

Surgical Automation for Multilateral Multi-Throw Suturing

Siddarth Sen

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-166

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-166.html>

November 23, 2016



Copyright © 2016, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Surgical Automation for Multilateral Multi-Throw Suturing

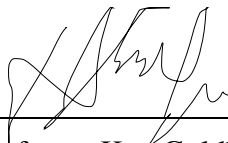
by Siddarth Sen

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:



Professor Ken Goldberg
Research Advisor

11/16/16

(Date)



Professor Pieter Abbeel
Second Reader

11/16/16

Surgical Automation for Multilateral Multi-Throw Suturing

by

Siddarth Sen

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair

Professor Pieter Abbeel

Fall 2016

Abstract

Surgical Automation for Multilateral Multi-Throw Suturing

by

Siddarth Sen

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

For supervised automation of multi-throw suturing in Robot-Assisted Minimally Invasive Surgery, we present a novel mechanical needle guide and a framework for optimizing needle size, trajectory, and control parameters using sequential convex programming. The Suture Needle Angular Positioner (SNAP) results in a $3\times$ error reduction in the needle pose estimate in comparison with the standard actuator. We evaluate the algorithm and SNAP on a da Vinci Research Kit using tissue phantoms and compare completion time with that of humans from the JIGSAWS dataset [6]. Initial results suggest that the dVRK can perform suturing at 30% of human speed while completing 86% suture throws attempted.

Contents

Contents	i
1 Introduction	1
2 Related Work	4
2.1 Automated Suturing	4
2.2 Suture Needle Path Planning	5
3 Problem: Formulation and Definitions	6
3.1 Assumptions	6
3.2 Input	7
3.3 Output	7
3.4 Curvature Constrained Kinematic Model	8
4 Suture Needle Path Planning	9
4.1 Optimization Model	9
4.2 Trajectory Optimization	11
5 Reducing Needle Pose Uncertainty	14
5.1 Suture Needle Angular Positioner (SNAP)	14
5.2 Real Time Needle Tracking	15
6 System Design	17
6.1 Multi-Throw Suturing: System Design	17
7 Experiments	19
7.1 dVRK: Hardware and Software	19
7.2 Experimental Evaluation of Needle Tracking	19
7.3 Evaluation of Suture Needle Angular Positioner (SNAP)	20
7.4 Robot Experiments: Four-Throw Suturing Task	21
8 Conclusion	24
8.1 Discussion and Future Work	24

8.2 Challenges and Open Questions in Surgical Automation	25
9 Appendix	28
Bibliography	30

Acknowledgments

This work represents just a fraction of my fantastic experience in robotics at Berkeley, and none of it would have been possible without the support I've received in the past four years. First I would like to thank my research advisor, Professor Ken Goldberg, for having confidence in me when I first joined the Automation Sciences lab as an undergraduate and for exposing me to the world of robotics research. I also want to thank Professor Goldberg for teaching lessons that will stick with me beyond academia. He has taught me the importance of having high standards and the ability to effectively articulate an idea. I also thank Professor Goldberg for being a fantastic role model in showing me that it is possible to be well balanced and also be good at what you do. I want to thank Professor Pieter Abbeel for his teaching and advice. I thank him for welcoming me to lab meetings. Googling his lecture slides on robotics continues to be an invaluable resource to me. I thank postdoctoral researchers Sachin Patil and Lauren Miller for being patient mentors. I thank Jeff Mahler and Michael Laskey for being my first mentors and friends in research. Their patience and enthusiasm made me excited to come to lab. I thank both Ben Kehoe and Sanjay Krishnan for having thoughtful advice on nearly everything from how to write papers to how to select a research problem. I'd like to thank Steve McKinley for his prompt and constant resourcefulness. I thank him for his help in designing our experimental setups, tissue phantoms and instruments that enabled much of this work. I also thank him for the quick solutions that bailed me out every time I broke something minutes before an important demo or deadline. I'd like to thank Animesh Garg for his constant guidance during this entire process. I thank him for guiding me to the right books and the right papers in robotics and optimization. I thank him for providing help while we ran experiments late into the night. I thank him for all our discussions. I thank him for painstakingly reviewing every single one of my poorly written sentences while writing a paper together. This work would also not have been possible without the help of my peers: David Gealy, Yiming Jen, and Adithya Murali. Their help in designing and running experiments, coming up with novel solutions, and hard work made this possible. I also thank Adithya Murali for being there from the start when our robot first arrived in boxes and for being my peer every step of the way. I want to thank my parents for their love and support. I thank them for always encouraging my curiosity and for allowing me to think independently. I thank them for enabling me to pursue my education without having to worry about anything else. Finally I want to thank Linda Shih for her patience, generosity, and love. I thank her for being there through all the ups and downs and for being supportive of everything I've wanted to do.

Chapter 1

Introduction

Robotic Surgical Assistants (RSA), such as Intuitive Surgical’s da Vinci System have facilitated over 570,000 procedures worldwide in 2014 [9]. RSAs are currently controlled by surgeons using pure tele-operation, requiring uninterrupted attention and control. Automation of surgical sub-tasks such as suturing has the potential to reduce surgeon tedium and fatigue, operating time, and enable supervised tele-surgery.

We have explored a variety of different modes of surgical automation including palpation, tumor resection, and autonomous suturing. The results of the first two works are summarized below, with autonomous suturing being the main focus of this work.

One of the disadvantages of robotic surgery today is the lack of haptic feedback in robotic tele-operation. In traditional surgery, surgeons can palpate tissue in order to localize subcutaneous inclusions. In prior work [22] we developed a low-cost, single use, disposable haptic probe which mounts onto the tip of a standard da Vinci needle driver that allows the measurement of tissue stiffness through sliding motions across the tissue surface. The haptic probe contains a displacement-based contact sensing tip which allows for quasi-static motion similar to surgeon fingertip motion. Our results show that the probe is capable of measuring inclusions inside a silicone tissue phantom at depths of up to 5mm. We extend this work in [7] and explore efficient methods for continuous tissue palpation using the haptic probe. We developed a Gaussian Adaptive Sampling algorithm, Implicit Level Set Upper Confidence Bound, that prioritizes sampling tissue stiffness near a level set of an inclusion boundary estimate. Physical experiments on flat silicon phantoms with hidden inclusions suggest that our method outperforms raster scanning techniques while using at least 10x fewer measurements.

We have also explored how a series of interchangeable tool tip mounted instruments can complement the palpation probe in tasks requiring long term autonomy such as tumor resection [23]. Tumor resection is a multi-step multilateral surgical procedure to localize, expose, and debride a subcutaneous tumor. We demonstrated the task on a silicon phantom using the da Vinci Research Kit and a series of tool-tip mounted instruments. The palpation probe described in the previous paragraph is used to localize vein shaped inclusions underneath the phantom surface. A tool-tip mounted scalpel is used to make an incision in the tissue.

Finally a standard needle driver is used to debride the vein-shaped inclusion embedded in the silicone phantom. Our autonomous system successfully completed the entire procedure in five of ten trials. The most common failure mode was the final debridement phase, and we showed that this phase can be made more robust using visual feedback.

Now we will introduce the main surgical subtask of interest in this work: multi-throw suturing. The Fundamental Skills of Robotic Surgery (FSRS) defines a representative set of procedures for surgical training and evaluation [38]. FSRS includes Multi-Throw Suturing (MTS) where each MTS throw includes five steps as illustrated in Figure 1.1. A curved needle with suture thread is repeatedly pushed through a pair of tissue boundaries with one actuator, then pulled through with a second actuator until the thread is taut, then is transferred back to the first actuator to begin the next throw/suture [8, 3]. In Robot-Assisted Minimally Invasive Surgery (RMIS), MTS is a tedious subtask and it can be difficult for the surgeon to maintain proper needle pose during insertion and transfer as haptic feedback is not available.

We present initial results toward automating MTS with new hardware and a novel optimization algorithm. Our approach includes (1) a mechanical device, the Suture Needle Angular Positioner (SNAP), designed to align and hold the needle in a known orientation, (2) computer vision software to track needle pose, and (3) a sequential convex optimization formulation of needle motion planning. Initial results suggest that SNAP can reduce error in needle orientation by $3\times$ and that the combined system can successfully complete 86% of attempted suture throws at 30% the speed of human operators [6].

