ICT Call 7
ROBOHOW.COG
FP7-ICT-288533

Deliverable D5.2:

Automatic constraint extraction and its use in learning manipulatory tasks

SEVENTH FRAMEWORK PROGRAMME

January 31, 2014

Project acronym:	ROBOHOW.COG
Project full title:	Web-enabled and Experience-based Cognitive Robots that Learn Complex Everyday Manipulation Tasks
Work Package:	WP 5
Document number:	D5.2
Document title:	Automatic constraint extraction and its use in learning manipulatory tasks
Version:	1.0
Delivery date:	January 31st, 2014
Nature:	Report
Dissemination level:	Public
Authors:	Ana Lucia Pais, Nadia Figueroa, Miao Li and Aude Billard (EPFL) François Keith, Don Joven Agravante, Abderrahmane Kheddar (CNRS)

Contents

3

Summary

This deliverable reports on work performed as part of WP5 during the second year of the project. The main theme was on "Automatic extraction of constraints" and its use in learning manipulatory tasks.

We report on three strategies to address the problem of constraints' extraction. In Section 1 we present an extension of the constraint extraction method presented in the first year, see Deliverable D5.1 and whereby we use the notion of invariance across variable to determine the importance of each variable to the success of the task. The relative importance of each variable is then used as a soft-constraint in an impedance controller. Additionally, it is also used as a means to determine the most appropriate frame of reference into which to express the variables (the frame in which the variables are the most invariant). Transitions across frames of reference is used to segment the task. This framework is implemented to extract constraints in a bimanual coordination tasks, *bowl mixing*, that is relevant to the Robohow scenario.

In Section 2, we present an alternative way of automatically segmenting the task based on an analysis of the statistical distribution of the whole group of variables. We also explore the use of a representation of the data based on screw theory to enable segmentation to be robust to translational and rotational transformation. We report on preliminary implementation of this approach in simulation with theoretical sets of data showing complex distributions.

Then, in Section 3, we explore mechanisms by which one can learn and model the task impedance when manipulating an object. While in the approach of Section 1, the impedance is learned and controlled from the human or robot standpoint, i.e. from the actor's point of view, here, we model the impedance from the object's point of view, i.e. at task level. The advantage of such approach is that it can then be easily ported to robotic platforms that differ importantly in kinematic from the human.

Finally, in Section 4, we detail how motion primitives can be extracted from the monitoring of a human-human joint transportation task. It can then be improved to more complex motions using haptic cues without further human monitoring, and extended by including visual cues in order to handle more constraints, such as the balance of an object placed on the transported beam, and infer human intentions during the manipulation task.

Work on automatic task constraint and automatic segmentation of the task participates in *Task 5.1 Learning bootstrapping information to guide imitation learning*. Work on extracting the impedance from both robot and object' viewpoints participate in *Task 5.3 Learning adaptive stiffness control that has the desired effects*. Work on learning of human-robot joint transportation participates to *Task 5.5 Learning of haptics interaction*.

Chapter 1

Automatic extraction of constraints in bi-manual tasks

Introduction   To recall, in this work, we consider task constraints to be features of the motion that the robot should reproduce for achieving successful task execution. In the previous year of the project we proposed an approach for extracting these task constraints from human demonstration using a variance-based method (Pais et al, 2013).

The procedure is devised in two steps: robot data (end effector position and forces) is recorded from kinaesthetic demonstrations, then analyzed with respect to the observed variance across all demonstrations and to the way this variance changes in time. We extract the relative importance of force and position at each time step and use this information to continuously modulate the robot's stiffness throughout the task. We automatically determine the object of interest in each part of the task and express all the variables in this local frame.

For encoding the motion we use a time-independent representation. The motion profile is encoded as a non-linear dynamical system of the form $\dot{x} = f(x)$, where x and $\dot{x} \in \mathbb{R}^D$ represent the cartesian position and velocity of the end effector. The function $f : \mathbb{R}^D \mapsto \mathbb{R}^D$ is a continuous and continuously differentiable function stable only at the attractor $\overset{*}{x}$, estimated from demonstrations using a mixture of k Gaussians. The parameters of the GMM model are specified by a vector $\theta_x^k = [\pi_x^k, \mu_x^k, \Sigma_x^k]$, representing the priors, means, covariance matrices. Based on this encoding the velocity $\dot{x}$ is thus computed as $\dot{x} = E\{p(\dot{x}|x; \theta_x^k)\}$.

Similarly we encode the rotation specified by an axis-angle representation, with respect to an estimated attractor, and estimate a coupling function between the position and orientation. This ensures that the observed temporal–correlated behavior of the two variables is preserved and thus a perturbation will not cause an unsynchronized behavior. The system follows the original demonstrated dynamics and ensures asymptotical stability at the target of both position and orientation (Shukla and Billard, 2012). The force and stiffness profiles are encoded separately, as a function of the position, using a Gaussian Mixture Model.

Finally we generate a finite state machine based on the change of the extracted constraints. Each state contains the information for the parametrization of the task: (a) the local constraints and (b) the motion encoding. The procedure is described in Pais et al (2013).

Encoding bi-manual coordination patterns   Most of the common daily chores are performed bi-manually by humans, however executing bi-manual tasks by robots requires extending the current learning framework, as well as the control architecture and task representation. An advantage is that it allows executing tasks that cannot be addressed otherwise, such as molding dough, or tying shoe laces. Here we focus on a particular sub-set of bi-manual tasks that require force control during the arm coordination. In such task, the "master" hand manipulates the object (e.g. mould the dough), while the other hand provides assistance (such as compensating forces to keep the bowl steady on the table).

Directly programming bi-manual tasks might be difficult because of the high dimensionality of the variable set and modeling their coordinated behavior, however learning such tasks from demonstration poses few challenges, both on designing the experimental setup, as well as on analyzing the resulting data.

For a complete task representation, the recorded data should include information about the arm movement, the hand configuration, the force sensed on the arm and the tactile signature on the hand, the grasp configuration used and so on, which would easily be recorded from kinesthetic teaching in which the robot directly experiences the task. We propose a setup that combines two approaches: the user performs the demonstration by kinaesthetically guiding the robot arm with one hand and by wearing a data-glove covered with tactile sensors on the other hand. We alternate between the arms performing the active/passive tasks, recording complementary information from the robotic arm and from the glove.

We exemplify this on a task of stirring in a bowl, that requires completing a sequence of actions for each arm. To determine the coordination components we focus on determining the dominant hand at each moment during the demonstration, in either position or force applied in the task. This is similar to results on human subjects showing that the hands can change the active and passive roles during manipulation and that this is caused by a force-motion relation, rather than routine in performing the task Johansson et al (2006). Hand dominance allows us to encode differently the motion of the active and passive arms. For the active arm we encode the motion and force profile as described above. However for the passive arm the motion is insignificant, while the forces sensed on the arm are reaction forces responsible for keeping the object in place, therefore we choose to encode its force and stiffness profiles as dependent on the forces sensed on the active arm. This allows the passive arm to apply compensating forces to the ones applied by the active arm.

On-going work focuses on firstly extending the current approach from arm motion to hand and finger motion, analyzing coordination on multiple levels, between different systems, and subject to different dynamics: (a) between arms coordination; (b) arm-hand coordination; (c) coordination between fingers. Secondly we focus on determining causalities between variables involved in the task in order to automatically determine the best way of encoding the task.

This chapter is based on the following two papers, given in appendix:

> Pais, A. L., Umezawa, K., Nakamura, Y., and Billard, A. (2013) Task Parametrization Using Continuous Constraints Extracted from Human Demonstrations. *Submitted to IEEE Transactions on Robotics, November 2013*.
> Pais, A. L. and Billard, A. (2014) Encoding bi-manual coordination patterns from human demonstrations. *9th ACM/IEEE International Conference on Human-Robot Interaction (HRI) 2014*

Chapter 2

Discovering primitive motion categories from unsegmented demonstrations

Everyday human tasks (or complex robot tasks) such as baking a cake, making an omelette or preparing coffee can be decomposed into a sequential combination of simple motion primitives. Much research has been focused on the representation, recognition and learning of these motion primitives, with the ultimate goal of having a robot learn and reproduce the sequence of motions autonomously. However, many of these approaches take into consideration assumptions such as a priori knowledge of the expected motion primitives (pre-specified number and type of motions) for segmentation and recognition or use pre-segmented motion sequences for learning.

Our goal is to avoid these assumptions by proposing a framework that autonomously discovers motion primitives and groups them into primitive motion categories from unsegmented task demonstrations without any a priori knowledge. To incrementally learn and build primitive motion categories we need to tackle two main problems: unsupervised segmentation of motion primitives and clustering of the extracted primitives.

Unsupervised Segmentation To bypass the need of a priori knowledge from the expected motion primitives, we apply a Bayesian nonparametric approach for relating multiple time series, proposed by Fox et al (2009) called the Beta Process Hidden Markov Model (BP-HMM), with some extensions and modifications for our specific task. This approach follows the intuition that non-linear dynamical phenomena (such as human motion) can be modeled by a set of switching dynamical processes (Fox et al, 2008). The aim of the BP-HMM is to discover and incrementally model an unbounded collection of switching dynamical processes (behaviors) which are shared throughout multiple time series. In our setting, these dynamical behaviors correspond to motion primitives that are shared throughout different continuous complex task demonstrations and belong to an unbounded set of primitive motion categories.

Sharing features using the Beta Process (Fox et al, 2009; Hughes et al, 2012): In order to allow for an unbounded set of behaviors, the BP-HMM uses a feature-based nonparametric Bayesian approach based on the beta process, where the set of behaviors is represented as a list of features. Each i-th time-series is described with a sparse binary vector $\mathbf{f}_i = [f_{i_1}, f_{i_2}, ...]$ which indicates the presence/absence of features (behaviors). Given N time series, the matrix $\mathbf{F} = [\mathbf{f}_i; ...; \mathbf{f}_N]$ is the

binary feature indication for the full set of time series. $\mathbf{F}$ is generated by the beta-process (BP), which induces a predictive distribution on features known as the Indian Buffet Process (IBP), modeled as follows:

$$B|B_o,\gamma,\beta \sim BP(\beta,\gamma B_o), B = \sum_{k=1}^{\infty} b_k \delta_k \tag{2.1}$$

$$\mathbf{f}_i|B \sim BeP(B), i = 1...N \tag{2.2}$$

The realization B of the BP contains the set of global weights which determines the potentially infinite number of features. Each k-th feature is represented by its generative model parameters θ_k, while $b_k \in (0, 1)$ indicates its presence/absence. Each binary feature vector $\mathbf{f}_i$ is then determined by independent Bernoulli draws parametrized by B, dependent on its mass hyper-parameter γ which determines a Poisson(γ) distribution for the number of active features in each i-th time series and concentration hyper-parameter β which control how often features are shared between time series. By setting $\beta = 1$, the predictive distribution of the features is the IBP.

Dynamic behavior modeling using the BP-HMM (Fox et al, 2009; Hughes et al, 2012): This previously derived feature model is then combined with an HMM, to create the BP-HMM. $\mathbf{f}_i$ now determines the set of finite states available for the i-th time-series. Each t-th time step is assigned a state $z_{it} = k$ from the set $\{k : f_{ik} = 1\}$, which determines the parameters θ_k that generated $\mathbf{y}_{it}$ (observed data of i-th time series at step t, where $\mathbf{y}_{it} \in R^d$ for a d-dimensional time series). In the original implementation of the algorithm, Emily Fox et al. use the first-order vector autoregressive (VAR) process to model the switching dynamical behaviors in motion capture data. However, the algorithm is not limited to other types of generative models. Hughes et al (2012) extended the formulation of the BP-HMM to model the distinct behavior dynamics with Multivariate Gaussian and Multinomial distributions. In this work we use and compare the VAR process and Gaussian distribution. Following the generative processes of these two models:

	BP-AR(1)-HMM	BP-Gaussian-HMM
State Dynamics	$z_{it} \sim \pi_{z_{it-1}}, \mathbf{y_{it}}|z_{it} = k$	$z_{it} \sim \pi_{z_{it-1}}, \mathbf{y}_{it}|z_{it} = k$
Model Parameters	$\theta_k = \{A_k, \Sigma_k\}$	$\theta_k = \{\mu_k, \Sigma_k\}$
Observation Dynamics	$\mathbf{y}_{it} = A_{z_{it}} y_{it-1} + \mathbf{e}_{it}(z_{it})$	$\mathbf{y}_{it} \sim \mathcal{N}(\mu_{z_{it}}, \Sigma_{z_{it}})$
	$\mathbf{e}_{it} \sim \mathcal{N}(0, \Sigma_{z_{it}})$	

The transition distribution π_i is independent for each time series and is derived from a normalized collection of gamma-distributed random variables (active feature $\mathbf{f}_i$) with sticky parameters biasing the model to match high self transitions. The model parameters are drawn from conjugate priors on the mean and covariance of the data, namely the matrix normal inverse-Wishart, defined by $\Sigma \sim \mathcal{IW}(\nu_o, \mathcal{S}_o)$ and $A|\Sigma \sim \mathcal{MN}(0, \Sigma, R_o)$. $\mathcal{S}_o$ is a scaling matrix, ν_o are the degrees of freedom and R_o is the precision matrix. These parameters are derived from the actual data, except for the $\mathcal{S}_o$, this value is user-specific and its value might have a great impact on the performance of the algorithm depending on the statistical properties of the data. This happens due to the underlying properties of the $\mathcal{IW}$ prior, even though they are popular because of their conjugacy to Gaussian likelihoods, they assume strong dependence between variance and correlation (i.e high variance implies high correlation and low variance implies low-to-moderate correlation). This is bad for inference, because it means that correlation will be extremely high if variance is higher than expected. Thus, for sequences of motions where the variance is similar for each motion,

the inference is robust, however, when motions have different variance, the algorithm fails and different values of $\mathcal{S}_o$ may under or over-segment the trajectories. To alleviate this we use a non-informative hierarchical prior recently proposed by Huang and Wand (2013). This distribution comes from a family of prior distributions for covariance matrices whose standard deviation and correlation parameters are marginally non-informative for particular hyper-parameters. Hence, Σ is now drawn from:

$$\Sigma | a_1, ..., a_p \sim \mathcal{IW}(\nu_o + p - 1, 2\nu \text{diag}(1/a_1, ..., 1/a_p)) \tag{2.3}$$

$$a_k \sim^{ind.} Inverse - Gamma(1/2, 1/A_k^2), k = 1, ..., p \tag{2.4}$$

where p is the dimension of the time series data. In the original implementation, the BP-HMM model is learned using a Markov Chain Monte Carlo (MCMC) method that alternates between re-sampling $\mathbf{f}_i$ given $\mathbf{y}_i$ and θ_k, and θ_k given $\mathbf{f}_i$ and $\mathbf{y}_i$. This is done using a combination of Metropolis-Hastings and Gibbs samplers. This method, even though robust, needs many iterations, requires careful initialization and only modifies a small subset of variables at each step. Recently Hughes et al (2012), introduced an improved MCMC method for BP-HMM learning which rapidly discovers new and shared behaviors using split-merge moves based on sequential allocation. We use this implementation made available by Michael Hughes[1].

We examine the performance of the previously described variants of the BP-HMM on a toy dataset of 5 features (behaviors), where each behavior is a primitive motion category (PMC) from the following set: {hyperbolic parabloid, line, sign wave, helixoid, pure screw} (Figure 2.1a).
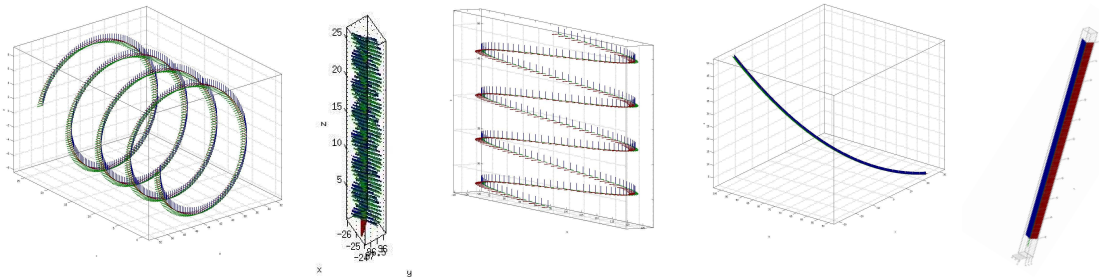
(a) Instances of each Primitive motion category (PMC)
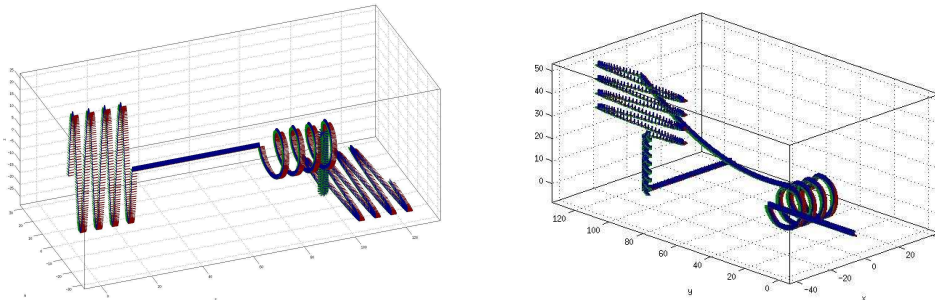
(b) Randomly generated trajectories from PMCs

Figure 2.1: PMCs and Random Toy Trajectories

Each PMC is a set of instances of motion primitives that share the same underlying dynamics, but are subject to translation, rotation and scaling in space and time. For example, a sign

---

[1] http://michaelchughes.github.io/NPBayesHMM/

wave constrained to the X-Y plane in a world coordinate system belongs to the same PMC as a sign wave constrained to the X-Z plane with shorter duration and larger amplitude. We create sequential random combinations of these primitive motion categories, to form N-trajectories with shared PMCs (Fig. 2.1a). Our generated toy trajectories are 7-dimensional time series, which represent the rigid motion (3d for Cartesian position and 4d for quaternion orientation) of a human/robot end-effector or object within a continuous demonstration (Fig. 2.1b). We present results from 3 experiments, each with increasing complexity (degree of variation within PMCs) for the segmentation algorithm (Fig.2.2 and Table 2.1).
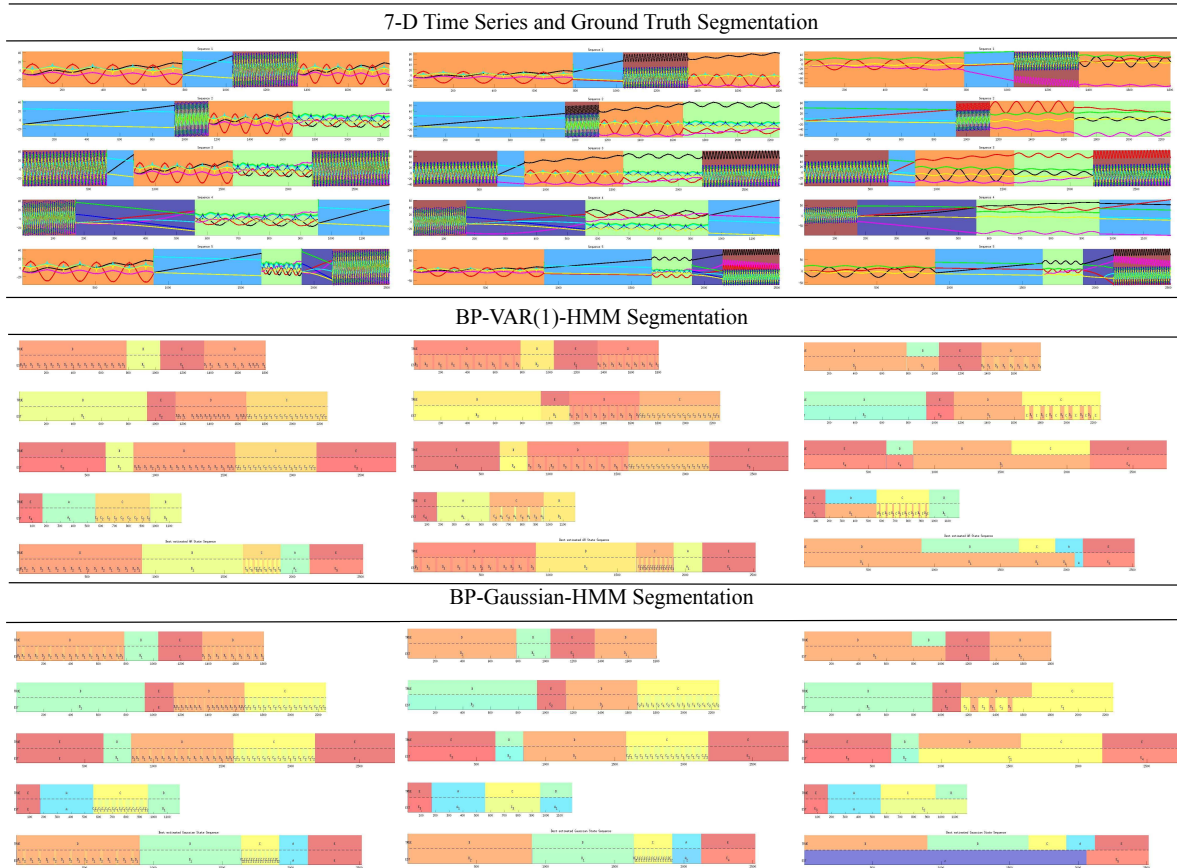

Figure 2.2: Segmentation results for incremental experiments (left column) Exp-1 Invariant PMCs (middle column) Exp-2 PMC subject to t and (left column) Exp-3 subject to t and R

Pure segmentation performance is evaluated with precision, recall and F-measure on the identified segment boundaries, where TP: correctly identified boundary, FP: incorrectly identified boundary and FN: unidentified boundary. To measure the performance of behavior sharing we use metrics based on conditional entropy analysis for clustering evaluation (i.e. homogeneity, completeness and V-measure introduced by Rosenberg and Hirschberg (2007)). Homogeneity is a score defining if each cluster contains only members of a single category, completeness is the score describing that all members of a given category are assigned to the same cluster and the V-measure is their harmonic mean. The range for the error metrics (both segmentation and behavior sharing) is bounded to [0,1].

