# An Online Method for Tight-tolerance Insertion Tasks for String and Rope

Weifu Wang[1], Dmitry Berenson[2] and Devin Balkcom[3]

***Abstract*— This paper presents a fast tight-tolerance threading technique for string and rope. Instead of relying on simulations of these deformable objects to plan a path or compute control actions, we control the movement of the string with a virtual magnetic vector field emanating from the narrow openings we wish to thread through. We compute an approximate Jacobian to move the tip of the string through the vector field and propose a method to promote alignment of the head of the string to the opening. We also propose a method for re-grasping the string based on the relationship between the string's configuration, the orientation of the opening, and direction of gravity. This re-grasping method in conjunction with our controller can be used to thread the string through a sequence of openings. We evaluated our method in simulation (with simulated sensor noise) and on the Da Vinci surgical robot. Our results suggest that our method is quite robust to errors in sensing, and is capable of real-world threading tasks with the da Vinci robot, where the diameter of the string (3.5mm) and opening (4.9mm) differ by only 1.4 mm.**

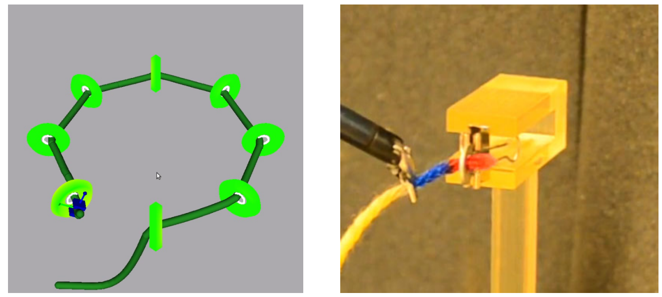


Fig. 1. Two examples of successfully threading deformable objects through of small openings online, in simulation and using a Da Vinci robot.

## I. Introduction

Manipulation of deformable objects is challenging due to the high dimensionality of these objects' state spaces, the difficulty of simulating deformation, and uncertainty in model parameters. Several approaches attempt to overcome these challenges through motion planning [8], [19], [23], [25], [28], but these methods rely on accurate simulations of deformation. While suitable for tasks that can be accomplished without needing high precision, these methods are not well-suited for tasks that require the object to be precisely aligned with environment features, such as threading a needle. In such tight-tolerance tasks, even slight inaccuracies in the simulation used for planning can lead to failure in execution.

This paper addresses tight-tolerance insertion tasks for string and rope like those shown in Figure I. Example tasks include but are not limited to, threading a needle, threading a belt, knotting knots, knitting, etc.

Rather than planning a path in simulation, we take a control approach to the problem. Recent work by co-author Berenson [2] shows that an *approximate Jacobian* can be computed that relates the motion of a gripper and control points on the deformable object using only geodesic distance information measured along the object. This Jacobian can then be used to servo the object toward a set of desired target points.

However, this method is not sufficient for robust tight-tolerance insertion, as, depending on the initial configuration of the string, it can easily mis-align the string with respect to the hole, converging to a local minimum that does not complete the task. The key contribution of this paper is to use the approximate Jacobian in conjunction with a virtual magnetic field emanating from the hole. This field can be used to align the string's tip to the hole regardless of the starting configuration. It also allows the controller to "re-try" insertion without failure checks or switches in the control strategy: although the controller may miss the hole several times, we simply execute the same controller until the insertion is complete. This controller, combined with our re-grasping strategy allows us to perform tasks which require multiple tight-tolerance insertions.

Although this work is still in early stages, and represents only a proof of concept, we have found the approach to be promising, and to have several advantages. The control strategy does not require simulation or modeling of the deformable objects beyond their basic geometry. Unlike in motion planning methods, computation of desired motion is fast enough to allow new gripper velocities to be computed on-the-fly, as new sensor information becomes available. Finally, the approach explicitly acknowledges and avoids the difficulty of effectively modeling the dynamic behavior of twisted string or rope, and is robust enough to allow insertion even in the presence of noise.

We evaluated our approach on a series of simulation (using the Bullet physics library [1]) and real-world (using a Da Vinci medical robot) experiments focusing on the problem of inserting a piece of string through a small opening. Figure I shows two examples of the achieved tasks. In the simulated tasks, significant artificial sensor noise is injected; preliminary experimental results show that the approach is quite robust to sensing errors.

## II. Related Work

Control for deformable objects has been investigated in a variety of industrial [11] and surgical [15], [16] contexts. Work on controlling deformable objects to tie knots dates back as far as Inoue and Inaba's 1985 paper [14]. Many of

[1] Dartmouth College, weifu.wang@dartmouth.edu
[2] Worcester Polytechnic Institute, dberenson@cs.wpi.edu
[3] Dartmouth College, devin@cs.dartmouth.edu

these approaches to deformable object manipulation require a fairly detailed model or simulation of the deformable object [9], [26].

The method for control proposed here is related to work in visual servoing for deformable objects [12], [22], [33], [36], though these methods do not use the vector field approach we propose, or attempt to accomplish tight-tolerance insertion tasks. Approaches that learn control policies for specific tasks from demonstration have also been investigated [21], [24], [29], [30]. Though effective at solving problems similar to those demonstrated, it is unclear how well these methods generalize to new tasks and environments.

