



Hand–object configuration estimation using particle filters for dexterous in-hand manipulation

The International Journal of
Robotics Research
1–15
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0278364919883343
journals.sagepub.com/home/ijr


Kaiyu Hang* , Walter G. Bircher*, Andrew S. Morgan and Aaron M. Dollar

Abstract

We consider the problem of in-hand dexterous manipulation with a focus on unknown or uncertain hand–object parameters, such as hand configuration, object pose within hand, and contact positions. In particular, in this work we formulate a generic framework for hand–object configuration estimation using underactuated hands as an example. Owing to the passive reconfigurability and the lack of encoders in the hand’s joints, it is challenging to estimate, plan, and actively control underactuated manipulation. By modeling the grasp constraints, we present a particle filter-based framework to estimate the hand configuration. Specifically, given an arbitrary grasp, we start by sampling a set of hand configuration hypotheses and then randomly manipulate the object within the hand. While observing the object’s movements as evidence using an external camera, which is not necessarily calibrated with the hand frame, our estimator calculates the likelihood of each hypothesis to iteratively estimate the hand configuration. Once converged, the estimator is used to track the hand configuration in real time for future manipulations. Thereafter, we develop an algorithm to precisely plan and control the underactuated manipulation to move the grasped object to desired poses. In contrast to most other dexterous manipulation approaches, our framework does not require any tactile sensing or joint encoders, and can directly operate on any novel objects, without requiring a model of the object a priori. We implemented our framework on both the Yale Model O hand and the Yale T42 hand. The results show that the estimation is accurate for different objects, and that the framework can be easily adapted across different underactuated hand models. In the end, we evaluated our planning and control algorithm with handwriting tasks, and demonstrated the effectiveness of the proposed framework.

Keywords

Dexterous Manipulation, Underactuated Manipulation, Soft Manipulation, Hand-Object Configuration Estimation, Particle Filtering

1. Introduction

Dexterous manipulation is an essential ability for robots to achieve small-scale object manipulation tasks without unnecessary large arm motions. However, owing to the complexity in the modeling of highly dynamic hand–object systems, the planning and control of dexterous manipulation still remains challenging (Bütepage et al., 2019; Okamura et al., 2000). To keep the problem tractable, the majority of research works have investigated various subproblems, ranging from contact modeling (Bicchi and Kumar, 2000; Han et al., 1997; Rojas and Dollar, 2016), grasp planning (Bohg et al., 2014; Hang et al., 2017), grasp control and stability maintenance (Li et al., 2014), to finger gaiting and sliding planning (Trinkle and Paul, 1990; Xu et al., 2010), tactile-based contact estimation (Koval et al., 2016), hand–object state estimation (Corcoran and Platt, 2010), etc. While some works have successfully demonstrated in-hand

manipulation capabilities, they usually rely on many sensing modalities, precise models of objects, simplified task requirements, or a huge dataset for training (Bütepage et al., 2019; OpenAI et al., 2018; Tahara et al., 2010).

In order to mitigate the difficulties of working with fully actuated dexterous hands, underactuated or soft robotic hands have been developed to simplify the problem of grasping (Deimel and Brock, 2016; Dollar and Howe,

Department of Mechanical Engineering and Material Science, Yale University, New Haven, CT, USA

*These authors contributed equally to this work.

Corresponding author:

Kaiyu Hang, Department of Mechanical Engineering and Material Science, GrabLab, 9 Hillhouse Avenue, Yale University, New Haven, CT 06511, USA.

Email: kaiyu.hang@yale.edu

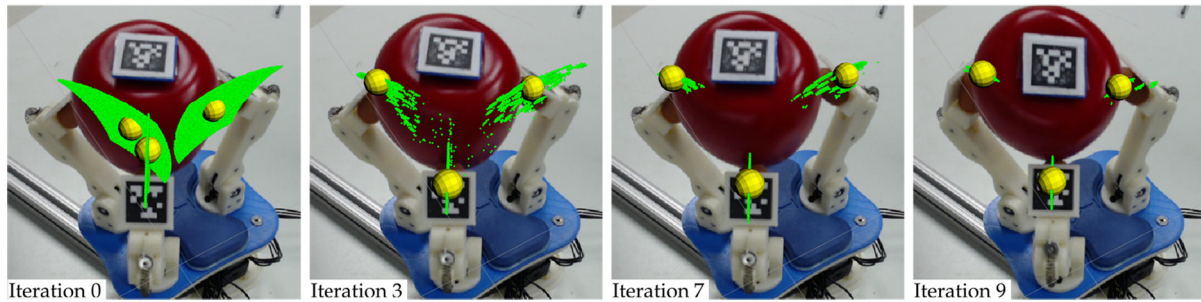


Fig. 1. An apple object is grasped by the Yale Model O underactuated hand, for which the hand joint configuration is unknown. Our particle filter estimates the hand configuration by randomly manipulating the object within hand, while observing only the movements of the object tip’s pose. The green points depict the potential contact positions hypothesized by the joint configuration particles, and the yellow spheres indicate the weighted average of all particles. After nine iterations, our particle filter has converged to the real hand configuration. The markers on the fingertips are used only for ground-truth comparison.

2010; Santina et al., 2015). Leveraging their passive reconfigurability, manipulation stability can be easily achieved, and hand–object motion models have been derived to control the in-hand movements of the grasped object (Calli and Dollar, 2019; Ma and Dollar, 2013). Nonetheless, since one cannot control each individual joint separately in an underactuated hand, there is usually no joint encoders. In addition, owing to the nature of underactuation, it is difficult to acquire precise information about underactuated hand–object configurations. As such, existing approaches have focused on only planar manipulation tasks, and often require precise object models to be known *a priori* (Bircher et al., 2017). However, the problem of underactuated manipulation in three dimensions, especially for novel objects, remains ignored.

This work builds upon our observation that, in many daily manipulation tasks, it is not necessary to know the object model in order to accomplish a task. For example, in order to insert a key grasped by human fingertips, we mainly care about how the tip of the key is posed. If the fingertip contacts are stable on the key, the control of the tip of key can be boiled down to the problem of regulating the positions of the grasping fingertips, which are constrained by each other’s relative positions. The same scenarios can be seen in many other applications, such as handwriting or peg-in-hole problems. Also note that, when manipulating rigid objects, as long as we can control the movement of one point on the object, all other points will move in a deterministic and predictable way, represented by the fixed transformations on the object body. In this work, we term the point on the object to be manipulated (e.g., the tip of key) as the *point of manipulation* (POM), and focus on the problem of controlling its movement.

Although in-hand manipulation can be achieved using different strategies, e.g., by finger gaiting (Xu et al., 2010), rolling or sliding contacts between the object and fingertips (Trinkle and Paul, 1990), dual-arm coordination to push and reconfigure object in-hand (Cruciani et al., 2018), or even extrinsic dexterity (Dafle et al., 2014), in this work we focus on small-scale manipulation of the object’s POM. In

particular, as seen in many daily life manipulation tasks such as key insertion or handwriting, we consider fixed contacts between the object and fingertips, and address the problem of manipulating the POM by regulating relative fingertip positions (Ma and Dollar, 2011). To ensure grasp stability and focus on the problem of in-hand manipulation of objects, we opt to adopt underactuated robotic hands, which can easily provide stable grasps and do not require complicated sensing or control during manipulation.

However, in order to precisely manipulate the POM using fingertips, the first problem one should consider is how to map the hand’s underactuated inputs to the movement of the fingertips, which then, in turn, map to the motion of the object. Different from the underactuated manipulation of known objects in planar setups (Calli and Dollar, 2019; Ma and Dollar, 2013), it is infeasible to model this 3D underactuated manipulation system by directly mapping the actuation inputs to the object movements, especially given that we aim at manipulating unknown objects. Instead, it is important to know the hand–object configuration to establish the mapping, in order to enable the manipulation planning and control. As such, a major problem we address is to estimate the hand configuration, without using any joint encoders. For this, we model the relationship between the POM and the grasping contacts as a rigid-body constraint, with which we analyze how the hand configuration can be modeled in a function mapping from the actuation inputs to the movement of the POM. Based on the motion model, we develop a particle filter-based framework to estimate the hand configuration. In brief, once the object is grasped, we sample a set of hand configuration hypotheses and randomly manipulate the object in a small range. Meanwhile, our system observes the movements of the POM as the evidence for the particle filter to calculate the likelihood of each hypothesis, and will finally obtain the real hand configuration when the particles are converged. Thereafter, the hand configuration can be tracked in real time in future manipulations. An example is shown in Figure 1.

In order to control the movement of the POM, we develop a method to compute a mapping between the POM’s motion and the actuation inputs, and show that we can manipulate the POM to trace specified waypoints. In the rest of this article, we review related works in Section 2, explain the generic particle filter-based framework in Section 3, and describe the underactuated manipulation planning and control in Section 4. Thereafter, we provide an instantiation of the proposed framework using the Yale OpenHand project in Section 5, present experimental results in Section 6, and finally conclude in Section 7.

2. Related work

In relation to the scope of this work, we review related works with respect to the necessary components in manipulation systems, the modeling of hand–object systems, and underactuated manipulation in previous works.