The first experiment involves validating the applicability of this segmentation approach without

Table 2.1: Error Analysis of Segmentation and Behavior Sharing.

Experiments	Segmentation			Behavior Sharing		
	Precision	Recall	F-measure	Homogeneity	Completeness	V-measure
Exp.1 (VAR)	1.0	1.0	1.0	1.0	0.747	0.855
Exp.1 (Gau)	1.0	1.0	1.0	1.0	1.0	1.0
Exp.2 (VAR)	1.0	1.0	1.0	0.959	0.619	0.753
Exp.2 (Gau)	1.0	1.0	1.0	0.959	0.603	0.7404
Exp.3 (VAR)	0.928	0.765	0.839	0.609	0.479	0.536
Exp.3 (Gau)	0.9	0.563	0.692	0.701	0.436	0.538

variation between the motion primitives of each PMC (i.e. no translation, rotation nor scaling). This experiment is completely unrealistic but it proves that if motion primitives from the same PMC have no variations between them, the segmentation and behavior sharing perform perfectly for both VAR and Gaussian models (F-measure=1.0, V-measure=1.0). For the second experiment, we apply only translation between the PMCs. This dataset is more realistic, since it is a continuous trajectory through space and time, however the only difference between motion primitives within a PMC is their position in space. Regarding segmentation, both generative models still perform with high accuracy (F-measure=1.0), however motion grouping increasingly deteriorates (V-measure-VAR=0.753,V-measure-Gau=0.7404). In the third experiment, we increased the variation within PMCs, now subject to translation and different rotations. The performance of both algorithms decreases dramatically, not only is motion grouping worse (V-measure-VAR=0.536,V-measure-Gau=0.538), but the segmentation is less accurate as well (F-measure-VAR=0.839, F-measure-Gau=0.692).

This leads us to the conclusion that the BP-HMM segmentation approach is only robust for PMCs with a minimal level of variation, which in a realistic application (i.e. tracking data of a human demonstrating a recipe or kinesthetic teaching of a complex task to a robot) is very hard to find. However, since the algorithm shows unprecedented performance for segmenting and grouping *invariant* PMCs, the next step in our work is to find an invariant representation of the primitive motion trajectories in order to feed invariant PMCs to the BP-HMM algorithm. Moreover, this approach will be contrasted to the automatic segmentation based on variance developed by our group (Pais et al, 2013) and on the ROBOHOW data for the tasks of flipping pancakes and of grating carrots.

Chapter 3

Learning object-level impedance control

Our goal in this part of work is to learn the object-level impedance control for robust grasping and dexterous manipulation. This work comes in the context of Task 5.3 in WP5, i.e., *learning adaptive stiffness control that has the desired effects on object*, which includes learning the dynamics of the hand and finger motion to ensure stable grasp and desired object manipulation. In both of these two tasks, appropriate grasping forces need to be applied on the grasped or manipulated object, either to keep the grasp stable under perturbation or to move the object to a desired configuration. This force pattern can be extracted directly using contact forces information or indirectly using impedance information.

Our approach is based on extracting the desired impedance characteristics from human demonstration, in which the robust grasping and dexterous manipulation are demonstrated by a human expert. Since the involvement of coordination of multiple degree of freedoms and multiple contacts in these tasks, it is usually very difficult to directly obtain the desired impedance pattern from the robot's perspective. To this end, we proposed an approach that learns the desired impedance pattern from object's perspective and validated this object-level impedance controller on both robust grasping and dexterous manipulation, see Fig. 3.1. In the following parts, we will briefly present our approach on learning the object-level impedance from human demonstration.
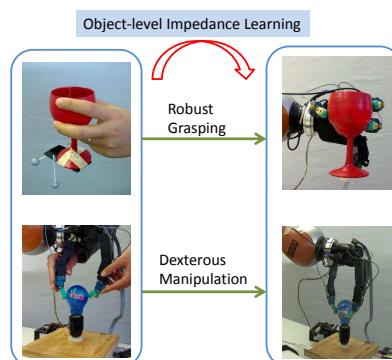

Figure 3.1: The object-level impedance for robust grasping and dexterous manipulation. The **left** figures are showing the human demonstration, while the **right** figures are the implementations on Allegro hand.

Relative Impedance for Robust Grasping   Our method of impedance selection for robust grasping is quite intuitive: the object stiffness in one direction is inversely proportional to the variance of displacement under perturbation in the corresponding direction. From this assumption, we can learn the relative stiffness for robust grasping in different directions from human demonstration. During the demonstration, an object is grasped by a human demonstrator with eyes closed. The grasped object is perturbed by another person randomly and the displacement of the object is recorded $\{\mathbf{x}^i, i = 1...N\}$. Then the object stiffness can be specified as follows:

$$K = \alpha\{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^i - \mathbf{x}_r)(\mathbf{x}^i - \mathbf{x}_r)^T\}^{-1} \tag{3.1}$$

where $\alpha \in \mathbb{R}^+$ is a ratio parameter that needs to be set manually and $\mathbf{x}_r \in \mathbb{R}^6$ is the object initial (and desired) position and orientation.

Variable Impedance for Dexterous Manipulation   For dexterous manipulation, besides the desired impedance, a time-varying reference trajectory $\mathbf{x}_r, \dot{\mathbf{x}}_r$ will also be required. This problem can be formulated as an optimization problem that learns the reference trajectory and the desired object impedance simultaneously. During the demonstration, at each sample instant, the motion of the object $\{\mathbf{x}(i), \dot{\mathbf{x}}(i)\}$ and the sum of manipulating forces $\mathbf{f}_{f,o}(i)$ applied on the object are recorded. Consider $t = 1...N_t$ consecutive samples of data obtained over a short time window. Assuming the impedance parameters and reference trajectory remain constant over this time window, the relationship between the object motion and the force exerted on object is given by:

$$\mathbf{f}_{f,o}(i) = D(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)) + K(\mathbf{x}_r - \mathbf{x}(i)), i = 1...N_t; \tag{3.2}$$

Since the object's impedance parameters and the reference trajectory are not changing with time over this time window, they can be obtained by minimizing the following objective function:

$$\min_{D,K,\dot{\mathbf{x}}_r,\mathbf{x}_r}\sum_{i=1}^{N_t}\|\mathbf{f}_{f,o}(i) - \{D(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)) + K(\mathbf{x}_r - \mathbf{x}(i))\}\|^2 \tag{3.3}$$

The additional constraints on the impedance parameters (positive-definite) and reference trajectory (upper bound) can be also taken in account in this optimization framework. For more details, one can refer to our recent submitted paper (Li et al, 2014), in Annex.

Li M., Yin H., Tahara K., Billard A. (2014) Learning object-level impedance control for robust grasping and dexterous manipulation. *International Conference on Robotics and Automation (ICRA), 2014. (Submitted)*.

Chapter 4

Learning of haptics interaction

We consider the case where the robot has to work in direct physical contact with the human via an object, more specifically, for a joint beam transportation task with a human operator. We seek to prove that segmentation of the task is also a valid approach in this case.

To this end, we monitor the task performed by two human partners on one degree of freedom and extract a model of segmentation of the task: the beam's trajectory can be segmented in constant velocity phases in the world frame, which we call *motion primitives*. This segmentation has been validated by implementing a pro-active behavior of the HRP-2 robot: the robot uses haptic signals to guess the human's intention, and then sequences the motion primitives to help him performing the task. This segmentation has then been extended by including lateral and rotation motion primitives, without monitoring humans. This extension allows the robot to perform a more complex task than the one that was monitored, which is also validated on the HRP-2 robot. The velocity of the beam on the frontal axis is similar in shape to the one observed with human dyads. Those works are presented more in detail in Bussy et al (2012b,a). This approach shows that segmentation of the task, in motion primitives in this case, is a valid approach even for physical collaborative tasks.

Yet, by evaluating this work, we found that the main issue was in detecting the change of intention from the human partner - more specifically how to detect the transition between states of the finite state machine. The main concern being that haptic information alone is lacking. Because of this, we continued this work by adding the use of visual data in human-humanoid joint actions. In addition to the observation of the human operator to follow his intention, this would also allow more complex tasks that can take advantage of vision. For example, we can now place objects on top of the table and add task constraints that try to keep these objects from falling of the table. In Agravante et al (2013), a height control using visual servoing was implemented to maintain a static object on the table while the human operator changed its height. This was expanded further in Agravante et al (2014), by using a ball - an extreme case of a moving object on top of the table.

These tasks show how vision data is complementary to haptic data and how a more complex task can be added while still retaining the carrying task created previously. Furthermore, we also investigate how the task knowledge from vision and the haptic information on human intention affect the impedance control framework used in the context of the collaborative task Agravante

et al (2014). In parallel to increasing task complexity, investigations are also in progress on using vision to monitor the human partner during the task and extract possible useful features that can help to determine the finite state machine transitions.

This chapter is based on the following publication that has been accepted in the International Conference on Robotics and Automation:

Agravante, D. J., Cherubini A., Bussy A., Gergondet P., Kheddar A. (2014) Collaborative Human-Humanoid Carrying Using Vision and Haptic Sensing *International Conference on Robotics and Automation (ICRA), 2014.* (*Accepted*).

Task Parametrization Using Continuous Constraints Extracted from Human Demonstrations

Ana Lucia Pais, Keisuke Umezawa, Yoshihiko Nakamura, and Aude Billard

Abstract—**Performing daily tasks requires a robot to combine representations of individual actions to achieve the task goal. In this work we propose an approach for learning task specifications automatically, from observing human demonstrations. We hypothesize that task specifications consist in tasks variables that present a pattern of change that is invariant across demonstrations. We identify these specifications at different stages of task completion. Changes in task constraints allow us to identify transitions in the task description and to segment the into subtasks. We use a variance-based method for extracting task-space constraints, and identify: (1) the reference frame in which to express the task variables, (2) the variable of interest at each time step, position or force at the end effector; and (3) a factor that can modulate the contribution of force and position in a hybrid impedance controller. The approach was validated on a 7 DOFs Kuka arm, performing a kitchen task, grating vegetables. Generalization was shown with respect to transferring the control strategy to other vegetables and across grating positions. A quantitative evaluation was conducted.**

Index Terms—**learning and adaptive systems, motion control, constraints extraction, programming by demonstration.**
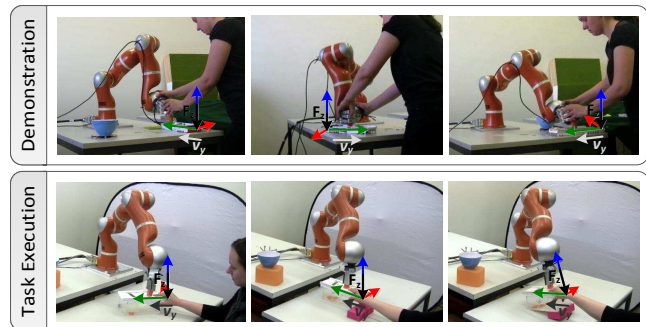

Fig. 1: From recording different human demonstrations, we detect the relevant frame of reference and the direction in which a hybrid force and position controller must be applied. In this figure the robot has correctly extracted that the frame of reference is attached to the grater and that force has to be applied along the vertical axis, whereas position control is needed along the horizontal plane of the grater. This allows the robot to reproduce the task even when the grater is moved in a different position and orientation.

I. INTRODUCTION

DAILY activities such as dish washing or preparing a meal often require completing a series of atomic actions while interacting with multiple objects. When performing such tasks, humans are able to focus on the key aspects necessary for achieving the goal. For example when grating a vegetable they naturally push against the grater, and focus on maintaining a certain speed and contact force with the grating surface. Moreover, humans naturally introduce variability by repositioning objects or by using different paths between two objects. Consequently, obtaining a feature-based representation for such high-level tasks assumes firstly to account for the large variability between demonstrations and to decide what feature should be reproduced (extracting task constraints with respect to trajectories and force profiles), and secondly to relate these features to the objects in the task (extracting the suitable frame of reference).

In this work we propose an approach for automatically extracting continuous task constraints required for successfully completing the task. We use Programming by Demonstration (PbD) to record kinesthetic demonstrations while using various initial positions of the robot and spatial configurations of the used objects (see Fig. 1, top row). Our approach exploits this variability between demonstrations to learn a criterion

A. L. Pais and A. Billard are with the Laboratory of Learning Algorithms and Systems (LASA), EPFL, Switzerland e-mail: {lucia.pais, aude.billard}@epfl.ch.

K. Umezawa and Y. Nakamura are with the Graduate School of Information Science and Technology, University of Tokyo, Japan e-mail:{umezawa, nakamura}@ynl.t.u-tokyo.ac.jp

for determining a notion of coherence in the demonstration. Specifically, a task variable (such as the force perceived at the end effector) might have a large variability within a demonstration, thus indicating that it becomes important in only a given region of the task. Moreover, regions in which a variable changes very little throughout a set of sequential demonstrations prove that the demonstrator was coherent in that part of the task. Therefore we focus on extracting such behaviors as the task constraints that should be reproduced.

Based on this criterion we learn a decomposition between force and position control and the frame of reference where this applies at all time. We fix a priori the type of controller, using a hybrid cartesian impedance controller throughout the task, and learn a weighting factor between the force and position that modulates their contribution by adjusting the controller's stiffness. This approach is designed as a bootstrapping process preceding learning a task model, that extracts the constraints without requiring any prior information about the type or goal of the task, nor models of the objects.

For learning a task model we consider that a change in the extracted constraints indicates a change of the atomic action to be performed, thus implicitly segmenting the demonstration data. This allows us to learn a model for each atomic action, in the local frame of reference, using the data between two changes of constraints. When encoding the motion profile we aim to preserve the exact behavior seen during demonstration. We therefore choose to encode variables that show a temporal coupling (like position and orientation, that change synchronously towards a target posture (*the attractor*)) using our Coupled Dynamical Systems (CDS) approach [1]. This encompasses the following advantages: (a) the motion is encoded in a time independent manner and ensures asymptotical

stability at the target of both dynamical systems; (b) the motion follows the demonstrated dynamics even if the execution starts from unknown regions of the space, far from the demonstrated motion, without the need to replan or re-scale the trajectory; (c) the temporal–correlated behavior of the two variables is preserved and thus a perturbation in one of the systems does not cause an unsynchronized behavior, the robot being able to adapt online to changes in the environment. The force profile is encoded separately, as a function of the position. This allows the robot to execute the task in changing conditions and to generalize to situations that were not seen during training (see Fig. 1, bottom row).

For reproducing the task we assume that the flow of actions (i.e. the task sequence) is implicit in the demonstration. Thus, a Finite State Machine (FSM) is generated for reproduction. The states are not known a priori but extracted. They correspond to the atomic actions identified previously and encode their corresponding constraints. The task is executed using a single controller and embedding the constraints online.

Automatically extracting this constraint–based task parametrization has several advantages. First, the constraints are extracted from variables that can directly be used for control, while learning the task in cartesian space makes the skill easily transferable to a new robotic platform. A single controller is used throughout the task and no switching occurs. Moreover, extracting the suitable frame of reference and expressing all the variables in this local frame allows simplifying the control (i.e. perform force and position control on perpendicular axis with respect to the object), and makes the skill easily generalizable to different locations or similar objects. Finally, by encoding the task in a time independent manner the system is robust to perturbations.

The proposed approach achieves a unitary task representation, as described in Section III, which we validate on a kitchen task, *grating vegetables* in Section IV. We discuss the advantages and limitations of our approach in Section V and present related work in Section II.

II. RELATED WORK

When encoding robotic tasks, for achieving generality of the learned model, the representation of the motion is often related to task space constraints, either *natural constraints* that arise from the physical properties of objects, or *artificial constraints*, such as following a specific motion profile [2]. In addition constraints can be *soft constraints* that allow to loosely follow a behavior with a certain flexibility [3], [4] (i.e. a slack in the desired velocity, motion or force profiles); or *hard constraints* that depict parts where the motion has to be exactly followed [5], [6]. An innate ability to understand such constraints also drives human learning (see Section II-A). In our work we focus on extracting artificial task constraints, and encode in a continuous manner soft constraints based on the variance existing in the demonstration data. Furthermore we reconstitute the task from an extracted sequence of states, parameterized with the extracted constraints. We review related work with respect to similarities with human learning, automatic extraction of constraints, task segmentation, and constraint–based motion planning.

A. Human inspiration

In imitation learning, coherence between demonstrations is an important factor for properly generalizing a task, therefore the variance encodes key information, also exploited in human learning [7]. Infants learn by detecting "statistical regularities" in the stimuli they face (such as auditory, linguistic [7], or visual cues [8]). These regularities draw their attention [9] and allow them to form an understanding about an environment whose structure was not otherwise specified [10]. Therefore, by detecting features encoded in the signal itself they can learn a task without being aware of the conceptual goal. Moreover infants have an innate ability to segment the action performed by other people that they observe, which gives them a high-level understanding of what the task consists of [11], and the intention behind it [12]. However, aside the implicit cues found in the demonstration, the ability to link these to the correct objects is important to achieve generalization. Studies show that adults are able to infer the correct reference frame in order to (a) build a personal perspective on a spatial scene [13], and (b) to perform spatial reasoning based on relations between objects [14].

The work that we propose follows the same principle, namely to focus on important cues in the signal in order to infer a notion of coherence that the demonstrator had. Based on this assessment implicit segmentation is performed, rather than using the engineering approach of seeking explicit segmentation. Moreover humans often form their understanding of how to properly manipulate objects, what forces should be applied and how to adjust the compliance of the arm, by actually trying to perform the task. Therefore we use kinesthetic demonstrations that provide the robot with information about forces and additionally facilitate the interaction by being similar to the way humans teach their children [15]. Moreover enhancing these methods with the capability of interpreting a task in a way that is akin to human learning might lead to increasing the robot's acceptance [16].

B. Automatic extraction of task-space constraints

Previous approaches at extracting task constraints focus mainly on the suitable frame of reference and encode the motion in a time dependent manner. Exploring low variance regions for determining task constraints or segmentation points has been previously explored, however we depart from these approaches by increasing the complexity and number of the encoded constraints. Moreover we consider continuous constraints that may apply throughout or only on a subpart of the task. Finally, we use a single controller throughout the task execution, while the constraints identify values taken by the variables of the impedance controller as the task unfolds.

In our previous work [17] we proposed extracting the reference frame in a manipulation task with respect to a proposed metric of imitation. Data recorded from demonstrations (joint angles, hand cartesian position relative to the objects and gripper status) is projected into a lower dimensionality latent space and further encoded in a time-dependent manner using a Gaussian Mixture Model (GMM). Gaussian Mixture Regression (GMR) is used to reproduce the motion. In a previous attempt,

temporal variations are encoded in an Hidden Markov Model (HMM) and implicit segmentation is performed through HMM states. [18]. These implementations have the limitations of encoding the motion in a time-dependent manner and lack the information about forces exerted on objects, that in some tasks might be key to achieving successful reproduction.