Manipulation of deformable objects like wire and thread has also been explored using motion planning [6], [20], [27], [28]. However, without a controller that can adapt to errors in sensing and execution, it is unlikely open-loop execution of planned paths can accomplish tight-tolerance insertion tasks for deformable objects. The controller we use is inspired by the magnetic field generated by a current-carrying wire; Haddadin *et. al.* have also used magnetic fields for control [10].

In the simplest case where there is only one ring into which we insert the string, the insertion task is similar to a classical problem: peg-in-hole [**?**], [**?**]. Our approach focuses on simple online controllers intended to be robust to deformation of the string, and relies on a global vector field that repeatedly drives the endpoint of the string to the hole if the first attempt is not successful.

Using sequences of controllers to achieve motions of complex dynamical systems has a rich history. Burridge *et. al.* [7] generated sequential motions for robots using fields. Tedrake *et. al.* proposed LQR-trees [34], [35] that construct funnels to generate motions to reach goal regions. In our work, we use a sequence of virtual magnetic fields to generate motions for the string.

In some sense, there is a parallel between finding a path for a point robot within a homotopy class, and our problem of achieving insertion of string through a small opening. Bhattacharya *et. al.* used a magnetic field to identify different homotopy paths in configuration spaces [3], [4], [5].

## III. Outline of the approach

The starting point for our method is a geometric model of an environment and one or a sequence of small openings for threading. First, we place a small virtual loop inside each of the small openings; if there are obstacles in the environment, we may place additional loops to avoid the obstacles. The objective is to use a sequence of virtual magnetic fields generated by the loops to drive the string around obstacles and through the openings. This paper focuses on guiding the string through the series of loops, not on determining the poses and radii of the loops, so we specify the loops manually in our test scenarios.

At a given time, the velocity and rotational velocity control of the gripper(s) grasping the string is computed as follows. A single loop is considered *active*; the desired velocity of the tip of the string is computed using magnetic field equations that simulate running a current through the active loop (Section V).

To ensure insertion, a second reference point on the string is used to control orientation of the string near the tip. The selection of this second point and computation of the desired motion for this point is introduced in Section V-A.

We next compute an approximate Jacobian that encodes the relationship between motions of the gripper and motions of the reference points (Section IV). We do not use the rigid body Jacobian because when the grasping point is far from the reference points the rigid body Jacobian will over-estimate the effect the gripper has on reference points' motion. Using this approximate Jacobian, we compute velocity and rotational velocities for the gripper that we expect to (approximately) give the desired motion of the reference points on the string. If the desired motion of the gripper would cause collision, a re-grasping strategy is applied (Section VI), and a new velocity is computed.

We consider insertion successful once the tip of the string has successfully penetrated a disc placed coincident with the active loop, but with larger radius (Section V-B). After successful insertion through the active loop has been achieved, the next loop in sequence becomes the new active loop.

## IV. Controlling deformable objects using an approximate Jacobian

We would like to determine the motion of one or more grippers grasping a deformable object that will cause certain reference points along the object to move in desired directions. Let $\mathcal{P}$ be a vector representing the locations (in the world frame) of the reference points.

We use a quasi-static model, so that the the equilibrium configuration of the flexible object is completely determined by the configuration(s) of the grippers (represented by a vector $q$) together with internal spring and external gravity forces. We have

$$\mathcal{P} = F(q), \tag{1}$$

where $F$ is the function that computes the locations of the reference points based on the configuration of the grippers. (Since the intent is to control the string locally, we will omit the fact that there might be multiple equilibrium configurations of the string for a given $q$.)

There is a Jacobian of $F$ that locally relates changes in the gripper configuration and changes of locations of reference points, $\mathcal{P}$:

$$\dot{\mathcal{P}} = J(q)\dot{q}. \tag{2}$$

If we have sufficiently many grippers relative to the number of points, then we expect there to be at least one set of gripper configuration-space velocities for a particular desired velocity of the reference points; if we knew $J$ for the configuration, we could compute such a gripper velocity by solving Equation 7 for $\dot{q}$. If there are not sufficiently many grippers, we might hope to find at least a best-possible motion (in the least-squares sense) using the pseudo inverse $J(q)^+$.

However, modeling string or rope sufficiently well to find $F$ or $J$ accurately poses several challenges, and we expect $F$ to be quite dependent on the physical structure and instantaneous configuration of the string. For example, the motion of wound string in a twisted configuration is perhaps extremely hard to predict.

Nonetheless, it is possible to *approximate* the Jacobian *without* a detailed model; Berenson [2] shows an approximation technique. In [2], the key idea is that even deformable objects behave "more rigidly" near the gripper. Experimentally [2], the amount of motion of a reference point due to a gripper diminishes exponentially with its distance from the gripper.

Let us call a Jacobian calculated based on this assumption a *diminishing rigidity Jacobian*, which can be denoted by $\tilde{J}(q)$. We briefly review the key points of a method for calculating this approximation $\tilde{J}(q)$; more detailed discussion can be found in [2].