2.1. System components

Dexterous in-hand manipulation is a multifaceted system-level problem. Dependent on the hand’s mechanical properties and the sensors available in a system, the difficulties of developing a robust and accurate system reside in a variety of subproblems. For example, for a fully actuated robot hand, acquiring a grasp and maintaining its stability during manipulation, even without the consideration of task requirements, is already a challenging problem (Hang et al., 2016). If the target object is unknown, approaches have been developed to estimate the object’s geometry before any grasping or manipulation can be considered (Li et al., 2016; Sommer et al., 2014). On the other hand, when the grasp planning and manipulation stability problems are mitigated by soft manipulators, the problem is then shifted towards the coordination between hand’s degrees of actuation, hand synergies, and controllable hand–object motions (Calli and Dollar, 2019; Deimel et al., 2017; Santina et al., 2018).

From a system design perspective, most research works systematically address the problem of dexterous manipulation in three steps (Okamura et al., 2000). First, in the pre-grasping phase, grasp planning algorithms are developed to find contact positions (Bohg et al., 2014), hand configurations (Borst et al., 2002), and arm reaching motions (Eppner and Brock, 2017; Haustein et al., 2017; Kimmel et al., 2018; Mahler et al., 2017). Thereafter, the generated plan is executed in the grasping phase, wherein perception uncertainties are managed, so that the object pose can be more accurately estimated (Corcoran and Platt, 2010; Koval et al., 2016), and the grasp can be locally adjusted based on the vision or tactile feedback to improve stability (Chebotar et al., 2016). Finally, the post-grasping phase deals with the manipulation planning and control, in order to use the grasped object for specific tasks (Calli and Dollar, 2019; Tahara et al., 2010).

In this work, because the grasp acquisition is relatively easy for our underactuated hands, we focus on the post-grasping phase to estimate the hand–object configuration for manipulation planning and control.

2.2. Hand–object system modeling

Hand–object systems are commonly modeled based on the precise hand model, object model, and contact model. When a fully actuated dexterous hand is used, synergies can be generated to reduce the control complexity (Bicchi et al., 2011; Ciocarlie et al., 2007). By modeling contacts as point, rolling, sliding, or area contacts, in-hand object motions can be analytically synthesized (Bicchi and Kumar, 2000; Han et al., 1997; Sundaralingam and Hermans, 2017). Within an object-centric virtual frame, tripod manipulation can be controlled with an impedance controller (Li et al., 2014). To handle novel objects and uncertainties in perception, recent research has been focused on using sensors to address object shape completion (Li et al., 2016), object pose estimation (Koval et al., 2016), and grasp stability maintenance (Hang et al., 2016).

Based on sequential importance resampling (SIR), particle filters have been adopted for modeling the nonlinear system dynamics, such as tactile sensing-based contact estimation (Corcoran and Platt, 2010; Koval et al., 2016), and vision-based sequential multi-object configuration estimation (Sui et al., 2015). However, existing filter-based approaches mostly only focus on larger-scale manipulation tasks, including non-prehensile manipulation or whole-hand grasping. In the problem of in-hand manipulation, especially for systems with limited perception capabilities, the problem of estimating the hand–object system to deal with the uncertainties has still not been addressed. In comparison with the existing approaches, in this work we aim at manipulating novel objects without knowing their geometric models, and will establish the mapping between the actuation inputs and the POM’s motion online while the object is being manipulated by an underactuated robotic hand. For this, we will show that our system can first estimate the hand–object configuration, and then model it into the mapping to enable the underactuated in-hand manipulation.

In addition, different from the efforts in human hand motion tracking, which is usually used for gesture recognition and human–computer interaction purposes, we are more interested in accurately estimating the robot hand configuration in order to control the hand in more complex manipulation tasks. Moreover, as opposed to the human hand tracking approaches, which can use special devices or purely RGB-D input to only track the hand motion and do not model the hand kinematics (Kim et al., 2009; Sridhar et al., 2013), we aim to understand the true hand–object configuration, including both object and joint configurations. We accomplish all of this using minimal sensor input, and model the object manipulation process in a more explicit framework to enable in-hand manipulation.

2.3. Dexterous underactuated manipulation

In order to alleviate the sophisticated modeling of the dynamics and uncertainties involved in grasping and manipulation, soft manipulators have been developed to shift the paradigm of precise fully actuated planning and control to the modeling of underactuated and passive mechanisms, which can provide robust solutions to sensorless compliant stability maintenance (Deimel and Brock, 2016; Dollar and Howe, 2010; Santina et al., 2015). Owing to their compliant nature, soft hands are particularly useful in manipulation tasks where physical interactions between the hand, object, and the environment are expected (Della Santina et al., 2017; Eppner et al., 2015). Leveraging the compliant interactions featured by the underactuated hands, certain manipulation tasks can be robustly achieved in an open-loop manner, which can be very challenging using a fully actuated hand, such as using environmental constraints to grasp objects, or interacting with both the object and the environment at the same time (Hang et al., 2019; Páll et al., 2018; Salvietti et al., 2015).

Although underactuated hands are not commonly considered to be as good at dexterous manipulation as their fully actuated counterparts, they are shown to be able to successfully achieve certain manipulation tasks. Assuming a simple circular object, one can analytically plan the motion of the object by minimizing the system energy (Ma and Dollar, 2013). With vision feedback and learning techniques, planar underactuated manipulation of shape primitives have been demonstrated (Calli et al., 2018; Liarokapis and Dollar, 2016). However, the state-of-the-art underactuated manipulation solutions usually rely on learning-based approaches, which require time-consuming data collection and training, to represent the motion primitives of the hand-object system (Liarokapis and Dollar, 2019), and many of them have only demonstrated their capabilities in planar manipulation tasks (Sintov et al., 2019). The limitations in learning-based approaches mostly come from the lack of modeling of the exact hand-object system, and as such the manipulation motions could be only implicitly described by the learned model, without an explicit understanding of how the actuation inputs will change the object configuration. To push the limits of the manipulation capabilities, in this work we develop a particle filter-based framework, and show that the hand configuration can be estimated without joint encoders, and that underactuated manipulation can be extended to three dimensions with explicit actuation planning and control.

3. Hand-object configuration estimation

In this section, we first introduce preliminaries of particle filters (Thrun et al., 2005), and then formulate a generic hand-object motion model based on necessary grasp constraints. By integrating the hand motion model with the POM, we finally establish the hand configuration estimation framework.

3.1. Particle filters

Particle filters are a family of non-parametric Bayes filters based on SIR. In a broad range of robotic applications, it is adopted to localize mobile robots (Thrun et al., 2005), estimate object poses (Koval et al., 2016), approximating hand-object configurations (Corcoran and Platt, 2010), etc. As the key process, a particle filter approximates the distribution of a system's hidden states by a set of random state samples, and recursively propagates and refines the sampled set in terms of the observations, with especially good functionality for nonlinear systems. Intuitively, the hidden states of a system are usually some system properties that cannot be directly obtained from the available sensors, such as the joint configurations of underactuated robotic hands when only motor positions are known. However, since we can, for example, observe the pose of the POM, we can evaluate how likely a hypothesized joint configuration is the true configuration in terms of our control inputs and observations. As such, the refinement of the sampled set can be done by importance resampling in terms of the likelihood of hypotheses.

Ideally, denoted by $\chi_t = \{x_t^1, x_t^2, \dots, x_t^M\}$ the M random samples, called particles, of the hidden states at time t , the likelihood of having a particle x_t^m in χ_t is determined by the posterior:

$$x_t^m \sim p(x_t | z_{1:t}, u_{1:t}) \quad (1)$$

where $z_{1:t}$ is the observations until time t , and $u_{1:t}$ is the control inputs. As implied by (1), if we have a sufficient number of particles that are drawn from the true underlying posterior distribution, the particle set χ_t can be seen as a non-parametric representation of the state distribution. However, in practice it is often infeasible to explicitly derive a true posterior distribution $p(x_t | z_{1:t}, u_{1:t})$, a particle filter needs to work with a *proposal distribution* π to recursively represent the approximation of the distribution:

$$\pi(x_t) \sim p(x_t | x_{t-1}, u_t) \pi(x_{t-1}) \quad (2)$$

Assuming a set of particles χ_t has been sampled from π , based on the definition of importance sampling, the relationship between p and π is given by

$$\int_S p(x_t) dx_t = \eta \sum_{m=1}^M I(x_t^m \in S) w^m, \quad x_t^m \in \chi_t \quad (3)$$

where $\eta = \left[\sum_{m=1}^M w^m \right]^{-1}$ is a normalization factor, S is an arbitrary valid region in the state space, and $I(\cdot) \mapsto [0, 1]$ is an indicator function denoting whether a sample $x_t^m \in \chi_t$ is in S . Note that there is a weight factor, $w^m = p(x_t^m) / \pi(x_t^m)$, associated with each particle. This weight factor is used to offset the difference between p and π , so that the probability distribution represented by the weighted χ_t resembles the true probability of p . Assuming the system is Markovian, the weight factor is derived in an expanded

Algorithm 1. Particle filter.

