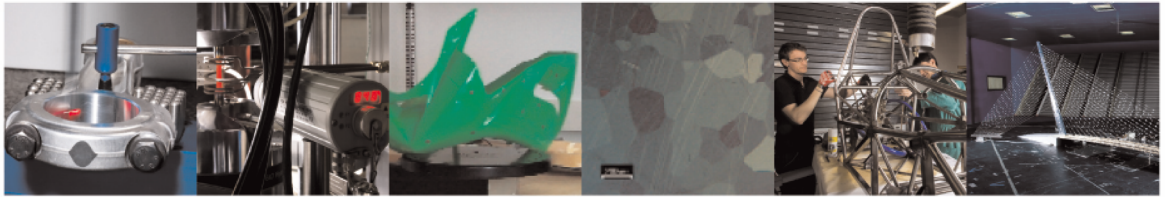




POLITECNICO
MILANO 1863

DIPARTIMENTO DI MECCANICA



Teaching contact-rich tasks from visual demonstrations by constraint extraction

Hegeler, Christian;Rozzi, Filippo;Roveda, Loris;Haninger, Kevin

This is a post-peer-review, pre-copyedit version of Hegeler, Christian;Rozzi, Filippo;Roveda, Loris;Haninger, Kevin, "Teaching contact-rich tasks from visual demonstrations by constraint extraction," 2025 IEEE 21st International Conference on Automation Science and Engineering (CASE), doi: 10.1109/CASE58245.2025.11164000

2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

© 2025. This manuscript version is made available under the [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) license <https://creativecommons.org/licenses/by-nc-nd/4.0/>



Teaching contact-rich tasks from visual demonstrations by constraint extraction

Christian Hegeler¹, Filippo Rozzi², Loris Roveda³, and Kevin Haninger¹

Abstract Contact-rich manipulation involves kinematic constraints on the task motion, typically with discrete transitions between these constraints during the task. Allowing the robot to detect and reason about these contact constraints can support robust and dynamic manipulation, but how can these contact models be efficiently learned? Purely visual observations are an attractive data source, allowing passive task demonstrations with unmodified objects. Existing approaches for vision-only learning from demonstration are effective in pick-and-place applications and planar tasks. Nevertheless, accuracy/occlusions and unobserved task dynamics can limit their robustness in contact-rich manipulation. To use visual demonstrations for contact-rich robotic tasks, we consider the demonstration of pose trajectories with transitions between holonomic kinematic constraints, first clustering the trajectories into discrete contact modes, then fitting kinematic constraints per each mode. The fit constraints are then used to (i) detect contact online with force/torque measurements and (ii) plan the robot policy with respect to the active constraint. We demonstrate the approach with real experiments, on cabling and rake tasks, showing the approach gives robust manipulation through contact transitions.

1 Introduction

Contact modeling and control are essential for robots to physically interact with their environments [1]. When the environment is less structured, the contact conditions will

vary and the ability to detect and reason about contacts is needed for robust and dynamic manipulation, such as for drawers, doors, and cables [2].

Contact constraints are important for the safety, performance, and robustness of manipulation [3–5]. For safety, violating the current constraint can result in excessive force and transitions between contact with excessive velocity can result in collision forces. For performance, constraints can be exploited to reduce variation in trajectories, and trajectory planning with respect to the constraints can also exploit dynamics (*e.g.*, Coulomb friction, constraint-space inertia). For robustness, changes in contact modes can be used to monitor a task to identify task success, failure, or other discrete modes. Methods exist to plan through constrained motion, but they need models of the constraints.

1.1 Related works

Contact models - *e.g.*, the contact normal or signed distance function - are key for modern planning-based approaches to locomotion and contact-rich manipulation [6]. These are application specific and often extracted from CAD models. However, they can also be learned from demonstrations, such as by using an instrumented tool that measures the pose and forces of a human [7].

Learning from demonstration (LfD) is a powerful tool, developing methods to specify robot tasks in an efficient and natural way [8]. Classical LfD approaches such as dynamic movement primitives can learn and adapt free-space trajectories to new start and goal positions, and have also been extended to consider contact [9]. However, demonstrating force trajectories can be complex: either a 2nd force/torque sensor is needed [10, 11], or the robot must be teleoperated [12, 13], limiting ease of demonstration and dexterity in fine tasks, respectively. Furthermore, this approach must balance whether the force or position trajectory should be tracked, a question of the robot’s impedance [14]. This can be optimized when the robot reproduces the trajectories [11, 15], however, introducing additional complexity in the definition

¹Department of Automation at Fraunhofer IPK, Berlin, Germany

²Politecnico di Milano, Department of Mechanical Engineering, Milano, Italy

³Istituto Dalle Molle di Studi sull’Intelligenza Artificiale (IDSIA), Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Università della Svizzera Italiana (USI) IDSIA-SUPSI, Lugano, Switzerland loris.roveda@idsia.ch

Corresponding author: Christian-Hegeler@proton.me. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement 101058521 – CONVERGING.