In this work we build on these existing approaches by additionally extracting constraints with respect to force profiles and robot stiffness in different regions of the task. Moreover we encode the motion using a Coupled Dynamical Systems (CDS) approach [1], which presents several advantages over encoding the motion using a GMM (as in [17]) or Dynamic Movement Primitives (DMP) (such as [19]). Firstly it is time invariant making the robot robust to perturbations, and ensures stability at the target, while reaching the attractor in a coordinated manner that resembles the original demonstrated dynamics of the end effector's position and orientation. This makes the motion very human-like, so there is no need for additional processing (such as synthesizing variance during execution [20]) to achieve a natural–looking movement. Secondly, from an execution point of view this encoding provides the ability to generalize to new contexts and adaptation to perturbations occurs in a reactive manner without replanning.

An alternative way of encoding a motion is based on an invariant representation [21]. While the motion in this form is not directly usable to reproduce the motion in changing contexts, it has the advantage of facilitating motion recognition regardless of the task frame. A different method is described in [22] in which a task space is selected based on three criteria: a variance–based analysis of object trajectories, attention focus on objects in the task and an evaluation of the teacher's discomfort during demonstration. While this method tends to be more complex, it is applied solely to vision–tracked human demonstrations. In our case the demonstrations are performed kinesthetically in order to allow the robot to experience forces that should be applied on objects. Moreover in [22] the authors perform an indirect evaluation of the data by suggesting that if a human maintains an uncomfortable posture during demonstration then this might be important for the task. In our case a direct evaluation is done on robot's proprioceptive data, while the user chooses an arbitrary position for demonstration.

Moreover the ability to successfully perform complex tasks resides in making use of additional sensing. For example, assessing joint torques values can be an indicator of whether the motion of the end effector is constrained [23]. Therefore the second aspect that we address is detecting axes in task space where force control applies and encoding these force profiles. Additionally varying the robot's stiffness according to the task ensures safer interaction [24]. However, adding the force information, while of high importance for the task, can be challenging depending on the platform. A method is proposed in [25] for using kinesthetic teaching for demonstrating the motion, and using a haptic device for demonstrating the force that should be applied while the robot is replaying the trajectory. In our work, we extract the task constraints from a low number of demonstrations. Moreover we extract atomic actions, thus making the approach suitable for tasks that encompass switching between multiple atomic actions.

C. Task segmentation

Complementary to the constraints extraction topic is that of performing task segmentation which on the long term offers the possibility to easily recognize, classify and reuse motions [26]–[28]. In our work we do not explicitly seek to segment the data, however segmentation occurs naturally when the task constraints change. This allows a flexible representation of the task, exploiting the local behavior in each sub-task. A vast majority of recent works focus on segmenting motion data represented by sets of joint positions or hand positions and orientation retrieved by motion capture systems in the case of human motion and by robots proprioception in the case of robotic motions. However very few works focus on segmenting task data that includes force information.

Current existing approaches for motion segmentation [26] rely on either (1) classification based on existing motion primitives used for prior training [29]–[31]; (2) looking for changes in a variable, like zero-crossings [32]; or (3) clustering similar motions by means of unsupervised learning [33]. The downside of these approaches is the need of prior task knowledge, which may be poor and incomplete in real-life situations. Furthermore most segmentation approaches rely on other techniques for human motion analysis which include [26]: Dynamic Time Warping (DTW) used in the temporal alignment of recorded data; or HMM for analyzing data that varies in time (such as hand movements sign language [34]). Additionally when a human demonstrates a task to a robot, he/she may stop during the demonstration to rearrange an object or teach in a different manner. In these cases the above mentioned approaches over-segment the data. Regions of low variance have been alternatively used to determine segmentation points [35]. We take a similar approach, but extend this method to analyze other motion constraints, while removing the over-segmentation.

The first approach for segmentation can ease robot control because of the existence of motion primitives. However while it is safe to assume that human motions are likely to follow a specific pattern in a known context, rather than being random (as shown in [36]), a major drawback is the need to include prior knowledge. It also restricts the scope of segmentation by knowing what the task is about. An example is described in [30] for segmenting motions used in robot assisted surgery.

The second segmentation involves searching for zero ve-locity crossings (ZVC) [32] or other changes in a variable compared to a known state [23]. This approach is sensitive to the variables encoded while one needs to find a way that would ensure optimal segmentation across all task dimensions. The third approach encompasses a more complex view of human motion, such as learning and clustering motion primitives in an incremental manner, from observing human motion [33]. The method in [33] performs unsupervised segmentation based on motion encoded through an HMM. The obtained segments are clustered according to a measure of relative distance and organized in a tree structure. It encodes generic motions at the root, that gradually become more specialized close to the leaves. The algorithm allows to change the model according to known primitives [37], and to use the same learned model
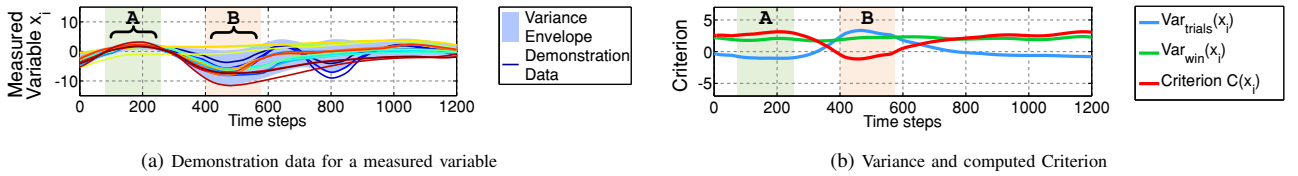
(a) Demonstration data for a measured variable

(b) Variance and computed Criterion

Fig. 2: Example of recorded data and computed variance over trials (Var_{trial}) and over a time window (Var_{win}) for a measured variable x_i. Region **A** shows data with little variance across trials (i.e. a feature of that should be reproduced). Region **B** shows data with large variance over trials, and low variance over a time window (almost constant).

not only for recognizing, but also for generating motions [31]. While being one of the most robust implementations to date, the approach lacks time independency in motion encoding.

These approaches, while efficient, have the shortcoming of not only being task specific and requiring a considerable amount of prior knowledge (and thus achieve little generalization across a wide range of tasks), but they also fail to model specific features of the motion, focusing mainly on changes in position. Moreover these algorithms focus on extracting motion primitives, as opposed to learning a parametrization of a control system that remains the same all along the task, as in your approach. This allows learning and reproducing a task in a seamless manner.

D. Constraint based motion planning

Alongside motion segmentation, learning systems should encompass features such as [33]: (1) the possibility to recognize a motion similar to one that was previously learned, (2) the ability to correctly categorize a new motion and (3) structuring the learned models such that the data can be easily accessed and used. The first two features can be explored in long-term learning, while the last aspect is essential as it allows using learned actions for generating higher–level planning strategies.

Previous ways of encoding the task sequence use: Petri nets, Markov Models [35], [38], graph representations [39]. In our work we consider the sequence of atomic actions implicit in the demonstration and extract a finite state machine to execute the task, in which the states encode the extracted constraints. Moreover, representing a task with respect to its goals allows a tree representation of skills [40], or a similar graphs-based representation for strategies [39]. While this means incorporating external knowledge in the representation, our implementation takes a lower level approach by encoding constraints that guarantee the task success without knowing the conceptual goal, and isolating atomic actions thus allowing individual reuse [41].

Properly segmenting and encoding atomic motions can be used for constrained motion planning [42], [43], while a constraint based representation of a more complex task can be used by a high level planner [44] for executing plans or for inferring motion grammars [45] for a high-level representation. A constraint based framework for specifying a task is described in [46], where a task is defined and executed as a finite state machine (FSM) based on the existing constraints. In our approach the parametrization of the FSM comes from the constraints extracted from demonstration.

III. METHOD

In this work we identify task constraints directly from variables that can be used for control (end effector position and force). This enables a consistent way of encoding all the subparts of the task for using a single controller and ensures a smooth reproduction by directly embedding the constraints during the execution. We use a hybrid impedance controller:

$$\tau = J^T(K(x - \tilde{x}) + F) \qquad (1)$$

in which the following variables: $\tilde{x} \in \mathbb{R}^6$ – the desired cartesian trajectory, $F \in \mathbb{R}^6$ – the desired force and $K \in \mathbb{R}^{6 \times 6}$ – the stiffness matrix are extracted from user demonstrations.

Moreover we extract a frame of reference RF in which the variables are most consistent and that in some cases may represent a quasi-orthogonal decomposition of position and force control along the axes of the object, although we continuously use a hybrid controller.

$$\tau = J^T RF \begin{bmatrix} K_1(x_1 - \tilde{x}_1) \\ K_2(x_2 - \tilde{x}_2) \\ F \end{bmatrix} \qquad (2)$$

A set of kinesthetic demonstrations is recorded, in which a user physically guides a passive robot throughout the task. Data are recorded from the robot and expressed in a fixed referential RF_0 located in the robot's base: end effector position and orientation as well as external forces estimated at the end effector. Additionally the objects positions with respect to the robot's base are tracked by a vision system.

From the demonstration we learn a desired path and stiffness profile for the directions along which position is the variable of interest. Secondly for the directions along which force is important, we learn a dependency between the desired force profile and the desired trajectory, i.e. F becomes a function of other variable, e.g. in Eq. 2, $F = f(x_1, x_2)$.

The proposed method for extracting the task constraints is illustrated below, on an uni-dimensional measurement ($d = 1..D$, where $D = 1$) of two variables: force $F \in \mathbb{R}$ and cartesian position $x \in \mathbb{R}$ of the robot's end effector. The data set is a vector of $k = 2$ components: $\xi_d^i = \{F_d^i, x_d^i\}$ considered to be recorded over a number of N demonstrations of a task (see Fig. 2 (a)). The upper indices correspond to representing the data in the reference frame of each object $o_i, i = 1..N^o$, from the total N^o objects involved in the task. The ξ^0 corresponds to the original recorded data (in RF_0), the fixed referential in the base of the robot. The data was temporally aligned using Dynamic Time Warping (DTW), resulting in a set of length T. For the purpose of this example
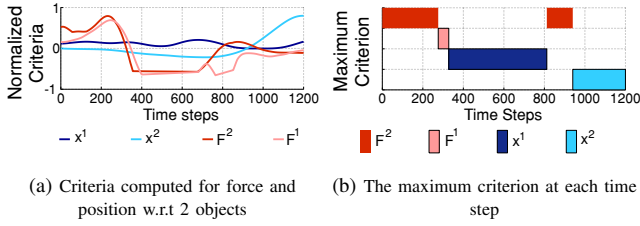
(a) Criteria computed for force and position w.r.t 2 objects

(b) The maximum criterion at each time step

Fig. 3: Comparison between the criteria computed for force (F) and position (x) in 2 reference frames (RF_1 and RF_2).

(a) The extracted reference frame at each time step

(b) The extracted variable of interest at each time step

Fig. 4: The reference frame and variables of interest are given by the maximum criterion in a time window ω_i.

we drop the lower index d. Note that in a typical robotic task a minimum of $k = 6$ variables have to be compared if using a 3D measurement of position and force. The data is analyzed with respect to the variability that exists in the motion. For all objects o_i we compute the variance of each recorded variable k in two ways: (1) as the averaged variance of the signal over trials (i.e. consecutive demonstrations):

$$Var_{trial}\left(\xi_{d,k}^i\right) = \frac{1}{N_D}\sum_{i=1}^{N_D}\left(Var(\xi_{d,k}^i)\right) \qquad (3)$$

and (2) the averaged variance of the signals over a time window ω. An example is given in Fig. 2 (b).

$$Var_{win}\left(\xi_{d,k}^i\left(t:t+\omega\right)\right) = \frac{1}{N_D}\sum_{i=1}^{N_D}\left(Var(\xi_{d,k}^i(t:t+\omega))\right) \qquad (4)$$

The size of the time window is chosen arbitrarily as being the shortest time period in which we see noticeable changes in the task flow. The values of the two variances are normalized such that $Var_{trial}, Var_{win} \in [-1,1]$.

We postulate that if a variable (a) changes value significantly within a single demonstration and (b) changes this value in a *systematic* way across demonstrations then this variable is *significant* for the task. It hence becomes a task constraint that should be reproduced. We thus develop a criterion given by the difference between the variance over the time window and that over trials, that allows comparing the task variables in a relative manner, without setting any hard thresholds. At each time step the criterion is computed as:

$$C\left(\xi_{d,k}^i\right) = Var_{win}\left(\xi_{d,k}^i\right) - Var_{trial}\left(\xi_{d,k}^i\right) \qquad (5)$$

and the obtained value is normalized, such that the criteria $C(\xi_{d,k}^i) \in [-1, 1]$. The total number of criteria to be computed for a task is given by $N^C = N^o \cdot k \cdot D$. In this example $N^C = 4$, as seen in Fig. 3 (a).

A. Determining the Task Constraints

Using the defined criterion we extract the following task constraints: the frame of reference and the relative positioning within an object space (as explained in Section III-A1), relative importance of position and force on each axis of the object (see Section III-A2), and a weighting factor between the two, used to modulate the controller's stiffness throughout the task (Section III-A3). The procedure is summarized in Alg. 1.

1) Extraction of the Reference Frame: Expressing the control variables in the local reference frame attached to the object
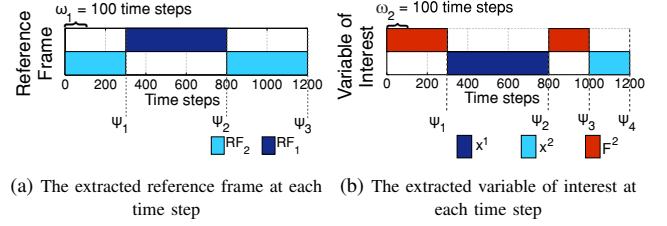
on which manipulation is performed at a given time, allows the robot to properly execute the task when the positions of the objects change in the scene. Moreover this allows us to consider constraints not only as factors that limit the robot's motion [47], but that also add meaning to the motion (i.e. a grating motion, characterized by a given force and motion profile, is only meaningful when performed on a grater and in the context of a grating scenario).

For choosing a frame of reference we compare the computed criteria and choose at each time index t, $t = 1..T$ the value of the highest criterion for all the variables considered $max(C(\xi_{d,k}^i))$, see Fig. 3 (b). Thus the vector of obtained maximum values $max(C(\xi_{d,k}^i))$ is analyzed separately for each dimension d, using a time window of arbitrary size (in this case $w_1 = 100$ time steps). We consider that in each time window the reference frame is given by the object o with the highest number of occurrences of its corresponding criterion $max(C(\xi_k^o))$. In this example there are two changes of reference frame, as shown in Fig. 4 (a): for the first 100 time steps the RF is given by object o_2, for the next 200 time steps there is a change to o_1, and for the rest of the motion the RF is changed to o_2.

The changes in the reference frame determine a set of *segmentation points* ψ_s which delimit the actions performed on each object. In this example there are 3 actions (one performed on object 1 and two performed on object 2) determined by the change of RF. Each segmentation point corresponds to a state that contains the time index t_s when the change occurred and the id of the reference frame used up to that point $\psi_s = [t_s, RF_s]$.

2) Extraction of the Relevant Task Variables: The criterion defined in Eq. 5 allows us to compare in a relative manner the influence of variables of different types (like force vs. position), and that vary across different scales, see Fig. 3 (a). The aim is to be able to quantify their relevance with respect to the task, so as to give more importance to the variable of interest in the controller and to adjust it when a change occurs.

For determining the relevant task variables, we analyze the criterion on each dimension d using a time window of arbitrary size (in this case $w_2 = 100$ time steps). Similarly to extracting the reference frame, we consider the relevant variable in each time window to be the one that has the highest occurrence of its corresponding maximum criterion in that interval. In the given example, there are several changes between position and force as variables of interest (see Fig. 4 (b)).

The changes in the variable of interest determine additional *segmentation points* which together with the initial points

Algorithm 1 Task Constraints Extraction

Bootstrapping(*Set of N demonstrations:* $\xi_{d,k}^i = \{F_d^i, x_d^i\}$)
$$\qquad\qquad\qquad\qquad\qquad\qquad {1 \rightarrow N}$$

Do DTW, dataset length T
Criteria: $C(\xi_{d,k}^i) = Var_{win}(\xi_{d,k}^i) - Var_{trial}(\xi_{d,k}^i)$
$s = 0$ % *number of segmentation points*
% *Determine the reference frame:*
for $t = 1 : \omega_1 : T$ **do**
$\quad RF(t) = RF_i$ for which $C_{max} = \underset{t:t+\omega_1}{max}(C(\xi_k^i))$
$\quad$ **if** $RF(t) \neq RF(t-1)$ **then**
$\quad\quad s = s + 1$; % *Create a new segmentation point*
$\quad\quad \psi_s = [t_s, RF_i]$ % *add the current constraints*
$\quad$ **end if**
end for

% *Determine the variable of interest:*
for each dimension $d = 1 : D$ **do**
$\quad$ **for** $t = 1 : \omega_2 : T$ **do**
$\quad\quad$ add $\xi_{d,k}^i$ to the current constraints vector
$\quad\quad \psi_s = [t_s, RF_i, \xi_{d,k}^i]$ for which $C_{max} = \underset{t:t+\omega_2}{max}(C(\xi_{d,k}^i))$
$\quad\quad$ **if** $\xi_{d,k}^i(t) \neq \xi_{d,k}^i(t-1)$ **then**
$\quad\quad\quad$ *Insert a new segmentation point*
$\quad\quad$ **end if**
$\quad$ **end for**

$\quad$ % *Determine the stiffness modulation factor:*
$\quad$ **for** each segment s **do**
$\quad\quad$ add $\lambda_{d,s}(t) = C(\xi_{d,1}(t) - \xi_{d,2}(t))$ to the constraints vector
$\quad\quad \psi_s = [t_s, RF_i, \xi_k^i, \lambda_{d,s}]$
$\quad$ **end for**
end for

return $\psi_{1:s}$
end

---

determined by the change of RF delimit individual atomic actions such as reaching movements. In the example described above, there are 3 segmentation points corresponding to the change of the variable of interest (see Fig. 4 (b)). The first two points are identical to the segmentation points ψ_1 and ψ_2 found by the change in the reference frame. The next point ψ_3 marks a change from a force-based part of the task to a position based part. The final point ψ_4 concludes the motion. The points are sorted according to the time index when the segmentation occurred. The information about the variable of interest is added to the vector $\psi_s = [t_s, RF_s, \xi_{d,k}^s]$.

3) Extraction of the Stiffness Modulation Factor: Determining the axis-specific relative importance between our two variables can be done by computing a weighting factor λ that balances the contribution of the force and position according to the relevance determined above. Thus, for each dimension d the value of $\lambda_d \in \mathbb{R}^D$ is given by the normalized difference between the criterion computed for position and the one computed for force

$$\lambda_d = C(x_d) - C(F_d) \qquad (6)$$

Thus the value of λ becomes a weighting factor for the controller's stiffness K. Therefore we can use an impedance controller for reproducing the motion with the factors described above representing continuously constraints, directly embeddable in the robot's control.

$$\tau = J^T \cdot RF \cdot (\lambda K(x - \tilde{x}) + F) \qquad (7)$$

The corresponding λ profile for each segment of the motion is added to the constraints vector $\psi_s = [t_s, RF_s, \xi_{d,k}^s, \lambda_s]$

B. Constraint–based Motion Learning and Execution

In our work, segmentation of the demonstrated data occurs implicitly whenever there is a change in the extracted task constraints. This is a natural manner of segmenting as the points in which either the reference frame or the variables of interest change, delimit atomic actions (e.g. the force sensed at the end effector might be relevant in the first part of the task while after the segmentation point, end effector's position could become more relevant). Segmenting and interpreting the data in a stochastic manner allows regenerating the motion according to the measures determined to be important as well as finding optimal control strategies with respect to the variables of interest (see Table I, Columns 1 and 2).

