2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Online Estimation of Articulated Objects with Factor Graphs using Vision and Proprioceptive Sensing

Russell Buchanan[1], Adrian Röfer[2], João Moura[1], Abhinav Valada[2], and Sethu Vijayakumar[1]

***Abstract*— From dishwashers to cabinets, humans interact with articulated objects every day, and for a robot to assist in common manipulation tasks, it must learn a representation of articulation. Recent deep learning methods can provide powerful vision-based priors on the affordance of articulated objects from previous, possibly simulated, experiences. In contrast, many other works estimate articulation by observing the object in motion, requiring the robot to already be interacting with the object. In this work, we propose to use the best of both worlds by introducing an online estimation method that merges vision-based affordance predictions from a neural network with interactive kinematic sensing in an analytical model. Our work has the benefit of using vision to predict an articulation model before touching the object, while also being able to update the model quickly from kinematic sensing during the interaction. In this paper, we implement a full system using shared autonomy for robotic opening of articulated objects, in particular objects in which the articulation is not apparent from vision alone. We implemented our system on a real robot and performed several autonomous closed-loop experiments in which the robot had to open a door with unknown joint while estimating the articulation online. Our system achieved an 80% success rate for autonomous opening of unknown articulated objects.**

I. Introduction

Articulated objects are ubiquitous in everyday environments: dishwashers, microwaves, and cabinets are all objects that robots must interact with to perform useful tasks like cooking or cleaning. To interact with these objects, a robot needs an understanding of the articulation – either as an analytical model (e. g. revolute or prismatic joint) or as an implicitly learned model through a neural network. Many recent works apply deep learning to predict affordance from vision; however, articulation prediction from vision alone may not be feasible. For example, in Fig. 1, each door appears identical until a person or robot interacts with them. This is a problem for robotic systems which rely exclusively on vision for understanding articulation. In our work, we propose a method for joint optimization of vision and proprioceptive sensing for online estimation of articulated objects.

Some early work on manipulating articulated objects focused on parts-based estimation using only proprioception [1]. Their method achieved impressive reliability in 2010, opening commonplace doors and drawers using only kinematic sensing and a grasp pose provided by a user. Vision has also been used, usually by tracking features or fiducial markers on the moving part and estimating articulation from the observed motion [2], [3]. This can be combined with proprioceptive sensing to manipulate a wider variety of objects [4]. The limitation of these methods is that sensing (both vision and proprioceptive) only occurs while the articulated part is in motion. The robot has no initial guess of how the object opens and so must rely on human direction or random guessing.

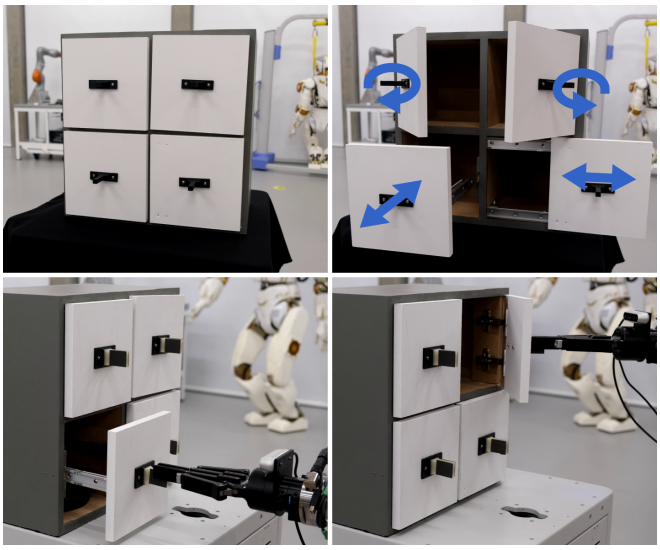

Fig. 1. Top row: a cabinet with a set of visually identical doors. Their different articulations are only revealed once open. It would not be possible from visual inspection alone to predict how each door opens. Bottom row: the robot autonomously opens the cabinet while estimating articulation online.

In response to this, the research focus has shifted recently to deep learning with only visual information used to predict articulation affordance [5], [6], [7]. The benefit of this approach is in estimating how to interact with the object, without the robot needing to physically touch it. However, this introduces additional limitations such as computational cost, poor generalization to previously unseen categories of objects, and an inability to predict articulation for objects without obvious visual cues of articulation, such as Fig. 1. These works also lack the quality of *interactive perception* [8] present in previous motion observation methods.

In this work, partly inspired by Lips and wyffels [9], we seek to revisit the use of proprioceptive sensing in manipulating articulated objects, while exploiting the latest deep learning advances. We propose a system that uses a neural network to predict object affordance as an initial guess and then updates the model estimate from proprioceptive sensing while interacting with the object. This is analogous to how a human might initially try to pull open a drawer only to realize during interaction that it is a revolute door.

[1] School of Informatics, University of Edinburgh, UK
[2] Department of Computer Science, University of Freiburg, Germany
This work is supported by the Alan Turing Institute, EU H2020 Project Harmony (Grant No. 101017008), the Carl Zeiss Foundation ReScaLe project, and the BrainLinks-BrainTools center of the University of Freiburg.