Input: χ_{t-1}, u_t, z_t
Output: χ_t

- 1: $\chi_t \leftarrow \emptyset$
- 2: **for** $m = 1 : M$ **do**
- 3: sample $x_t^m \sim \pi(x_t^m | x_{t-1}^m, u_t)$ ▷ Propagation
- 4: $w^m = p(x_t^m) / \pi(x_t^m)$ ▷ Equation (4)
- 5: **end for**
- 6: normalize $\{w^1, \dots, w^M\}$
- 7: **for** $m = 1 : M$ **do**
- 8: sample i with probability $\propto w^i$ ▷ Resampling
- 9: $\chi_t.append(x_t^i)$ ▷ New Particles
- 10: **end for**
- 11: **return** χ_t

form to utilize the observation z_t when the system's state x_t is not directly available:

$$w^m = \frac{p(z_t | x_t^m) p(x_t^m | x_{t-1}^m, u_t)}{\pi(x_t^m | x_{t-1}^m, u_t, z_t)} \quad (4)$$

where the probability $p(x_t^m | x_{t-1}^m, u_t)$ is called a motion model, and describes how the system transitions under the control input u_t . See Thrun et al. (2005) for more details.

Using the particle weight in (4), a particle filter will then recursively refine the particle set χ_t using importance resampling. Briefly, each time after a control input is given to the system, the particles will be propagated using the motion model to transition every x_{t-1}^m to x_t^m , and then the weight of each particle is calculated. Based on the new particle weights, the particle set is resampled with the probability proportional to the weights. After sufficient iterations, the particles which are incorrect hypotheses of the state will be eliminated, and the surviving particles will converge and will be re-distributed according to the true underlying distribution p . The basic process of a particle filter is summarized in Algorithm 1.

3.2. Hand–object motion model

For most robotic hands, either fully actuated or underactuated, there exists an underlying function that maps from the actuation inputs to the hand configurations. For fully actuated hands, this function is the forward kinematics function, which can usually be obtained in an analytical form. For underactuated hands, although a forward kinematics function can exist for free-swing motions, such a function is in general much more complex since the hand configuration is also affected by the external contacts on different links of the hand. As such, in order to derive a hand–object motion model in terms of the actuation inputs, most systems require tactile sensors to localize and measure the magnitude of contact forces (Ajoudani et al., 2014), or need precise object models to analyze the pose and motion of the object (Koval et al., 2016).

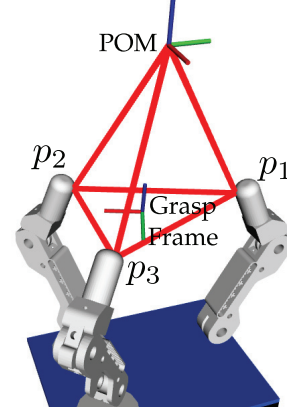


Fig. 2. A grasp frame defined by three contact points on the Yale Model O hand. Without an object model, the manipulation of the POM is determined by the relative positions between the fingertips and the POM.

However, for the purpose of manipulating an object in-hand, especially after the object is firmly grasped by the fingertips, we found that the manipulation can be described purely by the relative positions between the grasping points and a point on the object, termed as POM. These relative positions can uniquely define a grasp constraint and enable us to explicitly model the mapping between the actuation inputs and the underactuated hand configurations. In this work, we focus on small-scale in-hand manipulations that are used to finely adjust the pose of the object, and do not consider the larger-scale manipulations where contacts could move on the object surface, such as finger gaing, sliding, or regrasping by extrinsic dexterity. Therefore, we herein assume that the contacts between the fingertips and the object are fixed point contacts.

Concretely, let us denote by $\mathcal{P} = \{p_1, \dots, p_N\}$, $p_n \in \mathbb{R}^3$, the N fingertip contacts, we will first define a grasp frame based on the contact positions. For now, we assume a grasp is composed of at least three fingertip contacts. For robotic hands with $N = 2$, this definition can be adapted as demonstrated later in Section 5.2. Similar to Hang et al. (2016), the grasp frame can be defined using any three contacts in \mathcal{P} . Assuming we use the first three contacts, the grasp frame \mathcal{G} is defined as

$$\begin{aligned} \mathcal{G} &= [\rho_x, \rho_y, \rho_z | \rho_o] \in SE(3) \\ \rho_o &= \frac{1}{3}(p_1 + p_2 + p_3) \\ \rho_x &= \frac{p_2 - p_1}{\|p_2 - p_1\|} \\ \rho_z &= \frac{(p_3 - p_2) \times \rho_x}{\|(p_3 - p_2) \times \rho_x\|} \\ \rho_y &= \rho_z \times \rho_x \end{aligned} \quad (5)$$

As illustrated in Figure 2, we can see that those three contacts uniquely define a frame at the center of the grasp.

Since all fingertip contacts are firmly fixed on the object, additional grasp contacts, $\{p_4, \dots, p_N\} \subset \mathcal{P}$ (not shown in the figure which has a three-fingered hand), will keep fixed positions in \mathcal{G} . This further gives us a grasp constraint during manipulation that, as long as the contact positions do not change, the motion of any three contacts actually uniquely describes the motion of all other contacts.

If the grasped object is rigid, the POM point on the object also has a fixed relationship between itself and all grasp contacts. Formally, denoting the position of POM by $p_o \in \mathbb{R}^3$, we have the manipulation constraint:

$$\forall p_i, p_j \in \{\mathcal{P}, p_o\} : \|p_i^{\mathcal{G}} - p_j^{\mathcal{G}}\| = \text{const}(d_{i,j}) \quad (6)$$

where $p_i^{\mathcal{G}}$ is the position of p_i in the grasp frame \mathcal{G} , and $\text{const}(d_{i,j}) \in \mathbb{R}^+$ is the constant distance between points. This constraint shows that, see Figure 2, the motion of POM is fully constrained by the motion of grasp contacts, and can be uniquely described by the motion of any three contacts in \mathcal{P} . Although it is not a concern for three-fingered hands, for which the three contacts cannot be colinear to maintain grasp stability, depending on the kinematic structure and the hand mechanism designs, it is important to ensure that the chosen three contacts are never colinear in order to construct the grasp frame, especially for hands with more than three fingers. Even if it is difficult to derive a motion model for underactuated hands, with the constraint in (6), the motion of the hand-object system can be modeled as a parallel mechanism, for which the actuation inputs can be explicitly mapped to the object movements (Borràs and Dollar, 2015; Ma and Dollar, 2013).

Let $\mathcal{A} \subset \mathbb{R}^{d_a}$ be the hand actuation space of dimension d_a , and $\mathcal{C} \subset \mathbb{R}^{d_c}$ be the hand configuration space of dimension d_c . This mapping is conducted in three steps. First, once a grasp is obtained with hand configuration $c_{t-1} \in \mathcal{C}$ at time $t-1$, given an actuation input $u_t \in \mathcal{A}$, the new hand configuration is obtained by a function:

$$c_t = \Gamma(c_{t-1}, u_t) \text{ s.t. Equation(6)} \quad (7)$$

Second, based on the hand configuration c_t , we can calculate the grasp contacts and then use (5) to obtain the grasp frame \mathcal{G}_t at time t . Finally, as the transformation from the grasp frame to the POM frame is fixed, the POM's pose $\phi_t \in SE(3)$ at time t is calculated by

$$\phi_t = \mathcal{G}_t \cdot \text{inv}(\mathcal{G}_{t-1}) \cdot \phi_{t-1} \quad (8)$$

where $\text{inv}(\cdot)$ is the inverse transformation. In fact, this three-step process composes the motion model of a hand-object system. In this work, we represent this motion model $\Lambda : \mathcal{C} \times \mathcal{A} \times SE(3) \rightarrow SE(3)$ as

$$\phi_t = \Lambda(c_{t-1}, u_t, \phi_{t-1}) \text{ s.t. Equation(6)} \quad (9)$$

We can see that (9) is a recursive motion model, which describes the object POM's movement based on its current

hand configuration and the actuation inputs. This is particularly useful when we want to locally predict the motion of POM, and it enables us to plan the motion by purely regulating our control in the \mathcal{A} space.

However, there is a *causality dilemma* in the calculation of the Γ function. In practice, before we already know the constraints in (6), we cannot calculate $\Gamma(\cdot)$ to obtain the hand configuration. On the other hand, if we cannot calculate the hand configuration $\Gamma(\cdot)$, the contact positions in (6) would not be available for us to compose the constraint. Therefore, it prevents us from starting the recursion of (9). In order to break this dilemma, we will next develop a particle filter-based method to estimate the hand configuration c_t .

3.3. Hand configuration estimation

For most underactuated robotic hands, there are no encoders in the finger joints, and the hand configuration cannot be obtained directly from other sensors. In addition, since the hand-object motion model is usually highly nonlinear, it is difficult to estimate and track the joint configurations. Using the motion model derived in Section 3.2, we propose to estimate the hand configuration indirectly using particle filters.