1) Learning the motion profile: We choose to encode the motion using a coupled dynamical system approach, as described in [1], which allows us to preserve the coupled evolution of position and orientation towards the target posture, that was observed in the demonstrations. Each individual variable is encoded as a non-linear dynamical system of the form $\dot{x} = f(x)$, which encodes the mapping between a variable and its first derivative thus removing the explicit time dependency. Here x and $\dot{x} \in \mathbb{R}^D$ represent the cartesian position and velocity of the end effector. The function $f : \mathbb{R}^D \mapsto \mathbb{R}^D$ (initially unknown, but implicit in the demonstrated behavior) is a continuous and continuously differentiable function stable only at the attractor $\overset{*}{x}$. The non–linear behavior of function f is encoded using a mixture of k Gaussians, specified by a vector $\theta_x^k = [\pi_x^k, \mu_x^k, \Sigma_x^k]$, representing the parameters of the GMMs (priors, means, covariance matrices), such that $P(x, \dot{x}|\theta_x^k)$ represents the dynamics of system 1. Based on this encoding the velocity $\dot{x}$ is thus computed as $\dot{x} = E\{p(\dot{x}|x; \theta_x^k)\}$. The model is learned through maximization of likelihood under stability constraints (see [1] for details).

In our case the absolute position of the attractor in each segment is estimated from the initial set ξ_0 (in RF_0) as the average of all the points from the N demonstrations, on each dimension d, at the segmentation time t_s, resulting: $\overset{*}{x}_d = \underset{1 \rightarrow N}{avg}(x_d(t_s))$. The motion is encoded in the attractor's reference frame RF_*, such that the attractor becomes $\overset{*}{x}_i = 0$. The attractor has the meaning of a relative positioning with respect to the reference frame RF_i in that segment of the task. Several attractors can be defined with respect to a single object. In a grating task for example there are two attractors with

---

Algorithm 2 CDS [1] motion generation for a segment ψ_i

CDS motion(C_{ψ_i}, current robot position, current object position)
if *current attractor* $\overset{*}{x}, \overset{*}{r}$ *not reached* **then**
$\quad$ % *Compute next end effector position*
$\quad\quad \dot{x} = E\{p(\dot{x}|x; \theta_x^k)\}$;
$\quad\quad x(t+1) = x(t) + \dot{x}(t)\Delta t$
$\quad$ % *Infer orientation based on current position*
$\quad\quad \tilde{r} = E\{p(r|\gamma(x); \theta_c^k)\}$;
$\quad$ % *Compute next end effector orientation*
$\quad\quad \dot{r} = E\{p(\dot{r}|\beta(r - \tilde{r}); \theta_r^k)\}$;
$\quad\quad r(t+1) = r(t) + \alpha \dot{r}(t)\Delta t$
end if
% *Return next desired position and orientation*
return $\{x(t+1), r(t+1)\}$
end

State	Constraints	Motion Encoding
ψ_1	$[t_{s_1}, RF_1, F, \lambda_1]$	$C_{\psi_1} = [C_x, \theta_F^k, \theta_\lambda^k]$
ψ_2	$[t_{s_2}, RF_2, x, \lambda_2]$	$C_{\psi_2} = [C_x, \theta_\lambda^k]$
ψ_3	$[t_{s_3}, RF_2, F, \lambda_3]$	$C_{\psi_3} = [C_x, \theta_F^k, \theta_\lambda^k]$
ψ_4	$[t_{s_4}, RF_1, x, \lambda_4]$	$C_{\psi_4} = [C_x, \theta_\lambda^k]$

TABLE I: Final task parametrization for the given example, consisting of states ψ_s, the extracted constraints and the corresponding statistical encoding to be used by the controller in each segment, C_{ψ_s}.
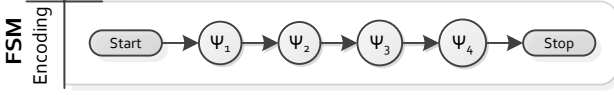
Fig. 5: Finite State Machine used for executing the task. Each state encodes the determined constraints. We consider that the order of the demonstrated actions is implicit for the task flow.

respect to the grating surface: the *top* (initial point touched on the grater) and the *bottom* (after passing the blade).

Similarly we encode the rotation specified by an axis-angle representation $r \in \mathbb{R}^4$, as $P(r, \dot{r}|\theta_r^k)$, with respect to an estimated attractor $\overset{*}{r}$. Finally $P(\gamma(x), r|\theta_c^k)$ represents a coupling function between the two systems, learned using maximization of likelihood. During the execution the system updates the dynamics of system 1 through GMR, second the coupling is updated and this determines updating the second system (see Alg. 2).

The model can be further parameterized to control the speed and amplitude of the robot's behavior under perturbation, using two scalars α, β. While in the original implementation in [1] these parameters are learned from recording good trials and perturbed demonstrations, here we can estimate them based on the variance information, such that in regions with high variability the adaptation is slower than in regions with low variability. Thus, in the proposed impedance controller, the reference trajectory for the reaching segments is given by the learned CDS model. This ensures that the learned model follows the original dynamics of the demonstrated motion, it is stable at the target. The synchronous evolution is ensured through a coupling function. The complete CDS encoding of the motion in a sub-part of the task is thus specified by the vector: $C_x = [\theta_x^k, \theta_r^k, \theta_\xi^k, \overset{*}{x}, \overset{*}{r}, \alpha, \beta]$.

2) Learning the force profile: For segments of the task, and across dimensions in which the force becomes important, we use GMM to learn a joint distribution of the variables F and x. We use a model comprising a mixture of K Gaussian components, such that: $p(F, x) = \sum_{k=1}^{K} (\pi_F^k \cdot p(F, x; \mu_F^k, \Sigma_F^k))$, where π_F^k, μ_F^k and Σ_F^k represent the priors, the mean and the covariance matrix for the Gaussian model. These parameters are learned through (EM) Expectation – Maximization algorithm. The vector $\theta_F^k = [\pi_F^k, \mu_F^k, \Sigma_F^k]$ is added to the $C_{\psi_s} = [\theta_F^k]$. During the execution, GMR is used for predicting the force to be applied based on the current position: $E\{p(F|x)\}$. Unlike the encoding of position, for the force there is no attractor, as force control is performed along a trajectory. Following the desired force in the running controller is ensured by a PD controller.

Algorithm 3 Constraint–based task execution

FSM Execution(ψ_i, C_{ψ_i}, $i = 1 : s$)
  do
  read robot current position $\xi_{d,1}$ and EE force $\xi_{d,2}$
  read objects positions
  for *all task segments s* **do**
    Use current state's constraints $\psi_s = [t_s, RF_s, \xi_k^d, \lambda_s]$
    Transform data to RF_s
    % Compute next desired robot position
    $\{x(t+1), r(t+1)\}$ = CDSmotion(robot position, object position)
    % Determine stiffness modulation based on current position
    $\lambda = E\{p(\lambda|x)\}$
    if *Force important on dimension d* **then**
      % Predict force based on current position
      $F = E\{p(F|x)\}$
    end if
    Transform all data back to RF_0
    Update robot's motion (according to eq. 7)
    Else *Go to the next state*
  end for
  until *Task completed*
end

3) Learning the stiffness profile: We encode the stiffness modulation factor λ similarly to encoding the force, by learning a joint distribution $p(\lambda, x)$ using a mixture of k gaussians. The model is parameterized by the vector $\theta_\lambda^k = [\pi_\lambda^k, \mu_\lambda^k, \Sigma_\lambda^k]$, representing the priors, means and covariance matrices.

4) Constraint–based Execution: We assume that the flow of atomic actions is implicit in the demonstration, thus the reproduction is based on the determined sequence of $\psi_{1:s}$ points. A finite State Machine containing the inferred states is generated, as shown in Fig. 5. A state is generated for each change of constraints and contains: (a) the extracted constraints, and (b) the learned motion models, as they are summarized in Table I, Column 3. The transition to the next state occurs when the attractor of the current state is reached. The execution of the task based on the extracted constraints is presented in Alg. 3.

IV. EXPERIMENTS

This approach was validated on a common kitchen task, *grating vegetables*, using a KUKA Light Weight Robot (LWR) with 7 degrees of freedom (DOFs). Two objects were used in the task: a grater and a bowl. Data was recorded from the robot at 100 Hz, consisting of: end effector position and orientation given as an homogeneous transformation matrix ($x \in \mathbb{R}^{12}$), and external forces estimated at the end effector ($F \in \mathbb{R}^3$). The objects are tracked at 1KHz using an OptiTrack motion capture system. The objects' initial position was static, and the user could move them freely during the demonstration.

A. Task description

The task consisted of several atomic actions, presented in Fig. 6: reaching from the initial position to the slicer (the motion takes around 3 to 5 seconds, until the demonstrator finds a conformable position), a repetitive slicing motion (on average around 30 seconds), a reaching motion from the slicer to the trashing container (on average 2 seconds). Typical demonstration data are presented in Fig. 7.

The variability of the task consisted in:

(a) Reaching the grater   (b) Grating the whole vegetable   (c) Trashing the remains
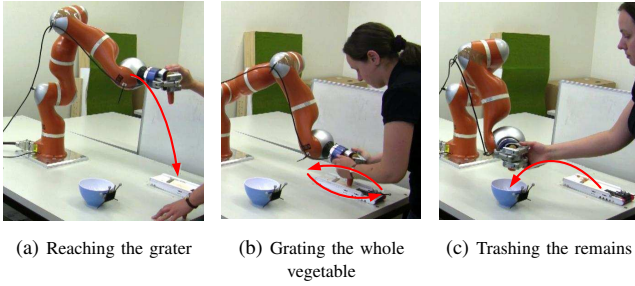
Fig. 6: Atomic actions in the *Vegetable Grating Task*. The user demonstrates the whole task at once, using different starting configurations of the objects and the robot.
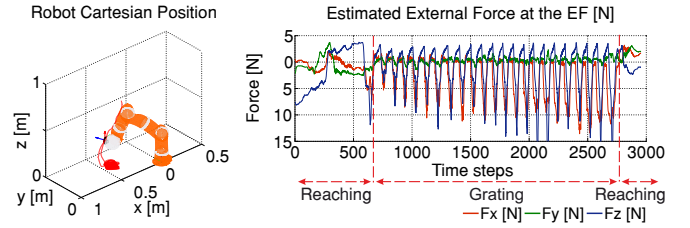

Fig. 7: Typical data obtained from a human demonstration, for the whole duration of the task: robot's cartesian position (left) and the sensed end effector force (right). Vertical lines were manually added to highlight the 3 parts of the task: reaching the grater, grating and reaching the trash.

1) Starting each demonstration from a different initial position of the robot, and placing the objects in different positions in the reachable space of the robot (we recorded data for 3 different positions of the objects, placed on average 30, 45 and 65 cm apart from the initial position);

2) Using vegetables of different sizes and types (we recorded data for 3 types of vegetables (carrots, zukinis and cucumbers). The vegetables varied in length, from a minimum of 10 cm for a carrot to a maximum of 35 cm for a cucumber, and with about 2 cm in diameter); the variability of the manipulated object affected the force applied by the user when providing demonstrations and the duration of the demonstration. The task lasted until the vegetable was fully grated;

3) inherent user variability between demonstrations.

A total of $N = 18$ demonstrations were recorded, 6 for each vegetable type, using 3 different objects positions and orientations.

B. Extracted Constraints

For extracting the task constraints we evaluated the 3D measurements of position and force projected in the reference frame of each object (see Fig. 8). For each dimension we computed the variance over trials and averaged over a time window (see Fig. 9), and the criterion, computed according to the equation 5. Following the approach described in Section III, the criterion on each axis was evaluated in a time window of width $w = 200$ time steps (2 seconds) for determining the reference frame (see Fig 10 (a)). This resulted in one segmentation point. The motions of reaching and grating were expressed in the reference frame of object 1, the grater, and the motion of reaching the trash container was expressed in the reference frame of object 2, the bowl.

Similarly, we evaluated the criterion on each dimension, using a time window of width $w = 300$ time steps (3 seconds) for determining the variable of interest (see Fig. 10 (b)). The results showed that the force on the vertical axis became important in the second part of the task (grating and trashing), while only position was important in the first part of the motion (corresponding to reaching the grater). The change in the variable of interest determined a new segmentation point. A final point concludes the motion.

C. Task execution

Three segmentation points ψ_s were determined for this task (see Fig. 11 (a)), involving 3 different states: (1) reaching using a position-weighted controller expressed in the RF of object 1, (2) grating using the RF of object 1 and (3) reaching for the trash bowl using the reference frame of object 2. Two attractors were determined relative to the grater: one near the handle (*Grater Top*); and one at the bottom of the grater (*Grater bottom*), after passing over the blade. Similarly at the end of the motion the positioning was relative to the trashing bowl. This allowed us to have an attractor based encoding of the task. A finite state machine was generated as described in Section III-B4, see Fig. 11 (b). For evaluation purposes the number of times the grating should be performed was added as an additional condition for the transition to the next state. The advancement of the FSM happened when the current attractor was reached, or when the number of grating passes was completed.

D. Framework Evaluation

We performed two different assessments with respect to (1) the correct extraction of task constraints; and (2) the ability of the system to generalize to new object locations and different vegetables.

1) Evaluation of the extraction of constraints: We validated whether the model had correctly extracted the dimensions onto which to provide either force or position control, by comparing the robot's quantitative performance in executing the task when using the proposed approach or other simple control schemes.

For evaluating the framework we compared our approach with standard control modes: a position controller and an impedance controller with fixed stiffness values. For these two control modes, 5 different demonstrations were provided ($D_i, i = 1..5$), using gravity compensation mode (*gcp*) and robot's execution was evaluated during motion replays ($R_i, i = 1..5$) in the different setups: position control (*pos*) and impedance control (*imp*). The performance under these control modes was compared to the developed approach (*amp*). Several replays were performed for each demonstrated motion. We constantly compensated for the decrease of the vegetable's height, during replays. Each group of 1 demonstration followed by 5 replays were performed on the same vegetable. A single vegetable type was used, and the task was demonstrated using 5 passes over the grating surface during each trial.

(a) Data on dimension 1
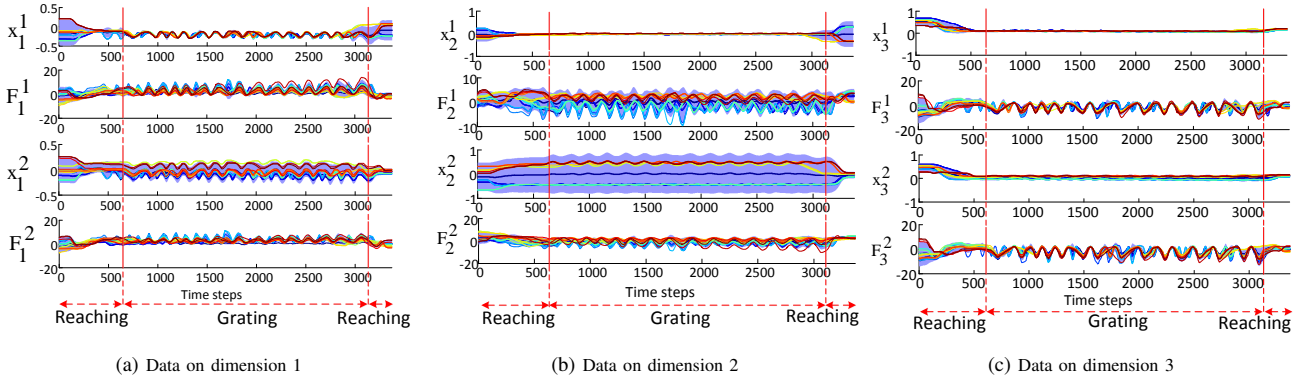
(b) Data on dimension 2

(c) Data on dimension 3

Fig. 8: Data recorded from demonstrations with the variance envelope. Data were aligned using DTW and projected in the reference frame of the two objects: the grater (RF_1) and the bowl (RF_2). Vertical lines were manually added to highlight the parts of the task.
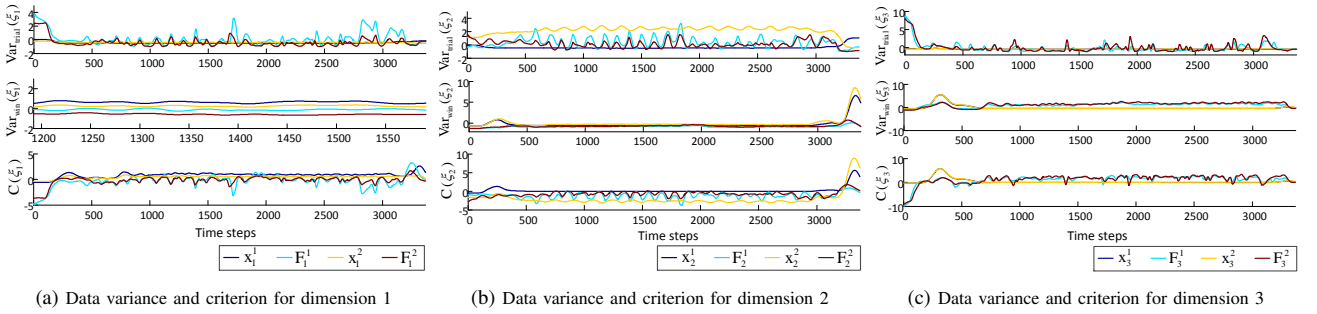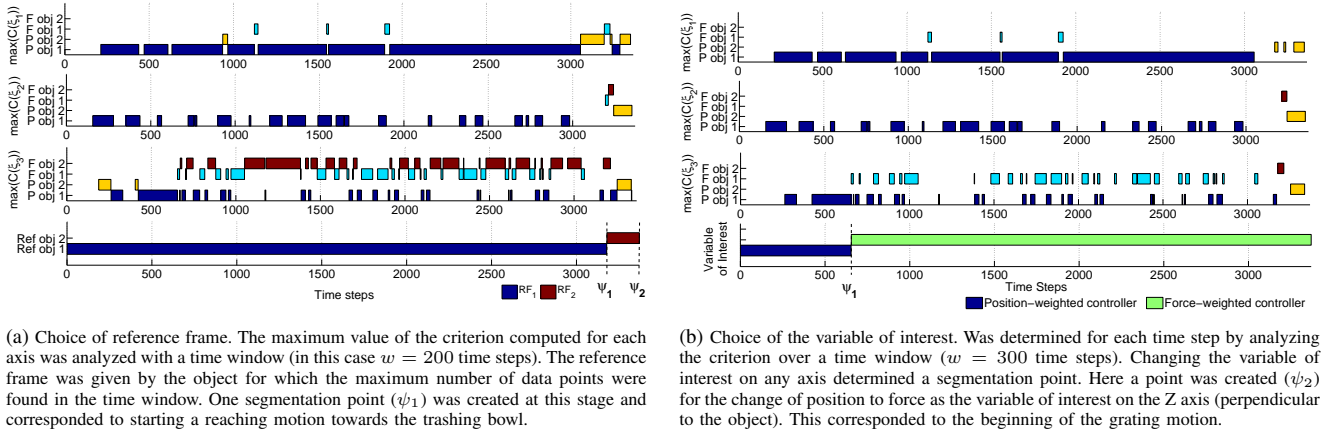

(a) Data variance and criterion for dimension 1

(b) Data variance and criterion for dimension 2

(c) Data variance and criterion for dimension 3

Fig. 9: The variance of the considered variables (Position and Force with respect to the two objects), averaged over a time window, and the corresponding criterion.

(a) Choice of reference frame. The maximum value of the criterion computed for each axis was analyzed with a time window (in this case $w = 200$ time steps). The reference frame was given by the object for which the maximum number of data points were found in the time window. One segmentation point (ψ_1) was created at this stage and corresponded to starting a reaching motion towards the trashing bowl.

(b) Choice of the variable of interest. Was determined for each time step by analyzing the criterion over a time window ($w = 300$ time steps). Changing the variable of interest on any axis determined a segmentation point. Here a point was created (ψ_2) for the change of position to force as the variable of interest on the Z axis (perpendicular to the object). This corresponded to the beginning of the grating motion.

Fig. 10: The extraction of constraints for the *Vegetable Grating Task*.
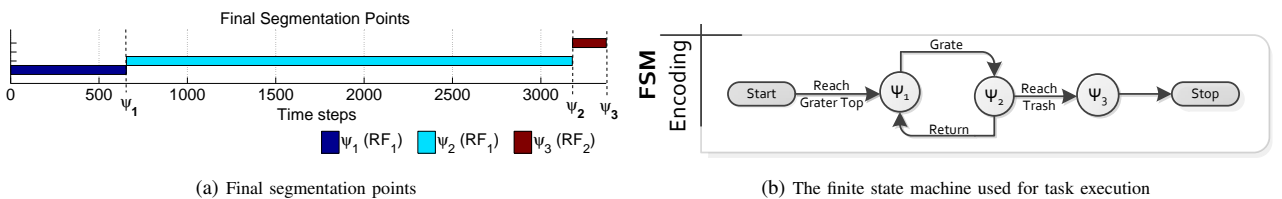
(a) Final segmentation points

(b) The finite state machine used for task execution

Fig. 11: Information used in the task encoding.

