

Learning Force-Based Manipulation of Deformable Objects from Multiple Demonstrations

Alex X. Lee

Henry Lu

Abhishek Gupta

Sergey Levine

Pieter Abbeel

Abstract—Manipulation of deformable objects often requires a robot to apply specific forces to bring the object into the desired configuration. For instance, tightening a knot requires pulling on the ends, flattening an article of clothing requires smoothing out wrinkles, and erasing a whiteboard requires applying downward pressure. We present a method for learning force-based manipulation skills from demonstrations. Our approach uses non-rigid registration to compute a warping function that transforms both the end-effector poses and forces in each demonstration into the current scene, based on the configuration of the object. Our method then uses the variation between the demonstrations to extract a single trajectory, along with time-varying feedback gains that determine how much to match poses or forces. This results in a learned variable-impedance control strategy that trades off force and position errors, providing for the right level of compliance that applies the necessary forces at each stage of the motion. We evaluate our approach by tying knots in rope, flattening towels, and erasing a whiteboard.

I. INTRODUCTION

In order to manipulate soft and deformable objects, it is often necessary for a robot to exert a particular force to bring the object into the desired configuration: tightening a knot requires pulling on the ends, flattening an article of clothing requires smoothing out wrinkles, and erasing a whiteboard requires applying downward pressure. Learning from demonstration has proven to be a successful approach for manipulating deformable objects [1], but transferring forces from a demonstration to a new situation poses some unique challenges: when there is a mismatch between the test and training situations, how does the robot decide how to trade off errors in position versus force?

In this paper, we present a method that combines geometric warping with statistical learning to compute target poses, forces, and time-varying gains for a new situation (e.g., a new arrangement of a rope that must be tied) by combining multiple demonstrations. We begin by gathering a number of demonstrations of the desired task, using either teleoperation or kinesthetic teaching. At runtime, these demonstrations are adapted to the current situation. This adaptation is performed by registering the initial state of the object from each of the demonstrations to the current scene, computing warping functions that map points from each demonstration into the current scene, and then applying this warp to each demonstrated trajectory. This corresponds to an *object-centric* warping of the demonstrations.

Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA. {alexlee-gk, armentai, abhigupta, sergey.levine, pabbeel}@berkeley.edu

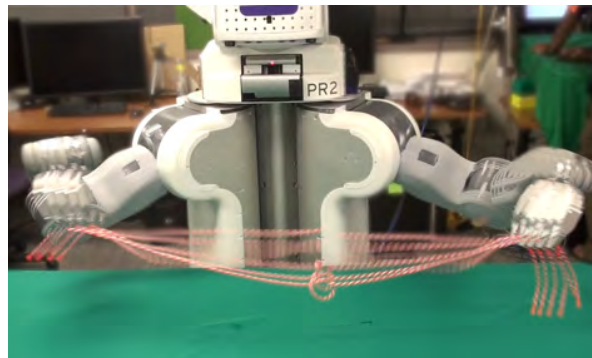


Fig. 1: PR2 applying strong lateral forces to tighten a knot.

Along with the geometric properties of the trajectory (positions and orientations of the end-effector), we also warp the demonstrated end-effector forces. Naïvely playing back these forces during execution is generally ineffective, as shown in our experiments, since the robot must make tradeoffs between position and force errors. For example, when attempting to tighten a knot in a rope that is longer than in the demonstrations, the robot cannot simultaneously exert the same force and maintain the same gripper position as in the (warped) demonstrations, since the rope will still have slack, and applying a force will push the grippers out of the desired pose. The key to successfully generalizing behaviors that involve both position-centric and force-driven stages is to learn the tradeoffs between force and position from the data. To this end, we present an algorithm that can be used to quickly fit time-varying feedback gains to the warped demonstrations. Since all of the demonstrations are aligned to the current scene using geometric warping, the variance in the demonstrations can be used to estimate the relative importance of maintaining position versus exerting force without concern for variations in the object configuration, which have been removed by the warp.

The main contribution of this paper is a method for generalizing demonstrations of manipulation behaviors that include both kinematic and force-driven phases. Our method first warps the demonstrated pose and force trajectories to align them with the current object configuration, and then extracts a single trajectory along with time-varying gains that trade off pose and force errors in a way that is consistent with the demonstrations. This learned variable-impedance controller is then used to perform the task in the new situation. Our experimental results show that this method is effective for tightening knots (see Figure 1), flattening and folding crumpled towels, and erasing a whiteboard.

II. RELATED WORK

Learning from demonstrations is an effective and practical approach for training robots to perform complex manipulation skills [2]. Manipulation of deformable objects with learning from demonstrations has in particular received considerable attention in recent years. Our approach builds on the work of Schulman et al. [1], which uses non-rigid registration of a point cloud of the manipulated object at the beginning of each demonstration to the current scene in order to warp the demonstrated trajectory to match the current setting. Other methods for adapting demonstrated trajectories to the current scene make use of task-specific features and Gaussian mixture models [3], [4]. The non-rigid registration approach has the advantage of requiring only a simple, task-agnostic perception pipeline.

Our main contribution is to combine learning from demonstration via non-rigid registration with force-based control strategies. Force control has long been recognized as a powerful tool for executing complex robotic motion skills [5]. A number of previous works have addressed learning variable-impedance control policies using reinforcement learning, which uses a reward function and trial-and-error [6], [7], [8]. Although such methods are highly automated, they typically require extensive system interaction during learning, and address simpler behaviors. Specifying a reward function for a complex compound behavior, such as tying a knot, poses a considerable challenge.