To this end, we model a hand configuration c_{t-1}^m as a particle to represent a hypothesis in the particle filter. Given the recursive motion model of (7), we can observe that our hand-object system is Markovian, and that the hand configuration changes resulted from each actuation input at a specific time point is a small amount of local motion. Therefore, although the global hand motion is quite nonlinear, its recursive local changes can be described by

$$p(c_t^m | c_{t-1}^m, u_t) \sim \mathcal{N}(\Gamma(c_{t-1}^m, u_t), \Sigma_{\text{motion}}) \quad (10)$$

where $\mathcal{N}(\cdot)$ is a multivariate Gaussian distribution, and $\Sigma_{\text{motion}} \in \mathbb{R}^{d_c \times d_c}$ is the covariance matrix that accounts for the system uncertainties caused by control and mechanical errors. Assuming that the system is able to observe the true pose $\bar{\phi}_t$ of the POM, the weight of this particle is then calculated in terms of the difference between the prediction of (9) and the true observation. Using the predicted c_t^m and (9), we can transition the pose of POM from ϕ_{t-1}^m to ϕ_t^m , and the weight of a particle is then updated by

$$w^m = \frac{1}{\sqrt{2\pi\sigma_p^2}} e^{-\frac{\|\phi_t^m - \bar{\phi}_t\|^2}{2\sigma_p^2}} \quad (11)$$

where $\sigma_p^2 \in \mathbb{R}$ is the variance factor accounting for the perception uncertainty. Again note that since POM's movement described by the recursive model in (9) relates to only the local pose changes, we choose Gaussian distribution in (11) to re-weight the particles despite its global nonlinear motion. From (11) we can observe an important property of our framework that our system does not require any

Algorithm 2. Hand Configuration Estimator.

Input: C_{t-1}
Output: C_t

- 1: $C_t \leftarrow \emptyset$
- 2: $u_t \leftarrow \text{Actuation.Random}()$
- 3: for $m = 1 : M$ do
- 4: predict $c_t^m \sim \mathcal{N}(\Gamma(c_{t-1}^m, u_t), \Sigma_{\text{motion}})$ ▷ Equation (10)
- 5: $\bar{\phi}_t = \text{Camera.Perceive}()$
 $\phi_t^m = \Lambda(c_{t-1}^m, u_t, \bar{\phi}_{t-1})$ ▷ Equation (9)
- 7: $w^m = \frac{1}{\sqrt{2\pi\sigma_p^2}} e^{-\frac{\|\phi_t^m - \bar{\phi}_t\|^2}{2\sigma_p^2}}$ ▷ Equation (11)
- 8: **end for**
- 9: normalize $\{w^1, \dots, w^M\}$
- 10: **for** $m = 1 : M$ **do**
- 11: sample i with probability $\propto w^i$ ▷ Resampling
- 12: $C_t.append(c_t^i)$ ▷ New Particles
- 13: **end for**
- 14: **return** C_t

calibration between the hand and the camera which is used to observe the POM's movement. This is because we only need to know the relative movement $\|\phi_t^m - \bar{\phi}_t\|$ of the POM, and it does not matter in what frame this is measured.

Based on the updated particle weights, the importance resampling procedure in Section 3.1 can be invoked to redistribute the particles towards the true underlying density distribution. In practice, leveraging the passive compliance of underactuated hands, we can generate random u_t to twiddle the object around its initial grasp without dropping the object. This enables us to physically trigger the particle propagations and estimate the hand configuration in a particle filter-based framework.

First, once the object is firmly grasped by the hand, we generate a set of hypothesis particles, $C_0 = \{c_0^1, \dots, c_0^M\}$, from a prior distribution $C_0 \sim \pi_0$. Thereafter, we randomly generate actuation input u_t , and predict how the particles will move. By measuring the difference between the predicted pose of the POM and the perceived one, we can update the particle weights using (11). Finally, the particles are resampled with the probability proportional to their weights to generate a new particle set. One iteration of this procedure is summarized in Algorithm 2. Note that in step 6, the POM's pose is transitioned from the perceived $\bar{\phi}_{t-1}$, instead of the predicted ϕ_{t-1}^m .

4. Manipulation planning and control

Once the hand configuration is estimated, the motion of the POM can be recursively described by (9), and we are able to predict where the POM will move to given an actuation input u_t . However, if we want to control the motion of the POM, we need to reverse this function, such that we can calculate what actuation input should be used if we want to move the POM to a certain pose. Formally, this reversed function $\tilde{\Lambda} : \mathcal{C} \times SE(3) \times SE(3) \mapsto \mathcal{A}$ takes the form:

$$u_t = \tilde{\Lambda}(c_{t-1}, \phi_{t-1}, \phi_{\text{goal}}) \quad (12)$$

where $\phi_{\text{goal}} \in SE(3)$ is the goal pose of POM. Given an estimated hand configuration c_{t-1} and the current POM's pose ϕ_{t-1} , this function should output the actuation input u_t , so that the POM will be moved to ϕ_{goal} . However, deriving this function analytically is often challenging since the hand-object system is nonlinear and usually constrained by some grasp configurations, as described by (9).

Although it is in general difficult to reverse such a motion function, one way to solve this problem is to leverage the recursive nature of the demanded function. Similar to what we discussed in Section 3.3, given a hand configuration c_{t-1} and a POM's pose ϕ_{t-1} , since (12) only represents where to move the POM to at the next time step t , it basically requires to output a control u_t that can move the POM towards the direction of ϕ_{goal} , rather than moving the POM directly to the goal pose in one step. Therefore, as a non-optimal solution, in order to locally control the hand to trace specified waypoints for the POM, we provide a simple local gradient-based approach to greedily reverse the motion function, and leave the development of a global planner for our future work.

For this, with respect to the forward motion model in (9), we calculate a Jacobian matrix $J \in \mathbb{R}^{6 \times d_a}$:

$$J = \left[\frac{\partial \Lambda}{\partial u^1}, \dots, \frac{\partial \Lambda}{\partial u^{d_a}} \right] \quad (13)$$

where each partial derivative $\frac{\partial \Lambda}{\partial u^i}$, $i = 1, \dots, d_a$, is

$$\frac{\partial \Lambda}{\partial u^i} = \left[\frac{\partial \Lambda_X}{\partial u^i}, \frac{\partial \Lambda_Y}{\partial u^i}, \frac{\partial \Lambda_Z}{\partial u^i}, \frac{\partial \Lambda_{roll}}{\partial u^i}, \frac{\partial \Lambda_{pitch}}{\partial u^i}, \frac{\partial \Lambda_{yaw}}{\partial u^i} \right]^T \quad (14)$$

which represents the partial derivative of the POM's pose against the actuation input u^i . This vector represents the partial derivatives in X, Y, Z translations, as well as in the roll, pitch, and yaw rotations. In this work, the Jacobian is numerically calculated with small perturbations of u on function $\Lambda(\cdot)$.

Using this Jacobian matrix, the manipulation can be conducted in an iterative manner and the reversed function (12) is modeled as shown in Algorithm 3. An important thing to note is that, owing to the nonlinearity of the hand-object system, it is not likely that a control input u_t can be directly calculated to move the POM to ϕ_{goal} in one step. Instead, it is more feasible to model this reversed function in an iterative manner as in Algorithm 3. Doing so, we not only can iteratively move the POM towards its goal pose, but also can correct the execution errors since we utilize the real time feedback as in steps 2 and 3. In practice, a particle filter usually requires a large number of particles to obtain precise estimation, and the estimation process in Section 3 may not satisfy real-time requirements. However, once the estimation has converged, we downsample the particles and only maintain a smaller amount of them for tracking, so that the hand configuration tracking can be performed in real time.

Algorithm 3. Manipulation Planning and Control.

Input: $\phi_{\text{goal}}, \text{maxIter}$

```

1: while  $\text{maxIter} > 0$  do
2:    $\phi_{t-1} \leftarrow \text{Camera.Perceive}()$ 
3:    $c_{t-1} \leftarrow \text{TrackHandConfig}()$   $\triangleright$  Section 3
4:    $\phi_{\text{diff}} \leftarrow (\phi_{\text{goal}} - \phi_{t-1})$ 
5:   if  $\|\phi_{\text{diff}}\| < E$  then
6:     return Success  $\triangleright$  Goal Reached
7:   end if
8:    $J \leftarrow \text{Jacobian}(\Lambda)$   $\triangleright$  Equation (13)
9:    $\Delta u = J^T \cdot \phi_{\text{diff}}$ 
10:   $\text{Hand.Execute}(\Delta u)$ 
11:   $t++$ ,  $\text{maxIter}--$ 
12: end while
13: return Failure

```

5. An instantiation using Yale OpenHand Models

In order to utilize the proposed particle filter for estimating the hand–object configuration, and use the Jacobian-based method for controlling the movement of the POM, we need to derive the hand’s motion model in terms of the grasp constraints and the actuation inputs. In this section, we instantiate those models using the Yale OpenHand Models as an example usage (Ma and Dollar, 2017). The Yale OpenHand project provides multiple hand models, such as the Model T42 with two fingers, and the Model O with three fingers. In both models, the fingers are composed of two joints, driven by a single actuator through a tendon. The fingers can be installed with either elastic flexure joints, or spring-loaded revolute joints. In this work, we describe the modeling details of a revolute–revolute Model O hand, and additionally show how this can be easily adapted to a revolute–revolute Model T42 hand.