Type	Control	w_{init} [g]	Δw [g]	w_{fin} [g]	w_{ratio} [%]	h_{init} [cm]	h_{fin} [cm]	h_{ratio} [%]	SP	SP_{ratio} [%]
Trial 1										
D1	gcp	100	4	60	21.00	14.5	8.4	42.06	5	100
R1	pos		1						1	20
R2	imp		2						3	60
R3	imp		3						2	40
R4	imp		4						4	80
R5	imp		7						4	80
Trial 2										
D2	gcp	74	7	48	21.62	11.5	7.4	35.65	5	100
R1	pos		2						1	20
R2	imp		2						2	40
R3	imp		5						3	60
Trial 3										
D3	gcp	74	9	43	31.08	10.0	6.5	35.00	5	100
R1	pos		1						1	20
R2	imp		7						4	80
R3	imp		6						4	80
Trial 4										
D4	gcp	90	6	55	17.78	13.0	7.5	42.30	5	100
R1	pos		0						0	0
R2	imp		5						4	80
R3	imp		3						2	40
R4	imp		1						1	20
R5	imp		1						1	20
Trial 5										
D5	gcp	83	6	52	18.07	13.2	9.7	26.92	5	100
R1	pos		0						0	0
R2	imp		2						2	40
R3	imp		1						1	20
R4	imp		1						1	20
R5	imp		5						4	80
Trial 6										
D_N	gcp	92	7	56	35.86	13.5	7	48.15	5	100
R1	amp		4						4	80
R2	amp		5						4	80
R3	amp		8						5	100
R4	amp		9						5	100

TABLE II: Evaluation of the control modes. For Trials 1 - 5 we compared the demonstrated motion D_i provided using the robots gravity compensation mode (*gcp*), with a standard position control mode (*pos*), and with an impedance controller with fixed stiffness (*imp*). Trial 6, illustrates the performance of the proposed controller, learned from the $N = 18$ demonstrations (*amp*).
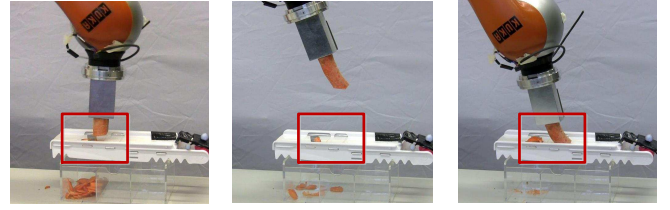

Fig. 12: Problems encountered when using standard control modes, mostly due to the size variation in the vegetable (from left to right): robot missed the grating target, incomplete slicing, high force applied, causes the vegetable to bend or break.
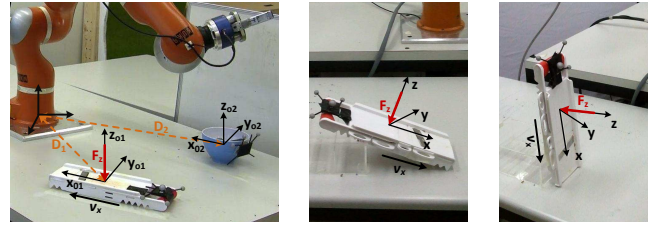

Fig. 13: The change of RF with respect to object's location.

For all the trials we measured: the original and final weight of the vegetable ($w_{init}, w_{fin}[g]$); the original and final height ($h_{init}, h_{fin}[cm]$). The original values were measured before the demonstration was performed, while the final values were measured at the end of the last replay round. For each round of demonstration and replay we measured the weight of the grated part ($\Delta w[g]$) with a precision of $\pm 1g$ and counted the number of successful passes (*SP*).

We evaluated the task performance with respect to the following computed measures:

1) $w_{ratio}[\%]$ the ratio of the grated vegetable ($w_{grated} = \sum \Delta w$) as a percentage of the initial weight. Note that the value of the $w_{init} - w_{grated}$ was often different than the final weight (w_{fin}) as the vegetable could break in the grating process. The broken part was not accounted for in the grated weight (Δw), but was reflected in a lower final weight.

2) $h_{ratio}[\%]$ the percentage of the vegetable length being grated ($h_{init} - h_{fin}$) with respect to the initial length.

3) $SP_{ratio}[\%]$ the percentage of successful passes (SP) out of the total passes performed.

Results are presented in Table II. Using a standard position controller (Trials 1 - 5) for replaying the motion gave good results in a very low number of cases: mean (M) = 12% and standard deviation (SD) = 10.95 successful passes, while the amount of vegetable grated was bellow one gram per trial (M = 0.80g, SD = 0.83). When replaying the recorded motion using an impedance controller the number of successful passes increased (M = 52.5%, SD = 25.16).

These results were compared against the proposed approach (see Table II, Trial 6), using the parametrization learned from the initial 18 demonstrations. The grating performance was assessed using the same performance metrics as for the standard control modes. The overall performance was better with respect to the amount of grated vegetable, and the number of successful passes. Common problems encountered when using standard control modes are presented in Fig. 12.

2) Evaluation of the generalization ability: We tested whether the automatic segmentation of the task and the extraction of RF was correct and led to a correct reproduction when the position of the objects was changed. The robot regenerated the complete sequence and managed to complete the overall task comprising the 3 segments even when the objects were located in arbitrary positions and orientations, none of which were seen during training.

The importance of being able to change the reference frame is illustrated in Fig. 13, when using different positions and orientations of the two objects. In this case we performed a pure qualitative assessment by placing the objects in random positions and orientations in the robot's reachable space, and using different vegetables (see Fig. 14). We measured the number of successful passes over the grater's surface. Similarly we tested the functionality over a larger grating surface.
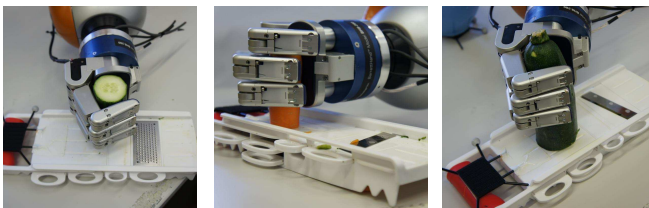
Fig. 14: Generalization to different grating surfaces and vegetables, while the position and orientation of the grater were changed.

Fig. 15: Segmentation points and controller type obtained when accounting for 4 variables: end effector position, force, torque, and velocity.

V. DISCUSSION

Our approach of extracting continuous soft constraints from human demonstration was tested on a cooking task encompassing 3 segments. The proposed method extracted the necessary information for performing the task and encoded the task flow without any prior knowledge. The task was reproduced from a time independent encoding, using an impedance controller parameterized by the continuous constraints.

From a Human-Robot Interaction (HRI) perspective, this method can facilitate teaching interactions as it allows the user to demonstrate the whole task rather than individual actions. A fragmented representation can be demanding when the user has to actively teach the robot how to perform the task. As multiple demonstrations are required for generalization, it is more convenient for the user to demonstrate the whole task, rather than individual actions, such as reaching movements.

We further discuss 3 aspects that could influence the behavior of our approach: (1) the number of variables accounted for when extracting the constraints, (2) the window size for analyzing the data and (3) the resulting criterion and the possibility to generalize a learned task to a new context.

a) Influence of other variables on segmentation points: The approach presented above can be extended by taking into account other variables. We computed the variance over trials and time window for 2 other measures: the torques sensed at the end effector, and the end effector velocity (a total of $k = 4$ variables). The analysis, using the same approach presented in Section III, shows that using the extra information provided by the velocity, or torque data does not significantly modify the segmentation points. The new segmentation points (see Fig. 15) roughly correspond to the ones determined before.

b) Choice of window size: In the current implementation the window size was chosen by the user as being the minimum time duration for observing noticeable changes in the task flow. The size of the window might influence the number and location in the data of the segmentation points obtained. We do not yet have an automatic way to determine the optimal time window, although it would be relatively straightforward to implement the heuristic described previously. This would however require to set a threshold of the minimum amount of change across all variables to determine the minimal time window and hence introduce yet another open parameter.

c) Task Generalization: We tested the developed controller for a different grating surface and a softer vegetable. This resulted in proper grating, as shown in Fig. 14. However
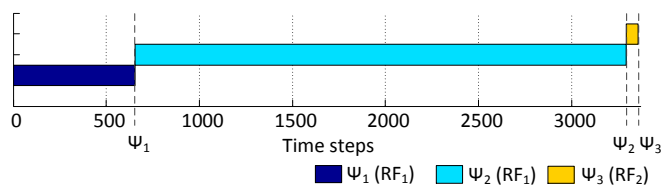
in the current implementation the choice of modeling the force as conditioned on the position was ad-hoc, prior information. Future work will extend this approach to learn and extract automatically that there is a correlation between these two variables and the directionality of the correlation.

VI. CONCLUSIONS

The presented approach for extracting task constraints takes advantage of the existing variance in the demonstrated data, and proposes a criterion for detecting regions of coherence across demonstrations. Objects upon which an action was performed are determined. The action is further encoded in the local frame of reference, in a time independent manner, preserving the task flow of actions.

In particular, we compared different measurements (like position and force) and modulated their contribution to the controller used in reproducing the motion, by using a weighting factor that adapts the robot's stiffness. Also by weighting the relative importance of each of the task variables when expressed in the reference system of the objects involved in the task we can determine the suitable reference frame to be used in each segment. Finally a set of segmentation points were obtained by splitting the motion whenever a change in the reference frame or in the variables on interest occured. The approach was validated on a kitchen task (grating vegetables), achieving good generalization results.

The advantages of using this segmentation and feature extraction method are firstly decreasing the task complexity by focusing on learning just the variables that are important for each region of the task (i.e. encode just end effector position for a reaching motion vs. accounting for position and force in manipulation sub-tasks) and secondly achieving efficient generalization when the position of the objects is changed.

REFERENCES

[1] A. Shukla and A. Billard, "Coupled dynamical system based arm-hand grasping model for learning fast adaptation strategies," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 424 – 440, 2012.

[2] L. Villani and J. De Schutter, "Force control," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 161–185.

[3] T. Kunz and M. Stilman, "Manipulation planning with soft task constraints," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 1937–1942.

[4] M. Ruchanurucks, S. Nakaoka, S. Kudoh, and K. Ikeuchi, "Humanoid robot motion generation with sequential physical constraints," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 2649–2654.

[5] D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation planning on constraint manifolds," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 625–632.

[6] P. Marayong, M. Li, A. Okamura, and G. Hager, "Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, 2003, pp. 1954–1959 vol.2.

[7] D. Baldwin, A. Andersson, J. Saffran, and M. Meyer, "Segmenting dynamic human action via statistical structure," *Cognition*, vol. 106, no. 3, pp. 1382 – 1407, 2008.

[8] N. B. Turk-Browne, B. J. Scholl, M. M. Chun, and M. K. Johnson, "Neural evidence of statistical learning: Efficient detection of visual regularities without awareness." *J. Cognitive Neuroscience*, vol. 21, no. 10, pp. 1934–1945, 2009.

[9] J. Zhao, N. Al-Aidroos, and N. Turk-Browne, "Attention is spontaneously biased toward regularities," *Psychological Science*, vol. 24, no. 5, pp. 667–677, 2013.

[10] R. Aslin and E. Newport, "Statistical learning: From acquiring specific items to forming general rules," *Current Directions in Psychological Science*, vol. 21, no. 3, pp. 170–176, 2012.

[11] J. R. Saffran, E. K. Johnson, R. N. Aslin, and E. L. Newport, "Statistical learning of tone sequences by human infants and adults," *Cognition*, vol. 70, no. 1, pp. 27 – 52, 1999.

[12] D. A. Baldwin and J. A. Baird, "Discerning intentions in dynamic human action," *Trends in Cognitive Sciences*, vol. 5, no. 4, pp. 171 – 178, 2001.

[13] N. Franklin, B. Tversky, and V. Coon, "Switching points of view in spatial mental models," *Memory & Cognition*, vol. 20, pp. 507–518, 1992.

[14] B. Claus, K. Eyferth, C. Gips, R. Hörnig, U. Schmid, S. Wiebrock, and F. Wysotzki, "Reference frames for spatial inference in text understanding," in *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*. Springer-Verlag, 1998, pp. 241–266.

[15] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: instructive demonstrations, generalization and practice," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS '03. ACM, 2003, pp. 241–248.

[16] J. Young, R. Hawkins, E. Sharlin, and T. Igarashi, "Toward acceptable domestic robots: Applying insights from social psychology," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 95–108, 2009.

[17] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, vol. 37, no. 2, pp. 286–298, 2007.

[18] ——, "On Learning the Statistical Representation of a Task and Generalizing it to Various Contexts," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2978–2983.

[19] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *Robotics, IEEE Transactions on*, vol. 26, no. 5, pp. 800–815, 2010.

[20] M. J. Gielniak, C. K. Liu, and A. L. Thomaz, "Task-aware variations in robot motion," in *ICRA*, 2011, pp. 3921–3927.

[21] J. De Schutter, E. Di Lello, J. De Schutter, R. Matthysen, T. Benoit, and T. De Laet, "Recognition of 6 dof rigid body motion trajectories using a coordinate-free representation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2071–2078.

[22] M. Muhlig, M. Gienger, J. Steil, and C. Goerick, "Automatic selection of task spaces for imitation learning," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 4996 –5002.

[23] V. Sukhoy, V. Georgiev, T. Wegter, R. Sweidan, and A. Stoytchev, "Learning to slide a magnetic card through a card reader," in *ICRA*, 2012, pp. 2398–2404.

[24] S. Calinon, I. Sardellitti, and D. Caldwell, "Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 249–254.

[25] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[26] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[27] J. Lin and D. Kulic, "Automatic human motion segmentation and identification using feature guided hmm for physical rehabilitation exercises," *Robotics for Neurology and Rehabilitation, Workshop at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[28] J. Shim and A. L. Thomaz, "Human-like action segmentation for option learning," in *RO-MAN*, 2011, pp. 455–460.

[29] O. Mangin and P.-Y. Oudeyer, "Learning to recognize parallel combinations of human motion primitives with linguistic descriptions using non-negative matrix factorization." in *IROS*. IEEE, 2012, pp. 3268–3275.

[30] L. Tao, E. Elhamifar, S. Khudanpur, G. Hager, and R. Vidal, "Sparse hidden markov models for surgical gesture classification and skill evaluation," in *Information Processing in Computer-Assisted Interventions*, ser. Lecture Notes in Computer Science, P. Abolmaesumi, L. Joskowicz, N. Navab, and P. Jannin, Eds. Springer Berlin / Heidelberg, 2012, vol. 7330, pp. 167–177.

[31] D. Kulic, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *I. J. Robotic Res.*, vol. 31, no. 3, pp. 330–345, 2012.

[32] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," in *Humanoids*, 2006, pp. 425–431.

[33] D. Kulic, W. Takano, and Y. Nakamura, "Combining automated on-line segmentation and incremental clustering for whole body motions," in *ICRA*, 2008, pp. 2591–2598.

[34] T. Matsuo, Y. Shirai, and N. Shimada, "Automatic generation of hmm topology for sign language recognition," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1 –4.

[35] S. H. Lee, H. K. Kim, and I. H. Suh, "Incremental learning of primitive skills from demonstration of a task," in *Proceedings of the 6th international conference on Human-robot interaction*, ser. HRI '11. ACM, 2011, pp. 185–186.

[36] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *Internationl Journal of Robotics Research*, vol. 24, pp. 31–48, 2005.

[37] D. Kulic and Y. Nakamura, "Scaffolding on-line segmentation of full body human motion patterns," in *IROS*, 2008, pp. 2860–2866.

[38] A. Arsenio, "Learning task sequences from scratch: applications to the control of tools and toys by a humanoid robot," in *Control Applications, 2004. Proceedings of the 2004 IEEE International Conference on*, vol. 1, 2004, pp. 400–405 Vol.1.

[39] R. Jäkel, S. R. Schmidt-Rohr, M. Lösch, and R. Dillmann, "Representation and constrained planning of manipulation strategies in the context of programming by demonstration," in *ICRA*, 2010, pp. 162–169.

[40] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *Int. J. Rob. Res.*, vol. 31, no. 3, pp. 360–375, 2012.

[41] R. Jkel, S. Rhl, S. Schmidt-Rohr, Z. Xue, and R. Dillmann, "Layered programming by demonstration and planning for autonomous robot manipulation," in *Advanced Bimanual Manipulation*, 2012, vol. 80, pp. 1–57.

[42] G. Oriolo and M. Vendittelli, "A control-based approach to task-constrained motion planning," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 297–302.

[43] G. Ye and R. Alterovitz, "Demonstration-guided motion planning," *International Symposium on Robotics Research (ISRR)*, 2011.

[44] M. Beetz, L. Mösenlechner, and M. Tenorth, "CRAM – A Cognitive Robot Abstract Machine for Everyday Manipulation in Human Environments," in *IEEE/RSJ International Conference on Intelligent RObots and Systems.*, 2010.

[45] N. Dantam and M. Stilman, "The motion grammar: Analysis of a linguistic method for robot control," *Robotics, IEEE Transactions on*, vol. 29, no. 3, pp. 704–718, 2013.

[46] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *I. J. Robotic Res.*, vol. 26, no. 5, pp. 433–455, 2007.

[47] M. Stilman, "Task constrained motion planning in robot joint space," in *in IROS*, 2007.

Encoding Bi-manual Coordination Patterns From Human Demonstrations

Ana Lucia Pais
Learning Systems and Algorithms Laboratory
École Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland
{lucia.pais}@epfl.ch

Aude Billard
Learning Systems and Algorithms Laboratory
École Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland
{aude.billard}@epfl.ch

ABSTRACT

Humans perform tasks such as bowl mixing bi–manually, but programming them on a robot can be challenging specially in tasks that require force control or on-line stiffness modulation. In this paper we first propose a user-friendly setup for demonstrating bi–manual tasks, while collecting complementary information on motion and forces sensed on a robotic arm, as well as the human hand configuration and grasp information. Secondly for learning the task we propose a method for extracting task constraints for each arm and coordination patterns between the arms. We use a statistical encoding of the data based on the extracted constraints and reproduce the task using a cartesian impedance controller.

Categories and Subject Descriptors

I.2.9 [**Robotics**]; I.2.6 [**Learning**]: Knowledge acquisition; H.5.2 [**User Interfaces**]: Interaction styles

General Terms

Robot Learning

Keywords

Programming by demonstration; Task constraints extraction

1. INTRODUCTION

Daily activities, such as dish washing or preparing a meal, require completing tasks that are implicitly bi-manual. A challenge in programming such tasks is accounting for all the task variables, for the motion of each arm, as well as for their coordinated behavior. Here we take a Programming by Demonstration (PbD) approach in which a human can directly demonstrate the task, and propose a method for determining and encoding bi-manual coordination patterns.

We exemplify this on a task (stirring in a bowl, as shown in Fig. 1) that requires completing a sequence of actions
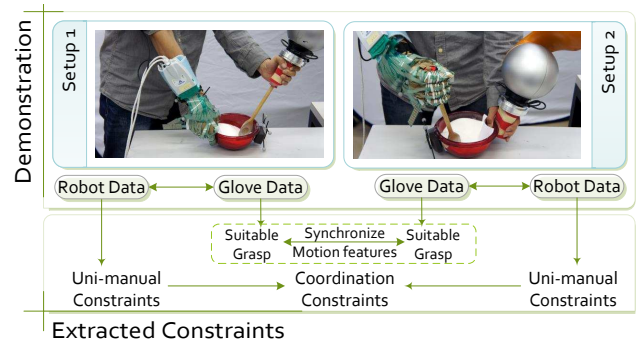
Figure 1: The two setups used in the demonstration phase. We alternate between the active/passive arms in the tasks, and record complementary information from the glove and robot arm.

for each arm. According to a taxonomy of bi-manual actions proposed in [6], the task subparts can be described as: (1) a discrete reaching motion from the initial position of each arm to the proper configuration to start mixing; (2) an asymmetrical coordinated motion, in which one arm is actively stirring while the other is passively assisting; (c) an uncoordinated reaching back action. The stirring action requires coordination not only in arm movement, but also with respect to the force and stiffness applied by each arm.