Previous methods that learn force-based behaviors directly from demonstrations typically use a least squares formulation to solve for time-varying impedances, either at each point in the demonstration or in the context of a Gaussian mixture model [9], [10], [11], [12], [13], [14]. The input to this least squares formulation is typically a temporal window over which the demonstration is assumed to be tracking the same setpoint. Some authors have also proposed intentionally introducing perturbations into the demonstrations to observe the human demonstrator’s recovery stiffness [13], as well as more sophisticated input modalities that make it easier for the demonstrator to directly specify the desired stiffness [15], [14]. Common applications for such techniques are tasks that require variable stiffness, such as assembling furniture [12] and ironing [11]. Our approach for fitting the feedback gains is based on a statistical model of a collection of demonstrations. It is mathematically similar to the least squares formulation, though we introduce several improvements, such as the use of a Bayesian prior. A key distinction between our approach and previous work is that the gains are fitted after non-rigid registration has been used to align all of the demonstrations relative to the object. This allows us to extract object-centric gains that indicate, for example, that the end-effector should be positioned very precisely during a grasp (corresponding to high position gains), even if each demonstration grasps the object at a different position in Cartesian space, because the point on the object is consistent. Unlike other object-centric approaches [12], our method does not require any task-specific features,

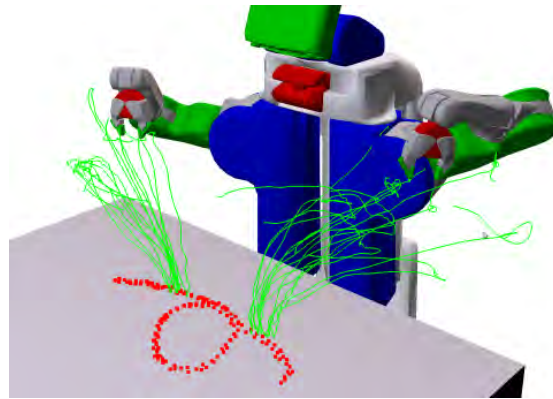


Fig. 2: Visualization of a collection of trajectories for knot tightening (the third stage of the knot tie) warped to match the current rope point cloud (shown in red). Note that all of the trajectories grasp the rope at roughly the same point, but diverge in different directions when tightening the knot.

operating directly on point clouds. Our learned feedback gains span a very large dynamic range, from high gains during grasping to gains that approach zero during force-driven parts of the task, such as the pull that is necessary to tighten a knot. Our method is also applied on a robot that lacks force sensors, using only measurements of its internal torques, which introduces unique challenges during the demonstration procedure.

We apply our method to manipulation of deformable objects, including rope and towels. Manipulation of rope, including knot tying, has previously been addressed in a task-specific manner by exploiting domain knowledge of the task [16], [17], as well as with motion planning techniques [18], [19]. Previous work has also proposed force-based methods for tying knots using hand-engineered manipulation strategies [20]. Cloth manipulation, including laundry folding, has similarly been addressed largely by task-specific methods, often focusing on accurate perception [21], [22] and grasp planning [23]. Unlike these methods, our approach is not specific to rope or cloth, and learns the manipulation strategy entirely from demonstrations.

III. OVERVIEW

For each manipulation behavior, we begin by recording a set of demonstrations for each stage of the task. The knot tying task is divided into three stages: first loop, second loop, and tightening the knot. Towel folding consists of two stages: straightening and folding. The erasing task consists of a single stage. We found that 5 demonstrations were sufficient for each stage to achieve adequate generalization, though larger libraries of demonstrations could further improve generalization. The demonstrations are stored along with a snapshot of the point cloud of the scene at the beginning of each stage, recorded with a Kinect depth sensor.

At test time, we use the thin plate spline robust point matching (TPS-RPM) algorithm [24] to register the initial point clouds from each of the demonstrations to the current scene. A color-based filter is used to mask out points that do

not belong to the object, though any image segmentation method could be used for this purpose. The TPS-RPM algorithm produces a warping function $\mathbf{y}_i = f(\mathbf{x}_i)$ that maps each point in the demonstration to a corresponding point in the current scene. We use this function to transform the corresponding demonstrated trajectory into the current scene, along with the demonstrated forces, which are warped by the derivative of $f(\mathbf{x}_i)$. A separate warp is computed for each demonstration, producing a collection of force-position trajectories that are aligned to the current scene. An image of one such collection is shown in Figure 2.

In addition to warping each of the demonstrations in space to match the current scene, we also warp the demonstrations in time to match one another. This is done by using dynamic time warping [25]. Once the trajectories are aligned in time and space, we extract a single mean trajectory that captures a denoised execution of the task, reducing the impact of accidental variations across demonstrations. We further analyze the demonstrations to also extract their covariance at each time step. This covariance is used to estimate the position and velocity feedback gains, which determine how much the robot prioritizes the kinematics of the task over matching the demonstrated forces. Intuitively, parts of the behavior where the position varies little between trajectories after registration, such as the point at which the robot grasps the rope, should track the demonstrated positions accurately, while parts where the position varies more should instead track the demonstrated forces.

Once we have extracted a mean trajectory and a sequence of time-varying feedback gains, we can execute the resulting trajectory with the learned gains to perform the manipulation behavior. The warped, averaged forces from the demonstrations are applied using the Jacobian transpose method, while the learned feedback gains are used to track the kinematic trajectory. When the learned gains are low, the force term dominates the behavior, regardless of the spatial discrepancy between the current and demonstrated pose. A diagram of our approach is shown in Figure 3, and each of the stages are described in detail in the following sections.

IV. BACKGROUND

In this section, we review the TPS-RPM algorithm for non-rigid point cloud registration [24], as well as the approach of Schulman et al. [1] for warping the demonstrated trajectories using the TPS warping function. The goal behind this method is to transform the demonstrated trajectories in an object-centric way without requiring any task-specific features, and the key idea is that the demonstrated trajectory should be warped in the same way that the demonstration object should be deformed to bring it into correspondence with the current scene. This has the effect of preserving the relationship between the end-effector motion and the object, which is crucial for many manipulation behaviors.