As mentioned previously, one benefit of using underactuated hands is the ability to easily achieve the manipulation stability, so as to randomly generate actuation inputs to manipulate the object for the hand–object configuration estimation. In addition, as will be seen in the experiments, since the fingertips are convex and the hands are compliant, the contact areas for precision grasps do not deviate much from one grasp to another, nor during small-scale in-hand manipulations, which ensures that our fixed contacts-based modeling is valid for those hands.

5.1. Model O

As shown in Figure 3, the Model O hand is composed of three fingers with springs and tendons. Each finger is driven by a single motor through a tendon, and the joint angles are compliantly determined by the springs and external contacts. Let us denote by r_m , r_p , and r_d the pulley radii at the motor, the proximal joint, and the distal joint, and by k_p and k_d the torsional spring constants of the proximal and distal joints. The free-swing motion of each finger is modeled in

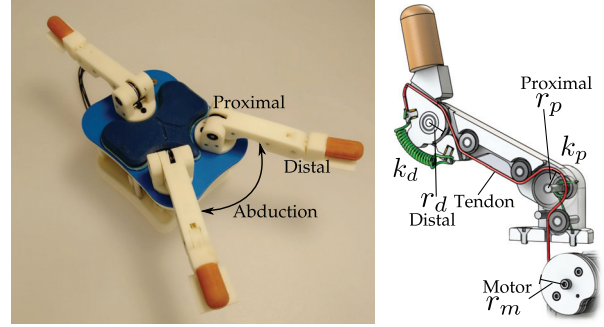


Fig. 3. Yale Model O Hand with two revolute joints on each finger.

terms of the energy stored in the springs. Intuitively, given the same actuation input, although the finger joints can reconfigure differently depending on the external contact forces, the finger’s configuration should always find equilibrium at its minimum energy state when there is no external contacts. In this work, our hand is controlled by the position of the motors in radians. When $u^i = 0$, the finger is at its resting position and the energy is 0. For $u^i > 0$, the i th finger’s energy is

$$E^i(c^i) = \frac{1}{2}(k_p \theta_{pi}^2 + k_d \theta_{di}^2) \quad (15)$$

where $c^i = (\theta_{pi}, \theta_{di})$ are the proximal and distal joint angles of the i th finger, and they together compose the hand configuration $c = (c^1, c^2, c^3) \in \mathcal{C}$. In addition, each finger’s configuration is constrained by

$$u^i r_m = \theta_{pi} r_p + \theta_{di} r_d \quad (16)$$

which enforces that the tendon length is conserved during finger motions. During manipulation, we assume the object is grasped by only the fingertips. As such, the hand configuration is not a free-swing configuration and cannot be decided by only (15) and (16). Fortunately, in terms of the manipulation constraint in (6), we know that the contacts have fixed positions in the grasp frame \mathcal{G} , and that the hand should still be configured at the state that minimizes its elastic energy, similar to the free-swing situation. Therefore, once a grasp is obtained with actuation input u , the hand configuration c^* is determined by

$$c^* = \underset{c}{\operatorname{argmin}} \sum_i E^i(c^i) \text{ s.t. Equations(6) and (16)} \quad (17)$$

Since this energy minimization is also constrained by (6), we will need the hand configuration c to obtain this constraint, so that this minimization can be solved. However, recall the causality dilemma in Section 3.2, since we cannot acquire the hand configuration directly from any sensors, we need to reformulate this process as a recursive motion model as in (7) to allow the particle filter to estimate the hand configuration. As such, instead of directly using the motor position u ,

we use Δu to control the relative changes of the motor positions, and the constraint in (16) becomes

$$\Delta u^i r_m = \Delta \theta_{pi} r_p + \Delta \theta_{di} r_d \quad (18)$$

Although there exists infinite combinations of $\Delta \theta_{pi}$ and $\Delta \theta_{di}$ that can satisfy this constraint, there exists only one combination that can minimize the energy in (15). Thus, this constraint can be incorporated into (15) and (17) to recursively calculate energy-minimized hand configurations, and then the particle filter proposed in Section 3 can be adopted. Note that since we do not change the abduction angle during manipulation, the hand is only controlled by opening and closing the fingers. In addition, since the abduction joint is rigidly coupled with its motor, its joint angle does not need estimation and can be directly read from the motor encoder.

5.2. Adaptation for the Model T42

Adapting the proposed framework to model the T42 hand is straightforward. Since it has only two fingers, we only need to adapt the modeling of the grasp frame defined in Section 3.2. As shown in Figure 4, we model one of the fingers as a combination of both fingertips p_2 and p_3 . In order to use (5) to calculate the grasp frame, we assign a very small distance between p_2 and p_3 in the direction of the fingertip’s width. As such, the hand–object system of the T42 can be modeled in the same way as in Section 3.2, and the particle filter can be directly adopted to estimate the hand configurations.

When controlling the motion of the fingers, the control inputs for finger p_2 and p_3 are degenerated to have a hard constraint $u^2 \equiv u^3$. Note that for the energy calculation in (15), the spring constants k_p and k_d should be reduced by half, so that the energy stored in the finger is not duplicated.

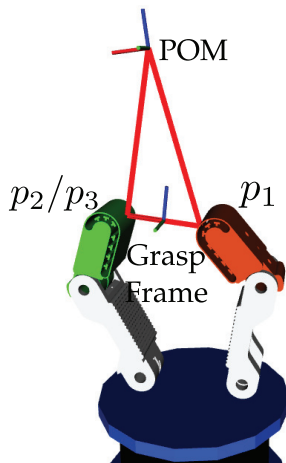


Fig. 4. Grasp frame adapted to the Model T42. The left fingertip is modeled as a combination of fingertips p_2 and p_3 , with a small distance between them.

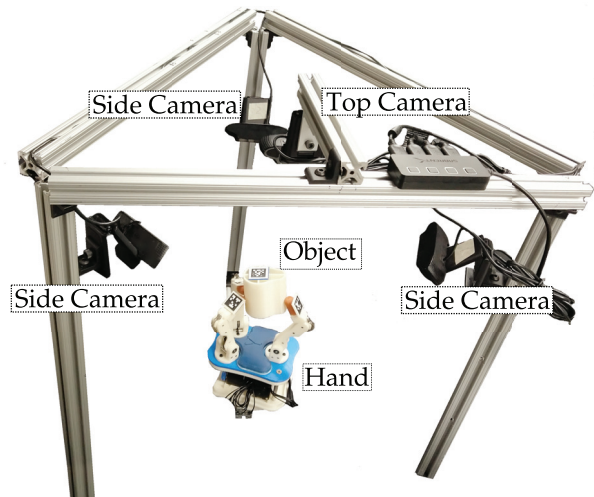


Fig. 5. Experimental setup: the top camera is used to track the motion of the POM, and the side cameras are used to track the fingertip positions, which are only used for ground-truth comparison.

6. Experiments

In this work, the algorithms were implemented in Python on a machine with Ubuntu 16.04 running on an AMD Ryzen Threadripper 1950X 16-Core Processor, which allows us to parallelize the particle propagation using 32 threads. Our experimental setup is shown in Figure 5. In order to provide particle visualization and enable quantitative evaluation of the estimations, all four cameras are calibrated in the hand’s frame. Based on this setup, we are able to track the POM’s movements using the top camera, and in the meanwhile track the positions of all the fingertips using the side cameras, which were used only for ground-truth comparison purposes. Next, we first evaluate the hand configuration estimator using both the Model O and the T42 hands, and then provide experimental results of in-hand manipulation using handwriting tasks.

6.1. Hand configuration estimation

We evaluated the configuration estimator using eight 3D printed objects shown in Figure 6. In all the experiments, we fixed the abduction angle to be 30° for the model O hand and randomly gave objects to the hand to grasp, so that the contacts on the object are randomized and the hand configurations cannot be known beforehand. The particles in our algorithm were uniformly sampled within a range that is large enough to contain the true hand configuration, and the number of particles was set to 10,000 for experiments on Model O, and set to 1,000 for experiments on Model T42. Once the estimation process starts, we generate small actuation inputs for the hand at random in each iteration, as modeled in Section 5, to twiddle the object in-hand. At the end of each iteration, the POM’s pose in six

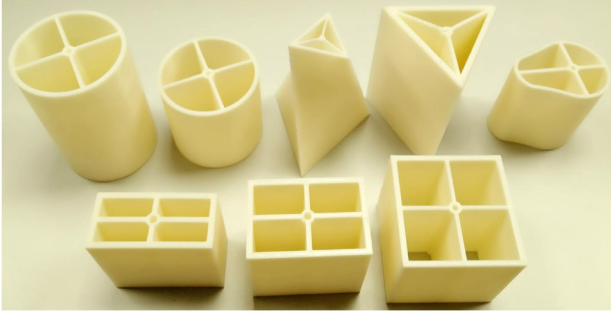


Fig. 6. Test objects. For the Model O experiments we used the cylinder (C), short cylinder (SC), twisted triangular prism (TTP), triangular prism (TP), and deformed cylinder (DC). For the Model T42 experiments we used the thin cuboid (TC), middle cuboid (MC), and large cuboid (LC).

dimensions is observed by the top camera and used as the input for the filter to propagate the particles and calculate their weights (see Algorithm 2). This process iterates until the particles are converged. In practice, we consider an estimation process as converged once the variance of all particles is below 0.01 rad. After convergence, we downsample the number of particles to 100 in terms of their weights, so as to enable real-time tracking.