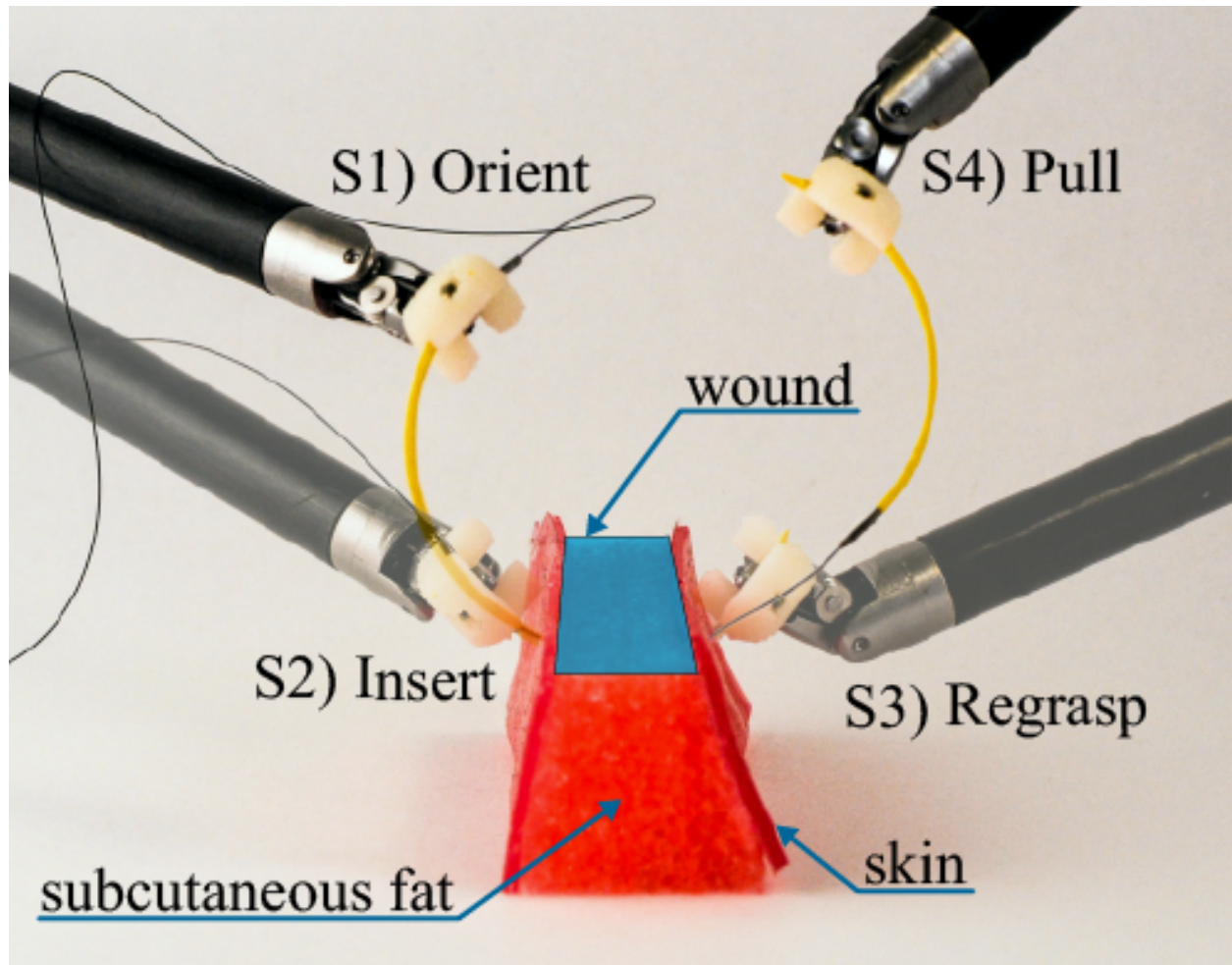


Figure 1.1: Each throw in Multi-throw Suturing (MTS) includes five steps: (S1) Needle placement in desired position and orientation by first actuator, (S2) Needle insertion through tissue by first actuator, (S3) Needle grasp by second actuator, (S4) Needle and thread pull until thread is taut, and (S5) Needle transfer back to first actuator. (note: S5 is not illustrated in this time-lapse image).

Chapter 2

Related Work

2.1 Automated Suturing

Automation of suturing has been studied in the context of hierarchical models for multi-step task planning [13], multilateral manipulation of needle and suture [37], and interaction with deformable tissue [10, 11].

While each of these studies made significant contributions as outlined below, challenges in combining these steps to achieve autonomy in longer tasks has not been sufficiently addressed. Kang et al. devised a specialized stitching device for RMIS which is capable of tying a knot [13]. Mayer et al. used a recurrent neural net as part of a controller to learn knot tying with three industrial arms using motion primitives from human demonstrations [21]. van den Berg et al. used iterative learning for performing knot tying at super-human speeds [2]. More recently, Schulman et al. used a learning by demonstration approach to warp recorded expert demonstrations and perform suturing in simulation [33]. Padoy et al. showed execution of collaborative human-robot suturing, but the key sections requiring interactions such as needle insertion and hand-off were performed manually [29]. Similarly, Staub et al. automated needle insertion into tissue for single-throw suturing [37].

Prior work in surgical automation has modeled the basis set of surgical motions as the “Language of Surgery” composed of surgemes (Hager et al.) [32]. Recent works have also explored the use of learning techniques to infer surgeme transitions from demonstration data [28, 19]. Many of the FSRs procedures, including MTS, are decomposable into short sequences of simpler sub-tasks. This decomposition allows the parametrization and building of Finite State Machines (FSM) for complex procedures using a learning by observation approach, for tasks such as tissue debridement [15], pattern cutting [24], and tumor localization & resection [23]. Our work on segmentation of multi-step task demonstrations [17] suggests that unsupervised learning of semantic transitions is feasible and can be analyzed to construct FSMs for these multi-step tasks.

2.2 Suture Needle Path Planning

Some preceding studies use a needle path of fixed curvature. Jackson et al. used a reference trajectory to create an analytical solution allowing for needle insertion without considering uncertainty or robot pose constraints [11]. However, needles do not always follow their natural curvature. Interaction with tissue may deflect the needle, and end point pose constraints necessitate non-orthogonal exit. The use of optimization-based planning has potential to address these limitations. Recent results in motion planning have shown that Sequential Convex Programming (SCP) based planning, such as [35] can be both faster and more successful in finding solutions than sampling based planners. This paper formulates suture needle path planning as a curvature constrained SCP-based optimization problem solved with a custom implementation.

This paper builds on prior work in optimization-based planning [5, 30], sub-task level segmentation of demonstrations [17, 19], gripper mounted interchangeable tools [23], and building robust finite state machines [24]. To the best of our knowledge, this is the first system to perform autonomous multi-throw suturing.

Chapter 3

Problem: Formulation and Definitions

The success of suturing is highly sensitive to needle pose uncertainty at entry point. Uncertainty in needle pose during insertion can result in tissue injury due to skin penetration at undesirable angles or the lack of sufficiently deep needle insertion to hold the suture securely. As illustrated by the several error cases in Figure 3.1, it is essential to maintain proper needle pose during insertion and handover to avoid dropping the needle or damaging tissue. Since the needle is thin and highly reflective, it is difficult to accurately detect its position and orientation with computer vision as noted in [11, 27, 36, 12]. Several medical device manufacturers offer needle-alignment devices for manual laparoscopic applications [20, 31] but, to the best of our knowledge, these are not available for RSAs.

Surgeons follow suturing task guidelines such as entering the tissue orthogonally, minimizing tissue-needle wrench, choosing the correct needle size for adequate suture depth, and inserting the needle to a sufficient depth to ensure needle protrusion for needle re-grasp. While a needle would follow a constant curvature path through rigid objects, tissue is deformable. Thus we model the needle path to allow bounded rotations about the needle tip while the needle is inserted. However, needle paths that do not follow the natural curvature of the needle can result in tissue damage, hence we define a bounded deviation (γ) from needle curvature (κ) that can be visualized as a cone at each point as illustrated in Figure 4.1. We monotonically reduce γ as the needle progresses to minimize tissue damage.

3.1 Assumptions

We assume that tissue is homogeneous and deformable. Real-time tracking and planning is used to account for departures from needle pose estimates during needle insertion. We assume that the needle is rigidly held in the gripper and can only move forward in the tangential direction of the tip. However, bounded reorientation of the needle tip is permitted as it is inserted through tissue. We assume that our system has access to a continuous range of needle sizes. In practice, needles vary in length in increments of 1 mm and vary in three different fractions of a circle.