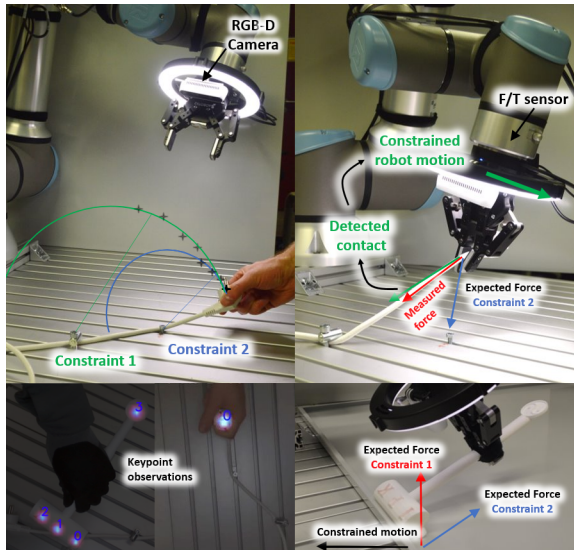


Figure 1: Setup of the approach, where the keypoint trajectory of an object is tracked by an RGBD camera (left), the data are clustered, and kinematic constraints are fit. The constraint model is then exploited by the robot online (right), where measured force is compared with the constraint Jacobians to identify constraint online and the robot’s trajectory is adapted to enter and maintain constraints.

of the reward function, data collection, and implementation.

For LfD methods which are variance- [16, 17] or stiffness-aware [18], the task representation is continuous, not explicitly representing changes in the contact conditions. Other methods which learn discrete modalities from demonstration such as [19, 20] do not build explicit contact models, limiting the ability to use force information or planning methods.

Demonstrations can also be used with behavior cloning in deep learning-based approaches, initializing a task-specific policy [21]. This has been applied to contact-rich tasks like peg-in-hole insertion from visual information [12]. While some deep learning policies can incorporate force information [22], the ability of these methods to handle tighter tolerance tasks and extrapolate beyond the data has not yet been shown. In fact, the ability of standard neural networks to directly model contact is currently limited [23].

1.2 Paper Contribution

In contact-rich manipulation tasks, object trajectories can be characterized by a set of key phases, depending on the type of constraint currently acting on the object. An autonomous robot should, therefore, be able to identify these constraints from such manipulation tasks. However, it is still difficult to get contact models for complex manipulation tasks, in particular avoiding the use of additional external sensors in the demonstrations. In addition, vision-only methods may lack robustness in reproducing contact-rich motion.

To address the above-mentioned issues, we consider using passive observations (*i.e.*, developing a visual-only method)

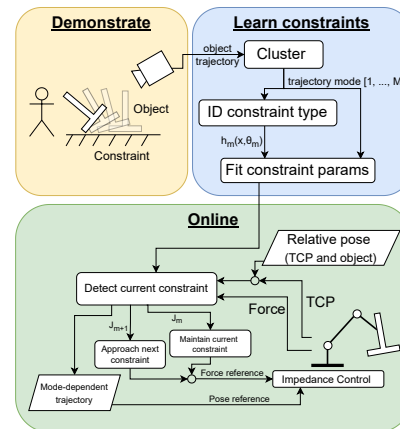


Figure 2: The components of the proposed approach, where the demonstrations and constraint learning are covered in Section III, and online control in Section IV.

of human contact-rich demonstrations and note that, due to the noise and low stiffness of typical human motions, repeatability in the trajectories will be induced primarily by contact constraints. We show that, by segmenting the demonstrations and fitting the constraints, constraint-aware manipulation can be improved. This happens in two ways: i) using the kinematic constraints to derive the direction of expected force, allowing online classification of the contact condition from F/T measurements, and ii) using contact models for the planning process, setting a desired force to maintain the current constraint and approach the next one.

We validate this approach with real experiments, using keypoint detection on objects without markers, and reproducing contact-rich tasks. We find that clustering and fitting parameterized models is effective even on noisy RGBD data, and that online detection with constraint Jacobians is effective provided the Jacobians are not parallel at the current pose.

The main novelties of the approach proposed here approach are: (i) a validated method for extracting a discrete collection of geometric constraints from vision-only demonstrations; (ii) a method for the online detection of the active constraint, based on F/T measurements and the learned constraint geometry; (iii) a higher-level controller which sets the force reference of an impedance controller to exploit the constraint information to improve task robustness.