Although there exist techniques to track the object's pose in real time based on the appearance or depth measurements of the object, to focus on the scope of this project and to simplify the perception of POM, we used AprilTag markers (Wang and Olson, 2016) and arbitrarily attached them on the top of objects. In order to obtain the ground truth for comparison, we also attached markers on the back of each fingertip for us to calculate the true hand configurations using inverse kinematics. However, those fingertip markers are not used for hand configuration estimation. Some example results of the estimation process are shown in Figure 7 (see also Extension 1). As indicated by the estimated contact positions, which are calculated by forward kinematics based on the estimated hand configurations, our approach is able to estimate the hand configurations with

good precision. Although there were little offsets from the estimated contacts to the real contacts, our estimation results were good enough to accurately represent the grasp frame.

A record of the hand configuration estimation process is illustrated in Figure 8. At the beginning of the estimation, since the particles were sampled randomly, we can see quite large errors in estimation. As the estimation iterates, the error decreased, and the estimation eventually converged to the true configuration. Since the estimator needs to balance between all joint angles, the errors in estimating each individual joint angles were not monotonically decreasing. However, we can see that the total error of all joint angles was almost monotonically decreasing, indicating the effectiveness of the particle filter. In addition, we can see that the estimation was able to follow the real hand configuration. This ensures that, once the estimation is converged, we are able to use the estimator as a tracker to track the hand configuration during future manipulations.

To quantitatively evaluate the performance, we have repeated the experiments on each object five times and report the statistics in Table 1. First, it is interesting to note that the number of iterations taken for the estimator to converge when manipulating different objects are similar, although the objects are differently shaped and sized. This is because our system does not model the object geometries. Instead, since the estimator considers only the contacts and the POM, differently shaped objects are not actually different, as seen by the estimator. Furthermore, every time an estimation was converged, we recorded the maximum error of all joint angles, and the results show that the errors were not significant. However, as will be seen in our manipulation planning and control experiments, the estimation errors will cause some noticeable imprecision in manipulation. In practice, we see that because the spring constants modeled in the hand motion model, see Section 5, cannot be accurately estimated, the estimation is inherently affected by this uncertainty, which we plan to address in future work.

Moreover, we also implemented the proposed framework on a Model T42 hand. As shown in Figure 9, using the

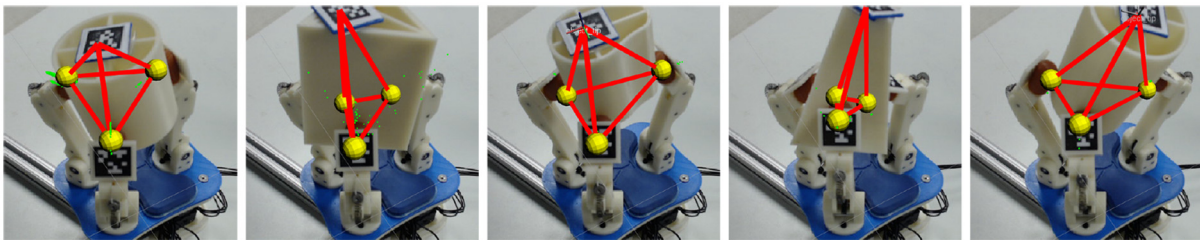


Fig. 7. Hand configuration estimations performed by our system using the Model O hand on the test objects shown in Figure 6. In the pictures, the yellow spheres indicate the contact positions according to the estimated hand configurations. The total estimation iterations taken for these examples were: 9, 5, 11, 7, and 12. The complete experiment records can be viewed in Extension 1.

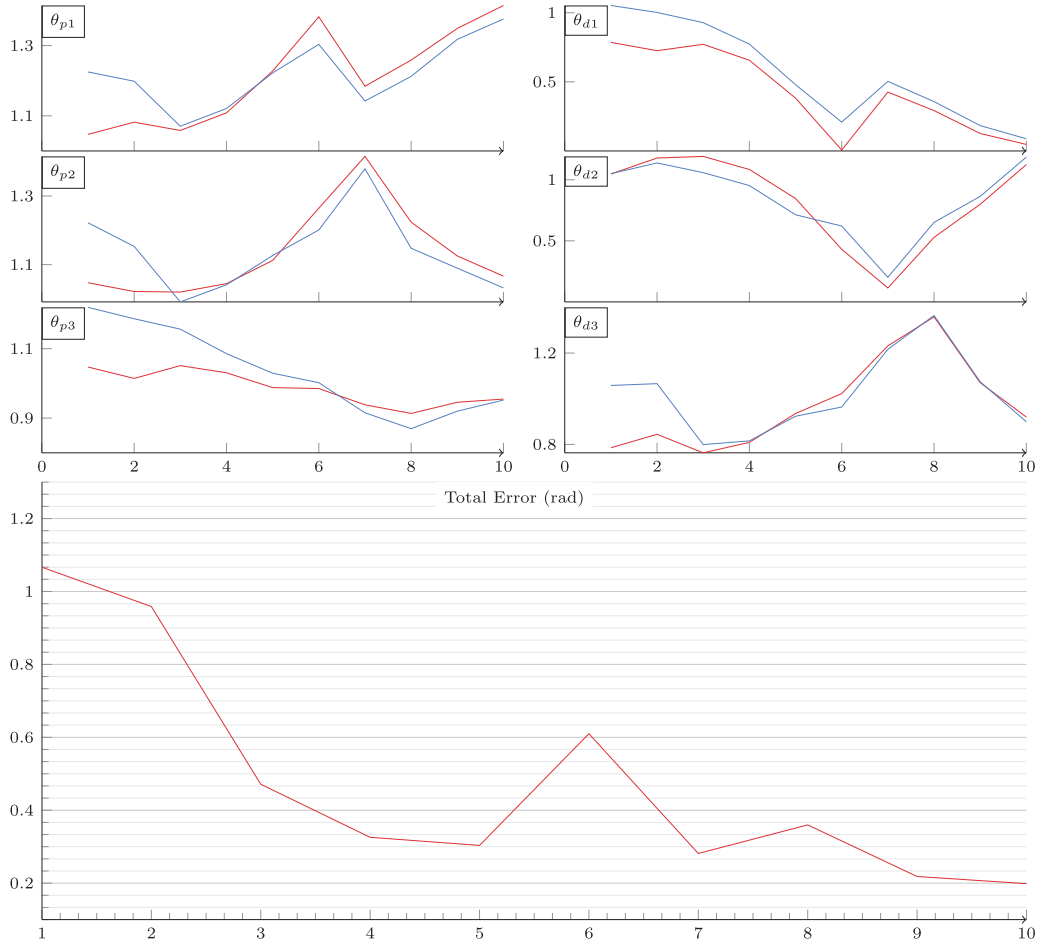


Fig. 8. Upper: Joint angle trajectories (y -axis in radians) of the Model O hand manipulating the cylinder object during estimation iterations (x -axis). (Red: real angles; blue: estimated angles). Lower: Total error of all joint angles (y -axis in radians) during estimation iterations (x -axis) .

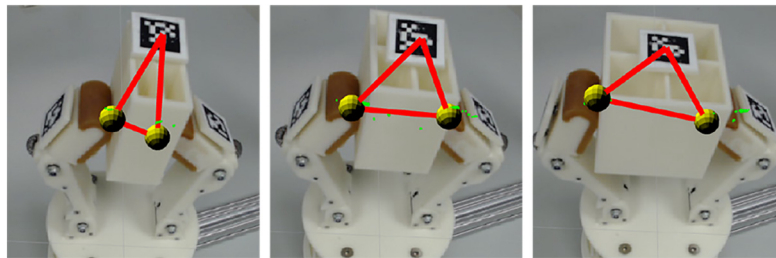


Fig. 9. Hand configuration estimations performed by our system using the Model T42 hand on the TC, MC, and LC objects shown in Figure 6. The total estimation iterations taken for these examples were 7, 9, and 10.

model adaptation from Section 5.2, our approach can also estimate the hand configuration for this two-fingered hand. Since this hand has fewer joints to estimate, we used only 1,000 particles and repeated each experiment five times. The statistics are reported in Table 1. We can see that the statistics of the T42 hand are similar to that of the Model O hand, this indicates that the estimation complexity was not decreased for this two-fingered hand, even if we need fewer particles as the problem dimension has become lower, the

number of estimation iterations are still similar due to the model adaptation does not intrinsically change the motion model.

6.2. In-hand manipulation

We evaluated the manipulation planning and control method proposed in Section 4 using the Model O hand. In the experiments, we first randomly give the cylinder object

Table 1. Statistics of the hand configuration estimation. Iterations: number of iterations taken until convergence; Error: the maximum joint angle errors; \pm : standard deviation.