A. Non-rigid Registration

We assume that the demonstration scene consists of N points $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_1 \dots \hat{\mathbf{p}}_N]^\top$ and the test scene consists of N'

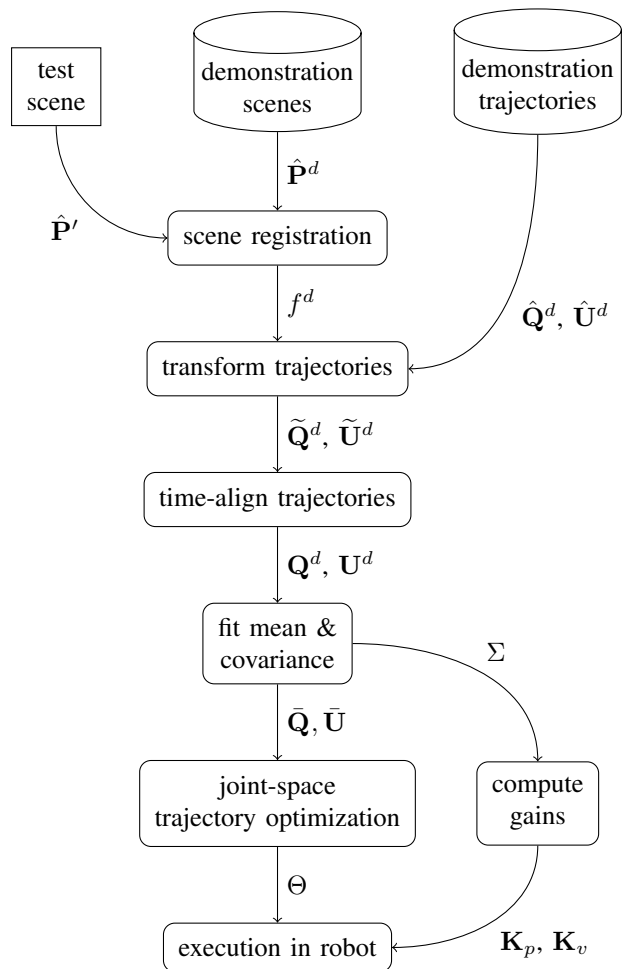


Fig. 3: Diagram of our approach. $\hat{\mathbf{P}}^d$ and $\hat{\mathbf{P}}'$ are points in the demonstration and test scenes, f^d is the TPS-RPM registration function, $\hat{\mathbf{Q}}^d$ and $\hat{\mathbf{U}}^d$ are the demonstrated poses and forces, $\tilde{\mathbf{Q}}^d$ and $\tilde{\mathbf{U}}^d$ are transformed by f^d , \mathbf{Q}^d and \mathbf{U}^d are further registered in time, $\bar{\mathbf{Q}}$ and $\bar{\mathbf{U}}$ denote the mean trajectory, and Θ is the final sequence of joint angles. \mathbf{K}_p and \mathbf{K}_v are the computed position and velocity gains. Note that all fitting is performed at test time, after each of the demonstrations are aligned with the current scene.

points $\hat{\mathbf{P}}' = [\hat{\mathbf{p}}'_1 \dots \hat{\mathbf{p}}'_{N'}]^\top$. We first formalize problem for the case when $N = N'$ and the correspondences between the points are known. The non-rigid registration problem is to find a function $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that maps the points $\hat{\mathbf{P}}$ to the points $\hat{\mathbf{P}}'$. To find a smooth mapping, we restrict f to be a thin plate spline [26], which is defined as the optimal solution f for the following optimization problem:

$$f = \arg \min_f \sum_{i=1}^N \|\hat{\mathbf{p}}'_i - f(\hat{\mathbf{p}}_i)\|_2^2 + \lambda \|f\|_{\text{TPS}}^2, \quad (1)$$

where the regularizer $\|f\|_{\text{TPS}}^2$ is the TPS energy function

$$\|f\|_{\text{TPS}}^2 = \int d\mathbf{p} \|D^2 f(\mathbf{p})\|_F^2.$$

This regularizer is a measure of the curvature of f , and encourages the mapping to be as smooth as possible, while λ

controls the trade-off between matching the correspondence points and the smoothness of the spline. The optimal f can be found analytically [27], and has the form

$$f(\mathbf{p}) = \sum_i \mathbf{a}_i k(\hat{\mathbf{p}}_i, \mathbf{p}) + \mathbf{B}\mathbf{p} + \mathbf{c}.$$

This corresponds to an affine transformation defined by \mathbf{B} and \mathbf{c} , plus a weighted sum of basis functions, which have the form $k(\hat{\mathbf{p}}_i, \mathbf{p}) = -\|\mathbf{p} - \hat{\mathbf{p}}_i\|^2$ in 3D. When one-to-one correspondences between the points are not available initially and $N \neq N'$, the thin plate spline robust point matching (TPS-RPM) algorithm [24] can be used to register the two scenes. This method alternates between choosing correspondences and minimizing a weighted version of Equation 1.

B. Learning from Demonstrations with Thin Plate Splines

A demonstration consists of a point cloud $\hat{\mathbf{P}}$ of the initial scene, which shows, for example, the initial arrangement of the rope or towel, and a sequence of end-effector poses $\hat{\mathbf{Q}} = [\hat{\mathbf{q}}_1 \dots \hat{\mathbf{q}}_T]^\top$. In our work, this demonstration is also augmented with a sequence of end-effector forces $\hat{\mathbf{U}} = [\hat{\mathbf{u}}_1 \dots \hat{\mathbf{u}}_T]^\top$, which we discuss in Section V. At test time, a point cloud $\hat{\mathbf{P}}'$ of the test scene is observed, and the TPS-RPM algorithm is used to find a warping function f that maps points from the demonstration scene to the test scene. This warp function is applied to the end-effector trajectory to obtain a warped trajectory $\tilde{\mathbf{Q}}$. This trajectory does not incorporate collision avoidance and joint limits, so trajectory optimization [28] is then used to find a feasible joint angle trajectory $\Theta = [\theta_1 \dots \theta_T]^\top$.