1.3 Proposed Approach vs State of the Art

Compared to related constraint-learning work [7], this method doesn’t need additional motion markers or F/T sensors to collect demonstrations and the learned models are applied online. Compared with dense visual descriptors used in teaching by demonstration [21], this method can handle contact-rich manipulation from human-readable visual information (pose from point correspondences). Compared

with contact-DMPs (dynamic movement primitives) [9], the demonstrations are purely visual and produce explicit contact models. Compared with variance-aware [16] or stiffness-aware [18] LfD, our method explicitly models constraints, and changes in contact, and can detect changes in contact online.

2 Problem Statement and Approach

This section formalizes the type of considered manipulation problem, where multiple kinematic constraints occur during its execution, then overviews the proposed approach.

2.1 Contact-rich Manipulation

The contact-rich manipulation problem is expressed over the pose or local features of a manipulated object, $x \in \mathbb{R}^n$, where n is the dimension of the position or pose observations. During a task, the object has a trajectory of N points, $\mathcal{X} = [x_1, \dots, x_N]$. This trajectory has discrete changes in contact conditions, denoted sequentially as $m \in [1, \dots, M]$. Each mode has a holonomic constraint of the form $h_m(x) = 0$, where $h : \mathbb{R}^n \rightarrow \mathbb{R}^{n_m}$ and n_m is the degree of the constraint in m [24]. A constraint $h_m(x)$ is enforced with constraint forces of the form $F = \lambda^T J_m(x)$, where $F \in \mathbb{R}^n$ are the forces expressed in the coordinate system of x , $\lambda \in \mathbb{R}^{n_m}$ the Lagrange multipliers representing the magnitude of the force and $J_m = \partial h_m / \partial x$ the constraint Jacobian.

2.2 Problem Statement and Approach

Given only pose trajectories \mathcal{X} , where a sequence of contact modes $[1, \dots, M]$ implicitly occur, it is desired that the robot can safely reproduce this sequence of contacts over small geometric variations in the constraints.

The proposed approach can be seen in Figure 2, where pose trajectories \mathcal{X} are used to find kinematic constraint models. First, the data is clustered to produce M datasets, corresponding to each constraint mode. Then, a constraint model is fit for each data cluster, where it is assumed that the constraint is one of a library of parameterized models $h_m(x, \theta_m)$, where the parameters θ_m are parameters fit to the clustered data. The fit models then give Jacobians J_m , which give the direction or subspace in which the mode’s constraint forces are expected. Force measurements allow the identification of the constraint, and can inform the robot policy.

3 Visual Demonstrations

This section describes how the visual demonstrations are collected and the data segmented.

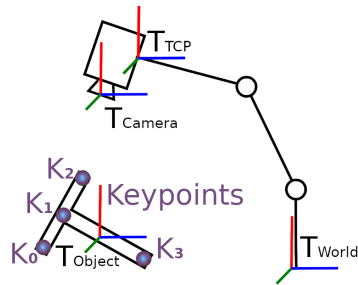


Figure 3: Overview of involved coordinate systems for object pose estimation

3.1 Pose estimation

Poses of objects, such as the plug and rake seen in Figure 1, are estimated with the means of keypoints. Keypoints have a fixed relation to local visual object features and are ideally invariant to rigid transformations of object poses. The keypoint location is estimated from RGB images in 2D pixel space based on the work of [25] and are subsequently transformed into 3D world coordinates following the transformations shown in Fig. 3. Using depth information and calibrated camera intrinsics, keypoints locations are lifted from pixel space into 3D space relative to the camera. The camera poses relative to world coordinates are estimated with the known tool center point (TCP) and the known fixed transformation from TCP to camera initially calibrated with the work of [26].

Keypoints are detected using deep convolutional neural network models trained on object-specific data sets that consist of image data with corresponding keypoint labels. The image data is collected autonomously by the robot. The labeling process is semiautomatic, keypoints labeled in one image, then projected from pixel space to 3D world coordinates and back to pixel coordinates in the other images. When using three or more distinguishable keypoints, related object poses can be unambiguously described in $SE(3)$. For an initial keypoint configuration, an initial object pose is assigned with the center at the mean of all related keypoints and identity as rotation. When detecting keypoints for any new object pose, the pose is described by a transformation of the initial object pose that aligns both sets of keypoints.

3.2 Segmentation

In this section we propose an autonomous segmentation framework to extract a set of segmentation points by dividing a multi-dimensional motion trajectory obtained from a single passive demonstration; the aim is to define a temporally ordered sequence of clusters in which just one constraint is present, such that each cluster can be used for the constraint learning. To this extent, we model the overall trajectory including the time component as a Gaussian Mixture Model (GMM) [27] and we estimate the parameters of the model using the standard Expectation Maximization approach. Rep-

representing a multi-dimensional trajectory as a GMM provides a way of encoding the local direction and local relations (*i.e.*, covariance) between the variables. The main advantage of such an estimation framework is that there is no need to manually predefine any internal parameters according to the types of given tasks and/or trajectories.