Hand	Model O					T42		
	C	SC	TTP	TP	DC	TC	MC	LC
Iterations	9.2 \pm 1.4	10.3 \pm 1.0	11.6 \pm 1.6	8.7 \pm 1.2	9.6 \pm 0.9	8.8 \pm 1.9	12.1 \pm 2.4	9.9 \pm 1.2
Error (10^{-2} rad)	8.2 \pm 2.1	9.1 \pm 1.4	12.2 \pm 2.3	7.6 \pm 1.1	9.5 \pm 1.3	5.4 \pm 3.1	7.1 \pm 2.2	10.8 \pm 2.1

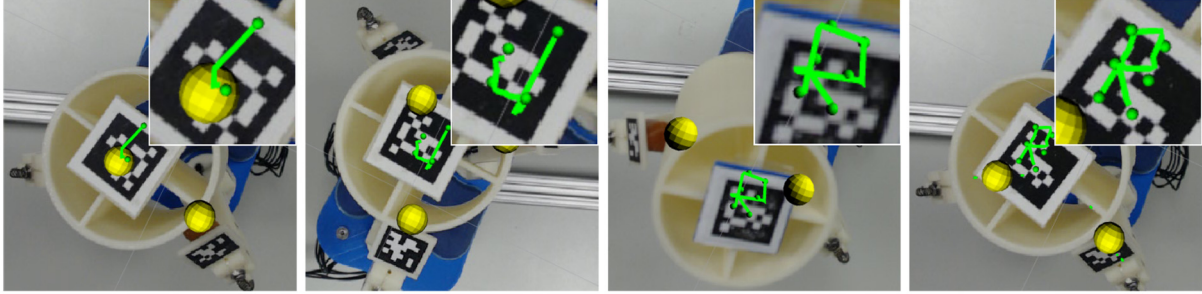


Fig. 10. The hand was tasked to manipulate the POM to trace the path indicated by waypoints (green dots), in order to write the letters “I,” “J,” “R,” and “R” (green trajectories). The distance between adjacent waypoints was set to 6 mm, and set to 12 mm only for the long edge of the “J.”

into the hand for grasping, and then use the particle filter to estimate the hand configuration. Once the estimation converged, we generated waypoints for the POM to trace. As shown in Figure 10, the hand was tasked to trace the writing path of English letters “I,” “J,” and “R.”

In this experiment, since we only needed to manipulate the POM to trace waypoints in \mathbb{R}^3 and did not have specific requirements for the orientation of the POM, we truncated the partial derivative vector in (14) to contain only the first three elements, so as to calculate the gradients only for translational manipulation. As such, the POM was moved towards each waypoint, with arbitrary orientation as long as the position could be reached. For reaching each waypoint, we set the *maxIter* in Algorithm 3 to 10, and turned to the next waypoint if the *maxIter* was exhausted. As seen in Figure 10, the manipulation was able to approximately pass through all the waypoints to achieve the handwriting tasks. Since the hand estimation could end up with different hand configurations due to the random manipulations, the letters were written differently each time.

We repeated the writing task for each letter five times and report the statistics in Table 2. Importantly, we can note positional errors for almost every waypoint. As mentioned previously, one of the major reasons for those errors was that the hand configuration estimation was imperfect, and the errors from the estimation were propagated to the manipulations. In addition, recall that the planning and control is based on the numerical gradient, which is merely a local representation describing the mapping from the actuation inputs to the POM’s movements. Therefore, owing to the nonlinearity of the hand–object system, it is

Table 2. Statistics of handwriting tasks. Iterations: number of iterations taken to reach a waypoint; Error: average positional errors; \pm : standard deviation.

Letter	“I”	“J”	“R”
Iterations	4.2 \pm 2.3	6.1 \pm 3.0	7.1 \pm 3.4
Error (mm)	0.73 \pm 0.47	1.02 \pm 0.60	0.96 \pm 0.27

possible that Algorithm 3 could not find a solution to move the POM to the exact waypoint.

Moreover, we also tested the manipulation by commanding the POM to trace random waypoints of both translations and orientations in $SE(3)$ using the cylinder object. In this experiment, the waypoints were sampled sequentially around the current pose of the POM within a range of $[-5 \text{ mm}, 5 \text{ mm}]^3 \times [0.2 \text{ rad}, 0.2 \text{ rad}]^2$. Note that due to the kinematic constraints of the Model O hand, the orientation in the yaw direction was not sampled. We repeated this evaluation for 30 times and the average error was 1.9 mm in position, and 0.09 rad in orientation. Since the controller had to balance between the positional errors and orientational errors, we noticed that the positional error was larger in this case in comparison with the handwriting tasks.

7. Conclusion

In this work, we have proposed a particle filter-based framework for hand–object configuration estimation, tracking, and manipulation planning. As an example usage of

our framework, we used underactuated hands without joint encoders, and leveraged the passive compliance of the hand for grasp stability and modeled the hand–object motion by formulating grasp constraints. In terms of the motion model, we developed a particle filter-based approach to estimate the hand configurations. Once an object is grasped, our system randomly manipulates the object in-hand and in the meantime observes the movements of the POM. By calculating the likelihood of the hand configuration hypotheses (particles) and iteratively filtering out the unlikely ones, the particles will finally converge towards the true hand configuration. Once the estimation is converged, we provided a method for iterative planning and control of in-hand manipulation, which is able to achieve underactuated manipulation by directly generating actuation inputs.

Although we have only shown an example usage with underactuated hands, our framework can be applied to more general cases including fully actuated hands. For example, even if the hand configuration can be directly obtained from joint encoders of fully actuated hands, we can instead model the exact contacts between the fingertips and the object using kinematic equivalents (Rojas and Dollar, 2016), for which the joint configurations are unknown, and then use our framework to estimate the exact contact positions to accurately register the object into the hand frame. However, it has to be noted that adapting the proposed approach to fully actuated hands requires additional efforts on manipulation stability maintenance. Different from the underactuated mechanism, a grasp controller has to be devised to enable the hand to actively ensure stable contacts during manipulation. Moreover, more sensing modalities may be required to improve the system’s robustness. For example, a tactile feedback-based impedance controller can be adopted for this purpose (Li et al., 2014).

7.1. Observations and limitations

Our experimental results have shown that the proposed system is able to estimate the hand configurations without needing the geometric model of the grasped object. In addition, we have shown that our framework can be easily adapted to different hands with different numbers of fingers. The manipulation capability was evaluated through handwriting tasks. The results showed that the proposed manipulation planning and control algorithm is effective in manipulating the POM to reach commanded positions and orientations. By analyzing the performance of the proposed framework, we hypothesize that the estimation errors were mainly caused by the physical uncertainties of the spring constants in the modeling of the hand–object motions, which could have distorted the propagation of the particles. For the same reason, the manipulation control was also affected. In addition, we have observed that the local Jacobian-based planning and control method was not able to find solutions in some cases, such as when the maximum number of execution iterations was exhausted, or when there exists singularities in the Jacobian matrix.

As a basic assumption of the proposed approach, fixed contact positions simplified the construction of the grasp frame, which enabled us to describe the relationship between contacts and the POM. Using this relationship, we have derived the motion model of the object, and integrated the motion model into the particle filtering framework for hand–object configuration estimation. However, assuming fixed contacts introduced two limitations to the system’s capability. First, although we do not need to know the geometry of the object and where the contacts are on the object, the contact positions on the fingertips have to be known *a priori*. In our experiments, since the fingertips are convex and the hands are compliant, the contact areas for precision grasps do not deviate much from one grasp to another, it was therefore possible for us to find out the locations beforehand. Second, although during the estimation process we require only small random manipulations, which do not have a major effect on our assumption of fixed contacts. Once the estimation has converged, our system is currently not able to provide larger manipulation ranges with high accuracy, as larger movements may require contact changes such as rolling, which can break the derived motion model.

In addition, it is worthwhile to note that, since the object motion model relies on the proposed grasp frame, which has to be rigid in order to uniquely describe the object motion, our framework does not work for soft or deformable objects. In addition, when manipulating heavy objects, it will be necessary to model the gravitational potential energy of the object into (15), as the object weight can also affect the minimum-energy hand configuration. However, as it is difficult to know the object mass or its mass distribution beforehand, this additional modeling can be very complicated and actually requires our system to also estimate these parameters along with the hand configuration. Limited by the scope of this work, we leave this to our future work.


7.2. Future work

In order to address the identified problems, we plan to extend the capability of our system to work into more general setups, including the estimation of the joint stiffnesses. In particular, we will formulate the effects of contact rolling and sliding into our recursive motion model, so that the hypotheses represented by particles can also model the dynamic changes of the grasp constraints. With this additional modeling, we hope to also address a broader range of application scenarios where finger contacts can be broken and remade during finger gaiting manipulations. To further tackle the problem of imperfect perception, instead of using a marker to track a single POM point, we plan to track multiple key points at the same time to filter the potential perception noise, as well as addressing the problem of occlusions which could happen in single point tracking.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Science Foundation (grant number IIS-1734190).