V. ALIGNING FORCE-AUGMENTED TRAJECTORIES

In this work, we use multiple demonstrations augmented with forces to obtain a single, denoised trajectory with force terms and variable impedances. This requires transforming force-augmented trajectories in both space and time in order to align them spatially to the current scene and temporally to each other.

A. Warping Force Trajectories

Each demonstration trajectory is augmented with a sequence of end-effector forces $\hat{\mathbf{U}} = [\hat{\mathbf{u}}_1 \dots \hat{\mathbf{u}}_T]^\top$. Each force $\hat{\mathbf{u}}_t$ is composed of a translational force and a torque. As before, the registration function f is applied to the trajectory. Since forces are vector quantities, they must be transformed by the derivative of f ,

$$\tilde{\mathbf{u}}_t = \frac{df}{d\mathbf{p}}(\hat{\mathbf{q}}_{\mathbf{p},t})\hat{\mathbf{u}}_t,$$

where $\frac{df}{d\mathbf{p}}(\hat{\mathbf{q}}_{\mathbf{p},t})$ is the derivative of f at $\hat{\mathbf{q}}_{\mathbf{p},t}$, and $\hat{\mathbf{q}}_{\mathbf{p},t}$ is the translational component of the end-effector pose $\hat{\mathbf{q}}_t$. This transformation rotates the forces to align them with the current test scene.

B. Aligning Multiple Demonstrations

In order to obtain a single denoised trajectory and appropriate gains, we use a collection of demonstrations $\mathcal{D} = [\mathcal{D}^1 \dots \mathcal{D}^D]$. Each demonstration \mathcal{D}^d consists of the scene points $\hat{\mathbf{P}}^d$, a pose trajectory $\hat{\mathbf{Q}}^d$, and a force trajectory $\hat{\mathbf{U}}^d$. For each demonstration \mathcal{D}^d , we first compute a registration function f^d that maps points from the demonstration scene $\hat{\mathbf{P}}^d$ to the test scene $\hat{\mathbf{P}}'$, as described in Section IV. We then apply the registration function to each demonstration trajectory to get $\tilde{\mathbf{Q}}^d, \tilde{\mathbf{U}}^d$:

$$\begin{aligned} \tilde{\mathbf{q}}_t^d &= f^d(\hat{\mathbf{q}}_t^d) \\ \tilde{\mathbf{u}}_t^d &= \frac{df^d}{d\mathbf{p}}(\hat{\mathbf{q}}_{\mathbf{p},t}^d)\hat{\mathbf{u}}_t^d. \end{aligned}$$

After this transformation, the trajectories $\tilde{\mathbf{Q}}^d$ and $\tilde{\mathbf{U}}^d$ are spatially aligned with respect to the test scene, but they may differ in length and are not aligned in time, which makes analyzing the variation across trajectories problematic. Let the first demonstration \mathcal{D}^1 be the one that is closest to the test scene in terms of the registration cost in Equation 1. We use dynamic time warping [25] to time-align the trajectories with respect to the trajectory of demonstration \mathcal{D}^1 , giving us a time index τ_t^d that aligns the trajectories of each demonstration. The corresponding time-aligned trajectories are then given by $\mathbf{q}_t^d = \tilde{\mathbf{q}}_{\tau_t^d}^d$ and $\mathbf{u}_t^d = \tilde{\mathbf{u}}_{\tau_t^d}^d$. We must also obtain a sequence of velocities $\dot{\mathbf{Q}}^d$ for each trajectory. These can be obtained analogously to forces, by transforming the demonstrated velocities by the derivative of f^d , or by performing finite differences on the transformed, time-aligned trajectories \mathbf{Q}^d . The latter approach, which we use in our implementation, has the advantage that no additional bookkeeping is necessary to compensate for the time warp.

VI. LEARNING VARIABLE-IMPEDANCE CONTROL FROM MULTIPLE DEMONSTRATIONS

Once all of the demonstrations for the current task have been aligned in both space and time, we use them to extract a single underlying trajectory consisting of poses, velocities, and forces, along with time-varying feedback gains \mathbf{K}_{pt} and \mathbf{K}_{vt} that determine how much the controller prioritizes correcting position and velocity errors over matching the demonstrated force. Both of these operations are performed by analyzing the mean and covariance of the poses, velocities, and forces in the warped and aligned trajectories.

A. Probabilistic Modeling

We model the demonstrations as arising from a time-varying linear-Gaussian control law, which determines the force to exert at the end-effector as a linear function of the poses and velocities, corrupted by Gaussian noise. The probability of observing a force \mathbf{u}_t^d is therefore modeled as

$$p(\mathbf{u}_t^d | \mathbf{q}_t^d, \dot{\mathbf{q}}_t^d) = \mathcal{N}(\mathbf{K}_{pt}(\bar{\mathbf{q}}_t - \mathbf{q}_t^d) + \mathbf{K}_{vt}(\dot{\bar{\mathbf{q}}}_t - \dot{\mathbf{q}}_t^d) + \bar{\mathbf{u}}_t; \mathbf{C}_t), \quad (2)$$

where $\bar{\mathbf{q}}_t$, $\dot{\bar{\mathbf{q}}}_t$, and $\bar{\mathbf{u}}_t$ are the position, velocity, and force along the estimated desired trajectory. Note that the mean of this linear-Gaussian controller corresponds to the usual form