Let us consider two points $p_i, p_j \in \mathcal{P}$, and denote the geodesic distance between them as $d(p_i, p_j)$. By geodesic distance, we mean the distance between these two points when the string (or other deformable object) is undeformed (a straight line). Let $c(i, g) \in \mathcal{P}$ be the point on the deformable object grasped by gripper $g$ that is closest to the $i$th point of $\mathcal{P}$. Let $w(i, g)$ be the rigidity weight of the $i$th point with respect to gripper $g$, which is calculated by:

$$w(i, g) = e^{-k(d(p_i, c(i,g)))}, \tag{3}$$

where $k$ is a constant determined experimentally for the system (for details, see [2]).

We take the translation component of $\tilde{J}(q)$ for the $i$th point with respect to gripper $g$ to be simply

$$\tilde{J}_{trans}(q, i, g) = w(i, g)\mathbf{I}_{3\times 3}. \tag{4}$$

Let the configuration of the gripper in the world frame be described by a $3 \times 3$ rotation matrix $\mathbf{R}^g$ and a translation vector $v^g$. We can then define the rotation component of the approximate Jacobian:

$$\tilde{J}_{rot}(q, i, g) = w(i, g)[\mathbf{R}^g[1] \times r, \mathbf{R}^g[2] \times r, \mathbf{R}^g[3] \times r], \tag{5}$$

where $r = (c(i, g) - v^g)$.

Then $\tilde{J}(q, i, g)$ for point $i$ with respect to gripper $g$ is

$$\tilde{J}(q, i, g) = w(i, g)[\tilde{J}_{trans}(q, i, g), \tilde{J}_{rot}(q, i, g)]. \tag{6}$$

Combining the Jacobians for all points and all grippers into a single matrix, we can find $\tilde{J}(q)$. To find $\dot{q}$ for a desired motion of the reference points, we multiply by the pseudo-inverse of the diminishing rigidity Jacobian:

$$\dot{q} = \tilde{J}(q)^+\dot{\mathcal{P}}. \tag{7}$$

Additionally, the calculated term of $\dot{q}$ can be combined with obstacle avoidance terms, calculated in Equations 11 to 13 in [2]; when multiple grippers are used, a stretch avoidance term can also be added to automatically correct excessive stretching. We will omit the details of these terms, which can be found in [2].

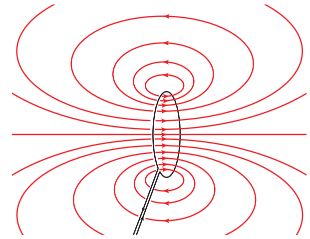


Fig. 2. An example of magnetic field induced by a current carrying planar circle loop.

## V. Constructing vector field

To perform insertion tasks, we construct vector fields in the workspace, such that the reference points on the string following these vector fields can lead the string through tight-tolerance spaces. These motions can then be used to generate appropriate motion for grippers.

The tip of the string is chosen as a reference point, since in order to thread the string through a tight opening, the tip usually needs to go through it first.

The vector field should satisfy certain properties to be useful. We would like all integral curves of the vector field to go through the tight opening, so that by following the vector field, the string can be inserted into the small opening regardless of the initial configuration. The vector field should also have a "direction" consistent with the the desired insertion task, so that all the field lines enter the opening from one side, and leave from the other. Finally, we would like to find a vector field that has a closed form solution everywhere in the workspace such that calculation of $\dot{\mathcal{P}}$ is fast.

One field that obeys these properties is a magnetic field induced by a current carrying loop. For simplicity, we let the wire be a planar circle (referred as a loop in the following context). An example of an induced vector field is shown in Figure 2.

For an arbitrary point $\mathrm{p}$ in the workspace, and a fixed current-carrying loop $S$, the magnetic force at $\mathrm{p}$ can be calculated as follows:

$$F(\mathrm{p}) = \frac{1}{4\pi}\int_S \frac{(x - \mathrm{p}) \times dx}{||x - \mathrm{p}||^2}. \tag{8}$$

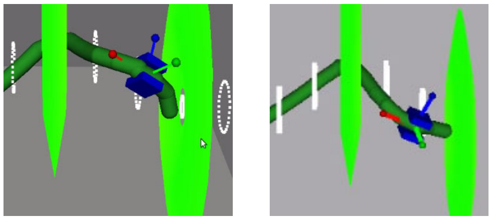
This vector is used as the desired motion for the tip of the string, which we denote as $v_1$.

It is convenient that the magnetic field shown in Figure 2 describes reasonable motions for the tip of the string globally. Even if the tip reaches the wrong side of the insertion loop, the field lines drive the tip of the string back around for another attempt using the same vector field. In some sense, the length of the integrated field line (from arbitrary point to the intersection with the interior of the loop) approximates the real distance to the goal.

In current work, we manually place these rings that induces the magnetic field. In future work we will study how to place these rings automatically.

### A. Orientation at the tip of the string

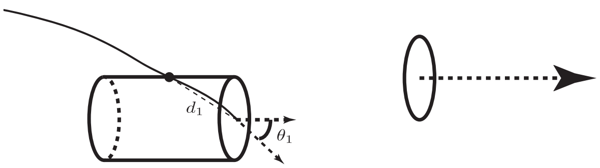
The orientation of the string tip is critical to the success of insertion. Figure 3 shows examples of two different angles



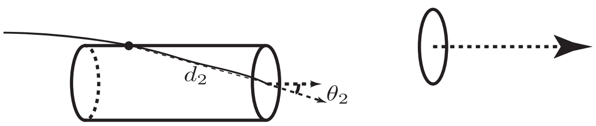
(a) Example of a bad angle for threading.