We assume the ergodicity of the training data and we do not need to consider temporal and spatial variations between multiple demonstrations while preparing the training data. A multi-dimensional trajectory, $\mathcal{X} \in \mathbb{R}^{(D+1) \times N}$, is extracted from a passive observation. Here, $(D + 1)$ denotes the D -dimensional spatial variables (*e.g.*, object pose, velocity, keypoints coordinates) and the one-dimensional temporal variable (*i.e.*, time step), and N is the length of the trajectory.

From the GMM, the segmentation points are detected in temporally overlapping points in-between two consecutive Gaussians; the points are located in all points intersected along the time component of the GMM. However, before detecting these intersections, the mean and the covariance of the time component need to be extracted from the output mean and covariance of the GMM. To this extent, the mean and covariance matrices of the i^{th} Gaussian cluster are represented as

$$\mu_i = [\mu_{i,t}, \mu_{i,X}], \quad \Sigma_i = \begin{bmatrix} \Sigma_{i,t} & \Sigma_{i,tX} \\ \Sigma_{tX,i} & \Sigma_{i,X} \end{bmatrix}, \quad (1)$$

where t and X refer to the one-dimensional temporal variable and the D -dimensional spatial variable in the $(D + 1)$ dimensional trajectory. All intersections between Gaussians are extracted by estimating the weights of Gaussians along the time component. Estimating the weights is defined as

$$\gamma_i(t) = \frac{w_i \mathcal{N}(t; \mu_{i,t}, \Sigma_{i,t})}{\sum_{k=1}^K w_k \mathcal{N}(t; \mu_{k,t}, \Sigma_{k,t})}, \quad (2)$$

where i and K refer to the indices of Gaussians and the total number of Gaussians, w_i are the output weights of the GMM and $\gamma_i(t)$ represent the weights projected along the time component for each cluster, as seen in Fig. 4. Each data point is assigned to a set based on its maximum weight, and the overall trajectory is thus split into M modes, with corresponding data \mathcal{X}^m for the m^{th} mode.

3.3 Constraint Fitting

Given a trajectory segment containing N data points that consists of a single contact mode m , the kinematic constraints which are imposed in that mode can be identified. We assume that the kinematic constraints are holonomic constraints in the form $h_m(x, \theta_m) = 0$, where θ_m are parameters describing the geometry. The constraint h_m is assumed to belong to a set of possible contact primitives [7], such as point-on-plane, or line constraint.

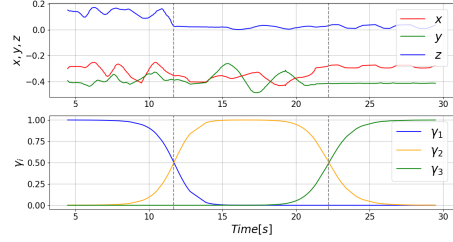


Figure 4: Segmentation framework for Zen Garden Raking. The upper plot represents the mean positional coordinates of the considered keypoints while the lower plot shows segmentation points extracted from weights projected along the time component.

The constraint for a mode is identified by fitting

$$\min_{\theta_m} \sum_{x_t \in \mathcal{X}^m} \|h_m(x_t, \theta_m)\|_2^2 + \beta_m(\theta_m) + Ns \quad (3)$$

$$\text{s.t. } \|h_m(x_t, \theta_m)\|_2^2 \leq s \quad \forall x_t \in \mathcal{X}^m \quad (4)$$

where β_m is a mode dependent regularization term and s a slack variable describing the maximum of h_m which adds an H_∞ norm penalty on $h_m(x_t, \theta_m)$.

3.3.1 Radius Constraint

The radius constraint describes a point $x \in \mathbb{R}^3$ with a fixed radius $r \in \mathbb{R}$ to a point $x_c \in \mathbb{R}^3$ in base coordinates. All possible values for x are therefore distributed on the surface of a sphere centered at x_c . All points x_t must therefore fulfill the equality constraint $\|x - x_c\|^2 = r$. The radius constraint is thus fit by using

$$h_m(x_t, \theta_m) = r - \|x_t - x_c\|_2, \quad \beta_m(\theta_m) = kr, \quad (5)$$

where k is a regularization constant.