To be able to record the interaction forces perceived on each hand in coordination and in conjunction with measurements of the arm and finger displacement, we developed an experimental setup displayed in Fig. 1 (see Section 3 for a description). We analyze the demonstration data and extract (1) continuous constraints for each arm, consisting of the variables of interest in each part of the task, expressed in the local frame of reference of the object on which we perform manipulation and a stiffness modulating factor; (2) coordination patterns between the variables of interest in each part of the motion. We represent the motion using a time independent statistical encoding which allows using the extracted features as continuous task constraints that can be embedded online in the robot's motion. For reproducing the task on a bi–manual robotic platform we use a cartesian impedance controller for each arm, parameterized with the extracted constraints.

2. RELATED WORK

In our previous work [1], we proposed a method for encoding arm–motion in discrete bi–manual tasks based on determining key postures during the demonstration. However

in the present work we focus on tasks that require coordinated force control. We determine continuous constraints that apply throughout the task or in parts of the task.

For each arm we extract task constraints using the method prosed in [7], based on analyzing the variance in the data. We extend this approach to determine arm dominance (i.e the relative importance of each arm). We further encode the whole task as a sequence of states describing each action. Alternative representations are graph-based [2], or Markov-model based [5].

3. METHOD

To execute the task on a robotic platform we consider a cartesian impedance controller for each arm, given by $\tau = J^T \cdot RF \cdot (\lambda K(x - x_d) + F)$. The desired position $x_d \in \mathbb{R}^3$, the force to be applied $F \in \mathbb{R}^3$, and a factor $\lambda \in \mathbb{R}^3$ that modulates the arm's stiffness K, are extracted from demonstration (as explained below) and are expressed in the local reference frame of the object of interest RF.

To demonstrate the task we designed a setup as shown in Fig. 1, in which the user can perform the task by kinesthetically guiding the robotic arm with one hand and by wearing a data glove on the other hand. This particular configuration has two main advantages: (1) it makes it easy for the user to provide demonstrations (i.e. rather than handling multiple degrees of freedom from two robot arms); (2) it allows simultaneously recording complementary information: end effector cartesian positions ($x_R \in \mathbb{R}^3$) and forces ($F_R \in \mathbb{R}^3$), from a KUKA LWR arm; hand configuration ($\theta_G \in \mathbb{R}^{23}$ joint angles), and wrist position ($x_G \in \mathbb{R}^3$) from the data glove. Additionally we recorded the object's cartesian position ($x_o \in \mathbb{R}^3$) using an Optitrack vision system.

The user performed the task in two phases, by alternating the roles of the active and passive hands. This allowed us to record both robot data and glove data for both the active and the passive arm. We recorded N = 6 demonstrations in each phase. We aligned the recorded data using Dynamic Time Warping (DTW). The final data set for phase 1 is $\xi^1 = \{x_R^A, F_R^A, x_G^P, \theta_G^P, x_o\}$, where the upper indices refer to the hand performing an active (A) or passive (P) task. Similarly a data set ξ^2 is obtained in the second phase.

Uni-manual constraints.

To extract the constraints of each arm, we consider for each phase $i = 1..2$, a subset $\xi_R^i = \{x_R, F_R\}$ of ξ^i. The glove wrist position x_G is used for aligning the robot motion in the two phases. We analyze the robot data in the reference frame of the object (i.e. the bowl), as described in [7]. For each recorded variable (position and force), across each dimension, we compute a criterion based on the observed variance in the data [7]. This allows us to compare in a relative manner variables of different types. We consider at each time step the variable of interest to be the one with the maximum computed criterion. When this changes a segmentation point is created, resulting in a set of states ψ_s. For the current task representation see Fig. 2. The arm motion in each segment is encoded as a non-linear dynamical system [4]. The force components are encoded in a Gaussian Mixture Model (GMM) as a function of position. We compute a stiffness modulation factor λ as the difference between the criterion computed for position and the one computed for force on each axis. Additionally for each state we determine a corresponding hand configuration $\theta_{G,s}$.
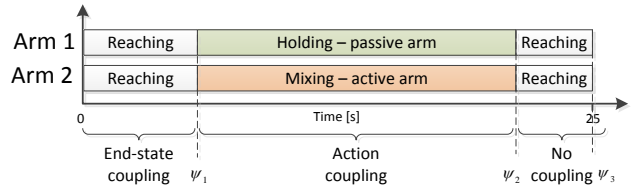

Figure 2: The identified motion segments for each arm, and corresponding coupling.

Bi-manual coordination.

Comparing the obtained criteria between the two arms allows us to determine at each time step which arm is dominating in either position or force applied in the task. This is similar to results on human subjects showing that the arms can change the active and passive roles during manipulation and this is caused by a force-motion relation, rather than prior knowledge or routine in executing the task [3].

Hand dominance thus influences the way we model the task subparts. For the active arm we encode the motion and force profile as described above. However for the passive arm the motion is insignificant, while the forces sensed on the arm are reaction forces responsible for keeping the object in place. Therefore we choose to encode using a GMM model its force $p(F_R^P, F_R^A)$, and stiffness profiles $p(K_R^P, F_R^A)$ as dependent on the forces sensed on the active arm. This allows the passive arm to apply compensating forces to the ones applied by the active arm.

4. CONCLUSION AND FUTURE WORK

We presented a procedure for recording bi–manual demonstrations that reduces user's effort and maximizes the obtained information. We analyze the data to extract constraints for each arm and encode coordination patterns.

Future work involves determining a two levels encoding of the task: (1) *skill level* as general knowledge about the action, and (2) *task level*, as a parametrization of the learned skill. This enables policy reusability for similar tasks, such as stirring in a bowl of dough, and applying the same skill for stirring coffee.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] E. Gribovskaya and A. Billard. Combining dynamical systems control and programming by demonstration for teaching discrete bimanual coordination tasks to a humanoid robot. In *HRI*, 2008.

[2] R. Jäkel, S. R. Schmidt-Rohr, M. Lösch, and R. Dillmann. Representation and constrained planning of manipulation strategies in the context of programming by demonstration. In *ICRA*, 2010.

[3] R. S. Johansson, A. Theorin, G. Westling, M. Andersson, Y. Ohki, and L. Nyberg. How a lateralized brain supports symmetrical bimanual tasks. *PLoS Biol*, 4(6):e158, 2006.

[4] S. M. Khansari-Zadeh and A. Billard. Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models. *IEEE Transaction on Robotics*, 2011.

[5] S. H. Lee, H. K. Kim, and I. H. Suh. Incremental learning of primitive skills from demonstration of a task. In *HRI*, 2011.

[6] C. L. MacKenzie and R. G. Marteniuk. Bimanual coordination. volume 23 of *Advances in Psychology*, pages 345 – 358. 1985.

[7] A. L. Pais, K. Umezawa, Y. Nakamura, and A. Billard. Task parametrization using continuous constraints extracted from human demonstrations. *Submitted*, 2013.

Learning Object-level Impedance Control for Robust Grasping and Dexterous Manipulation

Miao Li[1], Hang Yin[1], Kenji Tahara[2,1] and Aude Billard[1]

Abstract— Object-level impedance control is of great importance for object-centric tasks, such as robust grasping and dexterous manipulation. Despite the recent progresses on this topic, how to specify the desired object impedance for a given task remains an open issue. In this paper, we decompose the object's impedance into two complementary components– the impedance for stable grasping and impedance for object manipulation. Then, we present a method to learn the desired object's manipulation impedance (stiffness) using data obtained from human demonstration. The approach is validated in two tasks, for robust grasping of a wine glass and for inserting a bulb, using the 16 degrees of freedom Allegro Hand mounted with the SynTouch tactile sensors.

I. INTRODUCTION

Robust grasping and dexterous manipulation are two of the most important capabilities that a robot is expected to have. The main characteristic of a robust grasp is its ability to comply with external perturbations applied to the grasped object while still maintaining the grasp. In dexterous manipulation, the robotic hand, mainly the fingertips, have to physically interact with the object in order to move it to a desired configuration. In both scenarios, appropriate grasping forces need to be applied on the grasped or manipulated object, either to keep the grasp stable under perturbation or to move the object to a desired configuration.

To this end, various control algorithms have been proposed and ported to control multi-fingered robotic hand. These can be roughly divided into two groups. The first group encompass hybrid position/force control approaches that modulates the force explicitly to manage the interaction imposed by the environment [1], [2], [3]. Another group uses impedance control to regulate the interaction force implicitly by specifying the impedance of the grasped object [4], [5], [6], [7]. In general, the hybrid position/force control is more precise when controlling simultaneously the force and position. The main deficiency of hybrid control is the transition between position and force control when the contact state varies between non-contact and contact. A small delay in this transition may lead to a very large overshot contact force. In addition, the selection of accurate grasping forces for hybrid control that fulfil the friction constraints and task requirements is still a difficult planning problem [3]. In impedance controller, the object motion is realized by a desired object impedance that

[1]M. Li, H. Yin and A. Billard are with LASA, École Polytechnique Fédérale de Lausanne (EPFL), Switzeland {`miao.li,` `hang.yin,aude.billard`}`@epfl.ch`

[2]K. Tahara is with Faculty of Engineering, Kyushu University, 744 Moto'oka, Nishi-ku, Fukuoka 819-0395, Japan. He is currently a visiting scholar at LASA. `tahara@ieee.org,` `kenji.tahara@epfl.ch`
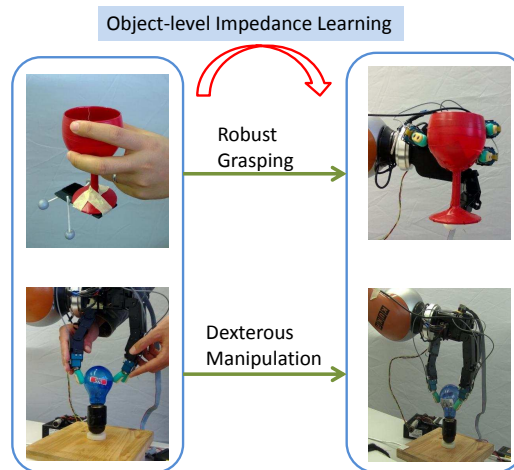

Fig. 1: The obejct-level impedance for robust grasping and dexterous manipulation are learned from human demonstration.

generates force to move the object to a desired configuration. It has the advantage that it will converge to the desired position in free motion and a stable equilibrium position in the case of interaction with the environment. This merit can be greatly beneficial for both robust grasping and dexterous manipulation. Therefore, we restrict the rest of this review to impedance controllers only. In [8], a fingertip Cartesian stiffness controller was introduced using fingertip force sensor. However, the stiffness controller can not actively control the whole system dynamics. To overcome this defect, Liu and Hirzinger [9] proposed a Cartesian impedance controller for the DLR hand based on the joint torque measurements. While these two controllers are in the fingertip Cartesian space, object-level impedance controllers are proposed by directly specifying the desired impedance in the object frame, which are usually more suitable for robust grasping and dexterous manipulation of an object. In [4], an object level impedance controller has been proposed for a multi-arm manipulator to directly control the internal object forces and compensate the system dynamics. The object is assumed to be rigidly grasped that can transmit bilateral contact forces between the fingertips and the object. Wimbock et al. [5], [7] recently presented their experimental evaluation of an intrinsically passive controller for multi-fingered hand, where the damping parameters are designed and implemented as a function of the object effective inertia and stiffness matrix. A similar impedance controller was also proposed in [6] and [10] by defining a virtual frame which depends only on the fingertip positions. The damping parameters are designed in

both the finger joint space and the object frame.

However, despite all the above-mentioned progress, one critical issue still remains unaddressed: how to specify the proper impedance for a given task? The specification of impedance is known as a difficult problem as it depends on the task at hand as well as the kinematic and dynamic limitation of the robot [11], [12]. Moreover, the impedance parameters may need to adapt to the task requirements or to variation in the environment, such as the bulb replacement task that the torsional resistance increases greatly during the last phase of the task. To this end, sensor feedback should be taken into account to monitor the status of task's completion and to vary the impedance accordingly.

In this paper, we attempt to address this problem by learning the impedance from human demonstration. In the following Section II, some related works regarding impedance specification are summarized. In Section III, an object-level impedance controller is reformulated. In Section IV, methods for learning impedance from human demonstration for robust grasping and dexterous manipulation are presented. Experiments on a multi-fingered robotic hand are demonstrated and discussed in Section V. Finally, we give a conclusion and an outlook on future work in Section IV.

II. RELATED WORK

a) Analytical Impedance Specification: In one of their early works, Mason and Salisbury [8] used the congruence transformation to obtain the desired object stiffness from joint stiffness. Based on this work, Cutkosky and Kao [13] expressed the compliance of a grasp as a function of grasp geometry, contact conditions and mechanical properties of the fingers. In order to choose the grasp compliance for a given task, Shimoga and Goldenberg [14] formulated a concept termed Grasp Admittance Center, which is the origin of a frame that impedance matrices will be diagonal. Also, A qualitative method has been developed to choose the relative magnitude of the impedance parameters for a set of tasks. In [15], [16], Kim et al. analysed the compliance characteristics for different tasks by considering the grasp geometry, which is the relation between the operational space and the fingertip space of multi-fingered hand. Their analytical results show that the non-diagonal terms in impedance matrices can not be specified arbitrarily and they also used a qualitative method (small and large value of stiffness) to specify the impedance parameter.

b) Impedance Learning: Learning of tasks is another approach by which desired impedance parameters can be specified. In [17], the impedance learning problem is formulated as a model-based reinforcement learning problem, where the impedance parameters can gradually change to improve the task performance. In [18], the authors accomplished a variable impedance controller with a model-free, sample-based reinforcement learning method. However, the reinforcement function needs to be carefully defined to capture the essence of the task, which will be difficult for complex tasks, such as dexterous manipulation. Sikka and McCarragher [19] presented a method that can learn
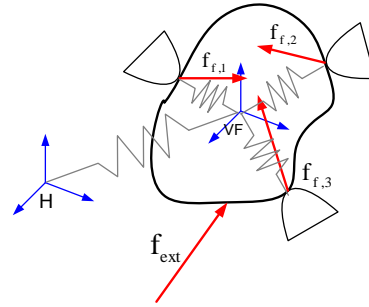

Fig. 2: An object grasped by 3 fingers. The object impedance and grasp impedance are shown as springs. The $\mathbf{f}_{f,i}, i = 1, 2, 3$ are the contact forces on each fingertips. $\mathbf{f}_{ext}$ is the external perturbation force. The frame **H** and **VF** are the inertial frame and the virtual frame, respectively.

the robot end-point stiffness of contact tasks from human demonstration. An online, incremental algorithm has been proposed in [20] to learn varying end-point stiffness from human demonstration. For a multi-fingered robotic hand, a implicit compliant controller [21] is learned to adapt the grasp under perturbation, which actually mapping the fingertips tactile response to finger joints. However, this method is hand dependent and difficult to generalize to manipulation tasks.

As discussed in [22], the tasks for multi-fingered hand are usually object-centric. In these cases, learning an object-level impedance is more suitable and can be easily applied to other hands. In this paper, we extend the object-level impedance controller in [10] with tactile feedback and object-level impedance learned from human demonstration.

III. OBJECT-LEVEL IMPEDANCE CONTROL

In this section, we will reformulate the object-level impedance controller proposed in [10], which is mainly composed of two parts, a stable grasp controller and an object manipulation controller.

A. Object Manipulation Impedance

Following the formulation of impedance control in [11], the dynamics of the object, as shown in Fig. 2, is governed by the equation:

$$\mathbf{f}_{f,o} + \mathbf{f}_{ext} = M_0 \ddot{\mathbf{x}} \qquad (1)$$

where $\mathbf{f}_{f,o}$ is the summation of manipulating forces $\mathbf{f}_{f,oi}$ exerted on the object from each fingertip, $\mathbf{f}_{ext}$ is the external perturbation force. All the forces are expressed in the inertial frame. M_0 is the actual inertia matrix and $\mathbf{x}$ is the position and orientation of the object. Usually, the position and orientation are controlled independently [10]. Here for simplicity, we put position and orientation in one vector $\mathbf{x}$ to introduce the controller.

The objective of impedance control is to modulate the interaction between the object and the environment by controlling the contact forces. The desired interaction of the system is given by:

$$\mathbf{f}_{ext} = M\ddot{\mathbf{x}} + D(\dot{\mathbf{x}} - \dot{\mathbf{x}}_r) + K(\mathbf{x} - \mathbf{x}_r) \qquad (2)$$

where $\mathbf{x}_r, \dot{\mathbf{x}}_r$ is the reference trajectory and M, D, K are the desired apparent inertia, damping and stiffness, respectively. From equation (1) and (2), we can derive the object-level impedance control law given as:

$$\mathbf{f}_{f,o} = ED(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}) + EK(\mathbf{x}_r - \mathbf{x}) + (E - I)\mathbf{f}_{ext} \quad (3)$$

where $E = M_0 M^{-1}$ and I is the identity matrix. In practice, it is often sufficient to keep the inertia unchanged, i.e., $M_0 = M$ and only shape the stiffness and damping. Then the equation (3) can be simplified as:

$$\mathbf{f}_{f,o} = D(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}) + K(\mathbf{x}_r - \mathbf{x}) \quad (4)$$

B. Stable Grasping Impedance

Up to now, only the object manipulation impedance has been considered. In order to make the grasp stable during manipulation, we need to design a stable grasping impedance, which can be used to change the grasping forces. In our paper, the contact model between the object and the fingertips is assumed to be point contact with friction, which can only transmit contact forces. Therefore, we only use one translational spring connecting each fingertip and the origin of the object (virtual) frame to represent the stable grasping impedance (stiffness), as shown in Fig. 2. The grasping forces can be expressed as:

$$\mathbf{f}_{f,gi} = K_{gi}(\|\Delta \mathbf{p}_i\| - L_i)\frac{\Delta \mathbf{p}_i}{\|\Delta \mathbf{p}_i\|} \quad (5)$$

where $\mathbf{f}_{f,gi}$ and K_{gi} are the grasping force and stable grasping stiffness at i-th fingertip. $\Delta \mathbf{p}_i = \mathbf{p}_o - \mathbf{p}_i$ with $\mathbf{p}_i$ as the position of contact point on i-th fingertip and $\mathbf{p}_o$ as the position of the object frame origin. L_i is the desired distance from the i-th fingertip to the object frame origin.

C. Implementation Issues

A rigorous implementation of the controller will require a lot of computational load [23]. To reduce it, the finger dynamics is not compensated and thus joint torques at each finger can be obtained from a simple Jacobian transpose.

$$\boldsymbol{\tau}_{f,i} = J_{f,i}^T \mathbf{f}_{f,i} \quad (6)$$

where $\boldsymbol{\tau}_{f,i}$ are the joint torques at i-th finger and $J_{f,i}$ is the Jacobian of the i-th finger. The contact force can be computed as: $\mathbf{f}_{f,i} = \mathbf{f}_{o,i} + \mathbf{f}_{g,i}$. The more rigorous way to compute the contact force using grasp mapping can be also use here [7], which is also more computational expensive.

In order to implement this controller, we need to address the following issues: (1) measure the object position and orientation $\mathbf{x}$; (2) design the reference trajectory $\mathbf{x}_r, \dot{\mathbf{x}}_r$; (3) choose the impedance parameters K and D. While (1) will be discussed in the remaining part of this section by introducing a *Virtual Frame*, the method to deal with (2) and (3) by learning from human demonstration will be presented in the next section.

D. Virtual Object Frame

Due to the occlusion of the hand, it is still very difficult to rely on vision to obtain the actual object position and orientation in the controller. To deal with this, the concept of Virtual Frame (VF) is adopted here, which is a function of all the contact points between object and fingertips. Virtual frame (VF) can be used to estimate the real object position and orientation if we assume that relative contact points between object and fingertips do not change [1]. Different from the definition in [10], the VF in this work is the function of real contact point on each fingertip, which can be obtained from tactile feedback. In our work, we only use three fingers, the origin of VF is:

$$\mathbf{p}_o = \frac{1}{3}\sum_{i=1}^{3} \mathbf{p}_i \quad (7)$$

The orientation of the frame is defined in the following way:

$$R_o = [\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z] \in SO(3) \quad (8)$$
$$\mathbf{r}_x = \frac{\mathbf{p}_3 - \mathbf{p}_1}{\|\mathbf{p}_3 - \mathbf{p}_1\|}$$
$$\mathbf{r}_z = \frac{(\mathbf{p}_3 - \mathbf{p}_1) \times \mathbf{r}_x}{\|(\mathbf{p}_3 - \mathbf{p}_1) \times \mathbf{r}_x\|}$$
$$\mathbf{r}_y = \mathbf{r}_z \times \mathbf{r}_x$$

With the defined VF, one can compute the translation and rotation difference between the VF and the desired or reference frame. Thus, from equation (4), (5) and (6), the desired joint torque for each finger can be calculated. For more details about the implementation, one can refer to [10].