(b) Example of a better angle for threading.

Fig. 3. Examples of bad and good angles of threading.




(a) A configuration where the second point is close to the tip of the string.




(b) A configuration where the second point is further from the tip of the string.

Fig. 4. Dynamic selection of the second reference point; $d_1 < d_2$, and $\theta_2 < \theta_1$.

of the string tip. The experiment of string having the angle in Figure 3(a) did not go through the small opening.

Although the magnetic field gives reasonable motions for the tip of the string, it is in the workspace rather than the configuration space. If multiple reference points on the string follow the same vector field, it may not be possible to exactly specify motions of the reference points (or the string) in configuration space. Nonetheless, we have found that by carefully choosing the location of a second reference point along the string (in addition to the tip), and computing a different vector field "on the fly" for this point, the orientation of the tip can be controlled effectively.

We choose to find this second point dynamically, using a cylinder centered at the tip with radius $S$. Let the second point be the first intersection between the string and this cylinder. Figure 4 shows an example of dynamically choosing the second reference point.

What is the desired motion for the second reference point so that the string achieves an orientation consistent with the insertion direction? Let the two reference points have locations $p_1, p_2$, and the distance between them be $d$. Let the calculated motion of the tip be $v_1$, and the normal of the loop to penetrate be $n$. We choose the motion for the second point to be

$$v_2 = p_1 + v_1 - d \cdot n - p_2, \tag{9}$$

so that if the tip moves exactly $v_1$ in one time step, and the

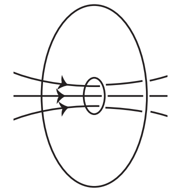


Fig. 5. A loop used for guiding the string, and a disk for detecting successful insertion.

second reference point moves $v_2$, then the orientation of the line segment between the references will be parallel to the insertion direction. Together, $v_1$ and $v_2$ forms $\dot{\mathcal{P}}$.

### *B. Radii of loops*

There are many sources of error in the insertion process. The expected motion is calculated using a Jacobian that is only approximate, and we expect errors in sensing and control. Following the designed vector field, even with error, the controller will keep inserting the string. So, given sufficient time, we expect the process to succeed eventually. However, to minimize the number of misses, certain principles apply in the design of the virtual loops.

A reasonable approach is to make the loops to be some fraction of the size of the small opening. A smaller loop ensures that the the induced magnetic field lines pass very close to the center of the opening; a larger loop allows more gradually-bending field lines.

Notice that in the presence of error, it is easily possible for the string to pass outside of the target loop (miss the target), since some of the field lines pass very close to the boundary of the loop. In order to ensure that the controller does not repeatedly "re-try" the insertion task near the boundary of the loop, we test if the string passes through a larger disk. As long as the small opening is larger than this larger disk, we may in this case consider the insertion to have succeeded. Figure 5 shows such an example; the smaller loop induces the vector field the string follows, while the larger disk is used to detect penetration.

## VI. Re-grasping

There are two reasons that release and re-grasping of the string may be needed during the insertion task. First, one of the primary sources of error in the insertion task is due to the approximate Jacobian. The further the gripper is from the reference points, the greater we expect this error to be.

We compare the expected motion and the actual motion of the reference points at each time step, and let the user select a tolerance, such that if the error of the motion for any reference point exceeds the tolerance, the gripper should then re-grasp closer to the reference points to reduce the error produced by approximate Jacobian.

The second reason for re-grasping is that the gripper cannot itself pass through the small opening, and must release the string and reposition to either push or pull the string completely through the opening after initial contact.

For this task, we choose to consider a simple version of the re-grasping problem. Given a string modeled as a sequence of $n$ links, a gripper may grasp the center of any link.

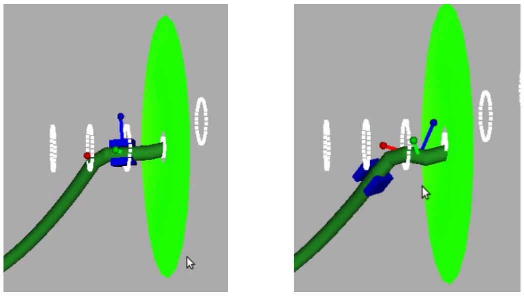


Fig. 6. Re-grasping strategy that considers gravity. Left: The configuration before re-grasp. Right: the configuration after re-grasp.

Since we are only controlling two reference points during the execution in this work, more than one gripper could result in stretching the string, thus violating the constraints; for simplicity, we use a single gripper for the insertion task.

We tried two approaches to find the placement of the gripper and times for re-grasping. First, at each time step, starting from the reference points, find the first link of the string that is accessible and after all of the reference points, and place the gripper there. This strategy results in frequent re-grasping near the obstacle when inserting the string.

The second approach we tried attempts to take advantage of the fact that the string tends to "droop" in the direction of gravity. If the target loop is lying more flat with respect to the direction of gravity, we expect the insertion task to be easier, and can grasp the string farther away, leading to fewer re-grasps. We used the following heuristic to compute the grasp location. We compute the angle between the normal to the target loop and the direction of gravity, $\theta$. We choose the grasp point to be a distance of $l/\theta$ beyond the final reference point, where $l$ is a constant that might be chosen experimentally.