3.3.2 Line on Plane Constraint

The line-on-plane constraint describes an object in line contact with a fixed plane. The object can be translated and rotated along the plane such that the line contact between the object and the plane is never broken. When describing the line with two points $p_0, p_1 \in \mathbb{R}^3$ with a fixed relative position to the object and a fixed plane parameterized with its normal vector $n_p \in \mathbb{R}^3$ and offset $d \in \mathbb{R}$ the described constraint can be formulated such that all poses in a data set \mathcal{X}_m that correspond to the described constraint mode must fulfill the equality constraint with

$$h_m(x_t, p_0, p_1, n_p, d) = \left[\left| \|n_p^T(T(x_t)p_0)\|_2 - d \right| \right] + \left[\left| \|n_p^T(p_0 - p_1)\|_2 \right| \right], \quad (6)$$

$$\beta_m(\theta_m) = k(|\|n_p\|_2 - 1| + |\|p_0 - p_1\|_2 - k_D| + \|p_0\|_2 + \|p_1\|_2), \quad (7)$$

where $T(x) \in \mathbb{R}^{4 \times 4}$ is the transformation matrix built from the pose x , k is a regularization constant and k_D is the desired distance of p_0 and p_1 in meter.

3.3.3 Hinge constraint

A single fixed line in the object coordinate system has a fixed position in world coordinates. The line contact is parameterized with two points $x_{c,0}, x_{c,1} \in \mathbb{R}^3$ in world coordinates and $p_0, p_1 \in \mathbb{R}^3$ in object coordinates. Using transformation matrices from the object to the world coordinate system $T(x_t)$ one can express the described relationship as $x_{c,i} = T(x_t)x_{p,i}$, $i = 1, 2$. The constraint is thus fit with

$$h_m(x_t, \theta_m) = \begin{bmatrix} \|x_{c,0} - T(x_t)x_{p,0}\|_2 \\ \|x_{c,1} - T(x_t)x_{p,1}\|_2 \end{bmatrix}, \quad (8)$$

$$\beta_m(\theta_m) = k(|\|p_0 - p_1\|_2 - k_D| + \|p_0\|_2 + \|p_1\|_2), \quad (9)$$

where k and k_D follow the description from 3.3.2. The upper bound s for this constraint by the H_∞ norm is omitted.

4 Online Control

This section describes how the contact models can be employed in the online control of the robot.

4.1 Contact Detection

Given a constraint $h_m(x, \theta_m) \in \mathbb{R}^{n_m}$, the forces which would enforce that constraint can be found as

$$F = J_m^T(x)\lambda + \epsilon, \quad (10)$$

where force $F \in \mathbb{R}^n$, Jacobian $J_m(x) = \frac{\partial h_m(x, \theta_m)}{\partial x} \in \mathbb{R}^{n_m \times n}$, $\lambda \in \mathbb{R}^{n_m}$ are Lagrange multipliers which enforce the constraint and are not measured, and ϵ additional forces. The additional forces ϵ can be from gravitational or inertial forces of the payload, friction in contact, or other errors. The forces F are expressed in the coordinate system of the pose x .

In order to efficiently detect the active constraint in the online framework we need a robust metric that could scale properly for constraints with higher dimensions ($n_m > 1$). We formulate the problem as a least-squares problem to exploit the closed-form solution through the pseudoinverse $J_m(x)^\dagger$ of the constraint Jacobian as

$$\epsilon(F, x) = \|F - J_m(x)^\dagger J_m(x)F\|_2. \quad (11)$$

This residual tells us the amount of measured force that is not explained by the constraint and gives a metric for online

detection: the constraint with a lower residual is selected as the current constraint.

As the measured wrench vector F_t is measured at the TCP, it needs to be transformed to match the orientation of the coordinate system of the pose x , expressed with XYZ extrinsic Euler angles. We transform the wrench into a frame at the origin of the object frame and with the same orientation as the base frame. The standard transform of the full wrench vector with respect to a new reference frame can be found in [28].

To reduce false contact detection from the acceleration of the payload we also model the free space as a type of contact mode, but we do not detect free space based on the force residual. Free space detection is done when $\|F_\tau\| < \bar{F} \quad \forall \tau \in [t - w, t]$, where $\bar{F} \in \mathbb{R}$ is a threshold, w a properly tuned time window and $\|F_\tau\|$ is the norm of the measured wrench at TCP for each time step.

4.2 Constraint-aware Control

The constraint information can also be directly used in the control strategy. We use a lower-level admittance controller, such that

$$M\ddot{x}_t^r + D\dot{x}_t^r + K(x_t^d - x_t^r) = F_t^d - F_t, \quad (12)$$

where $F^d \in \mathbb{R}^6$ is the force reference, F_t the measured force in robot TCP, x_t^d the desired pose of the robot, x_t^r the robot pose and $M, D, K \in \mathbb{R}^{6 \times 6}$ the mass, damping, and stiffness admittance parameters. All quantities in (12) are expressed in the TCP frame, and the matrices M, D , and K are diagonal. The following sections propose how the reference signals F_t^d and x_t^d can be adjusted online according to the mode information.