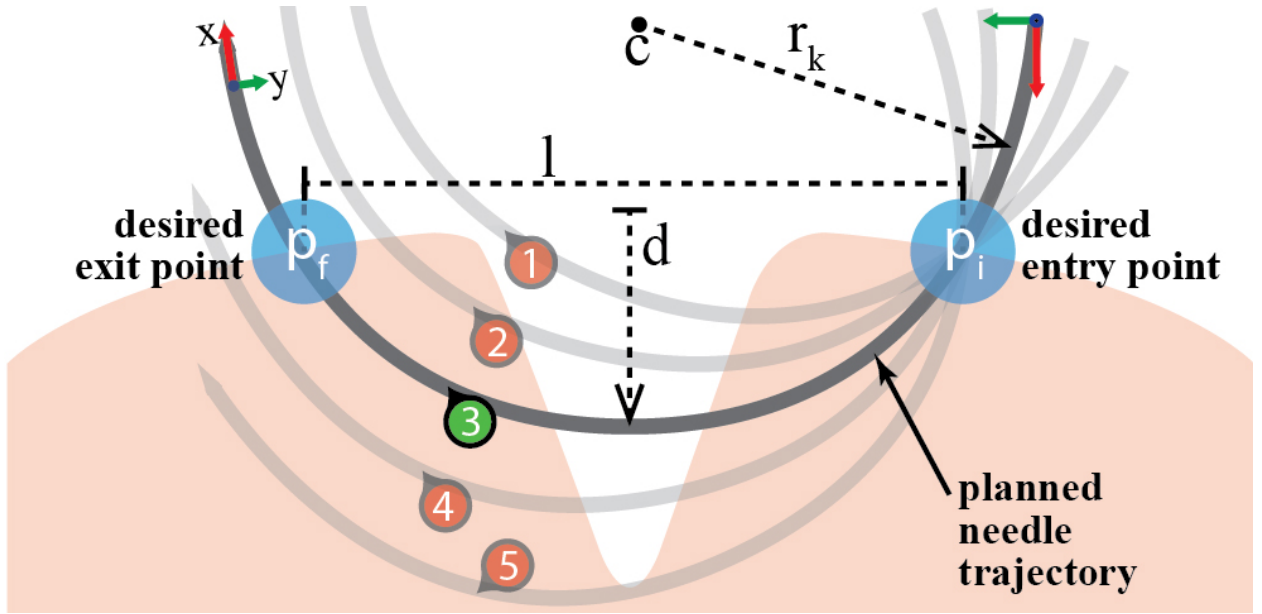


Figure 3.1: The needle trajectory labeled (3) shows the desired trajectory along with poses at the entry and exit points of the tissue. The success of suturing depends on correct orientation of the needle with respect to the tissue. For example, uncertainty in needle pose at the entry point may result in the needle not connecting opposite tissue sides (1), not making sufficiently deep insertion to hold the suture securely (2), not having enough length of needle at the other end to enable re-grasping (4), or passing completely under the wound and not exiting the tissue at all (5).

3.2 Input

The wound shape is provided as input, with the points $\mathcal{M} = [M_1, M_2, \dots, M_D] \in \mathbb{R}^3$ representing the wound surface as a spline. The system is also provided with suture depth d , suture width l , and a pair of entry/exit poses $(P_i, P_f \in SE(3))$ for the first throw as illustrated in Figure 3.1. Further, we are also given suture pitch w – the distance between consecutive suture throws.

3.3 Output

The system needs to find a set of suture throws \mathcal{S} , where $\forall S_j \in \mathcal{S}$, we need to calculate an optimized sequence of needle tip poses $\mathcal{X}_j \in SE(3)$ satisfying the the suture depth and suture width constraints or report that no such path plan exists. The system also needs to choose a needle curvature and length. The entry and exit positions at each suture throw S_j are obtained by linearly interpolating P_i, P_f along the spline while keeping the orientation constant.

3.4 Curvature Constrained Kinematic Model

The needle trajectory is discretized into time intervals $\mathcal{T} = \{0, 1, \dots, T\}$, where the needle moves a fixed length (Δ) at each time step. At each time step the needle's pose is parametrized as $X_t \in SE(3)$.

We model the needle trajectory as a sequence of $T - 1$ circular arcs with curvature κ_t between every consecutive pair of needle poses (X_t, X_{t+1}). We model our control of the needle at each time step as a rotation and insertion where at each time step the pose X_t is propagated a distance Δ to X_{t+1} . Although a needle naturally follows a path of constant curvature, the needle tip can be reoriented at each time step to change the local curvature by $\bar{\gamma}_t$. Thus at each time step the path curvature κ_t can be expressed as $\kappa_t = \kappa + \bar{\gamma}_t$ where κ is the curvature of the needle and $\bar{\gamma}_t$ is the change in curvature applied at each time step. The transformation between consecutive needle poses can be represented a twist in $\mathfrak{se}(3)$, $u_t = [\Delta \ 0 \ 0 \ 0 \ \Delta\kappa_t \ 0]^T$

The Lie group $SE(3)$ and the corresponding algebra $\mathfrak{se}(3)$ are related by the exponential and log maps $\exp : \mathfrak{se}(3) \rightarrow SE(3)$ and $\log : SE(3) \rightarrow \mathfrak{se}(3)$. Closed form expressions exist to compute these maps efficiently. Given an incremental twist $x = [p_x \ p_y \ p_z \ r_x \ r_y \ r_z]^T \in \mathbb{R}^6$, the corresponding Lie algebra element is given by the mapping $^\wedge : \mathbb{R}^6 \rightarrow \mathfrak{se}(3)$ as

$$x^\wedge = \begin{bmatrix} 0 & -r_x & r_y & p_x \\ r_z & 0 & -r_x & p_y \\ -r_y & r_x & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The reverse mapping $^\vee : \mathfrak{se}(3) \rightarrow \mathbb{R}^6$ can be used to recover the twist, x from an element of $\mathfrak{se}(3)$. Poses between consecutive time steps can then be related as:

$$X_{t+1} = \exp(u_t^\wedge) \cdot X_t \quad (3.1)$$

Further details about the Lie algebra can be found in the appendix.

Chapter 4

Suture Needle Path Planning

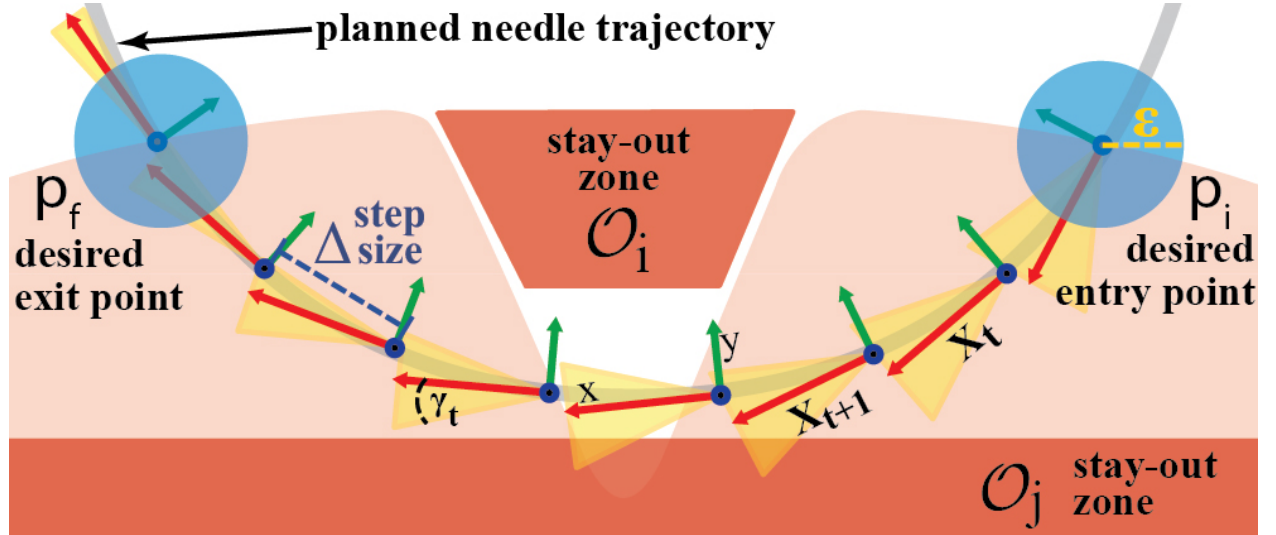


Figure 4.1: The optimization steps and non-holonomic motion at each time-step. The figure shows stay-out zones \mathcal{O}_i , trajectory poses X_t , step-size Δ , needle radius r , and γ -cone of allowed rotation at each X_t .

The Suture needle Path Planning (SPP) problem can be formulated as a non-convex, curvature constrained motion planning problem solved with a series of locally convex approximations using sequential convex programming (SCP). We begin by presenting the problem formulation.

4.1 Optimization Model

The following optimization model is based upon the curvature constrained motion planning formulation described by Duan et al. [5]. We extend this formulation to include parameters

to simultaneously optimize needle dimensions.

For notational convenience we concatenate the states from all time steps as $\mathcal{X} = \{X_t : t \in \mathcal{T}\}$ and control variables as $\mathcal{U} = \{\kappa, \Delta, \gamma_t : t \in \mathcal{T}\}$

$$SPP : \underset{\mathcal{X}, \mathcal{U}}{\text{minimize}} \quad \alpha_{\Delta} C_{\Delta} + \alpha_I C_I \quad (4.1)$$

$$\text{s.t.} \quad \log(X_{t+1} \cdot (\exp(u_t) \cdot X_t)^{-1})^{\vee} = 0_6 \quad (4.2)$$

$$|\bar{\gamma}_t| \leq \gamma_t \quad \forall t \quad (4.3)$$

$$T\Delta + 2l_g - \frac{2\pi l_n}{\kappa} \leq 0 \quad (4.4)$$

$$\text{sd}(X_t, \mathcal{O}_i) \geq d_s, \quad \forall i \quad (4.5)$$

$$X_0 \in \mathcal{B}(p_i, \epsilon), \quad X_T \in \mathcal{B}(p_f, \epsilon) \quad (4.6)$$

Each term in the above formulation is described below:

Costs (Eqn. 2):

We assume the volume of needle in tissue is proportional to tissue trauma and hence we penalize longer trajectories such that $C_{\Delta} = T\Delta$, the length of the trajectory. To penalize tissue damage we optimize for the shortest length needle trajectory that satisfies our constraints. We approximate the trajectory length as:

$$C_{\Delta} = T\Delta \quad (4.7)$$

Furthermore, surgical guidelines suggest that the needle entry pose should be orthogonal to the tissue surface. C_I penalizes deviations from an orthogonal start pose. The weights α_{Δ} and α_I are parameters that are tuned in the optimization.