of task-space impedance control. The maximum likelihood settings for the parameters $\bar{\mathbf{q}}_t$, $\dot{\bar{\mathbf{q}}}_t$, $\bar{\mathbf{u}}_t$, \mathbf{K}_{pt} , and \mathbf{K}_{vt} are obtained by maximizing the probability of each of the demonstrated trajectories, which can be done by fitting a joint Gaussian distribution to $\{\mathbf{q}_t^d, \dot{\mathbf{q}}_t^d, \mathbf{u}_t^d\}$ at each time step, using the mean to obtain $\bar{\mathbf{q}}_t$, $\dot{\bar{\mathbf{q}}}_t$, and $\bar{\mathbf{u}}_t$, and conditioning on \mathbf{q}_t and $\dot{\mathbf{q}}_t$ to obtain the feedback gains \mathbf{K}_{pt} and \mathbf{K}_{vt} . Formally, if the mean and covariance of $\{(\mathbf{q}_t^d, \dot{\mathbf{q}}_t^d, \mathbf{u}_t^d)^\top\}$ are given by μ_t and Σ_t , respectively, the parameters are:

$$\begin{aligned}\bar{\mathbf{q}}_t &= \mu_{\mathbf{q},t} & \mathbf{K}_{pt} &= -\Sigma_{\mathbf{uq},t}\Sigma_{\mathbf{qq},t}^{-1} \\ \dot{\bar{\mathbf{q}}}_t &= \mu_{\dot{\mathbf{q}},t} & \mathbf{K}_{vt} &= -\Sigma_{\mathbf{u}\dot{\mathbf{q}},t}\Sigma_{\dot{\mathbf{q}}\dot{\mathbf{q}},t}^{-1} \\ \bar{\mathbf{u}}_t &= \mu_{\mathbf{u},t},\end{aligned}$$

where subscripts indicate submatrices (e.g., $\mu_{\mathbf{q},t}$ is the top portion of μ_t , and $\Sigma_{\mathbf{qq},t}$ is the top-left submatrix of Σ_t). Intuitively, this method for recovering variable impedance parameters corresponds to assigning greater priority to minimizing position errors when the positions are consistent across the demonstrations, such as during a grasp. When the positions have high variance and do not correlate with force, the corresponding gains will be low, and the force term $\bar{\mathbf{u}}_t$ will dominate. This is similar to least-squares methods proposed in prior work for learning variable-impedance motions from demonstrations [9], [10], [13].

However, directly fitting the covariance to the samples at each time step often does not produce suitable gains, for two reasons. First, when the demonstrated trajectories diverge in different directions, such as after grasping an object very precisely, the learned gains will be negative, which will create an unstable controller. Second, since we use 5 demonstrations for each task, the number of samples is inadequate to accurately fit the covariance at each time step independently. To address the former issue, we take a Bayesian approach and impose a prior on the covariance Σ_t . To address the latter problem, we use additional samples from adjacent time steps with a weighted fitting scheme, under the assumption that the gains vary slowly over time.

B. Feedback Gain Priors

A standard approach to imposing a prior on the covariance of a Gaussian distribution is to use an inverse-Wishart distribution [29], which is determined by two parameters: the degrees of freedom parameter ν , and the prior matrix Ψ . The maximum a posteriori (MAP) estimate of the covariance Σ_t under this prior is given by

$$\Sigma_t = \frac{D\hat{\Sigma}_t + \Psi}{\nu + D + P + 1},$$

where P is the dimensionality of the data points (18 in task space), and $\hat{\Sigma}_t$ is the empirical covariance obtained in the usual fashion. The MAP covariance is essentially a weighted combination of $\hat{\Sigma}_t$ and Ψ . We set $\nu = P + 2$ in our implementation (a standard default choice). The prior matrix Ψ is often set to identity, but since we would like to use it to encode the prior belief that the feedback gains

should be positive, we instead construct it as following:

$$\begin{aligned}\Psi_{\mathbf{q}\mathbf{q}} &= \text{Cov}(\mathbf{Q})w_p & \Psi_{\mathbf{u}\mathbf{q}} &= -\mathbf{K}_p^0\Psi_{\mathbf{q}\mathbf{q}} \\ \Psi_{\dot{\mathbf{q}}\dot{\mathbf{q}}} &= \text{Cov}(\dot{\mathbf{Q}})w_v & \Psi_{\mathbf{u}\dot{\mathbf{q}}} &= -\mathbf{K}_v^0\Psi_{\dot{\mathbf{q}}\dot{\mathbf{q}}},\end{aligned}$$

where $\text{Cov}(\mathbf{Q})$ denotes the empirical covariance of the warped poses (to determine a global scale), while w_p and w_v are weights that determine the relative importance of the prior gains for poses and velocities, respectively. Note that $\Psi_{\mathbf{u}\mathbf{u}}$ is irrelevant for determining the gains. The prior gains \mathbf{K}_p^0 and \mathbf{K}_v^0 are set to 100 and 20, respectively, which is on the same order as the default PD gains on a PR2 robot, while the weights are set to $w_p = 0.1$ and $w_v = 10.0$ since, for low-velocity behaviors, we are more interested in fitting the position gains to data, while the velocity gains can reasonably be set by the prior at most times. At test time, the prior keeps most gains positive, and we clip any negative gains to zero. A more sophisticated method based on non-negative least squares can also be used, though we found that it produced no improvement over simple clipping.

C. Temporal Windowing

To increase the number of samples available for fitting the covariances, we include samples from nearby time steps when computing the empirical covariance $\hat{\Sigma}_t$. This is done by using a squared exponential weight on each sample based on that sample's time stamp τ : $w_\tau = \exp(-\frac{1}{2\sigma^2}(\tau - t)^2)$. We used $\sigma = 1.0$ in our implementation (corresponding to 1 second), since we found that smoother gains that incorporate information from a wide temporal window produced better results. Note that this windowing was only used for computing $\hat{\Sigma}_t$, not the mean μ_t , in order to preserve high-frequency details in the trajectory.