IV. IMPEDANCE LEARNING FROM HUMAN DEMONSTRATION

In this section, methods about how to specify impedance for robust grasping and dexterous manipulation will be presented.

A. Relative Impedance for Robust Grasping

A robust grasp should be able to comply with external perturbation from any directions. But in different directions, the extent of compliance will depend on the grasp configuration as well as the task requirement. For instance, grasping a screwdriver as a tool and grasping a pen to write will require totally different levels of rotational compliance along the axial direction.

Our method of impedance selection for robust grasping is quite intuitive: the object stiffness in one direction is inversely proportional to the variance of displacement under perturbation in the corresponding direction. From this assumption, we can learn the relative stiffness for robust grasping in different directions from human demonstration. This idea has also been utilized to learn the end-point stiffness for a single manipulator [20]. During the demonstration, an object is grasped by a human demonstrator with eyes closed. To mimic the fact that our controller will use solely

---

[1]This assumption will neglect the rolling and slipping effects.

proprioceptive and tactile information, with no vision. The grasped object is perturbed by another person randomly and the displacement of the object is recorded $\{\mathbf{x}^i, i = 1...N\}$. Then the object stiffness can be specified as follows:

$$K = \alpha\{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^i - \mathbf{x}_r)(\mathbf{x}^i - \mathbf{x}_r)^T\}^{-1} \qquad (9)$$

where $\alpha \in \mathbb{R}^+$ is a ratio parameter that needs to be set manually and $\mathbf{x}_r \in \mathbb{R}^6$ is the object initial (and desired) position and orientation.

Besides stiffness specification, object workspace modelling is also very important for robust grasping as it determines the extent of motion of a grasped object during perturbation. However, the workspace of a grasped object will depend on the hand kinematics and the grasp configuration. To this end, we teach the robot the extend to which it can stretch its fingers through kinaesthetic demonstration, by back-driving the fingers, see Fig. 3. All positions and orientations adopted by the hand during the demonstration are used to build a probabilistic model of the workspace of the hand. The use of a probabilistic model is advantageous as it accounts for the imprecision of the recording and allows to generalize outside the demonstrations. The latter is particularly important since demonstrations may not be exhaustive and may not explore all possible postures. Here, we use Gaussian Mixture Model (GMM). A GMM is a probabilistic model of density function composed of K Gaussian components. The likelihood of each position/orientation under this model is given by:

$$p(\mathbf{x}) = \sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad (10)$$

where π_k is the prior of the kth Gaussian component and $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. A new VF, computed using Eq. (7) (8), is said to lie in the object's workspace if its likelihood to belong to the model is greater than a fixed threshold, i.e. $p(x_*) > L_{thresh}$. This threshold, in our experiment, is quite conservative and is set as no more than 2 standard deviation, which means that about 95.45% training position/orientation of VF will be covered by the learned GMM. For more details about the parameters selection for training GMM, one can refer to [24].

With the object workspace model, one can design a desired reaction behavior to improve grasping stability, eg., increase object stiffness gradually when the object is approaching boundary of learned working space. This could be achieved in our model by increasing the ratio parameter α in equation (9). This would however increase stiffness by the same amount in all directions. It may often be useful to be able to shape this increase along particular directions, such as the direction the moves the object farthest away from the workspace's boundary.

B. Variable Impedance for Dexterous Manipulation

In the case of robust grasping, the reference frame can be easily set as the initial position and orientation, which

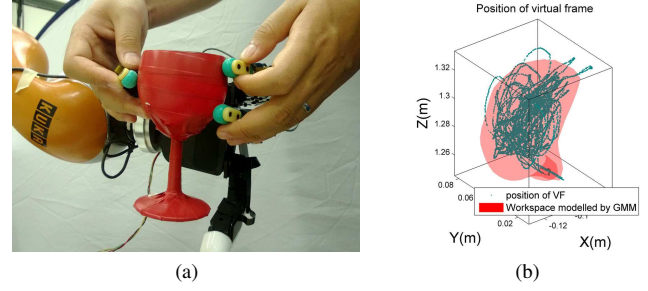(a)   (b)

Fig. 3: (a) Human teaching of object workspace. The object impedance are set to zero in all directions during the demonstration, which means human can move the object freely in its workspace. (b) The position and orientation of VF are recorded and trained using GMM. Here is shown the trained result in the subspace of VF position. The red surface is the iso-surface with the same threshold likelihood L_{thresh}

does not vary with time. For dexterous manipulation, a time-varying reference trajectory $\mathbf{x}_r, \dot{\mathbf{x}}_r$ will be required. In this section, we will present a method that learns the reference trajectory and the desired object impedance simultaneously.

The objective of human demonstration is to model the interaction between the object being manipulated and the environment. Thus, during the demonstration, at each sample instant $i, i = 1...N_s$, the motion of the object $\{\mathbf{x}(i), \dot{\mathbf{x}}(i)\}$ and the sum of manipulating forces $\mathbf{f}_{f,o}(i)$ applied on the object are recorded[2]. Consider $t = 1...N_t$ consecutive samples of data obtained over a short time window. Assuming the impedance parameters and reference trajectory remain constant over this time window, the relationship between the object motion and the force exerted on object is given by:

$$\mathbf{f}_{f,o}(i) = D(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)) + K(\mathbf{x}_r - \mathbf{x}(i)), i = 1...N_t; \quad (11)$$

During each time window, since we assume that the object's impedance parameters and the reference trajectory are not changing with time, they can be obtained by minimizing the following objective function:

$$\min_{D,K,\dot{\mathbf{x}}_r,\mathbf{x}_r}: \sum_{i=1}^{N_t}\|\mathbf{f}_{f,o}(i) - \{D(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)) + K(\mathbf{x}_r - \mathbf{x}(i))\}\|^2$$
$$(12)$$

In practice, the term from damping is usually ignored by assuming that the desired velocity trajectory is the same as the measured one and thus equation (12) can be simplified as:

$$\min_{K,\mathbf{x}_r}: \sum_{i=1}^{N_t}\|\mathbf{f}_{f,o}(i) - \{K(\mathbf{x}_r - \mathbf{x}(i))\}\|^2 \qquad (13)$$

One should note that the assumption that object's impedance parameters and reference trajectory are stationary for short period of time only works for slow task right now. For fast task, it may require to use a high speed (force and motion) sensor to collected enough to obtain a reasonable optimal solution for eq. (13). Also, from eq. (13), we can obtain

---

[2]In practice, only the contact forces on each fingertip can be measured, which include the grasping forces and the manipulating forces. the sum of grasping forces is very small in our setting, i.e., equation (5), which can be ignored.

the desired impedance parameters for each time window and their corresponding reference trajectory. In this framework, the desired impedance parameters and the reference trajectory will depend on time. To account for this, we define a variable $\phi \in [0, 1]$ to represent the completion of the task, which is a function of the desired trajectory, i.e., $\phi = \Phi(\mathbf{x}_r)$. In our experiments, Φ is given by the distance from the current configuration to the goal configuration. Thus, the impedance parameters and desired trajectory are expressed as a function of ϕ.

Since the system should be stable, additional constraints should be taken into account. First, the stiffness matrix should be positive semi-definite and its elements must be less than some maximum value since we assume that human will not demonstrate extremely large object stiffness. Also, for the reference trajectory, it should be close to the actual measured object trajectory. Thus we have:

$$
\begin{aligned}
K_{i,j} &\leq k_{lim}, \quad i = 1...6, j = 1...6; \\
\|\mathbf{x}_r - \mathbf{x}(i)\| &\leq \Delta x_{lim}, \quad i = 1...N_t; \\
\|\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)\| &\leq \Delta \dot{x}_{lim}, \quad i = 1...N_t;
\end{aligned}
\tag{14}
$$

where $\Delta x_{lim} \in \mathbb{R}^+, \Delta \dot{x}_{lim} \in \mathbb{R}^+$ is upper bound of the difference between the actual and real (position and velocity) trajectories. With the objective function (13) and the constraints (14), the optimized impedance parameters and the reference trajectory can be obtained in each time window.

V. EXPERIMENTS AND DISCUSSION

In the experiments, we use a 4-fingered Allegro hand[3] to test the object impedance specification for robust grasping and dexterous manipulation. The initial grasp and the grasping stiffness are predefined.

A. Setup

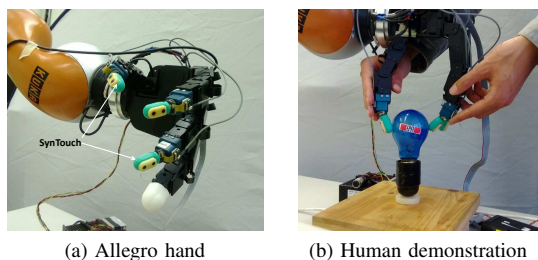

(a) Allegro hand   (b) Human demonstration

Fig. 4: (a) The Allegro hand mounted with the SynTouch tactile sensors on the fingertips; (b) Human demonstration of bulb replacement.

Each of the four fingers of the *Allegro hand* has 4 independent torque-controlled joints, see Fig. 4a. In our experiments, we only use 3 fingers even though our controller can be generalized to 4 fingers. Each fingertip of these 3 fingers has been mounted with a biometric tactile sensor from *SynTouch*[4], which has been calibrated to provide contact information such as contact position and contact force.
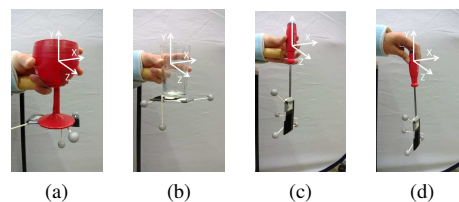
(a)   (b)   (c)   (d)

Fig. 5: Human demonstration of robust grasping on 3 different objects: glass, cup, screwdriver (side and top grasp). The motion of the object when perturbed is tracked by OptiTrack.

B. Robust Grasping

In the robust grasping experiment, a human expert demonstrates 4 grasps as shown in Fig. 5. The arm and wrist are fixated on the table so that the object motion will only come from the finger motion. During the experiments, for each object the perturbations are applied by another person randomly. The position and orientation for the objects are tracked using a motion capture system from *OptiTrack*[5] at a sampling rate of 240Hz. More than 10000 datapoints are collected for each object.

The recorded object orientation is transformed into RPY Euler angles. The relative impedance parameters for the 4 grasps in different directions are computed using equation (9). In general, the choice of frame of reference depends on the task. Here we compute a diagonal stiffness matrix in the reference frame of the object since this is also the frame of reference in our impedance controller. It is also possible to extract the principle directions and corresponding stiffness along these directions from eq. (9) [20], but then we need to transform the stiffness along these principle directions into the object's frame of reference in real time during implementation.

The relative stiffness for these grasps are shown in Fig. 6– Fig. 9. The results show that in different directions, the relative stiffness is indeed different, which means that an isotropic stiffness is not always suitable. Also, comparing the relative stiffness for two different grasps on the screwdriver (Fig. 5, (c) and (d)), see Fig. 8 and Fig. 9, we found that the rotational stiffness around Y-axis is totally different. In the top grasp, the rotational stiffness around Y-axis is much smaller than that of side grasp, which means that the top grasp requires smaller forces to rotate the screwdriver around Y-axis. This result coincides with our intuition. Here, we only show the implementation results from the grasp on the glass. The parameters are set as follows: $K_{gi} = 20N/m$, $L_i = 0.5\|\Delta \mathbf{p}_i\| m$, $k_{tx} = 20N/m$, $k_{ty} = 240N/m$, $k_{tz} = 30N/m$, $k_{rx} = 1.2 \times 10^{-3} Nm/\deg$, $k_{ry} = 6 \times 10^{-3} Nm/\deg$, $k_{tz} = 1.2 \times 10^{-3} Nm/\deg$. The snapshot of the implementation on the Allegro hand is shown in Fig. 10.

C. Dexterous Manipulation

For dexterous manipulation, we use the bulb replacement as an example, Fig. 4b. The bulb is initially on the socket

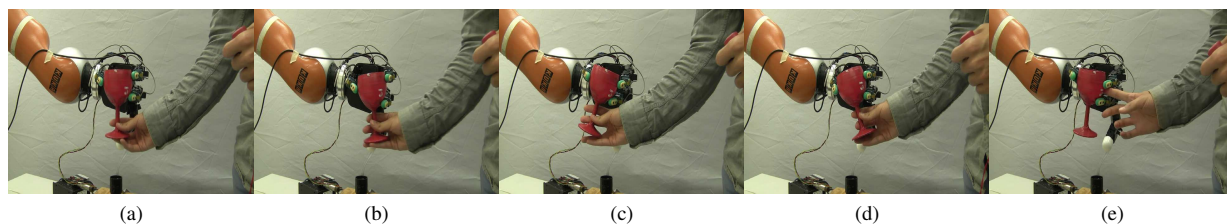<center>(a)   (b)   (c)   (d)   (e)</center>

Fig. 10: Testing of robust grasping: Snapshots of the response of our controller when a human perturbs the original position of the glass. The fingers adapt smoothly to follow the direction of motion induced by the human. The impedance was learned from former human demonstration, using results in Fig. 6. The video is available at: `http://lasa.epfl.ch/~miao/robust_grasping.wmv`
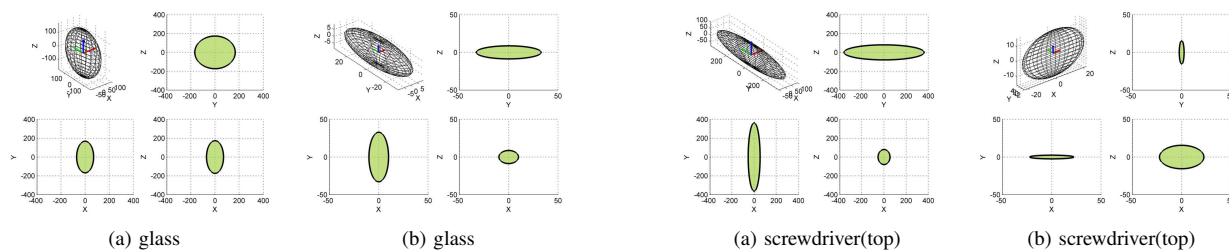

<center>(a) glass   (b) glass</center>

Fig. 6: (**a**): The relative translational stiffness for glass, $k_{tx} < k_{tz} < k_{ty}$; (**b**): The relative rotational stiffness for glass, $k_{rx} \approx k_{rz} < k_{ty}$.

<center>(a) cup   (b) cup</center>
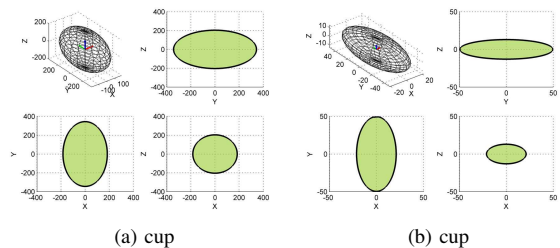
Fig. 7: (**a**): The relative translational stiffness for cup, $k_{tx} \approx k_{tz} < k_{ty}$; (**b**): The relative rotational stiffness for cup, $k_{rx} < k_{rz} < k_{ty}$.

<center>(a) screwdriver(side)   (b) screwdriver(side)</center>
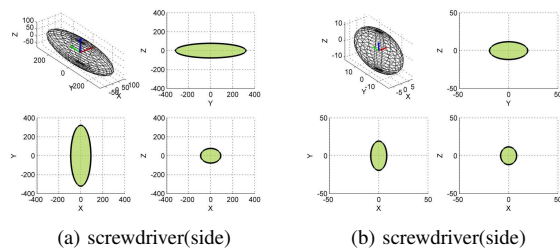
Fig. 8: (**a**): The relative translational stiffness for screwdriver (side grasp), $k_{tx} \approx k_{tz} < k_{ty}$; (**b**): The relative rotational stiffness for screwdriver (side grasp), $k_{rx} \approx k_{rz} < k_{ty}$.

already. During the human demonstration, only two fingers are used as the impedance is learned in object's frame of reference and using two fingers is easier to demonstrate. The manipulating forces are measured using SynTouch mounted on the fingertips. The object real trajectory is tracked using OptiTrack [6]. Using equation (13) and (14), with $\Delta x_{lim} =$

<center>(a) screwdriver(top)   (b) screwdriver(top)</center>
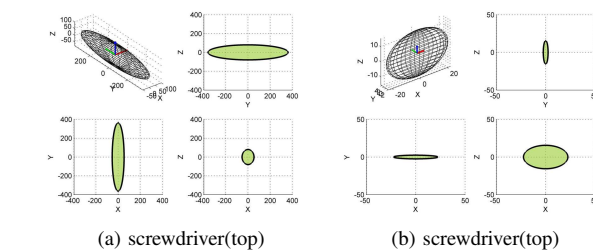
Fig. 9: (**a**): The relative translational stiffness for screwdriver (top grasp), $k_{tx} \approx k_{tz} < k_{ty}$; (**b**): The relative rotational stiffness for screwdriver (top grasp), $k_{ty} < k_{rx} \approx k_{rz}$.

$60\deg, k_{lim} = 100N.mm/deg$ [7], the reference trajectory and desired stiffness for bulb replacement are obtained and shown in Fig. 11 and Fig. 12a, respectively.

If we compare the desired rotation angle with the actual rotation angle, we see that the difference varies during the whole task. This means that human demonstrator indeed regulates the difference between the actual and reference trajectories as well as the stiffness parameter. When looking at the desired object stiffness, Fig. 12a, we see that the desired stiffness increases significantly during the last phase of the task. This is due to the fact that the resistance torque between the bulb and the socket increases significantly at the last phase. We repeated this demonstration 10 times, the obtained desired stiffness for each trial is shown in Fig. 12b.

If we could measure the rotational angle of the bulb using vision, then the status of task completion ϕ and the corresponding k and $\mathbf{x}_r$ can be obtained directly. Unfortunately, it is difficult to rely on vision for dexterous manipulation task as the hand often obstructs the object from the camera's view. For this reason, we rely on tactile information to guide the task process. Figure 12 indicates that the regulation of stiffness during this task seems to follow two distinct phases. During the first phase, which occurs before break point $\phi = 0.8$ (i.e. more than 2/3rd of the total duration of the task), the stiffness is quasi constant. Whereas in the second phase, it increases steadily. We model this by setting a constant value for the stiffness for the first phase and by increasing linearly the stiffness up to its upper bound for the remainder of the task, i.e., $4N.mm/deg$, see Fig. 12b. We noticed during the implementation that this breakpoint corresponds to the instant when one fingertip (usually the thumb of Allegro

---

[6]During the human demonstration, in order to track the object robustly, the experimenter must take care of not placing his fingertips on the vision markers.