Kinematic Constraints (Eqns. 3, 4):

The kinematic constraint in Eqn. 1 can be transformed using the exponential log map into the standard equality constraint in Eqn. 3. Eqn. 4 bounds the magnitude of $\bar{\gamma}_t$ to minimize tissue damage. We select γ_t to be monotonically decreasing with t because needle rotations away from its natural curvature cause greater damage the further the needle is inserted into tissue.

Needle Length Constraints (Eqn. 5):

The length of the insertion trajectory ($T\Delta$) is constrained to be less than the length of the needle ($2\pi l_n/\kappa$) and should allow for grippers to hold the needle on both ends ($2l_g$).

Collision Constraints (Eqn. 6):

We impose constraints to ensure that our trajectory avoids collisions with pre-defined stay out zones. We ensure that the signed distance between each X_{t+1} and each convex mesh in \mathcal{O} is greater than a safety margin parameter d_s . The stay out zones can be non-convex meshes that can be decomposed into convex sub meshes [4], $\mathcal{O} = \{\mathcal{O}_1, \dots, \mathcal{O}_i\}$. This is a mesh that can be constructed using wound shape (width, depth) provided by a surgeon. Our technique also allows this mesh to be general and non-convex. This would allow mesh construction through sensory feedback from endoscopic cameras or ultrasound. Convex decomposition can then be used to decompose a given point cloud of an incision into a set of convex stay out zone meshes.

Entry and Exit Point Constraints (Eqn. (7)):

We constrain the start and end poses of the trajectory to be within an ϵ -Ball of the calculated entry (p_i) and exit (p_f) poses. This can be expressed as $\log(p_i \cdot X_0^{-1})^\vee \leq \epsilon \cdot \mathbf{1}_6$ for the start pose of the trajectory. The end pose constraint follows a symmetric formulation.

We note that a constant of Δ is chosen for all time steps instead of having a different Δ_t for each time as the latter is experimentally found to disagree numerically with the findings of Duan et al. [5].

4.2 Trajectory Optimization

Sequential convex programming is a general approach for solving constrained, non-convex optimization problems. We refer the reader to [34] for the details of SCP-based motion planning. The optimization problem outlined in Eqn. (2) is, however, described directly over the set of poses X . Optimizing directly over these poses can lead to poor results due to the large number of free variables in the rotation matrix of each pose. We generalize sequential convex optimization to the case where the domain is the differentiable manifold, the $SE(3)$ Lie group, rather than \mathbb{R}^n by considering a local coordinate parametrization of the manifold. This parametrization is given by the Lie algebra $se(3)$, which is defined as the tangent vector space at the identity of $SE(3)$ [5]. Each iteration of the SCP algorithm takes gradient steps in this local tangent space allowing for better solution convergence.

Figure 4.2 shows the SPP output for three different sets of pose constraints. For #1, we restrict rotation about needle tip ($\gamma_t = 0, \forall t$). Coupled with the orthogonality constraint at entry/exit, this results in a constant curvature path along the needle radius. For #2, orthogonality is enforced only at entry pose, and γ_t is set to a monotonically decreasing sequence in t . This results in rotations about the needle tip that achieve an asymmetric trajectory satisfying pose constraints at entry. We also demonstrate a case with no pose constraints in #3, resulting in the shortest path trajectory, but with oblique entry angles.

Since minimizing trauma is one of the main factors in suturing, we consider two criteria: Trajectory Length (TL) and Swept Needle Volume (SNV). We assume that SNV is linearly

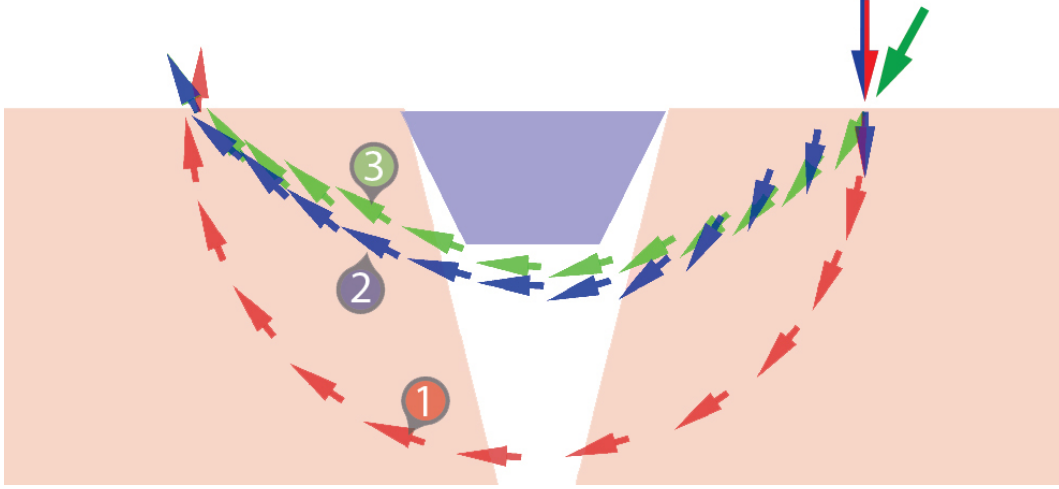


Figure 4.2: The side view of three needle trajectories generated by SPP. Trajectory 1 and 3 are constant curvature trajectories whereas trajectory 2 is a variable curvature trajectory.

proportional to volume swept by needle due to rotations at the tip. We define TL: $\mathcal{C} = T\Delta$ in eq(2), and SNV: $\sum_t \gamma_t t \Delta$. Comparing the lengths normalized to the longest trajectory (#1), we find that trajectories #2 and #3 have 0.72 and 0.70 lengths. We also note that #1 and #3 have an SNV of 0, while #2 has a SNV of 0.242.

We evaluated our Suture Path Planning algorithm with three different sets of constraints. The generated trajectories are rendered in the figure above. Trajectory 1 was generated using a start pose constraint and γ_t values set to zero. This prevents trajectory deviations from the natural curve of the needle and forces the trajectory to follow a constant curvature path that is deeper than Traj. 2 and 3. If we relax the pose constraint to just a position constraint, the SPP algorithm outputs Traj 3. The trajectory still follows a constant curvature path but is shorter and less deep than Traj. 1.

Trajectory 2 is generated by keeping the start pose constraint but relaxing the values of γ_t such that it is monotonically decreasing. Intuitively, this allows the needle's trajectory to deviate from its natural curvature as it is first entering the tissue. This added flexibility allows Trajectory 2 to satisfy the start pose constraint in Traj 1 while achieving a path length nearly as short as that of Traj 3.

Computing Gradients

For some of the functions in our optimization problem, it is difficult to compute explicit gradients. One such example, is the needle sweep cost as it passes through tissue. Instead we use finite differencing to compute an estimate of the gradient in each step of the gradient descent loop. We approximate the gradient using the centered finite differencing method where h is a small number:

$$\nabla f(x) \approx \frac{f(x + \frac{1}{2}) - f(x - \frac{1}{2})}{h}$$

We use the centered difference due to its higher accuracy in comparison to the forward or backward differences.

Chapter 5

Reducing Needle Pose Uncertainty

As stated in Section 3 and Figure 3.1, tissue damage is minimized with orthogonal needle entry and motions that are tangential to the needle tip. These guidelines require accurate needle pose estimates at the needle entry point and robust needle grasps.

5.1 Suture Needle Angular Positioner (SNAP)

Commercially available RMIS needle drivers allow handling of a variety of needle sizes. However an analysis of suturing trials in JIGSAWS dataset [6] reveals that multiple pairs of hand-offs are required for correct needle orientation. This is because the motion of a needle held within the needle driver jaws is not fully constrained. The flat gripper surface allows rotation and translation along the length of the needle, which can be hard to control without haptic or visual feedback.

There have been some commercial efforts to mitigate back-and-forth hand-offs and uncertainty in laparoscopic surgery through passively orienting the needle on gripper closure using a “self-righting” gripper jaw design [20, 31]. However, these are not designed for automation, and require a complete tool redesign.

We develop a design for a low-cost Suture Needle Angular Positioner (SNAP) for dVRK Classic 8 mm Needle Drivers with 6 mm jaws, which works to guide and passively orient a curved needle into a stable pose upon closure of gripper jaws as illustrated in Figure 5.1(d). SNAP reduces needle pose uncertainty along two rotational axes as shown in Section 7. This allows for a higher tolerances in relative positioning during needle hand-off, relaxing the accuracy requirements of needle tracking.