We also considered each of the six degrees of freedom of the end effector in isolation, without computing covariances between different degrees of freedom. This made the results easier to analyze, but prevented our approach from learning variable stiffness along non-axis-aligned directions. It would be straightforward to extend our method to use full covariances, and we plan to experiment with this in the future.

D. Joint-Space Control

Once we compute the mean end-effector trajectory and feedback gains \mathbf{K}_{pt} and \mathbf{K}_{vt} , we need to execute this trajectory on the robot. Directly following the mean of Equation 2 using, for example, Jacobian transpose control could lead the robot into kinematic singularities. Instead, we first jointly optimize a sequence of joint angles $\Theta = [\theta_1 \dots \theta_T]^\top$ to match the desired end-effector trajectory while conforming to kinematic constraints, as in prior work [1], [28]. This joint-space trajectory can then be tracked with variable impedance by converting the gains \mathbf{K}_{pt} and \mathbf{K}_{vt} into joint space gains \mathbf{K}_{pt}^θ and \mathbf{K}_{vt}^θ , using the Jacobian at θ_t :

$$\begin{aligned}\mathbf{K}_{pt}^\theta &= \mathbf{J}(\theta_t)^\top \mathbf{K}_{pt} \mathbf{J}(\theta_t) + \mathbf{K}_{p,\text{pd}}^\theta \\ \mathbf{K}_{vt}^\theta &= \mathbf{J}(\theta_t)^\top \mathbf{K}_{vt} \mathbf{J}(\theta_t) + \mathbf{K}_{v,\text{pd}}^\theta,\end{aligned}$$

where $\mathbf{K}_{p,\text{pd}}^\theta$ and $\mathbf{K}_{v,\text{pd}}^\theta$ are low PD gains (set to 1% of the default values) that ensure that the robot keeps moving along

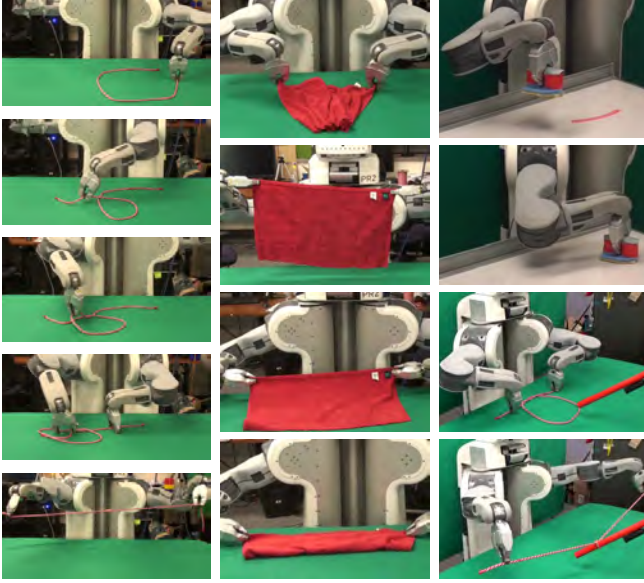


Fig. 4: Images of the tasks (progress top to bottom) in our experiments: tying a knot (left), folding a towel (middle), erasing a whiteboard (top right), and tying a rope to a pipe (bottom right). For videos, see <http://rll.berkeley.edu/icra20151fmd/>

the trajectory along directions that lie in the null space of $\mathbf{J}(\theta_t)$. In future work, these matrices could be projected into the null space of $\mathbf{J}(\theta_t)$ directly for more compliant behavior. Using the joint-space gains \mathbf{K}_{pt}^θ and \mathbf{K}_{vt}^θ , the torque applied at the joints at test time is given by

$$\mathbf{f}_t = \mathbf{K}_{pt}^\theta(\theta_t - \theta_{\text{obs}}) + \mathbf{K}_{vt}^\theta(\dot{\theta}_t - \dot{\theta}_{\text{obs}}) + \mathbf{J}(\theta)^\top \bar{\mathbf{u}}_t,$$

where θ_{obs} and $\dot{\theta}_{\text{obs}}$ are the observed joint angles and velocities, and \mathbf{f}_t is the torque to apply at each joint.

VII. EXPERIMENTAL RESULTS

We evaluated our method on four tasks: tying and tightening a knot, tying a rope to a pipe, flattening and folding a towel, and erasing a whiteboard. In each case, we generated 5 demonstrations for each stage of the task. The complete knot tie consisted of three stages, pipe tying consisted of one, towel folding consisted of two, and erasing consisted of a single stage. For erasing, registration was done directly on the red marks on the board, in order to erase the right regions. On each task, we compared our method to a baseline that tracked the warped trajectory with a standard high-gain PD controller, which matches the setup in prior work [1]. For the tightening phase of the knot task, we also evaluated an ablated version of our method that included the force term, but used the default proportional-derivative gains of the PR2 robot, in order to demonstrate the importance of learning time-varying feedback gains.