4.2.1 Contact-triggered Control

At each time step t , we infer the current mode m_t as shown in Section 4.1. We then suppose that each mode has its own motion strategy as

$$x_t^d = \pi_{m_t}(x_t^r), \quad (13)$$

where π_{m_t} is a mode-specific controller which produces a desired pose x_t^d .

4.2.2 Making and Maintaining Contact

Many constraints are unilateral constraints more accurately considered as $h(x) \geq 0$, for example sliding on the contact of a surface. This means the constraint can be lost by moving away from the constraint, which may be undesired for the task and limits the contact detection.

Additionally, the pose where contact is entered may vary. Additional motion in the direction of the next constraint, as given by the Jacobian of the next constraint, can help ensure

that contact is made. This is done by adding to the desired contact wrench.

We add a small desired force to the robot force controller as

$$F_t^d = T_w \left(J_{m_t}^T(x_t)\alpha + J_{m_{t+1}}^T(x_t)\alpha^+ \right), \quad (14)$$

where T_w transforms the wrench from object coordinates to TCP, m_t is the current mode, m_{t+1} the next contact, $\alpha \in \mathbb{R}^{n_{m_t}}$, $\alpha > 0$ specifies the force to apply to maintain the current constraint and α^+ the force to approach the next constraint.

5 Validation

The approach is validated on a UR16e robot, which has an integrated F/T sensor at the flange and a Realsense D435 camera. The CasADi framework is used for parameter fitting and calculating Jacobians [29]. The admittance control is implemented via ROS [13]. The code is available at <https://github.com/khaninger/contactmonitoring>.

5.1 Cable pulling

The data collected for the radius constraint parameter estimation consists of 3D positions of a plug that is constrained with a cable that is fixed via two different pivots to a workbench. The demonstration data is shown in Fig. 5, where the observed plug positions are segmented into: free space (yellow), constraint by cable fixed at pivot 0 (cyan) and pivot 1 (magenta). The segmentation process has some noise in detecting the transitions but provides sufficient accuracy for the constraint fitting.

The location of the fit pivot 1 and its ground truth is shown in blue respectively red. Additionally, the sphere for the estimated radius around pivot 1 is shown. The fit constraint results in an error of 3.2cm with respect to the measured pivot, where the position is measured at height of the workbench with the camera from pixel space just as the keypoints are measured.

Using the fit constraints, the controller is applied as seen in the attached video. The parameters used for constraint detection are $\bar{F} = 6$, window $w = 8$, and constraint forces are applied of $\alpha = 12$, $\alpha^+ = 6$. The stiffness parameters of the impedance controller are $K = [250, 250, 550, 20, 20, 20]$. The motion strategy $\pi_{m_t}(x_t)$ finds the closest point in the demonstration trajectory, generates x_t^d by transforming from the object frame to TCP, and advances along the trajectory when the distance to the current x_t^d is less than 1 cm. The demonstration trajectory moves from free space, pulling tight with the pivot 0 constraint, then moves directly to enter the pivot 1 constraint, similar to that seen in Fig. 5.

Results of the online detection can be seen in the attached video and Fig. 6(a), where the residual per constraint and

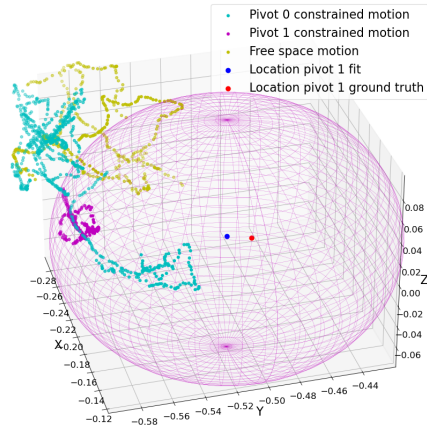


Figure 5: Segmentation of cable pulling data with fit parameters for front pivot

forces are shown. Near the starting pose, the residual between pivot 0 and pivot 1 can be clearly distinguished - at that point these constraints are in different directions. However, as the robot moves near when the two constraints are in a similar direction (around 20 sec), the residuals become more similar, and a few centimeters of error in the mode detection were noticed. This is attributed to the constraints being in a similar direction - the vectors J_0 and J_1 for pivot 0 and 1 are almost identical, and distinguishing the forces is difficult. Similar difficulty in distinguishing certain constraints has been noted by other authors [7, 30].

To validate the robustness of detecting changes in contact, the cable is unhooked by hand from pivot 1 multiple times. Additionally, at 60 seconds the robot is manually perturbed 20 cm from the path and at 80 seconds the cable slips on its own. The robot is able to recover from all these perturbations, as seen in the attached video. The force residuals can be seen in Fig. 6(b), where it can be seen that this online change is robustly detected, and the robot can recover with the mode dependent control. Furthermore, adding the desired force along the constraint direction allows the robot to keep the cable tight while executing the task, which does not always occur when using admittance control with $F_t^d = 0$.