Mode of Operation: SNAP is mounted axially on one of the needle driver jaws. It is designed to guide the needle towards a groove running perpendicular to the length of the gripper jaws Figure 5.1 (b), (c). Upon closing the jaws, the needle rolls to a stable pose, passing through contact points C_1 and C_2 as shown in the section view in Figure 5.1(b).

The size of the needle gripper is parametrized by the distance between contact points C_1 and C_2 which is dependent on the curvature of the needle, that is a needle with a larger radius

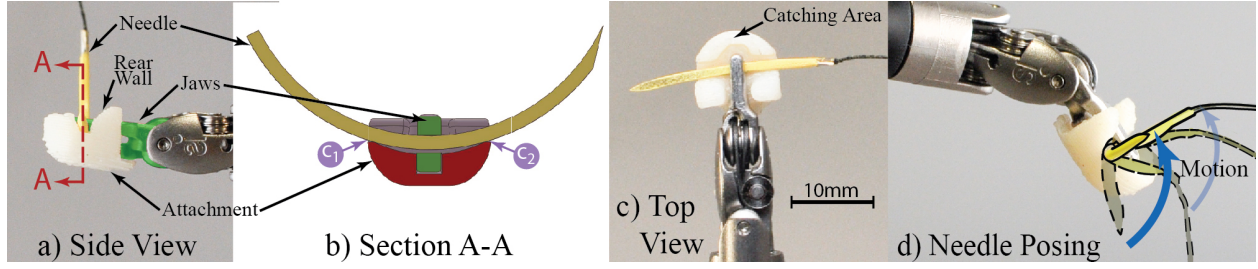


Figure 5.1: This figure illustrates the design and function of the 3D-printed Suture Needle Angular Positioner (SNAP). Figures (a) and (b) show a convex depression in which needle rests upon gripper closure. Figure (d) shows a time-lapse figure of the gripper closing action on needle orientation.

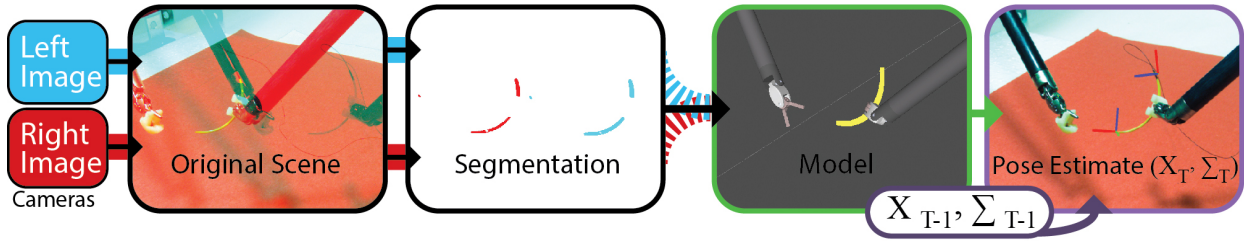


Figure 5.2: This figure shows an overview of the needle tracking pipeline, from stereo images to the final needle pose estimate overlaid onto the original scene. Each box contains an image from each step of the pipeline starting from the raw images from each camera and ending with the final pose estimate overlaid on the camera image. We fuse a Kalman Filter estimate with current camera estimate to compute the final estimate. The tracking system is robust to outliers and missing data in the segmentation masks.

needs a wider contact grasp to enable the needle rolling upon jaw closure. As illustrated in Figure 5.1(a), SNAP has a rear-wall that allows the gripper to overshoot during the pre-grasp approach. It also has a needle *catching area* in the front (Figure 5.1(c)) that guides the needle into the groove, compensating for undershoot during pre-grasp. Both of the above features increase the robustness of needle manipulation.

The SNAP is fabricated from ABS plastic using a *Stratasys uPrint* 3D printer. For an 8 mm classic needle driver, using a $\frac{3}{8}$ circumference, 39 mm length needle, we designed the SNAP with $C_1 - C_2$ span of 10 mm. Through experimental evaluation, we improved upon the SNAP design to include a larger rear wall. This enabled a wider jaw opening during approach allowing for larger tolerance in needle pose uncertainty.

5.2 Real Time Needle Tracking

We have developed a real-time needle tracking system to provide closed loop feedback during the suturing process as summarized in Figure 5.2. Due to tissue and tool specularities, per-

ception using RGB-D sensing is not feasible. Our system provides 3D needle pose estimates using a custom built stereo camera pair, composed of two Prosilica GigE GC1290C cameras with 6 mm focal length lenses. The needle tracking algorithm is implemented as a ROS node that publishes real time estimates of the needle’s pose. The tracking system works with partial occlusion for instance when the needle is inside the tissue or behind the robot arms.

We use a model-based tracking system leveraging the needle shape and color. The first step in the process is *Needle Segmentation*. We use a yellow painted needle to assist in foreground/background separation. We use HSV (Hue, Saturation, Value) separation to identify the needle in a cluttered environment with the open-source OpenCV library and create a set of image plane points \mathcal{P}_I .

We leverage the circular shape of the surgical needles and their elliptical projection. We create a small set of parametrically sampled points along the length of needle model \mathcal{P}_M , $|\mathcal{P}_M| = 12$, and then use affine point set registration to fit the \mathcal{P}_I to \mathcal{P}_M . We model the non-linear registration problem as point set matching. This creates robustness to outliers, missing data due to occlusions, and noisy data from incorrect segmentation masks. We use the Matlab library CPD2 for solving the registration problem [26].

Using the ellipse fits on the image pair, we generate a dense set of corresponding points along the needle. This creates a robust disparity map of 3D points on the needle. A plane is then fit to the 3D points, providing a normal vector, while an average tangential direction is calculated using the three points on the end of the needle. Using the end point of the needle and these two vectors, a pose $p_n \in \mathbb{R}^6$ is generated. We use a Kalman filter to smooth needle tip pose estimates.

The use of industrial Prosilica cameras with a wide baseline necessitated the use of a large workspace and consequently larger than average needles in order to enable robust needle tracking. Laparoscopic cameras have a smaller baseline and smaller field of view compared to our setup. The proposed tracking system should be transferable to a laparoscopic setup allowing the use of much smaller needles.

Chapter 6

System Design

6.1 Multi-Throw Suturing: System Design

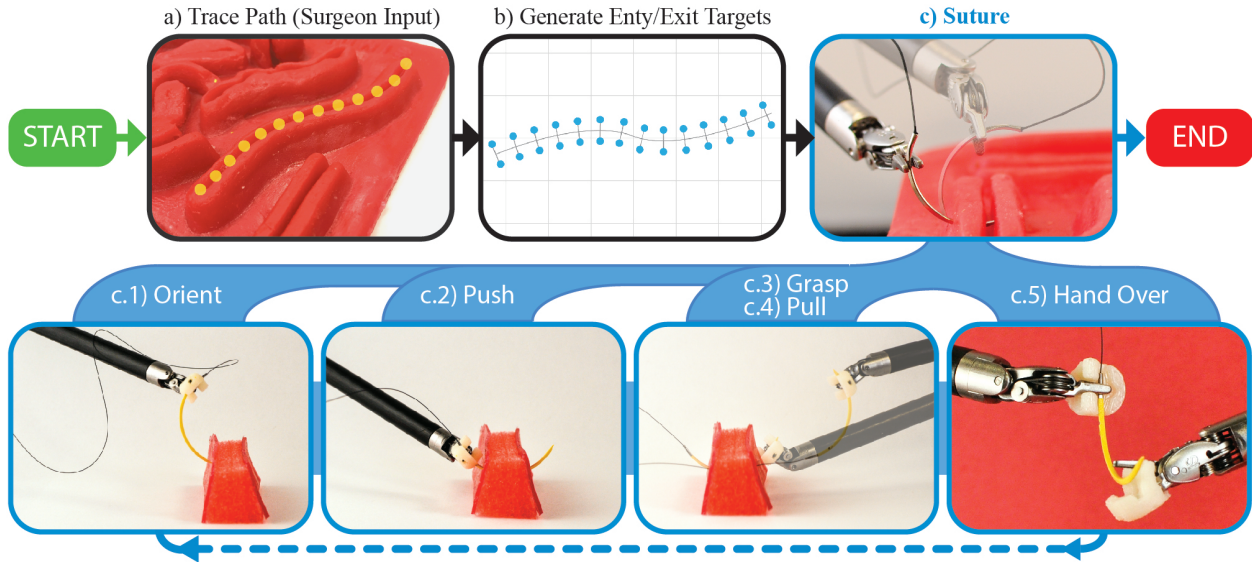


Figure 6.1: The figure outlines the Multi-Throw Suturing Finite State Machine. First the surgeon specifies a suture path with wound width & depth and suture pitch. The system then computes the number of suture throws required; and generates entry & exit points and optimized trajectories along with required needle size for each throw of the MTS. Each of the steps S1-S5 (see Figure 1.1) are repeated with visual feedback for each suture throw until all suture throws are completed.