We expect the walls of a physical small opening to have some depth. When should we decide that the string has penetrated the opening? Once the tip of the string has penetrated the target disk, then either the tip may be re-grasped on the other side of the opening, or it may not be (due to occlusion). If the tip is re-graspable, then we do the re-grasp, and declare success. Otherwise, we choose a new first reference point at the intersection of the target disk with the string, and continue pushing the string.

We present the entire online control strategy in Procedure Insertion strategy, which may be used to guide string through one or more small openings. The loop and disk configurations are inputs. We denote the actual motion of the rope as $\tilde{\mathcal{P}}$.

**Procedure** Insertion strategy

Find gripper location using the normal of first target loop and gravity;
**while** *last loop not penetrated* **do**
    Compare the actual motion $\tilde{\mathcal{P}}$ of the reference points and the expected motion $\dot{\mathcal{P}}$;
    **if** $|\dot{\tilde{\mathcal{P}}} - \dot{\mathcal{P}}| >$ *tolerance* **then**
        Re-grasp closer to reference points;
    **if** *string penetrates a target disk* **then**
        Calculate the gripper location $g$ using the normal of next loop and gravity;
        **if** *location $g$ is available* **then**
            Re-grasp at $g$;
            Activate next loop;
            Set the first reference point to be the tip;
        **else if** *Gripper is too close to the obstacle* **then**
            Set the first reference point to be the first point on the string that has not yet penetrated current target disk;
            Find gripper location $g'$ using new normal of active loop;
            Re-grasp $g'$ away from the reference point;
    Recalculate second reference point's location;
    Find the expected motion of reference points $\tilde{\mathcal{P}}$;
    Compute the approximate Jacobian $\tilde{J}$;
    Compute desired motion of string $\dot{\mathcal{P}}$;
    Calculate the motion of gripper $\dot{q}$; apply for $\Delta t$ ;

## VII. Experiments

We first conducted experiments in simulation using the Bullet Physics library [1] and constructed the string using a sequence of capsules (hinged short links). The simulator is used as a black box for our evaluation; no simulation model or parameters were known to our method.

We will first describe several experiments in simulation that we used to test the ability of this method to thread one or more small openings in sequence, with various loop orientations. We also tested our method's ability to deal with noise (added zero-mean Gaussian noise in sensing) and a changing environment (a large move of the target opening during execution) in simulation. Finally, we conducted physical experiments with a Da Vinci surgical robot, to insert yarn into washers of various sizes, with arbitrary initial configuration.

### A. Simulations

Figure 6 shows a close-up of a simulation experiment; white dotted lines show the loops. Figure 7 shows an example of threading string through several openings in sequence, with penetration normals parallel to the x-y plane. Figure 8 shows an example with normals that are not parallel to the x-y plane.

During simulation experiments, we observed that the gripper tends to grasp near the tip during most of execution, leading the tip of the string to behave almost like a rigid body. Even though this strategy is effective for threading purposes, it does not demonstrate the advantage of our approach for deformable objects.

We then enforced that the grasping location be further from the tip during insertion. In such experiments, the string demonstrates clear deformation. The controller was still able to accomplish the insertion task consistently, even though sometimes it missed the opening on several times, perhaps

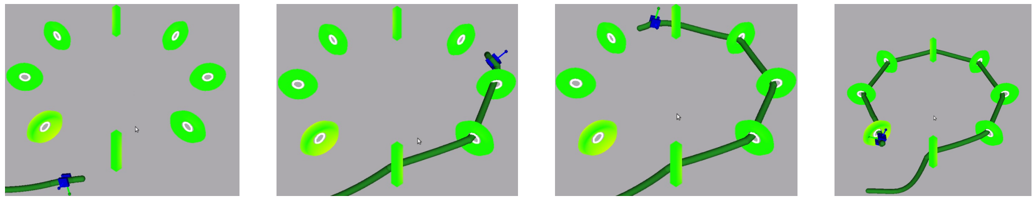


Fig. 7. Simulation results of threading a sequence of loops, Figures 7(a) to 7(d) show several snapshots from the simulation.

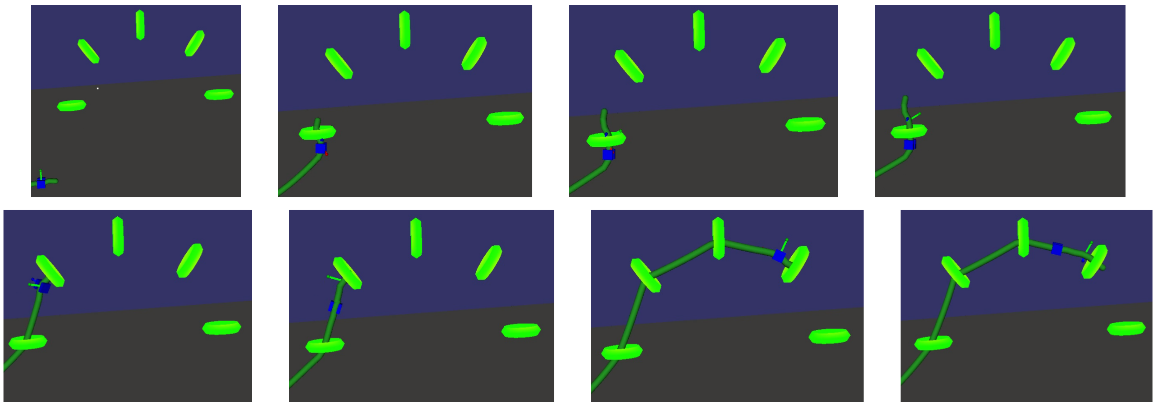


Fig. 8. Simulation results of threading a sequence loops with various ring orientations. Figures 8(b) to 8(d) show the frequent switches near the first loop, due to gravity. The rest of the figures show re-grasping at further and further locations.