We use a network trained in simulation using the Partnet-Mobility dataset [10] to predict articulation affordance from an initial point cloud. Then, at the grasp point specified by a user, the robot will interact with the articulated object and use proprioceptive sensing to estimate the articulation parameters while opening. We formulate this estimation problem as a factor graph and estimate screw parameters which are then passed to our symbolic math model for motion generation. In summary, the contributions of this paper are:

- Online estimation of articulation parameters using vision and proprioceptive sensing in a factor graph framework.
- Full system integration for shared autonomy between a human user and the robot for opening articulated objects.
- Validation of our system with extensive real-world experimentation, opening several articulated objects with the estimation and control running online in closed loop.

II. Related work

In this section, we briefly summarize related works on articulation estimation. First, we cover interactive perception [8] methods which have a long history of use with articulated objects. Then, we briefly cover the most relevant, recent deep learning methods for vision-based articulation prediction.

A. Interactive Perception Methods

Few works use solely proprioception for estimating articulation due to challenges in identifying a grasp point. Jain *et al.* [1] simplified the problem by assuming a prior known grasp pose and initial opening force vector which allowed them to open several everyday objects. They demonstrated that once a robot is physically interacting with an articulated object and is given a good initial opening direction, proprioception alone can be sufficient to open the objects.

More commonly, proprioception is fused with vision. Sturm *et al.* [3] introduced a probabilistic framework for estimating articulation. Their method explicitly classifies the type of joint by maximizing the likelihood from an observed trajectory. They tested their method using fiducial markers, depth images, and kinematic information. Later, other work proposed a method based on bundle adjustment of visual features [11]. Martin-Martin *et al.* [4] introduced a framework that can estimate online from vision and tactile sensing. Like previous work, they track the motion of visual features while the robot is interacting with the object. This is fused with force/torque sensing, haptic sensing from a soft robotic hand, and end effector pose measurements.

Heppert *et al.* [12] proposed a neural network to track the motion of the parts from vision. The tracked poses are connected by a factor graph to estimate the joint parameters. In their experiments, they estimated an unknown articulation; however, their controller used a prescribed motion to open the object, giving sufficient information to the estimator. In our work, we also use a factor graph that connects part poses to a joint screw model. However, we use predictions from a neural network to give the robot an initial estimate of the articulation which is then updated online from kinematic sensing. Our use of both interactive perception and learning-based predictions allows us to perform closed-loop control and estimation while opening unknown articulated objects.

B. Learning-Based Methods

Many works have focused on using only visual information with deep learning to predict articulation without the need for object interaction [13], [14], [15]. Often, these works use simulated datasets such as the PartNet-Mobility dataset [10] which contains examples of common articulated objects.

Recently, there has been focus on learning category-free articulation affordances [16], [17], [7] which describe how a user can interact with an object. This is typically parameterized as a normalized vector which describes the motion of a point on the articulated part of an object. Bahl *et al.* [18] used a neural network to predict both grasp pose and opening trajectory from human demonstrations.

Some recent learning-based works have incorporated interaction. Jiang *et al.* [19] used pointcloud data before and after a human interacted with the target articulated object. This is similar to the works described in Section II-A. Nie *et al.* [20] introduced a method which predicts articulation as well as proposes an interaction through which to observe the motion and update the articulation estimate.

All of these works have similar limitations which makes their use on real robots challenging. They use only visual information and have large computational requirements which prevents online estimation. When they are used with real data, they typically take a single "snapshot" of the object and make a single inference. However, due to the reliance on recognizing visual similarity in objects compared to past training experiences, these methods exhibit poor performance on an object like in Fig. 1 which has no visual indicators as to how it opens. If the predictions are wrong, then these methods are reliant on highly compliant controllers to account for the error due to a lack of online estimation.

C. Systems

In our work, we provide not only an estimation method but also a full system for opening articulated objects with shared autonomy. Therefore, we also mention some related systems work. Mittal *et al.* [5] introduced a system for whole-body mobile manipulation. They used the category-level object pose prediction network from [13]. This meant their method needed prior information about the category of object with which the robot interacted. Also, in their method, they make a single prediction before interaction and then rely on controller robustness to account for mistaken predictions.

A closed-loop learning estimation method was proposed by Schiavi *et al.* [6]. This method estimates articulation affordance from vision at multiple time steps during the interaction. A sampling-based controller solves for the optimal opening trajectory. When opening the object becomes stagnant due to torque limits, the robot releases the object and moves to a configuration to view the full object again, then makes a new vision-based estimate of the articulation.

These systems rely heavily on robust and compliant controllers to account for all errors in articulation estimation.

In contrast, our work updates the estimation of the articulation model seamlessly during interaction.