We present a closed loop Finite State Machine (FSM) for multi-throw suturing with needle tracking and multilateral needle hand-off as illustrated in Figure 6.1. Given the registration of the tissue phantom in the camera frame, a multi-throw suture plan is generated. The SPP algorithm is used to generate needle trajectories and a suggested needle curvature.

Each throw in the task consists of the following sequence of sub-tasks which were segmented on the basis of manual surgeme labels for suturing in the JIGSAWS dataset:

S1. Needle Orientation: The system generates pose estimates for both the front tip of the needle, N_T , and the tail connected to the suture thread, N_S . Starting with the needle held in the right gripper at N_S , the system creates an initial pose estimate. Using this estimate, the robot aligns the needle with the camera’s image plane, allowing for an occlusion-free view of the needle and an improved pose estimate.

S2. Needle Insertion: The system executes a trajectory for N_T using the planner described in the previous section. We note that at this point, the suture path can be re-planned after every user-specified rolling time horizon.

S3. Needle Grasp: After the right arm guides the needle through tissue, the left arm grasps the needle at N_T and pulls the needle tangentially to the needle tip, rotating around the center of curvature of the needle in order to minimize tissue trauma.

S4. Needle Pull: Once the needle is completely outside the tissue, it is pulled away sufficiently to tighten the suture. The system estimates how much slack is available in the suture thread by modeling the length of thread between consecutive entry points as a helical loop with radius equal to the radius of the needle and pitch equal to the suture pitch. This provides a conservative estimate of how much slack is lost in each throw and the system uses it to decrease the distance the needle is pulled away after each throw.

S5. Needle Hand-Off: Our needle tracking algorithm estimates the pose of the needle end N_S while it is grasped at N_T . Similar to step (S1), the left arm aligns the needle with the image plane to improve the needle pose estimate. This estimate is used to align the needle with the right arm in order to grasp the needle at N_S and perform the next suture throw.

Due to inherent pose errors in camera-robot registration and robot kinematics, the hand-off process is performed by simultaneously engaging the right arm at N_S while disengaging the left arm at N_T . A slight error in coordination will result in failed transfer due to stresses generated on the needle. The use of SNAP on both gripper ends facilitates this process because the grooves provide a space resulting in a partial cage instead of force closure during the hand-off.

Suture Thread Length Management: At the end of each suture throw, the needle needs to pull through sufficient suture thread in order to perform all future throws while maintaining tension in completed sutures in order to facilitate tissue apposition. In absence of haptic feedback and suture thread tracking, we keep a running track of available suture length by recording the distance the gripper pulls the needle away from its exit point after each throw. We conservatively estimate length of the suture thread used in each throw as the length of a helical loop between consecutive entry points with helical radius equal to that of needle r and helical pitch equal to suture pitch w . We subtract this amount from our estimate of thread length after each throw.

Chapter 7

Experiments

7.1 dVRK: Hardware and Software

We use the Intuitive Surgical da Vinci Research Kit (dVRK) surgical robot assistant as in [24], along with open-source electronics and software developed by WPI and Johns Hopkins University [14]. We use a pair of 8mm Needle Drivers with each gripper having one Suture Needle Angular Positioner (SNAP). The software system is integrated with ROS and allows direct robot pose space control, working in Cartesian space instead of commanding motor torques. We developed a high level controller for the dVRK platform that allows us to publish goal pose commands for the robot’s tool tip through ROS. This system linearly interpolates from its current position to the desired goal position. Spherical linear interpolation is used to interpolate the tool tip’s rotation in order to achieve constant rotational motion. Our controller is designed to allow speed control of the robot allowing us to slow down the robot during delicate motions, while maintaining a higher speed during coarse maneuvers. The high level controller can also take as input a robot trajectory, enabling us to feed motion plans computed by our optimization solver directly into the system.

7.2 Experimental Evaluation of Needle Tracking

The size and shape of needles makes it difficult to obtain ground truth pose estimates using techniques like fiducial-based motion capture. Instead we designed an experiment to indirectly verify the efficacy of our needle tracking system. The robot holds the needle rigidly in its gripper and moves the needle to random positions in the workspace. Note that the relative pose of the needle with respect to the gripper position never changes. At each random position the robot pauses and uses the needle tracking system to compute the needle’s relative pose with respect to the gripper pose (estimated from kinematics). Poses at 20 different random locations were recorded. Table I shows the standard deviation in x,y,z (in mm) and in roll, pitch, and yaw (in degrees) respectively in the needle’s relative pose. The low error in every dimension suggests that our estimates of the needle’s relative pose

are nearly identical at each random location. This matches with the ground truth that the needle’s relative pose never changes. The errors reported are not due to the needle tracker alone, but the composite error produced from needle tracking, camera-robot registration, and robot kinematics. However, the errors provide an upper bound on the needle tracking error and is representative of error that our system must be robust to.

Table 7.1: Error in Relative Needle Pose (Over 20 Trials)

	Position (mm)			Orientation (degrees)		
	x	y	z	Yaw	Pitch	Roll
Std. Dev	2.182	1.23	1.54	2.495	4.699	4.329

7.3 Evaluation of Suture Needle Angular Positioner (SNAP)

1. Stationary Needle Pick Up:

In this experiment we evaluate the SNAP’s ability to reduce variation in needle grasp pose. This variation is the result of small natural perturbations in the needle starting pose and noise in the robot’s kinematic chain. In each trial, a needle is placed in the same location and the robot is provided a constant known grasp pose to initiate pick up. Once the needle is grasped, the robot brings the needle to a known location and the needle’s pose is recorded using our needle tracker. We repeat this process over ten trials both with and without SNAP. The standard deviations in each degree of freedom of the needle’s pose is presented in Table II. The SNAP reduced needle pose variation in both position and orientation, in some cases by over one order of magnitude.

2. Perturbed Needle Pick Up:

In the second experiment we intentionally perturb the orientation of the robot’s grasp pose to evaluate robustness to uncertainty and variation in grasp orientation. Experiment 2 is a variation of Experiment 1 where the commanded grasp pose is perturbed from -30 degrees to 30 degrees in yaw, pitch, and roll. The perturbations are applied in increments of 10 degrees independently in each axis resulting in 19 trials total. Our results show that the use of SNAP results in a $3x$ reduction in needle pose uncertainty over the standard Needle Driver.

Table 7.2: SNAP Evaluation

Stationary Grasp Orientation		Error (Standard Deviation)					
	Succ. Grasps	x (mm)	y (mm)	z (mm)	yaw (deg)	pitch (deg)	roll (deg)
Without SNAP	100%	2.511	1.434	4.838	20.547	7.584	6.472
With SNAP	100%	0.199	0.158	0.177	0.926	1.094	0.664
Perturbed Grasp Orientation		Error (Standard Deviation)					
	Successful Grasps	x (mm)	y (mm)	z (mm)	yaw (deg)	pitch (deg)	roll (deg)
Without SNAP	100%	2.01	2.59	5.95	15.54	12.74	7.62
With SNAP	91.66%	1.58	1.15	1.19	5.55	3.97	6.34

7.4 Robot Experiments: Four-Throw Suturing Task

We used a suturing phantom made with foam to mimic subcutaneous fat tissue with a layer of 1mm thick skin using (shore hardness 2A) *DragonSkin 10 Medium* Silicone Rubber (*Smooth-On*). The tissue was designed to have the following characteristics. First, the suturing phantom was designed to be deformable. This mimics the natural deformability of tissue and introduces uncertainty into the needle insertion process. Second, the dynamics of the needle changes when it transitions from traveling through free space to traveling through tissue. By building the phantom around a foam core we ensure that the phantom can be penetrating by the needle while still providing resistance to torques applied to the needle tip. We also performed experiments on a second version of our phantom that replaced the silicon skin with chicken skin in order to improve its clinical accuracy. The mechanical design of the dVRK robotic arms ensure that the arms do not move at the point where they would enter a human body ensuring that the kinematic motions of our system remain feasible in-vivo in a minimally invasive surgical (MIS) setting. Due to the wide baseline of our stereo cameras, the size of our phantom, needles, and workspace were constrained to be larger than those found in a nominal MIS setting.

In this experiment, the system tries to complete a closed loop four throw suturing task similar to the suturing task found in the JIGSAWS data-set [6]. We initialized the system with entry and exit poses on opposite surfaces of the tissue phantom and with a desired suture depth. Our system generates insertion trajectories and based on the output optimal needle curvature, we selected a 39mm long, 3/8 reverse cutting needle to perform the suturing throws. For each trial we record time to completion as well as the failure mode if necessary. The robot moves at a top speed of 3cm/s. The results of each trial are found in Tables 7.3 and 7.4.



Figure 7.1: The following is an image of our experimental setup. We also see a revision of our tissue phantom where the silicone skin is replaced with raw chicken skin.