5.2 Zen Garden Raking

This task involves a rake over a planar surface, with the additional constraint of reaching the edge of the raking area. These two constraints are extracted from demonstrations, where the pose of the rake is tracked as the rake moves in free space, line contact with the workbench, then a hinge constraint with the border of the raking area is met. The rake pose is tracked with four keypoints, and a keypoint is saved only when all four are observed, and the residual error of the correspondence to the reference keypoints is less than 3 mm.

A subset of the observed poses of the rake in contact including the fit plane and projected contact points for all

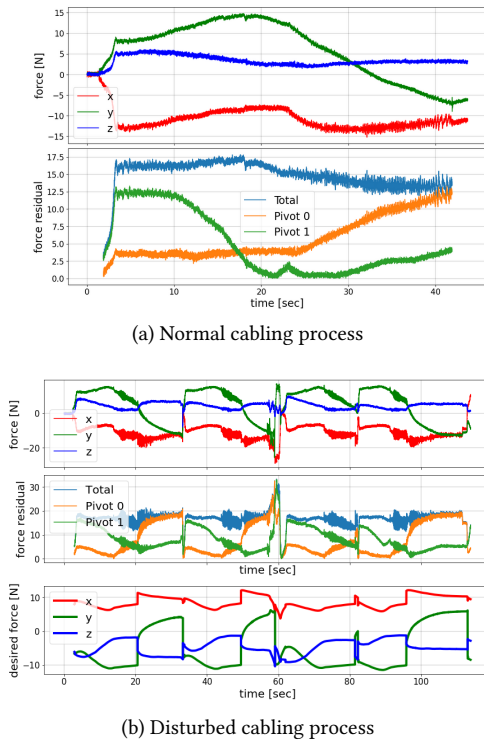


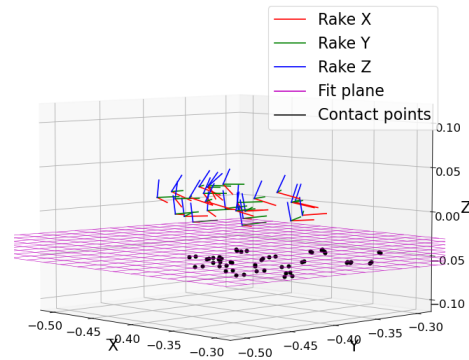
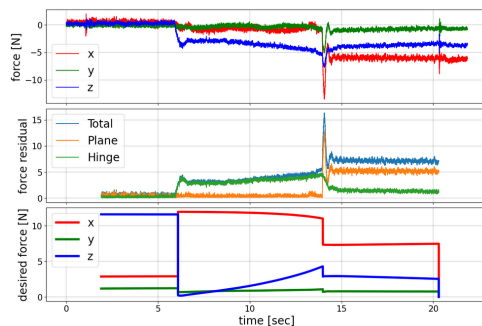
Figure 6: Forces and contact residual in the cabling task

shown poses are visualized in Fig. 7, along with the points of the identified contact point on the line constraint (black points) and the fit plane location (purple).

The results of the live control experiments on the rake can be seen in Fig. 8 and the attached video. The parameters used are $\bar{F} = 1.4$, $w = 6$, $\alpha = 4$ and $\alpha^+ = 6$. The first contact, just after 5 seconds, results in an overshoot, but the residual for the plane constraint approaches zero with no overshoot, indicating most of the collision forces are aligned with the expected constraint. The contact with the second constraint (hinge) results in an impulse in the forces, but again this impulse does not appear in the residual for the hinge constraint, indicating the impulse is occurring primarily in the subspace expected by the constraint.

6 Conclusion

The paper has proposed a pipeline for learning constraints from the passive observation of markerless human demonstrations. A general constraint fitting process is introduced, and several example constraints are validated from real data. The accuracy of a keypoint-based pose estimation, GMM-based clustering, and constraint model identification is shown to have sufficient accuracy for some contact-rich tasks. These constraint models are shown to result in effective online detection of constraint from F/T information, provided the Jacobian between them is sufficiently distinct.

Figure 7: Poses from rake data with fit parameters for the plane contact, with corresponding contact points p_0 , p_1 Figure 8: Measured forces in TCP, constraint residual, and desired force F_t^d in the rake task

The usefulness of this information was demonstrated by simple mode-dependent trajectory following with a desired force which maintains the current constraint.