III. Screw Theory Background

Screw theory is the geometric interpretation of twists that can be used to represent any rigid body motion (Chasles theorem) [21]. Screw motions are parameterized by the twist $\xi = (\mathbf{v}, \boldsymbol{\omega})$, where $\mathbf{v}, \boldsymbol{\omega} \in \mathbf{R}^3$. The variable $\mathbf{v}$ represents the linear motion and $\boldsymbol{\omega}$ the rotation. We can convert this to a tangent space to SE(3) using $\hat{\xi}$ as

$$\hat{\xi} = \begin{bmatrix} \hat{\boldsymbol{\omega}} & \mathbf{v} \\ 0 & 0 \end{bmatrix} \in \mathfrak{so}(3), \tag{1}$$

where the hat operator $\hat{(\cdot)}$ is defined as:

$$\hat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & -\omega_x & 0 \end{bmatrix}. \tag{2}$$

This can be converted to the homogeneous transformation $\mathbf{T}^{twist}(\hat{\xi}, \theta) \in \mathrm{SE}(3)$ using the exponential map Exp:

$$\mathbf{T}^{twist}(\hat{\xi}, \theta) = \mathrm{Exp}(\hat{\xi}\theta), \tag{3}$$

where $\theta \in \mathbf{R}$ is the articulation configuration.

Now if we define a fixed world frame $\mathtt{W}$, the pose of the moving part of an articulated object $\mathbf{T}_{\mathtt{WA}} \in \mathrm{SE}(3)$ is related to the other, non-moving part $\mathbf{T}_{\mathtt{WB}} \in \mathrm{SE}(3)$ by

$$\mathbf{T}_{\mathtt{WA}} = \mathbf{T}_{\mathtt{WB}} \mathbf{T}^{twist}(\hat{\xi}, \theta). \tag{4}$$

IV. Problem Statement

The goal of this work is to estimate online the Maximum-A-Posteriori (MAP) state of a single joint from visual and proprioceptive sensing. We define the state $\boldsymbol{x}(t)$ at time t as

$$\boldsymbol{x}(t) \triangleq [\xi, \theta(t)] \in \mathbb{R}^7, \tag{5}$$

where ξ are the screw parameters which are constant for all time and $\theta(t)$ is the angle of articulation at time t. We assume the object is composed of only two parts connected by a single joint. This encompasses the vast majority of articulated objects and therefore is a reasonable simplification. We define the pose of each part in the world frame $\mathtt{W}$ as $\mathbf{T}^A_{\mathtt{WA}}, \mathbf{T}^B_{\mathtt{WB}} \in \mathrm{SE}(3)$ where $\mathbf{T}^B_{\mathtt{WB}}$ is the bottom part that is static and $\mathbf{T}^A_{\mathtt{WA}}$ is the articulated part which the robot grasps. We estimate K poses at time indices k; so the set of all estimated states and articulated part poses can be written: $\mathcal{X} = \{\boldsymbol{x}_k, \mathbf{T}^A_k, \mathbf{T}^B_k\}_{k \in \mathsf{K}}$, dropping the transform reference frames for clarity.

We use P point clouds at times p which are each associated with a prediction on ξ. Without loss of generality, we set $\mathsf{P} = 1$ with one visual measurement at the beginning, although in future work we could add multiple predictions. We use K kinematic measurements at times k which are each associated with a pose estimate of the grasp point. The times k are only selected while the robot is in contact with the object and after the articulated part has been moved a certain distance d to avoid taking too many measurements. The set of all measurements are then grouped as $\mathcal{Z} = \{\mathcal{P}_p, \mathcal{K}_k\}_{p \in \mathsf{P}, k \in \mathsf{K}}$ where $\mathcal{P}$ are the point clouds and $\mathcal{K}$ the pose measurements from kinematics.

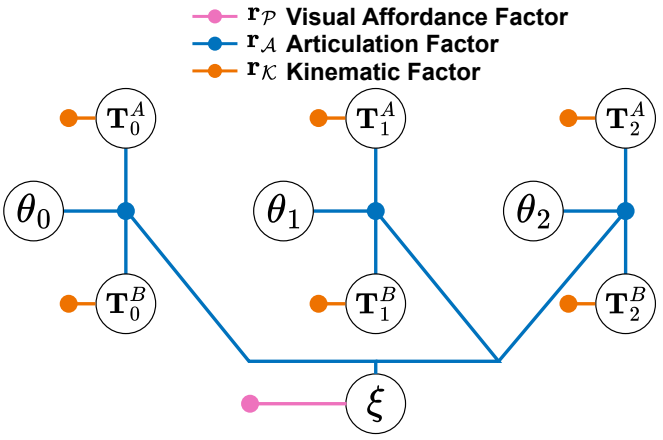

Fig. 2. The factor graph shows the variables we are estimating: $\mathbf{T}^A(t), \mathbf{T}^B(t), \theta(t)$ and ξ which exists at only one time step in the factor graph. We show three time steps including the initial visual affordance factor which provides a prior estimate on ξ as a unary factor.

V. Factor Graph Formulation

We maximize the likelihood of the measurements $\mathcal{Z}$, given the history of states $\mathcal{X}$:

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} p(\mathcal{X}|\mathcal{Z}) \propto p(\boldsymbol{x}_0) p(\mathcal{Z}|\mathcal{X}), \tag{6}$$

where $\mathcal{X}^*$ is our MAP estimate of the joint.