Table 7.3: Results for Four-Throw Suturing. 14 trials were performed, with a 50% success rate. For failed states, “N.I” represents incorrect needle orientation or insertion, “G.P.” represents incorrect needle re-grasp and pull after insertion, and “H.O” represents failure in needle hand-off respectively. The test setup was varied with translation of simulated wound along the wound axis.

Trial	4-Throw Success	# of Throws Completed (Attempted)	Failure Mode	Trans. in X	Suture Pitch	Total Time(s)
1	Failure	1 (2)	G.P.	-3mm	3mm	-
2	Failure	2 (3)	G.P.	-3mm	3mm	-
3	Failure	3 (4)	G.P.	-2mm	3mm	-
4	Success	4 (4)		-1mm	3mm	387
5	Success	4 (4)		0mm	3mm	380
6	Success	4 (4)		0mm	3mm	380
7	Success	4 (4)		0mm	3mm	383
8	Failure	2 (3)	H.O.	1mm	3mm	-
9	Failure	2 (3)	N.I.	1mm	3mm	-
10	Failure	3 (4)	G.P.	2mm	3mm	-
11	Success	4 (4)		3mm	3mm	393
12	Success	4 (4)		4mm	3mm	383
13	Success	4 (4)		5mm	3mm	382
14	Failure	3 (4)	G.P.	6mm	3mm	-
Mean	50%	3.14				384
Std Dev		1.027		Single Throw Success Rate: 86.3%		

Table 7.4: This table compares the performance of our autonomous suturing system with different skill levels of surgeons in the JIGSAWS dataset[6]

Operator Mode	Average Time for 1-Throw (s)	Average time for 4-throw Task (s)
Expert	19.03	87.02
Intermediate	18.57	87.89
Novice	32.14	136.85
Autonomous (Our Approach)	112.33	383.00

Chapter 8

Conclusion

8.1 Discussion and Future Work

Initial experiments in this paper confirm that the system presented can computationally plan and execute a multi-throw suturing task with four-throws in closed-loop operation. The combination of our needle tracking system and SNAP enables our system to minimize needle pose uncertainty. This allows our system to perform multilateral needle hand-off, enabling the execution of multi-throw suturing.

The system completes 86.3% of individual suture throws attempted at approx. 30% of the average speed of manually tele-operated demonstrations as listed in Tables 7.3 and 7.4. Our results also show that the proposed needle tracking system can provide robust estimates of needle pose in near real-time with an empirical error of up to 5 degrees. Furthermore the use of SNAP improves repeatability in needle grasping by $10\times$ and grasping is robust to up to 30 degrees error in needle estimate.

However, we note that the system completes on average 3.14 of the intended 4 throws, with a 50% completion rate for the four-throw task. It is worth noting that 5 out of the 7 failures were due to incorrect needle re-grasp and pulling after the insertion step. Some of these failures were due to incorrect needle estimate after the needle exits the tissue in unexpected locations. The visual needle tracker could not recognize the needle due to large occlusions. Additional failures were due to entanglement of the suture thread during needle pulling.

We see two directions of future work that could improve the speed, robustness, and reliability of our autonomous surgical suturing system. The first direction is suture thread management. In this work we do not track the position or deformation of the suture thread. This limitation prevents us from tying knots and can lead to failure modes where the grippers or needle get tangled with the suture thread. Thread tracking combined with trajectory planning for thread management has the potential to allow our system to avoid entanglement and tie knots using a general framework.

Another exciting direction of work is real-time re-planning for needle insertion. Currently

one of our most common failure modes occurs when the needle tip exits the tissue in an unexpected location, preventing the robot from pulling it out of the tissue. This failure is common because even small errors in needle tip positioning before insertion become magnified by the deformation of the tissue phantom. Using visual feedback to re-plan the trajectory during the insertion would provide the system with better control of the the needle tip's exit point. Since tissue deformation is difficult to model, learning based approaches could be used to train the system to guide the needle through deformable material.

Our work presents initial results toward automating MTS with a combination of new hardware and a novel optimization algorithm. The paper describes the mechanical device, the Suture Needle Angular Positioner (SNAP), designed to align and hold the needle in a known orientation, and an SCP formulation of needle motion planning. Initial results suggest that SNAP can reduce error in needle orientation by $3\times$ and that the combined system can successfully complete 86% of attempted throws at 30% the speed of human operators [6].

8.2 Challenges and Open Questions in Surgical Automation

We conclude by discussing some of the general challenges and open questions in surgical automation.

Unique Challenges in Surgical Automation

While many of the challenges in the automation of robotic surgery are broadly applicable to robotics in general, some are either unique to robotic surgery or unique to a small subset of the field. For example, many elements of a surgical environment are highly deformable. Surgical systems interact with their environment through bending, cutting, stretching, peeling, and puncturing a variety of tissues and materials. Sensing and modeling such a varied environment remains a challenge in surgical settings. It is important to note the specific challenges that exists in a surgical environment. While existing sensors like the Kinect that have had great success in mapping and reconstruction in the macro world, such sensors perform poorly in small surgical environments due to focusing limitations and the highly specular nature of surgical environments. Much of our own work abstracts away the challenges of environmental sensing and modeling. In our palpation work, we assume a known tissue shape and configuration. Similarly our autonomous suturing system ignores deformable suture thread and models the suturing incision as a rigid mesh. These limitations need to be addressed in order to build safe and robust systems.

Another challenge in surgical automation is the difficulty in defining how surgical sub-tasks should be performed. Much of our work uses finite state machines to complete tasks. However these require explicit descriptions of how the robot needs to interact with its environment. For example, in our work in autonomous suturing, the costs and constraints for a

good suturing motion plan had to be carefully constructed from a series of surgeon recommendations and guidelines. However for more complicated procedures, such task structure can be unclear. For example, if the task is to excise a subcutaneous tumor, it can be challenging for even a human to define a general purpose set of guidelines to complete the task. While state machines provide autonomous systems with predictability and semantic meaning, they can be limiting in scenarios where it's difficult to construct concrete guidelines on how to perform a task. In prior work we explored how surgeon demonstrations can be used to automate surgical cutting [24]. As more kinematic and visual data from robotic surgery becomes more readily accessible, figuring out how to learn from this data is an exciting area of research. In [18], [16],[25], Krishnan and Garg et al explore how surgeon demonstrations can be used to learn task milestones for the construction of state machines.

Hardware for Automation

Current robotic surgical assistants are designed with human tele-operation as their primary mode of control. As result, errors in tool tip pose estimation and camera registration are easily compensated for by a human operator. However such errors can have a huge effect on system performance on autonomous tasks. For example, constructing accurate surface stiffness maps with a palpation probe requires the robot to maintain a constant tool tip orientation as it moves through the workspace. However systematic bias and hysteresis in tool position and orientation results in a decrease in the tool's signal to noise to ratio. Physical experiments showed that while the probe is capable of detecting inclusions 5mm deep when mounted on a vertical CNC machine, its sensitivity decrease to approximately 1-2mm when mounted on the dVRK due to positional noise inherent in the surgical system. These results suggest the need to develop surgical systems with autonomy in mind with complementary hardware and software. In this work the use of both SNAP(a hardware solution) and needle tracking (a software solution) enables robust transfer of the needle between robot tool tips. Without such a system, performing multiple suture throws would be significantly less robust. Further exploration of how hardware and software solutions can complement each other may enable us to build systems that are greater than the sum of their parts.

Experimental Setups

The design and development of experimental setups and tissue phantoms is an important component for the evaluation of autonomous systems. It's not always feasible to evaluate a surgical system on live tissue. Nevertheless, it's important to perform tests on experimental setups that are clinically relevant. In some instances the limitations of tissue phantoms can adversely affect system performance. For example, we used silicone based tissue phantoms for our palpation experiments because they mimic the deformability and stiffness of real tissue. However the high coefficient of friction of silicone made cutting the surface with a scalpel challenging. As a result, the autonomous system was designed to perform a sawing motion

while cutting, rather than the simple straight line motion that one could apply to real tissue. In some sense the system was designed to solve an artificial problem; one that existed in our experimental setup but not in real clinical settings. Using tissue phantoms constructed from real tissue such as chicken skin can mitigate some of the problems introduced by purely artificial phantoms but can introduce their own challenges. Using animal tissue can slow down development and iteration speed due to the necessity of maintaining a sterile environment around the experimental setup. It can also be challenging to maintain consistent tissue properties over multiple trials. For example, chicken skin tends to dry rapidly, resulting in a membrane that becomes more challenging to pierce with a needle as time progresses. Ideally tissue phantoms need to be consistent, durable, and easy to construct while mimicking key properties of living tissue. It's an open question how to find the right balance between these opposing factors.

Chapter 9

Appendix

Generating constrained curvature trajectories in 3D space is challenging because it requires planning in the Special Euclidean Group in 3D, i.e. $SE(3)$ configuration space. $SE(3)$ is a semidirect product of Special Orthogonal Group ($SO(3)$) and \mathbb{R}^3 . Using homogeneous coordinates, we can represent $SE(3)$ as follows:

$$SE(3) = \left\{ \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \in GL(4) \mid R \in SO(3), t \in \mathbb{R}^3 \right\} \quad (9.1)$$

The action of an element $g \in SE(3)$ on a point $p \in \mathbb{R}^3$ is given by $g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$, $g \cdot p = Rp + t$.