ORCID iD

Kaiyu Hang  <https://orcid.org/0000-0003-4132-1217>

References

- Ajoudani A, Godfrey SB, Bianchi M, et al. (2014) Exploring teleimpedance and tactile feedback for intuitive control of the Pisa/IIT SoftHand. *IEEE Transactions on Haptics* 7(2): 203–215.
- Bicchi A, Gabbicini M and Santello M (2011) Modelling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 366(1581): 3153–3161.
- Bicchi A and Kumar V (2000) Robotic grasping and contact: A review. In: *IEEE International Conference on Robotics and Automation (ICRA)*.
- Bircher WG, Dollar AM and Rojas N (2017) A two-fingered robot gripper with large object reorientation range. In: *IEEE International Conference on Robotics and Automation (ICRA)*. *IEEE*, pp. 3453–3460.
- Bohg J, Morales A, Asfour T and Kragic D (2014) Data-driven grasp synthesis – a survey. *IEEE Transactions on Robotics* 30(2): 289–309.
- Borràs J and Dollar AM (2015) Dimensional synthesis of three-fingered robot hands for maximal precision manipulation workspace. *The International Journal of Robotics Research* 34(14): 1731–1746.
- Borst C, Fischer M and Hirzinger G (2002) Calculating hand configurations for precision and pinch grasps. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*.
- Bütepage J, Cruciani S, Kocic M, Welle M and Kragic D (2019) From visual understanding to complex object manipulation. *Annual Review of Control, Robotics, and Autonomous Systems* 2(1): 161–179.
- Calli B and Dollar AM (2019) Robust precision manipulation with simple process models using visual servoing techniques with disturbance rejection. *IEEE Transactions on Automation Science and Engineering* 16(1): 406–419.
- Calli B, Kimmel A, Hang K, Bekris K and Dollar A (2018) Path planning for within-hand manipulation over learned representations of safe states. In: *International Symposium on Experimental Robotics*.
- Chebotar Y, Hausman K, Su Z, Sukhatme GS and Schaal S (2016) Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1960–1966.
- Ciocarlie M, Goldfeder C and Allen P (2007) Dimensionality reduction for hand-independent dexterous robotic grasping. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 3270–3275.
- Corcoran C and Platt R (2010) A measurement model for tracking hand-object state during dexterous manipulation. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4302–4308.
- Cruciani S, Smith C, Kragic D and Hang K (2018) Dexterous manipulation graphs. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2040–2047.
- Dafle NC, Rodriguez A, Paolini R, et al. (2014) Extrinsic dexterity: In-hand manipulation with external forces. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1578–1585.
- Deimel R and Brock O (2016) A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research* 35(1–3): 161–185.
- Deimel R, Irmisch P, Wall V and Brock O (2017) Automated co-design of soft hand morphology and control strategy for grasping. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1213–1218.
- Della Santina C, Bianchi M, Averta G, et al. (2017) Postural hand synergies during environmental constraint exploitation. *Frontiers in Neurobotics* 11: 41.
- Dollar AM and Howe RD (2010) The highly adaptive SDM hand: Design and performance evaluation. *The International Journal of Robotics Research* 29(5): 585–597.
- Eppner C and Brock O (2017) Visual detection of opportunities to exploit contact in grasping using contextual multi-armed bandits. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 273–278.
- Eppner C, Deimel R, Álvarez Ruiz J, Maertens M and Brock O (2015) Exploitation of environmental constraints in human and robotic grasping. *The International Journal of Robotics Research* 34(7): 1021–1038.
- Han L, Guan YS, Li ZX, Shi Q and Trinkle JC (1997) Dexterous manipulation with rolling contacts. In: *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 2, pp. 992–997.
- Hang K, Li M, Stork JA, et al. (2016) Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation. *IEEE Transactions on Robotics* 32(4): 960–972.
- Hang K, Morgan AS and Dollar AM (2019) Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands. *IEEE Robotics and Automation Letters* 4(2): 662–669.
- Hang K, Stork JA, Pollard NS and Kragic D (2017) A framework for optimal grasp contact planning. *IEEE Robotics and Automation Letters* 2(2): 704–711.
- Haustein JA, Hang K and Kragic D (2017) Integrating motion and hierarchical fingertip grasp planning. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3439–3446.
- Kim J, Thang ND and Kim T (2009) 3-D hand motion tracking and gesture recognition using a data glove. In: *2009 IEEE International Symposium on Industrial Electronics*, pp. 1013–1018.
- Kimmel A, Shome R, Littlefield Z and Bekris KE (2018) Fast, anytime motion planning for prehensile manipulation in clutter. *CoRR* abs/1806.07465.
- Koval MC, Pollard NS and Srinivasa SS (2016) Pre- and post-contact policy decomposition for planar contact manipulation under uncertainty. *The International Journal of Robotics Research* 35(1–3): 244–264.
- Li M, Hang K, Kragic D and Billard A (2016) Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems* 75: 352–364.

- Li M, Yin H, Tahara K and Billard A (2014) Learning object-level impedance control for robust grasping and dexterous manipulation. In: *IEEE International Conference on Robotics and Automation (ICRA)*.
- Liarokapis M and Dollar AM (2019) Combining analytical modeling and learning to simplify dexterous manipulation with adaptive robot hands. *IEEE Transactions on Automation Science and Engineering* 16(3): 1361–1372.
- Liarokapis MV and Dollar AM (2016) Learning task-specific models for dexterous, in-hand manipulation with simple, adaptive robot hands. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2534–2541.
- Ma R and Dollar A (2017) Yale OpenHand Project: Optimizing open-source hand designs for ease of fabrication and adoption. *IEEE Robotics and Automation Magazine* 24(1): 32–40.
- Ma RR and Dollar AM (2011) On dexterity and dexterous manipulation. In: *International Conference on Advanced Robotics (ICAR)*, pp. 1–7.
- Ma RR and Dollar AM (2013) Linkage-based analysis and optimization of an underactuated planar manipulator for in-hand manipulation. *ASME Journal of Mechanisms and Robotics* 6(1): 011002.
- Mahler J, Liang J, Niyaz S, et al. (2017) Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In: *Robotics: Science and Systems*.
- Okamura AM, Smaby N and Cutkosky MR (2000) An overview of dexterous manipulation. In: *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 1, pp. 255–262.
- OpenAI, Andrychowicz M, Baker B, et al. (2018) Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*.
- Páll E, Sieverling A and Brock O (2018) Contingent contact-based motion planning. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 6615–6621.
- Rojas N and Dollar A (2016) Classification and kinematic equivalents of contact types for fingertip-based robot hand manipulation. *Journal of Mechanisms and Robotics* 8: 041014.
- Salvietti G, Malvezzi M, Gioioso G and Prattichizzo D (2015) Modeling compliant grasps exploiting environmental constraints. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4941–4946.
- Santina CD, Grioli G, Catalano M, Brando A and Bicchi A (2015) Dexterity augmentation on a synergistic hand: The Pisa/IIT SoftHand+. In: *IEEE International Conference on Humanoid Robots (HUMANOIDS)*, pp. 497–503.
- Santina CD, Piazza C, Grioli G, Catalano MG and Bicchi A (2018) Toward dexterous manipulation with augmented adaptive synergies: The Pisa/IIT SoftHand 2. *IEEE Transactions on Robotics* 34(5): 1141–1156.
- Sintov A, Morgan AS, Kimmel A, Dollar AM, Bekris KE and Boularias A (2019) Learning a state transition model of an underactuated adaptive hand. *IEEE Robotics and Automation Letters* 4(2): 1287–1294.
- Sommer N, Li M and Billard A (2014) Bimanual compliant tactile exploration for grasping unknown objects. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6400–6407.
- Sridhar S, Oulasvirta A and Theobalt C (2013) Interactive markerless articulated hand motion tracking using RGB and depth data. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Sui Z, Jenkins OC and Desingh K (2015) Axiomatic particle filtering for goal-directed robotic manipulation. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 4429–4436.
- Sundaralingam B and Hermans T (2017) Relaxed-rigidity constraints: In-grasp manipulation using purely kinematic trajectory optimization. In: *Robotics: Science and Systems*.
- Tahara K, Arimoto S and Yoshida M (2010) Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand. In: *IEEE International Conference on Robotics and Automation (ICRA)*.
- Thrun S, Burgard W and Fox D (2005) *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, MA: The MIT Press.
- Trinkle J and Paul R (1990) Planning for dexterous manipulation with sliding contacts. *The International Journal of Robotics Research* 9(3): 24–48.
- Wang J and Olson E (2016) Apriltag 2: Efficient and robust fiducial detection. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198.
- Xu J, Koo TKJ and Li Z (2010) Sampling-based finger gaits planning for multifingered robotic hand. *Autonomous Robots* 28(4): 385–402.

Appendix. Index to multimedia extensions

Archives of IJRR multimedia extensions published prior to 2014 can be found at <http://www.ijrr.org>, after 2014 all videos are available on the IJRR YouTube channel at <http://www.youtube.com/user/ijrrmultimedia>

Table of Multimedia Extensions

Extension	Media type	Description
1	Video	The complete experiment records