We assume the measurements are conditionally independent and corrupted by zero-mean Gaussian noise. Therefore, Eq. (6) can be expressed as the following least squares minimization:

$$\begin{aligned} \mathcal{X}^* = \arg\min_{\mathcal{X}} & \|\mathbf{r}_0\|^2_{\Sigma_0} + \\ & \sum_{p \in \mathsf{P}} \|\mathbf{r}_{\mathcal{P}_p}\|^2_{\Sigma_\mathcal{P}} + \sum_{k \in \mathsf{K}} \left(\|\mathbf{r}_{\mathcal{A}_k}\|^2_{\Sigma_\mathcal{A}} + \|\mathbf{r}_{\mathcal{K}_k}\|^2_{\Sigma_\mathcal{K}} \right), \end{aligned} \tag{7}$$

where each term is a residual $\mathbf{r}$ associated with a measurement type and assumed to be corrupted by zero-mean Gaussian noise with covariance according to measurement. A factor graph can be used to graphically represent Eq. (7) as shown in Fig 2 where large white circles represent the variables we would like to estimate and the smaller colored circles represent the residuals as factors. The implementation of the factors is detailed in Sec. VI-B.

VI. Proposed Method

This section describes the full system as shown in Fig 3.

A. Initialization

For the initialization module, we use the latest advances in deep learning for articulated objects and introduce a system of shared autonomy. First, a user is presented with a video feed of the object and clicks on the desired grasp point. With this query point, we use Kirillov *et al.*'s Segment Anything (SAM) [22] to segment a mask[1] of the non-static part.

The image mask and associated point cloud are then passed to the network which predicts the articulation affordance for each masked point. This affordance is parameterized as a

[1]As an additional, minor contribution, we open source the ROS wrapper for SAM which was used for this work `https://github.com/robot-learning-freiburg/ros_sam`

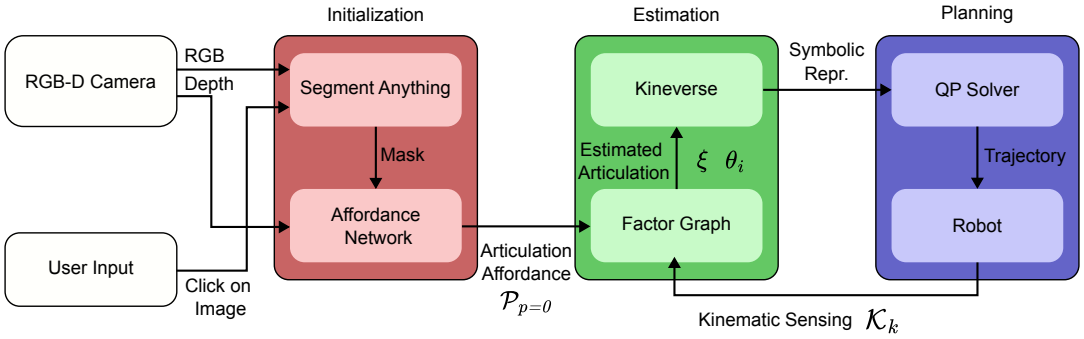

Fig. 3. Full system with information flow. An RGB-D camera provides RGB images which are segmented with the click prompt from a human user. This generates a mask on the articulated part which, with depth information from the camera predicts initial articulation parameters. This is provided to the factor graph which also uses kinematic measurements of the end effector to estimate the object articulation. The estimated articulation updates the symbolic math representation of both robot and object which is then formulated as a quadratic programming (QP) problem to solve for robot trajectory.

point cloud with the predicted, normalized acceleration of each point given a force opening the object. This idea was first introduced by Zeng *et al.* [23] where the concept was described as motion residual *flow* and later improved by Eisner *et al.* with Flowbot3D [7]. Our network is identical to Flowbot3D but, importantly, we change the prediction of flow to be in the camera frame, not the global frame. We found this necessary to enable the method to work with real objects of arbitrary pose, otherwise, the predicted flows were always directed away from the camera's optical center. Therefore, the output of the initialization step is the point cloud affordance $\mathcal{P}_{p=0}$, and the 3D point associated with the user's click which will be used as the first planning goal for the robot.

B. Factor Graph Estimation

For estimating the screw parameters, we use the factor graph in Fig. 2. This is similar to [12], however, instead of visually tracking the different parts we use kinematic sensing which enables online estimation. We also introduce a new articulation affordance factor to integrate the visual-based prediction from a neural network. Our method makes no assumption about what type of articulation is being estimated, e.g. revolute or prismatic.

1) Affordance Factor: To incorporate the predicted affordances we introduce an articulation affordance factor. First, we fit a plane to the affordance point cloud $\mathcal{P}$. We then create a new point cloud by adding the affordance to each point in $\mathcal{P}$, scaled by a small increment. This results in a second point cloud $\mathcal{P}^+$ which has been slightly shifted as though the object were opened. We fit a second plane to $\mathcal{P}^+$.