The Lie algebra of $SE(3)$, as in Agarwal [1], is given by

$$\mathfrak{se}(3) = \left\{ \begin{bmatrix} \hat{\omega} & u \\ 0 & 0 \end{bmatrix} \in GL(4) \mid \hat{\omega} \in \mathfrak{so}(3), u \in \mathbb{R}^3 \right\} \quad (9.2)$$

Here $\hat{\omega}$ is the skew symmetric form of the rotation vector $\omega = (\omega_x, \omega_y, \omega_z)^T$ and is an element of the Lie algebra for $SO(3)$

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (9.3)$$

The logarithm map $SE(3) \rightarrow \mathfrak{se}(3)$ is given by:

$$\log \left(\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \right) = \begin{bmatrix} \log(R) & A^{-1}t \\ 0 & 0 \end{bmatrix} \quad (9.4)$$

where

$$A^{-1} = I - \frac{1}{2}\hat{\omega} + \frac{2 \sin \|\omega\| - \|\omega\|(1 + \cos \|\omega\|)}{2\omega^2 \sin \|\omega\|} \hat{\omega}^2,$$

$$\log(R) = \frac{\phi}{2\sin(\phi)}(R - R^T) \equiv \hat{\omega}$$

and ϕ satisfies $Tr(R) = 1 + 2\cos(\phi)$, $|\phi| < \pi$. The matrix exponential to map $\mathfrak{se}(3) \rightarrow SE(3)$ is given by

$$\exp \left(\begin{bmatrix} \hat{\omega} & u \\ 0 & 0 \end{bmatrix} \right) = \begin{bmatrix} \exp(\hat{\omega}) & Au \\ 0 & 0 \end{bmatrix} \quad (9.5)$$

where $\exp(\hat{\omega})$ can be computed with Rodrigue's formula:

$$\begin{aligned} \exp(\hat{\omega}) &= I + \frac{1 - \cos(\|\omega\|)}{\|\omega\|} \hat{\omega} + \frac{\sin(\|\omega\|)}{\|\omega\|^2} \hat{\omega}^2, \\ A &= I + \frac{1 - \cos(\|\omega\|)}{\|\omega\|^2} \hat{\omega} + \frac{\|\omega\| - \sin \|\omega\|}{\|\omega\|^3} \hat{\omega}^2 \end{aligned}$$

With equations (9.4) and (9.5), we can convert an element in $SE(3)$ to $\mathfrak{se}(3)$ and vice-versa.

In each gradient descent iteration, we can update our current solution by taking gradient steps in $\mathfrak{se}(3)$. This let's avoid the problems with other pose representations presented above. Since we can convert efficiently between $SE(3)$ and $\mathfrak{se}(3)$, we can still represent our costs and constraints in $SE(3)$.

Bibliography

- [1] Motilal Agrawal. *A Lie Algebraic Approach for Consistent Pose Registration for General Euclidean Motion*. IEEE, 2006.
- [2] Jur van den Berg et al. “Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (2010).
- [3] Jon M Burch et al. “Single-layer continuous versus two-layer interrupted intestinal anastomosis: a prospective randomized trial”. In: *Annals of surgery* (2000).
- [4] Fernando De Goes, Siome Goldenstein, and Luiz Velho. “A hierarchical segmentation of articulated bodies”. In: *Computer Graphics Forum*. 2008.
- [5] Yan Duan et al. “Planning locally optimal, curvature-constrained trajectories in 3D using sequential convex optimization”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014.
- [6] Y. Gao et al. “The JHU-ISI Gesture and Skill Assessment Dataset (JIGSAWS): A Surgical Activity Working Set for Human Motion Modeling”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2014.
- [7] Animesh Garg et al. “Tumor Localization using Automated Palpation with Gaussian Process Adaptive Sampling”. In: (2016).
- [8] Julio Hochberg, Kathleen M Meyer, and Michael D Marion. “Suture choice and other methods of skin closure”. In: *Surgical Clinics of North America* 89.3 (2009), pp. 627–641.
- [9] Intuitive Surgical. *Annual Report 2014*. 2014. URL: <http://investor.intuitivesurgical.com/phoenix.zhtml?c=122359&p=irol-IRHome>.
- [10] M. Cenk Jackson Russell C. and Cavusoglu. “Modeling of needle-tissue interaction forces during surgical suturing”. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*. 2012.
- [11] Russell C. Jackson and M. Cenk Cavusoglu. “Needle path planning for autonomous robotic surgical suturing”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2013.

- [12] Russell C Jackson et al. “Automatic initialization and dynamic tracking of surgical suture threads”. In: *IEEE Int. Conf. Robotics and Automation*. 2015.
- [13] Hyosig Kang and John T Wen. “Autonomous suturing using minimally invasive surgical robots”. In: *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*. IEEE. 2000, pp. 742–747.
- [14] P Kazanzides et al. “An Open-Source Research Kit for the da Vinci Surgical System”. In: *IEEE Int. Conf. Robotics and Automation (ICRA)*. 2014.
- [15] Ben Kehoe et al. “Autonomous multilateral debridement with the Raven surgical robot”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014).
- [16] Sanjay Krishnan et al. “HIRL: Hierarchical Inverse Reinforcement Learning for Long-Horizon Tasks with Delayed Rewards”. In: *arXiv preprint arXiv:1604.06508* (2016).
- [17] Sanjay Krishnan* et al. “Transition State Clustering: Unsupervised Surgical Trajectory Segmentation For Robot Learning”. In: *Int. Sym. of Robotics Research*. Springer STAR. 2015.
- [18] Sanjay Krishnan* et al. “Transition State Clustering: Unsupervised Surgical Trajectory Segmentation For Robot Learning”. In: *International Symposium of Robotics Research*. Springer STAR. 2015.
- [19] Colin Lea, Gregory D Hager, and René Vidal. “An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks”. In: *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE. 2015, pp. 1123–1129.
- [20] D T Martin et al. *Articulating needle driver*. Tech. rep. 2012.
- [21] Hermann Georg Mayer et al. “A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks.” In: *IROS* (2006).
- [22] Stephen McKinley et al. “A disposable haptic palpation probe for locating subcutaneous blood vessels in robot-assisted minimally invasive surgery”. In: *IEEE CASE* (2015).
- [23] Stephen McKinley et al. *Autonomous Tumor Localization and Extraction: Palpation, Incision, Debridement and Adhesive Closure with the da Vinci Research Kit*. Hamlyn Surgical Robotics Conference, London. 2015. URL: j.mp/palpation-vid.
- [24] Adithyavairavan Murali* et al. “Learning by Observation for Surgical Subtasks: Multilateral Cutting of 3D Viscoelastic and 2D Orthotropic Tissue Phantoms”. In: *IEEE Int. Conf. Robotics and Automation*. 2015.
- [25] Adithyavairavan Murali* et al. “TSC-DL: Unsupervised Trajectory Segmentation of Multi-Modal Surgical Demonstrations with Deep Learning”. In: (2016).
- [26] Andriy Myronenko and Xubo Song. “Point set registration: Coherent point drift”. In: *IEEE Tran. on Pattern Analysis and Machine Intelligence* (2010).

- [27] Florent Nageotte et al. *Circular needle and needle-holder localization for computer-aided suturing in laparoscopic surgery*.
- [28] Scott Niekum et al. “Learning Grounded Finite-State Representations from Unstructured Demonstrations”. In: *Int. J. of Robotics Research* (2015).
- [29] N. Padoy and G.D. Hager. “Human-Machine Collaborative Surgery using Learned Models”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2011, pp. 5285–5292.
- [30] Swapnil Patil et al. “Gaussian belief space planning with discontinuities in sensing domains”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE. 2014, pp. 6483–6490.
- [31] S.U. Qureshi, K.M. Rupp, and B. Thompson. *Needle holder with suture filament grasping abilities*. US Patent 5,951,587. Sept. 1999.
- [32] Carol E Reiley and Gregory D Hager. “Task versus Subtask Surgical Skill Evaluation of Robotic Minimally Invasive Surgery”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2009.
- [33] John Schulman et al. “A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario”. In: 2013.
- [34] John Schulman et al. “Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization”. In: *Robotics: Science and Systems (RSS)*. 2013.
- [35] John Schulman et al. “Motion planning with sequential convex optimization and convex collision checking”. In: *The International Journal of Robotics Research* (2014).
- [36] S Speidel et al. “Image-based tracking of the suturing needle during laparoscopic interventions”. In: *SPIE Medical Imaging*. 2015.
- [37] C. Staub et al. “Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. 2010.
- [38] Andrew P Stegeman et al. “Fundamental skills of robotic surgery: a multi-institutional randomized controlled trial for validation of a simulation-based curriculum”. In: *Urology* (2013).