Since our PR2 robot cannot sense the actual force and torque on the end-effector, we computed the end-effector forces from the torques exerted at the joints by the motors during the demonstration. This method is straightforward to use with teleoperation, but presents a problem for kinesthetic

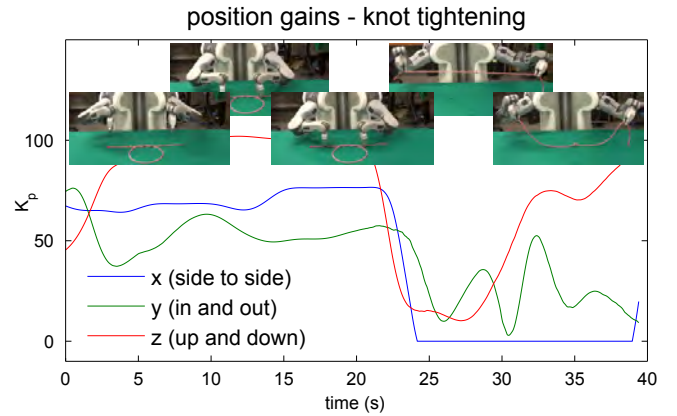


Fig. 5: Position gains learned for knot tightening. Images of the rope at the corresponding times shown above the plot. Note that the lateral gains drop to zero during tightening, when the behavior is dominated by the pulling force.

teaching, where the external force of the teacher on the joints is not registered by the motors. To generate demonstrations with kinesthetic teaching, we simply had the robot replay the demonstration in an identical situation kinematically, using high position gains to match the demonstrated trajectory and recording the resulting forces. This method proved sufficient for the tasks we tested, though demonstrating subtle variations in force was difficult.

Videos of the robot performing each of the tasks using our method and each of the baselines are included in the supplementary materials,¹ and images are provided in Figure 4. We conducted a quantitative evaluation on the tightening stage of the knot task to determine how well our method, the kinematic baseline, and the ablated variant could generalize to ropes of varying length. Purely kinematic execution of the warped trajectory will often fail to tighten the knot fully, especially when the rope at test time is longer than the one used in the demonstrations, since the arms must move further apart for longer ropes. The tightening of the knot is defined primarily by the force that is exerted, not the positions. Our learned gains for this task, shown in Figure 5, indicate that our method was able to learn this nuance successfully. The region between 24 and 28 seconds corresponds to tightening the knot, as shown in the inset, and also has the lowest gains.

In Table I, we show results for our method and the two baselines on varying lengths of test rope, using the same demonstration set. The experiments were conducted with ropes of lengths 130, 160, and 200 cm. For each experiment, we provide an image of the knot after tightening, and the diameter of the loop of rope. A fully tied knot has a diameter of zero. Our method was much more successful at producing tight knots than either of the baselines. This indicates that not only is a purely kinematic strategy insufficient for this task, but that simply including a force term with standard proportional-derivative gains is inadequate.

In Figure 6, we show the result of tying the rope to the pipe

¹See <http://rll.berkeley.edu/icra20151fmd/>










rope length	final loop diameter (cm)		
	kinematic only	default gains and force	our method
130 cm	 5.5 cm	 3.0 cm	 0.0 cm
160 cm	 8.0 cm	 6.0 cm	 0.0 cm
200 cm	 13.0 cm	 11.0 cm	 3.5 cm

TABLE I: Performance of our method, an ablated variant that used standard fixed gains, and a kinematic baseline that only tracked the demonstrated position. Experiments were conducted on ropes of varying length.

with our method and the kinematic baseline. Our method was able to tighten the rope around the pipe, while the kinematic baseline failed to do so, causing the rope to slip off the pipe.

In Figure 7, we show the whiteboard after erasing it with our method and the kinematic baseline. Without forces, the robot simply matched the desired height of the eraser, which is insufficient to erase the board. Our approach also applied a downward force, which cleared the board almost completely.

In Figure 8, we show a variety of towels before and after folding. The images include both the towel used in the demonstrations (left), and new towels that differ in their dimensions (middle and right). The demonstrations were performed on a towel 68 by 42 cm in size, and tested on towels 68 by 42 cm, 76 by 45 cm in size. Our method was able to straighten each towel and fold it over, while the kinematic baseline was consistently unable to straighten the towel in the first stage of the task.



Fig. 6: Tying rope around a pipe experiment: the rope after being tightened by the kinematic baseline (left) and our method (right). Since our approach applied an outward force, the rope was taut and the rope stayed in the pipe.

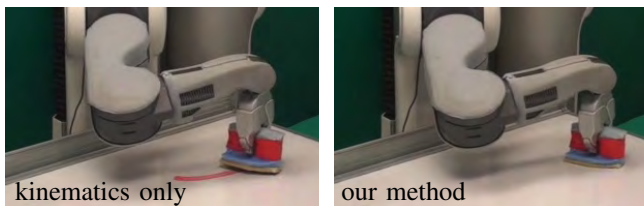


Fig. 7: Erasing experiment: the whiteboard after being erased by the kinematic baseline (left) and our method (right). Since our approach applied a downward force, the board was erased much more cleanly.

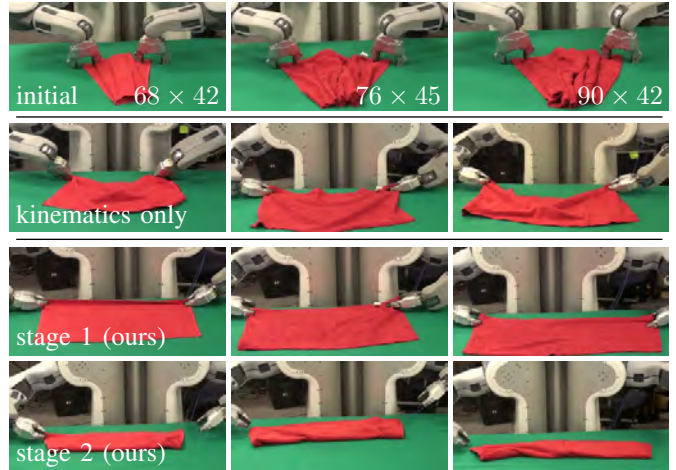


Fig. 8: Towel experiments: the top row shows the initial layout, the second row shows the kinematic method after the first stage, and the third and fourth rows show our method after the first and second stages.

Part of the reason for this was that, if towel is crumpled into even a slightly tighter pile than in the demonstration, the computed warp contracts the entire trajectory laterally, making it too narrow to straighten the towel, especially when the towel itself is wider than in the demonstration. The forces in our method were still able to stretch the towel despite the incorrect positions, but the kinematic baseline could not.