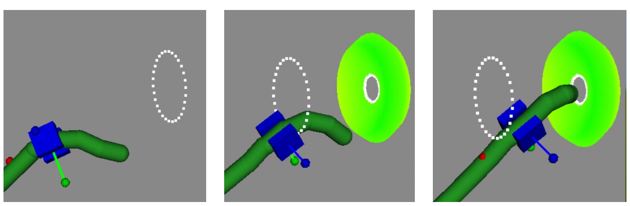


Fig. 9. Simulation results of threading while enforced grasping far from reference points, showing the deformation of string during execution.

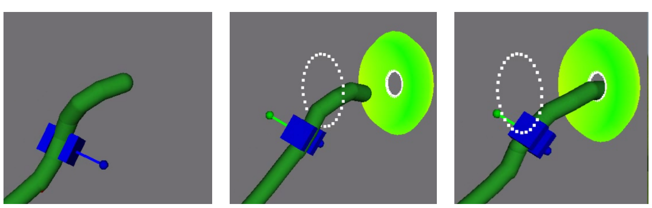


Fig. 10. Simulation results of threading while enforcing grasping even further from reference points, showing more deformation of string during the execution.

due to un-modeled un-simulated string deformation. Some configurations of the string during experiments are shown in Figures 9 and 10.

To test robustness to error, we conducted simulation experiments adding zero-mean Gaussian noise, observed the execution of the task, and time-to-completion for successful insertion with different given noise variances. Average times to completion acquired over 40 trials are shown in Table I. The number of misses was counted by the author watching the simulation. (A failure is reported after 180 seconds without penetrating the target; this only occurs in the presence of extreme noise.) Step size used was 0.05 unit.

Even with sensor noise with 0.3 unit variance applied to every coordinate of the reference points at each time step, our

| noise (variance) | Mean Completion time | completion time std. dev. | Mean # of misses | # Trials Failed |
|---|---|---|---|---|
| 0.01 | 7.82 | 0.39 | 0 | 0/40 |
| 0.03 | 8.03 | 0.41 | 0 | 0/40 |
| 0.05 | 8.92 | 0.44 | 0 | 0/40 |
| 0.1 | 10.24 | 0.37 | 0 | 0/40 |
| 0.3 | 12.47 | 4.36 | 0.025 | 0/40 |
| 0.5 | 14.659 | 5.57 | 0.05 | 0/40 |
| 1 | 29.12 | 16.07 | 0.3 | 0/40 |
| 1.5 | 55.42 | 39.33 | 0.57 | 0/40 |
| 2 | N/A | N/A | N/A | 40/40 |

TABLE I

STATISTICS EVALUATING THE EFFECT OF NOISE ON A SIMULATED INSERTION TASK.

method still succeeded in a mean time of 12.47 seconds. (For comparison, recall that the opening has radius 0.3, and the string has radius 0.2 in our simulation experiments.) As we increased the noise variance, the number of misses increased slowly, but the time to completion increased rapidly. When we added noise with variance 2, the gripper essentially moved randomly and did not penetrate the target.

We also experimented with moving the hole during the execution of the task to study robustness of the approach with respect to unexpected major changes in the environment. Figure 12 shows an example of how the controller followed the vector field after the target was moved.

### B. Experiments with Da Vinci robot

We also conducted physical experiments with one arm of a Da Vinci robot, with seven degrees of freedom. We used two HD webcams to track the location of the string. One webcam was placed over the workspace, while the other was placed facing the $x - z$ plane of the workspace. In the experiments,

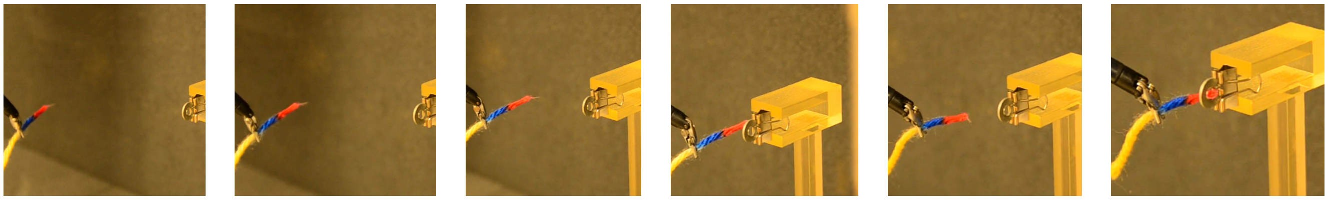


Fig. 11. Experiment using Da Vinci robot with the smallest washer. Washer diameter was 4.9 mm, and the yarn diameter was 3.5 mm. The first three frames show how the controller rotates the gripper to align the yarn with the insertion direction, but the first pass missed the washer, as shown in the fourth frame. Then, the controller made a second attempt (with no intervention) and successfully threaded the washer in the last frame.