From the two planes, we find the intersecting line from the cross product of the normals. This defines the axis of rotation of a revolute joint. If the cross product is zero, or very small, then we assume the joint is prismatic as shown in Fig. 4. A similar approach was done by Zeng *et al.* [23] and results in a prediction $\tilde{\xi}$ on the articulation which can be used in Eq. (7) as the affordance factor:

$$\mathbf{r}_{\mathcal{P}_p} = \xi - \tilde{\xi}. \tag{8}$$

In future work we intend to train the network to predict its own uncertainty similar to [24], however, at the moment we select a fixed value ($\sigma_{\mathcal{P}} = 1e^{-3}$).

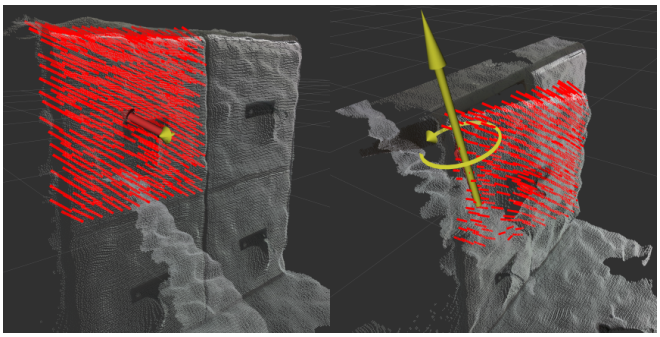

Fig. 4. Example affordance predictions from the neural network: prismatic left and revolute right. The small red lines are the output of the network, predicting articulation flow on the segmented points. The large red and yellow arrows indicated the resulting joint prediction from plane fitting.

2) Articulation Factor: From inspection of (4), we can see that all variables in the factor graph are related. The articulation residual can then be computed in Eq. (7) as:

$$\mathbf{r}_{\mathcal{K}_A} = \mathbf{T}^{twist}(\hat{\xi}, \theta_k) \boxminus \mathbf{T}_k^{B^{-1}} \mathbf{T}_k^A, \tag{9}$$

where $\boxminus$ is a pose differencing over the manifold using the logarithm map:

$$\mathbf{T}^A \boxminus \mathbf{T}^B = \mathrm{Log}(\mathbf{T}_k^{B^{-1}} \mathbf{T}_k^A) \in \mathfrak{so}(3). \tag{10}$$

3) Kinematic Factor: Kinematic measurements are added to the graph as unary pose factors on $\mathbf{T}^A$ and $\mathbf{T}^B$. We assume $\mathbf{T}^B$ doesn't move and reuse the initial grasp pose. The residual $\mathbf{r}_{\mathcal{K}_k}$ is the default $\mathrm{SE}(3)$ factor in GTSAM [25].

C. Online Motion Generation

This subsection covers the computation of the desired robot configurations $\mathbf{q} \in \mathbb{R}^7$, given the latest estimate of the articulation ξ. We model the forward kinematics of the robot end-effector as $\mathbf{T}_{\mathtt{WE}}(\mathbf{q})$, and the forward kinematics of our the articulated object as

$$\mathbf{T}_{\mathtt{WA}}(\theta, \xi) = \mathbf{T}_{\mathtt{WG}} \cdot \mathbf{T}^{twist}(\theta, \xi), \tag{11}$$

with $\mathbf{T}^{twist}$ computed using Eq. (3) with the latest estimated ξ and a goal θ. The pose $\mathbf{T}_{\mathtt{WG}}$ is a static transformation composed of the user defined grasp position $\mathbf{p}_{\mathtt{WG}}$ and a predetermined grasp orientation $\mathbf{R}_{\mathtt{WG}}$.

Once the robot grasps the object handle, we set $\theta_0 = 0$, which leads to $\mathbf{T}^{twist}(0, \xi_0) = \mathbb{I}$. We then progressively increment the desired articulation configuration $\theta_{t+1} = \theta_t + gv\Delta t$, with gv being a constant speed for opening/closing the articulation, up to the articulation limit after which we invert the sign of gv. For each θ_t, and given an estimate of ξ, we solve the inverse kinematics (IK) problem, subject to the condition $\mathbf{T}_{\mathtt{WE}}(\mathbf{q}_{t+1}) = \mathbf{T}_{\mathtt{WA}}(\theta_{t+1}, \xi)$. More specifically, we define the IK problem as a non-linear optimization problem where we encode the following task space constraints

$$\begin{aligned} \|\mathbf{p}_{\mathtt{WE}}(\mathbf{q}) - \mathbf{p}_{\mathtt{WA}}(\theta_t, \xi))\|_F^2 = 0 \\ \|\mathbf{R}_{\mathtt{WE}}(\mathbf{q}) - \mathbf{R}_{\mathtt{WA}}(\theta_t, \xi))\|_F^2 = 0 \end{aligned} \tag{12}$$

where $\|\cdot\|_F$ denotes a Frobenius norm.