VIII. DISCUSSION

We presented an approach for generalizing force-based demonstrations of deformable object manipulation skills to novel situations. Our method uses non-linear geometric warping based on point cloud registration to adapt the demonstrations to a novel test scene, and then learns appropriate feedback gains to trade off position and force goals in a manner consistent with the data, providing for variable-impedance control. Our results show that including forces in the manipulation tasks allows for significantly greater generalization than purely kinematic execution: knots can be tightened more tightly in ropes with greater length variation and can be tied to a pipe without slipping off, towels of varying geometries can be stretched and laid flat, and whiteboards can be erased effectively.

We chose our tasks to include both phases that are determined primarily by pose, such as positioning the gripper to grasp the rope, and phases that are primarily force-driven, such as tightening the knot. Performing such tasks kinematically is unreliable, because some parts are defined primarily by the force exerted on the object, while others require precise positioning. Automatically determining whether force or pose is important at each phase is essential for effectively generalizing demonstrations of such tasks.

Several future directions could extend this work to further improve generalization. When a model of the robot is available, model-based inverse reinforcement learning algorithms

can be used to infer a reward function for the task [30], [31], which can provide a more generalizable, goal-based representation of the variation in position and force than directly fitting gains. Optimal gains could then be recovered from the reward function by running an optimal control algorithm. Access to a model would also allow us to compute the external forces, based on the internal forces and accelerations. Performing feedback on these external forces would provide for closed-loop control of force, which would improve on the current open-loop Jacobian transpose scheme.

We generated our demonstrations using a combination of teleoperation and kinesthetic teaching. Since our robot does not have force sensors at the joints, we computed the demonstration forces based on motor efforts. For kinesthetic teaching, this required the robot to replay each demonstrated trajectory to obtain an accurate reading of the forces, which made demonstrating very precise force-based behaviors challenging. Even with teleoperation, we found it difficult to generate demonstrations with the right force profiles. Several authors have proposed alternative interfaces designed specifically for demonstrating force-based behaviors [11], [14]. Another way to alleviate these challenges is to allow demonstrations to be recorded by a human, rather than by directly moving the robot, for example by tracking the human demonstrator's hands and using force sensors. Learning complex force-based manipulation skills from human demonstrations is an exciting direction for future work that, combined with the techniques presented in this paper, could allow for robot manipulation skills that exceed the state-of-the-art in dexterity and finesse.

IX. ACKNOWLEDGMENTS

This research was funded in part by the AFOSR through a Young Investigator Program award, by DARPA through a Young Faculty Award, and by NSF under award #1227536.

REFERENCES

- [1] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *Proceedings of the 16th International Symposium on Robotics Research (ISRR)*, 2013.
- [2] S. Calinon, "Robot programming by demonstration," in *Springer Handbook of Robotics*. Springer, 2008, pp. 1371–1394.
- [3] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, April 2007.
- [4] S. Calinon, F. D'Halluin, D. Caldwell, and A. Billard, "Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, Dec 2009, pp. 582–588.
- [5] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [6] B. Yang and H. Asada, "Progressive learning and its application to robot impedance learning," *IEEE Transactions on Neural Networks*, vol. 7, no. 4, pp. 941–952, 1996.
- [7] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.
- [8] E. Rombokas, M. Malhotra, E. Theodorou, Y. Matsuoka, and E. Todorov, "Tendon-driven variable impedance control using reinforcement learning," in *Robotics: Science and Systems (R:SS)*, 2012.
- [9] P. Sikka and B. McCarragher, "Stiffness-based understanding and modeling of contact tasks by human demonstration," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 1997.
- [10] S. Calinon, I. Sardellitti, and D. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [11] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.
- [12] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *AAAI Conference on Artificial Intelligence*, 2013.
- [13] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning object-level impedance control for robust grasping and dexterous manipulation," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2014.
- [14] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE Transactions on Haptics*, vol. 7, no. 3, pp. 367–380, 2014.
- [15] —, "Online learning of varying stiffness through physical human-robot interaction," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2012.
- [16] H. Inoue and M. Inaba, "Hand-eye coordination in rope handling," *Robotics Research: the First International Symposium*, pp. 163–174, 1985.
- [17] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Knot planning from observation," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2003, pp. 3887–3892.
- [18] H. Wakamatsu, E. Arai, and S. Hirai, "Knotting/unknotting manipulation of deformable linear objects," *International Journal of Robotics Research*, vol. 25, no. 4, pp. 371–395, 2006.
- [19] M. Saha, P. Isto, and J.-C. Latombe, "Motion planning for robotic manipulation of deformable linear objects," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and D. Rus, Eds. Springer Berlin Heidelberg, 2008, vol. 39, pp. 23–32.
- [20] Y. Yamakawa, A. Namiki, M. Ishikawa, and M. Shimojo, "One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- [21] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Parametrized shape models for clothing," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2011.
- [22] B. Willimon, S. Birchfield, and I. Walker, "Model for unfolding laundry using interactive perception," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [23] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2011.
- [24] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 114–141, 2003.
- [25] H. Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, 1978.
- [26] J. Duchon, "Splines minimizing rotation-invariant semi-norms in sobolev spaces," in *Constructive Theory of Functions of Several Variables*, ser. Lecture Notes in Mathematics, W. Schempp and K. Zeller, Eds. Springer Berlin Heidelberg, 1977, vol. 571, pp. 85–100.
- [27] G. Wahba, *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics, 1990.
- [28] J. D. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *Robotics: Science and Systems (RSS)*, 2013.
- [29] A. O'Hagan, *Bayesian Inference*. Halsted Press, 1994.
- [30] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2004.
- [31] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *International Conference on Machine Learning (ICML)*, 2012.