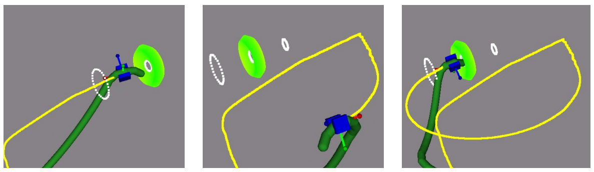


Fig. 12. Example of threading the needle after the target moved during execution (second frame). Gripper trajectory is shown in yellow.

we simply tested the performance of the insertion, without the re-grasping which was shown in previous simulation experiments. The Da Vinci robot is controlled by giving the locations of the end-effector at every time step, and Inverse Kinematics was used to calculate the joint angles.

We used three different washers with different sizes (shown in Figure 13), and attempted to thread a segment of yarn through the washers using the Da Vinci robot. The washer was placed on a stand that is not rigidly attached to the ground; thus, any aggressive motion towards the washer would knock the target over.

The Da Vinci robot successfully threaded all three washers. The inner diameter of the smallest washer was 4.9 mm, while the yarn had diameter 3.5 mm. Images from one of the experiments are shown in Figure 11. The robot successfully threaded even the smallest washer with no more than 4 passes on average.

## VIII. Conclusions and Future Work

In this paper, we presented a method to insert string through tight workspace openings online. We used an approximate Jacobian to estimate the motion of the string and control it. We constructed vector fields based on magnetic fields induced by current-carrying loops, which in some sense encode a measure of distance for insertion tasks.

Empirically, we found that the method has high tolerance to errors and noise. Both simulation and real robot experiments show good results. We also established a baseline criterion for grasping, so the controller can decide to re-grasp during the execution.

As part of this preliminary study, we placed the current-carrying loop used to compute desired velocities of reference points on the string manually. However, such loops could easily be placed automatically.

We hope to extend this approach beyond string. For example, we would like to thread belt-like deformable objects or ribbons through sequences of small openings next. We would also like to tie knots, where the loops formed by strings can

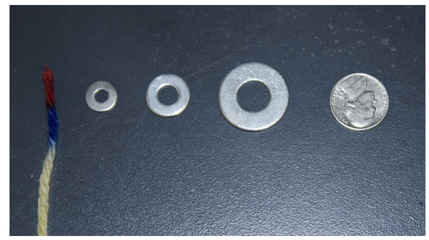


Fig. 13. The three washers used in experiments, the yarn next to a nickel for scale reference.

be thought as the target rings. A primary goal of our future work will be to conduct more thorough experiments with the Da Vinci robot, as well as other robots.

## IX. Acknowledgments

We would like to thank Calder Phillips-Grafflin, Nirav Patel and Adnan Munawar for help with the initial setup of the experiments. We would also like to thank Professor Gregory Fischer for letting us use the Da Vinci robot. This work was supported by NSF grants IIS-1217447 and IIS-1317462, and by the Office of Naval Research under Grant N00014-13-1-0735.

## References

[1] Bullet Physics library. `http://bulletphysics.org/wordpress/`.

[2] Dmitry Berenson. Manipulation of deformable objects without modeling and simulating deformation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4525–4532, Nov 2013.

[3] Subhrajit Bhattacharya, Maxim Likhachev, and V. Kumar. Topological constraints in search-based robot path planning. *Auton. Robots*, 33(3):273–290, 2012.

[4] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Identification and representation of homotopy classes of trajectories for search-based path planning in 3d. In Hugh F. Durrant-Whyte, Nicholas Roy, and Pieter Abbeel, editors, *Robotics: Science and Systems VII, University of Southern California, Los Angeles, CA, USA, June 27-30, 2011*, 2011.

[5] Subhrajit Bhattacharya, David Lipsky, Robert Ghrist, and Vijay Kumar. Invariants for homology classes with application to optimal search and planning problem in robotics. *Ann. Math. Artif. Intell.*, 67(3-4):251–281, 2013.

[6] T. Bretl and Z. McCarthy. Quasi-static manipulation of a Kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *The International Journal of Robotics Research*, 33(1):48–68, June 2013.

[7] Robert R. Burridge, Alfred A. Rizzi, and Daniel E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *I. J. Robotic Res.*, 18(6):534–555, 1999.

[8] Barbara Frank, Cyrill Stachniss, Nichola Abdo, and Wolfram Burgard. Efficient motion planning for manipulation robots in environments with deformable objects. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2180–2185, September 2011.

[9] Barbara Frank, Cyrill Stachniss, Nichola Abdo, and Wolfram Burgard. Efficient motion planning for manipulation robots in environments with deformable objects. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011, San Francisco, CA, USA, September 25-30, 2011*, pages 2180–2185. IEEE, 2011.

[10] Sami Haddadin, Bico Belder, and Alin Albu-Schaeffer. Reactive motion generation for robots in dynamic environments. In *Proceedings. IFAC 2011, World Congress, 28. Aug. - 02. Sep. 2011, Milano, Italy*, 2011.

