2000, vol 62, pp 353-388.

[16] Dominik Bertram, James Kuffner, Rüdiger Dillmann and Tamim Asfour, "An Integrated Approach to Inverse Kinematics and Path Planning for Redundant Manipulators", Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, May 15-19, 2006

[17] Jacques Denavit and Richard S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices", Trans ASME J. Appl. Mech 23: 215221, 1955.

[18] `https://github.com/tu-rbo/turbo-ros-pkg/tree/master/selected_points_publisher`.

[19] Der-Lin Chow, Wyatt Newman, "Improved Knot-Tying Methods for Autonomous Robot Surgery", Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, Wisconsin, USA. August, 2013, pp.461-465.