We exploit the differentiability of the constraints in Eq. (12) w.r.t. to $\mathbf{q}$, to linearize the problem, and solve it sequentially until constraint satisfaction as a quadratic program (QP):

$$\begin{aligned} \arg\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{C}\mathbf{x} \quad \text{s.t.} \quad & \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \\ & \mathbf{lb}_A \leq \mathbf{A}\mathbf{x} \leq \mathbf{ub}_A, \end{aligned} \tag{13}$$

where $\mathbf{x} = \langle \dot{\mathbf{q}}, \mathbf{s} \rangle$ is a vector of joint velocities and slack variables $\mathbf{s}$, and $\mathbf{A}$ is the Jacobian of the task constraints and association with the slack variables. Eq. (13) also encodes bounds on robot joint positions and velocities.

We use the Kineverse articulation model framework [26] for representing both the robot and the articulated object forward kinematics and constraints, computing the Jacobians, as well as encoding and solving the problem in Eq. (13). Kineverse uses the CasADi symbolic math backend [27], enabling effortless computation of gradients for arbitrary expressions, such as the articulation model.

Finally, we command the resulting joint positions $\mathbf{q}_{t+1}$ to the robot in compliant mode. Therefore, if the articulation estimation ξ is inaccurate, the robot can comply with the physical articulation, leading to an end-effector pose that is different from $\mathbf{T}_{\mathtt{WG}}(\theta_{t+1}, \xi)$. The actual end-effector pose $\mathbf{T}_{\mathtt{WE},t+1}$ is added to the graph as a measurement on $\mathbf{T}^A$.

VII. Experiments

A. Implementation details

For experiments, we used the compliant KUKA LBR iiwa robot. We used an Intel Realsense D435 camera and Robotiq 140 two finger gripper. We implemented the factor graph using the GTSAM library [25].

B. Hand Guiding Experiments

In these experiments, we investigated the accuracy of our factor graph estimation module. We compared our method against Heppert *et al.* [12] which also uses factor graphs to estimate a screw parameterization. The authors kindly granted us access to their code for a direct comparison.

We physically attached the robot's end-effector to the box lid and hand-guided the robot motion, in gravity compensation mode, to open and close the box. For this experiment, we recorded both the robot joint positions, measured by the encoders, and the respective box lid poses, tracked with Vicon motion capture, as shown in Fig. 5.

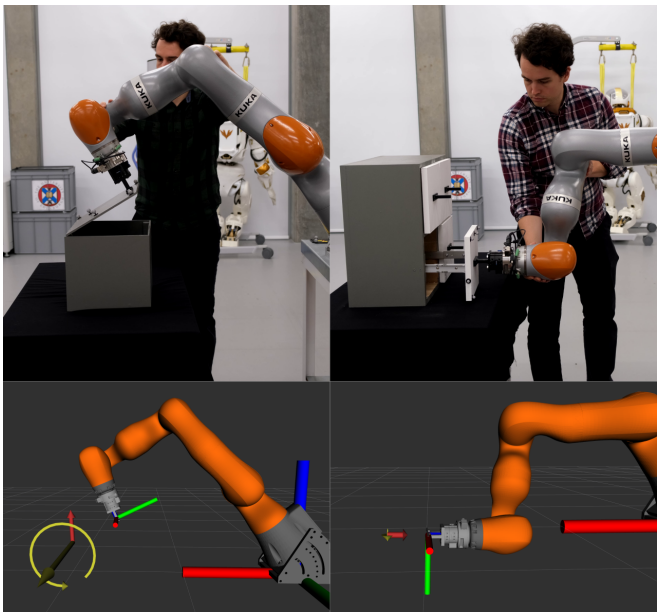

Fig. 5. Top: hand guiding experiments for revolute (left) and prismatic (right) joints. Bottom: the resulting estimated articulation. Yellow arrows show $\boldsymbol{\omega}$ while red arrows show $\mathbf{v}$. The large axis is the base frame of the robot which is used for $\mathtt{W}$ while the small axis is the estimated pose $\mathbf{T}^A$.

Similar to [12] we use the tangent similarity metric:

$$J(\mathbf{v}_{gt}, \mathbf{v}_{est}) = \frac{1}{\theta_{max} - \theta_{min}} \int_{\theta_{min}}^{\theta_{max}} \frac{\mathbf{v}_{gt}}{\|\mathbf{v}_{gt}\|} \cdot \frac{\mathbf{v}_{est}}{\|\mathbf{v}_{est}\|}, \tag{14}$$

where $\mathbf{v}_{gt}$ is the local linear velocity of the grasp point measured from Vicon and $\mathbf{v}_{est}$ is the estimated local velocity from the articulation model. We can compute $\mathbf{v}_{est}$ from ξ using the equation: $\mathbf{v}_{est} = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{c}$ where $\mathbf{c}$ is the contact point from kinematics. Since $\mathbf{v}_{gt}$ and $\mathbf{v}_{est}$ are normalized, they represent the direction of motion; therefore, their tangent similarity will be 1 when identical and 0 when perpendicular.