[11] Dominik Henrich and Heinz Wörn. *Robot manipulation of deformable objects*. Springer, 2000.

[12] S. Hirai and T. Wada. Indirect simultaneous positioning of deformable objects with multi-pinching fingers based on an uncertain model. *Robotica*, 18(1):3–11, January 2000.

[13] Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and Baoquan Chen. L1-medial skeleton of point cloud. *ACM Trans. Graph.*, 32(4):65:1–65:8, July 2013.

[14] Masayuki Inaba and Hirochika Inoue. Hand eye coordination in rope handling. *Journal of the Robotics Society of Japan*, 3(6):538–547, 1985.

[15] Hyosig Kang and J.T. Wen. Robotic assistants aid surgeons during minimally invasive procedures. *Engineering in Medicine and Biology Magazine, IEEE*, 20(1):94–104, Jan 2001.

[16] F Khalil and P Payeur. Dexterous robotic manipulation of deformable objects with multi-sensory feedback – a review. In In-Teh, editor, *Robot Manipulators, Trends and Development*, chapter 28, pages 587–621. 2010.

[17] Fouad F. Khalil and Pieere Payeur. Dexterous robotic manipulation of deformable objects with multi-sensory feedback - a review. *Robot Manipulators, Trends and Develpments*, In-Teh. Ed. 2010, ch. 28:587–621.

[18] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *Robotics and Automation, IEEE Journal of*, 3(1):43–53, February 1987.

[19] F. Lamiraux and Lydia E. Kavraki. Planning Paths for Elastic Objects under Manipulation Constraints. *The International Journal of Robotics Research*, 20(3):188–208, March 2001.

[20] Mark Moll and Lydia E Kavraki. Path Planning for Deformable Linear Objects. *IEEE Transactions on Robotics*, 22(4):625–636, 2006.

[21] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi. Knot planning from observation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[22] D. Navarro-Alarcon, Y. Liu, J.G. Romero, and P. Li. Visually Servoed Deformation Control by Robot Manipulators. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[23] Sachin Patil, J van den Berg, and Ron Alterovitz. Motion planning under uncertainty in highly deformable environments. In *Robotics: Science and Systems*, 2011.

[24] Matthias Rambow, Thomas Schauß, Martin Buss, and Sandra Hirche. Autonomous Manipulation of Deformable Objects based on Teleoperated Demonstrations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[25] S. Rodriguez and N.M. Amato. An obstacle-based rapidly-exploring random tree. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[26] Samuel Rodríguez, Jyh-Ming Lien, and Nancy M. Amato. Planning motion in completely deformable environments. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, May 15-19, 2006, Orlando, Florida, USA*, pages 2466–2471. IEEE, 2006.

[27] M. Saha, P. Isto, and J.-C. Latombe. Motion planning for robotic manipulation of deformable linear objects. In *Proc. International Symposium On Experimental Robotics (ISER)*, 2006.

[28] Mitul Saha, Pekka Isto, and Jean-Claude Latombe. Motion planning for robotic manipulation of deformable linear objects. In Oussama Khatib, Vijay Kumar, and Daniela Rus, editors, *Experimental Robotics, The 10th International Symposium on Experimental Robotics [ISER '06, July 6-10, 2006, Rio de Janeiro, Brazil]*, volume 39 of *Springer Tracts in Advanced Robotics*, pages 23–32. Springer, 2006.

[29] John Schulman, Ankush Gupta, Sibi Venkatesan, Mallory Tayson-Frederick, and Pieter Abbeel. A case study of trajectory transfer through non-rigid registration for a simplified suturing scenario. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013.

[30] John Schulman, Jonathan Ho, Cameron Lee, and Pieter Abbeel. Learning from Demonstrations Through the Use of Non-Rigid Registration. In *International Symposium on Robotics Research (ISRR)*, 2013.

[31] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and Control of Robot Manipulators*. Springer, January 2000.

[32] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *I. J. Humanoid Robotics*, 2(4):505–518, 2005.

[33] Jerzy Smolen and Alexandru Patriciu. Deformation Planning for Robotic Soft Tissue Manipulation. In *2009 Second International Conferences on Advances in Computer-Human Interactions*, pages 199–204, February 2009.

[34] Russ Tedrake. Lqr-trees: Feedback motion planning on sparse randomized trees. In Jeff Trinkle, Yoky Matsuoka, and José A. Castellanos, editors, *Robotics: Science and Systems V, University of Washington, Seattle, USA, June 28 - July 1, 2009*. The MIT Press, 2009.

[35] Russ Tedrake, Ian R. Manchester, Mark M. Tobenkin, and John W. Roberts. Lqr-trees: Feedback motion planning via sums-of-squares verification. *I. J. Robotic Res.*, 29(8):1038–1052, 2010.

[36] T Wada, S Hirai, S Kawarnura, and N Karniji. Robust manipulation of deformable objects by a simple PID feedback. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[37] Eiichi Yoshida, Mathieu Poirier, Jean-Paul Laumond, Oussama Kanoun, Florent Lamiraux, Rachid Alami, and Kazuhito Yokoi. Regrasp planning for pivoting manipulation by a humanoid robot. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, pages 2467–2472. IEEE, 2009.