References

- [1] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, “A survey of robot manipulation in contact,” *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022.
- [2] A. Bicchi and V. Kumar, “Robotic grasping and contact: A review,” in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.
- [3] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, “Dual arm manipulation—a survey,” *Robotics and Autonomous systems*, vol. 60, no. 10, pp. 1340–1353, 2012.
- [4] P. Jiménez, “Survey on model-based manipulation planning of deformable objects,” *Robotics and computer-integrated manufacturing*, vol. 28, no. 2, pp. 154–163, 2012.
- [5] R. Li and H. Qiao, “A survey of methods and strategies for high-precision robotic grasping and assembly tasks—some new trends,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2718–2732, 2019.
- [6] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, “Global Planning for Contact-Rich Manipulation via Local Smoothing of Quasi-dynamic Contact Models.” [Online]. Available: <http://arxiv.org/abs/2206.10787>
- [7] G. Subramani, M. Gleicher, and M. Zinn, “Recognizing geometric constraints in human demonstrations using force and position signals,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1252–1259, 2018.
- [8] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent advances in robot learning from demonstration,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 2020.
- [9] M. Deniša, A. Gams, A. Ude, and T. Petrič, “Learning Compliant Movement Primitives Through Demonstration and Statistical Generalization,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2581–2594, 2016–10.
- [10] T. Tang, H.-C. Lin, and M. Tomizuka, “A learning-based framework for robot peg-hole-insertion,” in *Dynamic Systems and Control Conference*, vol. 57250. American Society of Mechanical Engineers, 2015, p. V002T27A002.
- [11] C. Chang, K. Haninger, Y. Shi, C. Yuan, Z. Chen, and J. Zhang, “Impedance Adaptation by Reinforcement Learning with Contact Dynamic Movement Primitives,” in *2022 IEEE Intl. Conf. on Advanced Intelligent Mechatronics (AIM)*, 2022. [Online]. Available: [arXiv:2203.07191](https://arxiv.org/abs/2203.07191)
- [12] M. Vecerik, O. Sushkov, D. Barker, T. Rothörl, T. Hester, and J. Scholz, “A Practical Approach to Insertion with Variable Socket Position Using Deep Reinforcement Learning.” [Online]. Available: <http://arxiv.org/abs/1810.01531>
- [13] S. Scherzinger, A. Roennau, and R. Dillmann, “Contact Skill Imitation Learning for Robot-Independent Assembly Programming,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 4309–4316.
- [14] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. A. Theodorou, and S. Schaal, “Model-free reinforcement learning of impedance control in stochastic environments,” *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 4, pp. 330–341, 2012.
- [15] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret, “A survey on policy search algorithms for learning robot controllers in a handful of trials.” [Online]. Available: <http://arxiv.org/abs/1807.02303>
- [16] G. Franzese, A. Mészáros, L. Peternel, and J. Kober, “ILoSA: Interactive learning of stiffness and attractors,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7778–7785.
- [17] K. Haninger, C. Hegeler, and L. Peternel, “Model predictive control with gaussian processes for flexible multimodal physical human robot interaction,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6948–6955.
- [18] F. J. Abu-Dakka, L. Rozo, and D. G. Caldwell, “Force-based variable impedance learning for robotic manipulation,” *Robotics and Autonomous Systems*, vol. 109, pp. 156–167, 2018–11. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0921889018300125>
- [19] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, “Learning Manipulation Graphs from Demonstrations Using Multimodal Sensory Signals,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018–05, pp. 2758–2765.
- [20] A. S. Wang and O. Kroemer, “Learning Robust Manipulation Strategies with Multimodal State Transition Models and Recovery Heuristics,” in *Proceedings of the*

- 2019 *IEEE International Conference on Robotics and Automation*, 2019.
- [21] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, “Implicit behavioral cloning,” in *Conference on Robot Learning*. PMLR, 2022, pp. 158–168.
- [22] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, “Making Sense of Vision and Touch: Learning Multimodal Representations for Contact-Rich Tasks,” *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020-06.
- [23] M. Parmar, M. Halm, and M. Posa, “Fundamental Challenges in Deep Learning for Stiff Contact Dynamics,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021-09, pp. 5181–5188.
- [24] V. Acary and B. Brogliato, *Numerical Methods for Non-smooth Dynamical Systems: Applications in Mechanics and Electronics*. Springer Science & Business Media, 2008.
- [25] K. Blomqvist, J. J. Chung, L. Ott, and R. Siegwart, “Semi-automatic 3d object keypoint annotation and detection for the masses,” in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 3908–3914.
- [26] K. Daniilidis, “Hand-eye calibration using dual quaternions,” *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.
- [27] S. Calinon, F. Guenter, and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.
- [28] H. BO, “Six-Degree-Of-Freedom Active Real-Time Force Control of Manipulator,” 2001.
- [29] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [30] T. J. Debus, P. E. Dupont, and R. D. Howe, “Distinguishability and identifiability testing of contact state models,” *Advanced Robotics*, vol. 19, no. 5, pp. 545–566, 2005.