We recorded two hand guiding experiments, one for a revolute joint and one for a prismatic. First, we performed optimization over fixed increments, for example, optimizing over every 1° of rotation or $1\,\mathrm{cm}$ of translation. Next, we tested using fixed numbers of measurements equally spaced over the entire configuration range with full results shown in Fig. 6. In the factor graph, we make no distinction between prismatic or revolute. When estimating prismatic joints, $\boldsymbol{\omega}$ tends towards very small values. At the output, if $\|\boldsymbol{\omega}\| < 0.01$ we set $\boldsymbol{\omega} = 0_{3x1}$ and normalize $\mathbf{v}$.

We achieve good accuracy within a short window of measurements: after only 0.5° of rotation our estimator has an average tangent similarly of 0.90, after 1.0° this improves to 0.97. This enables online re-estimation in cases where the neural network prediction is wrong since the door will only need to be moved a small amount for the correct articulation to be estimated. Additionally, we show that for equally spaced measurements throughout the configuration range, as few as 3 measurements can be sufficient to accurately estimate the joint. In comparison with Heppert *et al.*, both methods have

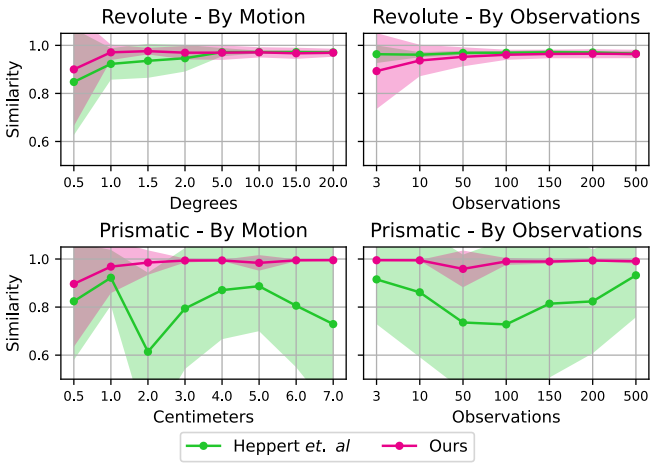

Fig. 6. Tangent similarity for hand guiding experiments. The solid line shows average error while the shaded region shows standard deviation.

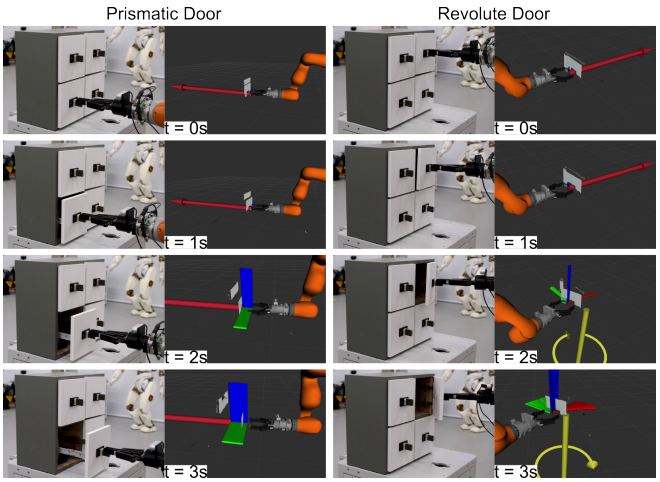

Fig. 7. Real-robot experiments: the red arrows indicate the estimated $\mathbf{v}$, yellow arrows show ω and are centered on the point q which lies on the axis of articulation. The axes show the measurements of $\mathbf{T}^A$. The left column shows the robot opening the prismatic drawer using an initial neural network prediction of the prismatic joint. The right column shows the robot opening the top right revolute door with an initial neural network prediction of prismatic.

similar performance for revolute joints while our method is better at distinguishing prismatic joints. This is likely because we check for the prismatic case whereas their method tended to confuse prismatic joints with very large revolute.

C. Full System Experiments

In these experiments, we tested the full pipeline, using vision and proprioceptive sensing together. The experiment protocol was as follows: the human user views the robot's camera feed which is looking at the same cabinet as in Fig. 1 and clicks on the image where to gasp. The robot then moves to the grasp goal and closes the gripper. Next, the robot moves using the learned articulation prediction from $\theta = 0$ to a specified upper bound. Estimation runs online, using kinematic sensing to update the model which is fed back to the controller in a closed loop. Eventually, the estimate converges to the correct estimate of the joint and the controller continues to open and close the door. We used a distance limit of $d = 2\,\mathrm{mm}$ or $d = 0.5°$ to trigger adding a kinematic measurement to the factor graph. We performed a new optimization after every 20 new measurements.

Two online estimation experiments are shown in Fig. 7. On the left, the robot initially received a correct prediction from the network that the joint was prismatic. Then, as the robot opened the drawer, the estimate of the joint was refined. On the right, the network incorrectly predicted prismatic and so the robot pulled backward on the handle. Because of compliance in the robot joints, the door opened by a few degrees at $1\,\mathrm{s}$. Opening the door by this small amount, rotated the end effector and allowed the correct articulation to be estimated from kinematics. By $3\,\mathrm{s}$ the joint estimate correctly converged and the robot was able to fully open the door. We repeated the full pipeline experiment 20 times on different doors and successfully opened the doors 16 times.

VIII. Discussion

Our method is able to accurately estimate articulation parameters so long as the door is able to move by a small amount, for example in the right column of Fig. 7. While this can correct for a poor neural network prediction, one limitation to our work is that certain incorrect predictions can lead to no motion of the robot in compliant mode, such as predicted articulations with an orthogonal direction of motion to the true articulation. For example, the robot was never able to open the bottom right door in Fig.1, which slides open to the right. This is because the network always predicted either a prismatic articulation, as a drawer, or a revolute articulation. As a result, the robot arm was unable to move and gain information when trying to open the door To account for this, in future work we will add an "exploration" module to the system. This could randomly apply forces in different directions until the arm is able to move and is similar to how humans discover articulations that are visually ambiguous.

Of the 20 full system trials attempted, 4 failed due to slipping of the gripper. Because we use kinematic measurements, we cannot differentiate between slipping and correct movement opening a door. In future work, we will investigate adding force/torque sensing and use a suction gripper.

IX. Conclusion

In this work, we present a complete system for opening of articulated objects using shared autonomy for which visual cues alone are insufficient to correctly estimate the articulation parameters. Our factor graph-based method incorporates both vision-based affordance predictions from a neural network with kinematic sensing in an analytical model for online estimation. We validated the system on a real robot and opened several articulated objects, using online estimates in a closed loop. We succeeded in 16/20 consecutive trials and properly estimated the articulation, despite of the original affordance prediction being incorrect for 13/16 of those trials. Nevertheless, in those cases, the robot was able to succeed due to the online re-estimation of the articulation through the compliant interaction. In future work, we will incorporate an exploration module to enable the discovery of articulations that might be undetectable from the network predictions.

REFERENCES

[1] A. Jain and C. C. Kemp, "Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1807–1814.

[2] P. Tresadern and I. Reid, "Articulated structure from motion by factorization," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 1110–1115.

[3] J. Sturm, C. Stachniss, and W. Burgard, "A probabilistic framework for learning kinematic models of articulated objects," *Journal of Artificial Intelligence Research*, vol. 41, no. 2, pp. 477–526, 2011.

[4] R. Martín-Martín and O. Brock, "Coupled recursive estimation for online interactive perception of articulated objects," *International Journal of Robotics Research*, vol. 41, pp. 741–777, 7 2022.

[5] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," *arXiv preprint arXiv:2103.10534*, 2021.

[6] G. Schiavi, P. Wulkop, G. Rizzi, L. Ott, R. Siegwart, and J. J. Chung, "Learning agent-aware affordances for closed-loop interaction with articulated objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5916–5922.

[7] B. Eisner*, H. Zhang*, and D. Held, "Flowbot3d: Learning 3d articulation flow to manipulate articulated objects," in *Robotics: Science and Systems (RSS)*, 2022.

[8] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. Sukhatme, "Interactive Perception: Leveraging Action in Perception and Perception in Action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[9] T. Lips and F. wyffels, "Revisiting proprioceptive sensing for articulated object manipulation," *arXiv preprint arXiv:2305.09584*, 2023.

[10] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 909–918.

[11] D. Katz, A. Orthey, and O. Brock, "Interactive Perception of Articulated Objects," in *Experimental Robotics: The 12th International Symposium on Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and G. Sukhatme, Eds. Berlin, Heidelberg: Springer, 2014, pp. 301–315.

[12] N. Heppert, T. Migimatsu, B. Yi, C. Chen, and J. Bohg, "Category-independent articulated object tracking with factor graphs," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3800–3807.

[13] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3703–3712.

[14] A. Jain, R. Lioutikov, C. Chuck, and S. Niekum, "ScrewNet: Category-Independent Articulation Model Estimation From Depth Images Using Screw Theory," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 670–13 677.

[15] H. Jiang, Y. Mao, M. Savva, and A. X. Chang, "Opd: Single-view 3d openable part detection," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 410–426.

[16] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 6793–6803.

[17] Z. Xu, Z. He, and S. Song, "Universal manipulation policy network for articulated objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2447–2454, 2022.

[18] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[19] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building Digital Twins of Articulated Objects from Interaction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5606–5616.

[20] N. Nie, S. Y. Gadre, K. Ehsani, and S. Song, "Structure from Action: Learning Interactions for 3D Articulated Object Structure Discovery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1222–1229.

[21] R. M. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. CRC Press, 1994.

[22] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[23] V. Zeng, T. E. Lee, J. Liang, and O. Kroemer, "Visual identification of articulated object parts," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2443–2450.

[24] R. L. Russell and C. Reale, "Multivariate uncertainty in deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7937–7943, 2022.

[25] F. Dellaert and GTSAM Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)

[26] A. Röfer, G. Bartels, W. Burgard, A. Valada, and M. Beetz, "Kineverse: A symbolic articulation model framework for model-agnostic mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3372–3379, 2022.

[27] J. A. E. 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.