[7]Δx_{lim} is chosen by considering the rotation limitation of human hand and the Allegro hand.

hand) starts slipping on the bulb. In order to detect the slippage, we use the contact forces $\mathbf{f}_c = [f_{cx}, f_{cy}, f_{cz}]$ from SynTouch on each fingertip, with f_{cx}, f_{cy} and f_{cz} being the tangential forces in two directions and the normal force, respectively. A slippage occurs at one fingertip if the contact forces that fingertip satisfying $\sqrt{f_{cx}^2 + f_{cy}^2} > \mu f_{cz}$, μ is the coefficient of friction that is set manually. In our task, we choose $\mu = 0.9$. Fig. 14 shows the computed coefficient of friction during one successful implementation. The resulting control strategy is given in Algorithm 1. The snapshot of the implementation on Allegro hand is shown in Fig. 13

Algorithm 1: Controller for bulb replacement task

---

1 Move fingers to initial positions: InitialGrasp();
2 **repeat**
    Impedance Control Mode:
    SetGrasp();
    Compute the VF (eq.(7));
    Set parameters:
    $L_i = 0.5\|\Delta\mathbf{p}_i\|m$, $k_{tx} = k_{ty} = k_{tz} = 0N/m$
    $k_{rx} = k_{rz} = 0Nm/\deg$, $k_{ry} = 1 \times 10^{-3}Nm/\deg$,
    $K_{gi} = 12N/m$
    $x_r = 60\deg$
    interpolate x_r to smooth the controller:
    for *i=1* **to** *1000* **do**
        Compute the current reference point:
        x_{cr}=Slerp(x_r,i);
        Send joint torques: ObjImp();
    Open Finger and move back to initial grasp:
    InitialGrasp();
  until *DetectSlip()*
3 **if** *DetectSlip()* **then**
4   Rotate the bulb for another 4 times:
    for *i=1* **to** *4* **do**
        Impedance Control Mode:
        SetGrasp();
        Compute the VF (eq.(7));
        Set parameters:
        $L_i = 0.5\|\Delta\mathbf{p}_i\|m$, $k_{tx} = k_{ty} = k_{tz} = 0N/m$
        $k_{rx} = k_{rz} = 0Nm/\deg$,
        $k_{ry} = 1 + i * 0.75 \times 10^{-3}Nm/\deg$,
        $K_{gi} = 12 + i * 2.5N/m$
        $x_r = 60\deg$
        interpolate x_r to smooth the controller:
        for *i=1* **to** *1000* **do**
            Compute the current reference point:
            x_{cr}=Slerp(x_r,i);
            Send joint torques: ObjImp();
        Open Finger and move back to initial grasp:
        InitialGrasp();
5 **return** 0;

D. Discussion

During the robust grasping, we didn't consider the problem of grasp stability. As studied in [25], the object dynamic
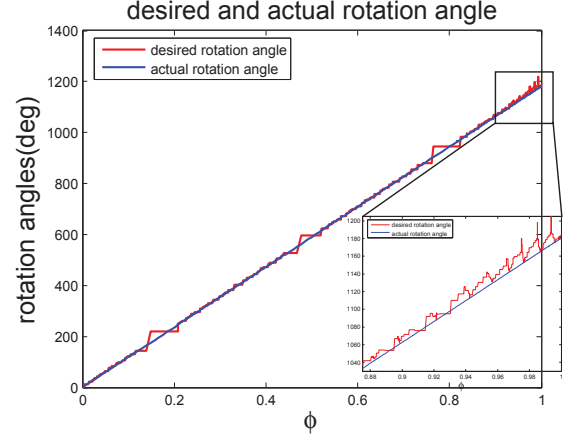

Fig. 11: The learned reference trajectory for trial 5. ϕ is the variable that represents the status of completion of the task, which is chosen as the ratio between current rotational angle and maximal rotational angle.
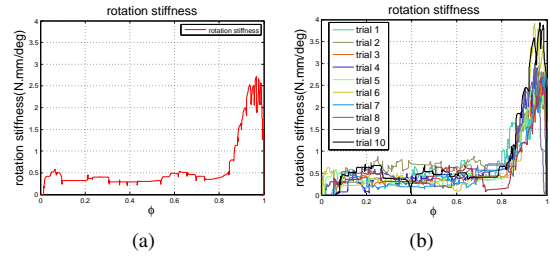

Fig. 12: (**a**) The learned desired object stiffness for trial 5. The stiffness will significantly increase at the last phase of bulb replacement. (**b**) The learned desired object stiffness for 10 different trials.
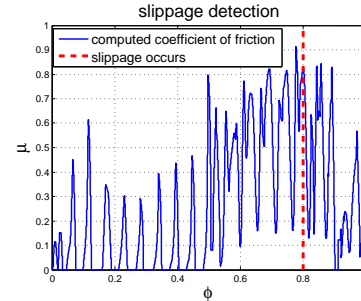

Fig. 14: The computed cofficient of friction on the fingertip of thumb during one successful implementation.

stability will be closely related to the choice of grasp stiffness. In future work, we will investigate ways in which to shape the stiffness while taking the grasp stability into account.

Second, currently the initial grasp and grasp stiffness are predefined in our experiments, which is based on the assumption that the given grasp can realize the desired object impedance. However, given the object impedance specification and a multi-fingered robotic hand, how to choose a grasp that can realize this desired impedance will be a challenging extension direction. One of the possible ways will be extending the optimization framework for grasp synthesis in our previous work [26].
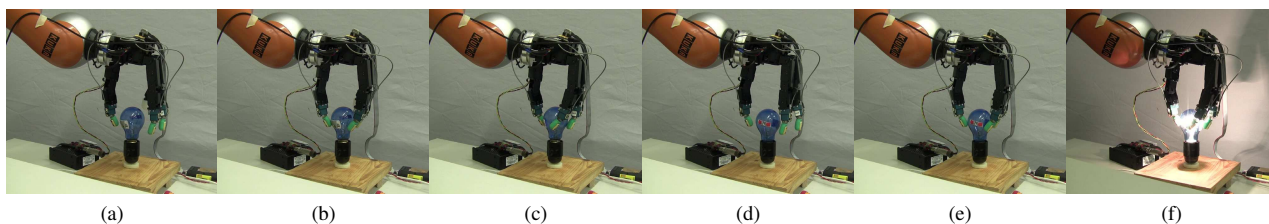
Fig. 13: The snapshots for dexterous manipulation. The video for this demo is available at: `http://lasa.epfl.ch/~miao/bulb_replace.wmv`

VI. CONCLUSIONS

In this paper, an object-level impedance learning approach was proposed for both robust grasping and dexterous manipulation. For robust grasping, the relative stiffness is specified by measuring the displacement of object under perturbation. For dexterous manipulation, the desired reference trajectory and the desired object impedance is learned through an optimization-based approach. The results show that a varying stiffness is more suitable in our task. Both of these approaches are validated on a multi-fingered robotic hand. We are currently working on integrating tactile feedback into the object impedance controller for grasping stiffness specification.

ACKNOWLEDGMENT

REFERENCES

[1] Z. Li, P. Hsu, and S. Sastry, "Grasping and coordinated manipulation by a multifingered robot hand.," *The International Journal of Robotics Research*, vol. 8, no. 4, pp. 33–50, 1989.

[2] T. Yoshikawa and X.-Z. Zheng, "Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 219–230, 1993.

[3] Z. Li, Z. Qin, S. Jiang, and L. Han, "Coordinated motion generation and real-time grasping force control for multifingered manipulation," in *Proceedings of International Conference on Robotics and Automation (ICRA), 1998*.

[4] S. A. Schneider and R. H. Cannon, "Object impedance control for cooperative manipulation: theory and experimental results," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 383–394, 1992.

[5] T. Wimbock, C. Ott, and G. Hirzinger, "Analysis and experimental evaluation of the intrinsically passive controller (IPC) for multifingered hands," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2008*.

[6] K. Tahara, S. Arimoto, and M. Yoshida, "Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2010*.

[7] T. Wimbck, C. Ott, A. Albu-Schffer, and G. Hirzinger, "Comparison of object-level grasp controllers for dynamic dexterous manipulation," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 3–23, 2012.

[8] M. T. Mason and J. K. Salisbury, *Robot Hands and the Mechanics of Manipulation*. The MIT series in Artificial Intelligence, Cambridge, Massachusetts: The MIT Press, 1985.

[9] H. Liu and G. Hirzinger, "Cartesian impedance control for the DLR hand," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS), 1999*.

[10] K. Tahara, K. Maruta, A. Kawamura, and M. Yamamoto, "Externally sensorless dynamic regrasping and manipulation by a triple-fingered robotic hand with torsional fingertip joints," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2012*.

[11] N. Hogan, "Impedance control - an approach to manipulation. i - theory. II - implementation. III - applications," *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, Mar. 1985.

[12] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st ed., 2008.

[13] M. Cutkosky and I. Kao, "Computing and controlling compliance of a robotic hand," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 151–165, 1989.

[14] K. Shimoga and A. Goldenberg, "Grasp admittance center: Choosing admittance center parameters," in *American Control Conference, 1991*, pp. 2527–2532, 1991.

[15] B.-H. Kim, B.-J. Yi, S.-R. Oh, and I. H. Suh, "Task-based compliance planning for multifingered hands," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2001*.

[16] B.-H. Kim, B.-J. Yi, S.-R. Oh, and I. H. Suh, "Fundamentals and analysis of compliance characteristics for multifingered hands," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2001*.

[17] B.-H. Yang and H. Asada, "Progressive learning and its application to robot impedance learning," *IEEE Transactions on Neural Networks*, vol. 7, no. 4, pp. 941–952, 1996.

[18] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.

[19] P. Sikka and B. J. McCarragher, "Stiffness-based understanding and modeling of contact tasks by human demonstration," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS), 1997*.

[20] K. Kronander and A. Billard, "Online learning of varying stiffness through physical human-robot interaction," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2012*.

[21] E. L. Sauser, B. Argall, G. Metta, and A. Billard, "Iterative learning of grasp adaptation through human corrections.," *Robotics and Autonomous Systems*, vol. 60, no. 1, pp. 55–71, 2011.

[22] A. Okamura, N. Smaby, and M. Cutkosky, "An overview of dexterous manipulation," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2000*.

[23] T. Yoshikawa, "Multifingered robot hands: Control for grasping and manipulation," *Annual Reviews in Control*, vol. 34, no. 2, pp. 199 – 208, 2010.

[24] B. Huang, S. El-Khoury, M. Li, J. J. Bryson, and A. Billard, "Learning a real time grasping strategy," in *Proceedings of International Conference on Robotics and Automation (ICRA), 2012*.

[25] C.-H. Xiong, Y.-F. Li, H. Ding, and Y.-L. Xiong, "On the dynamic stability of grasping," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 951–958, 1999.

[26] S. El Khoury, M. Li, and A. Billard, "Bridging the gap: One shot grasp synthesis approach," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS), 2012*.

Collaborative Human-Humanoid Carrying Using Vision and Haptic Sensing

Don Joven Agravante[1], Andrea Cherubini[1], Antoine Bussy[1,2], Pierre Gergondet[2] and Abderrahmane Kheddar[1,2]

Abstract— **We propose a framework for combining vision and haptic information in human-robot joint actions. The framework consists of a hybrid controller that uses both visual servoing and impedance controllers. This can be applied to tasks that cannot be done with vision or haptic information alone. In this framework, the state of the task can be obtained from visual information while haptic information is crucial for safe physical interaction with the human partner. The approach is validated on the task of jointly carrying a flat surface (e.g. a table) and then preventing an object (e.g. a ball) on top from falling off. The results show that the presented approach achieves its goal. Furthermore, the framework presented results in a more collaborative setup by imparting task knowledge to the robot as opposed to a passive follower.**

Index Terms— **Physical Human-Robot Interaction**

I. INTRODUCTION

Humanoid robots provide many advantages when working together with humans to perform various tasks. This is because humans have an extensive experience in physically collaborating with each other. Hence, humanoids can interact with humans because of their human-like range of motion and sensing capabilities. This reduces the need to learn how to interact with the robot. However, many challenges are still present in the various research areas that study physical human-robot collaboration. Here, the area of interest is using vision and force information together to enable human-robot joint actions, which are collaborative tasks requiring both parties to physically interact with each other (e.g. carrying a large object together). In such tasks, the robot:

1) must move safely and regulate interaction forces,
2) has shared human-in-the-loop control,
3) can only use its on-board sensors.

The first two items are the main aspect of all human-robot collaborative tasks. The last constraint is important for true autonomy. For example if vision has a limited field-of-view, external room cameras should not be used.

Physical human-robot collaboration has largely relied on the use of haptic data (force/torque) for control. This is because the main priority is the regulation of the interaction forces between the human and robot. For example, previous works [1]–[4] have demonstrated that using only haptic

[1]CNRS-UM2 LIRMM UMR 5506, Interactive Digital Human group, 161 Rue Ada, 34392 Montpellier, France {firstnames.lastname}@lirmm.fr.
[2]CNRS-AIST, JRL (Joint Robotics Laboratory), UMI 3218/CRT, Intelligent Systems Research Institute, AIST Central 2, Umezono 1-1-1, Tsukuba, 305-8568, Japan.

information, a humanoid robot can help a human carry large objects (e.g. a table or a beam or a panel). A possible future application of this is in construction sites [1]. The same scenario can also be applied to the household, such as moving furniture (e.g. table). While doing this task, one can imagine the need to prevent an object on top from falling off. For example in moving a table a short distance, it might be necessary to move it carefully with the objects on top rather than removing the objects, transporting the table and then placing the objects back on top. In this scenario, haptic information alone is not rich enough to give the robot knowledge about the state of the objects on top of the table. But vision can obtain such information, being largely complementary to haptics (analogous to human sight and touch). Using both information sources might enable a humanoid to perform more complicated tasks, similar to a human. Although the benefits are great, there are not many established methods integrating vision and force control.

In [5], three general categories for combining vision and force control are identified: traded, hybrid and shared. Traded control is the simplest, and switches between a pure visual servoing controller and a pure force control method given a certain threshold of the task error. In hybrid methods, a prior specification of a "task-frame" [6], [7] is required to decouple vision and force into orthogonal spaces. With this, the controllers can be designed separately. Finally, shared control methods aim at utilizing both vision and force information together in the same space, such that all available information is used [5]. For example, in [8], a force feedback is used to correct the visual servo control trajectory.

In this paper, the impedance control framework [9], is used. This allows a manipulator to be compliant by defining a virtual impedance. In this framework, vision can be used to provide a reference trajectory that is tracked in the absence of external forces [10]–[12]. When contact does occur, it has the properties of the shared control method where vision and force determine the control of the same degree of freedom (DOF) simultaneously. This approach is preferred over the other methods since it can allow for compliance in all DOF. This approach has been investigated previously in [10], [11] with experiments of a robot interacting with objects. Here and in our previous work [12], we use this approach for physical human-humanoid collaboration experiments.

Our work is partly based on the joint object transportation framework which was introduced in [3] and utilized in [12]. The overall system is hybrid - using the 3 DOF that are

controlled with only haptic information in [3] and designing separate controllers for the remaining DOF. However, these new controllers use the shared approach where both vision and force affect the DOF at the same time. This complicates the semantics to describe the whole system. In [12], vision was used together with haptic data to stabilize the height of the table, providing the reference trajectory of the impedance controller in 1DOF. As a continuation to this work, we use the same general framework of building a visual servoing controller for providing a reference trajectory to the impedance controller. However, we relax the constraint on having a static object in [12]: the object on the beam is free to move so control in 2 DOF is needed. This is a more difficult task and vision is indispensable to acquiring the state of the task. Another contribution of this work is the analysis of collaboration in joint actions when each agent has their own notion of how to do the joint task, as opposed to the common leader-follower strategy.

To continue, the general framework is described in Section II. The example task of human-humanoid table carrying while keeping an object on top from falling is then described in Section III. The details on implementing the general framework to this specific task is done in Section IV. Results from experiments are then presented in Section V. The novelty of equal collaboration and the challenges it poses is discussed in Section VI. Finally, Section VII concludes and outlines some future works to be done.

II. GENERAL CONTROL FRAMEWORK

Our general approach to combining vision and haptic cues is coupling a visual servoing controller to an impedance controller. This simplifies the design by decoupling the vision and force controllers in a systematic way. An overview of the complete control framework is shown in Fig. 1.
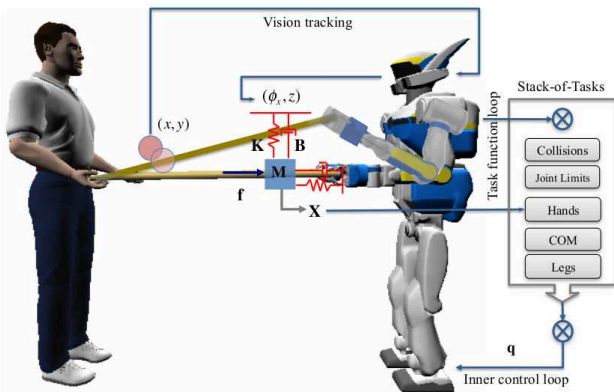

Fig. 1. The general control framework applied to the task of balancing an object on the table

Fig. 1 also shows the task example used in this paper - balancing an object on the table. The following subsections explain this general framework in a bottom-up approach starting from the lower level controllers and abstracting it higher to the cognitive level. The lowest level of control is the inner joint-level control. This is represented by $\mathbf{q}$

in Fig. 1. To abstract from the joint level to the "task level", the Stack-of-Tasks framework is used [13]. It is a generalized inverse kinematics abstraction layer that creates a hierarchical organization of different tasks to be executed giving higher priority to critical tasks [13]. It allows for easier integration with sub-tasks. For example, our experiments make use of the walking algorithm in [14] as a sub-task.

A. Impedance Control

The other sub-task concerns the grippers. In Fig. 1 the humanoid uses its grippers to co-manipulate an object with a human. To do this, it needs to be safe and intuitive to use. Here, impedance control [9] is used to regulate the contact interaction (for safety) between the robot and its environment. It also gives a simple physical analogy to the control - a virtual mass-spring-damper system [9]. This system is governed by the general equation:

$$\mathbf{f} = \mathbf{M}(\ddot{\mathbf{X}}_d - \ddot{\mathbf{X}}) + \mathbf{B}(\dot{\mathbf{X}}_d - \dot{\mathbf{X}}) + \mathbf{K}(\mathbf{X}_d - \mathbf{X}). \qquad (1)$$

The contact interaction is measured by the force-torque sensors in the robot grippers and is represented as $\mathbf{f}$. The vectors $\mathbf{X}_d, \dot{\mathbf{X}}_d$ and $\ddot{\mathbf{X}}_d$ are a desired pose and its first and second derivative. Correspondingly, vectors $\mathbf{X}, \dot{\mathbf{X}}$ and $\ddot{\mathbf{X}}$ represent an actual pose and its first and second derivative. Finally, matrices $\mathbf{M}, \mathbf{B}$ and $\mathbf{K}$ are the inertia, damping and stiffness parameters that define the desired virtual mass-spring-damper system [9]. Strictly following the terminology and causality from [9], our implementation on the HRP-2 humanoid, is an "admittance controller" since the robot is position-controlled by the Stack-of-Tasks, which uses the output of $\mathbf{X}, \dot{\mathbf{X}}$ and $\ddot{\mathbf{X}}$ from the impedance controller. These are obtained by solving the differential equation of Eq. (1) given the other variables. The parameters $\mathbf{M}, \mathbf{B}$, and $\mathbf{K}$ are determined empirically to provide comfort for the human collaborator. Finally, $\mathbf{X}_d, \dot{\mathbf{X}}_d$ and $\ddot{\mathbf{X}}_d$ are the desired pose and trajectory of the mass-spring-damper's reference position. These are detailed in the next subsection.

B. Proactive Behavior and Visual Servoing

For the general impedance controller of Eq. (1) a "passive" behavior is defined by setting the desired pose $\mathbf{X}_d$ as constant. This case is illustrated in Fig. 2(a) where only the human knows about the task to be done. This is the "classical" case in human-robot collaboration. In such a case (and considering constant impedance parameters $\mathbf{M}, \mathbf{B}, \mathbf{K}$), the robot's motion $(\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}})$ can only be initiated by an external force $\mathbf{f}$ due to Eq. (1). Recent research aims to make the robot a proactive follower to make the system more comfortable for the human. A way to achieve this is by creating a suitable desired pose and trajectory $(\mathbf{X}_d, \dot{\mathbf{X}}_d, \ddot{\mathbf{X}}_d)$ such that the human effort is minimized [3], [12], [15]. These works differ in the approach taken to produce the desired pose and trajectory. In [15], human motion is predicted by a minimum jerk model to give the desired pose. In [3], a human pair doing a joint transportation task was studied and it was observed from the data that the pair moves in constant velocity phases during this task. A finite state machine (FSM)

is then created by using the contant velocity assumption, giving the desired pose and trajectory. Haptic cues are used to determine the switching of states in the FSM [3]. Our latest work [12] takes the same approach as the one in this paper and is illustrated by Fig. 2(b). Here, the humanoid is given knowledge of the task. This is done by designing a visual servoing controller specific to the task and using the output as the desired pose and trajectory of the impedance controller. This also means that the robot has some autonomy in doing the task driven by its own knowledge of the state of the task. With the reasonable assumption that during the collaborative task human motion is task driven, the source (human intention to do the task) is taken into account rather than the result (human motion). This differentiates our approach from those that aim to model/predict human motion such as the early work described in [15].

Fig. 3.   Human-humanoid table carrying task with reference frames.

a reference "control frame" $\{cf\}$, rigidly linked to the table, to a desired pose with respect to a local frame $\{l\}$, rigidly linked to the robot torso. This pose is represented by the homogeneous transformation matrix $^l\mathbf{T}_{cf}$. To achieve this, the hand poses $\{rh\}$ and $\{lh\}$ are controlled in the local frame according to:

$$^l\mathbf{T}_h = {}^l\mathbf{T}_{cf}{}^{cf}\mathbf{T}_h \qquad h = \{rh, lh\}.$$

Assuming a rigid grasp of the table, the homogeneous transformation matrices $^{cf}\mathbf{T}_{rh}$ and $^{cf}\mathbf{T}_{lh}$ are constant and known once $\{cf\}$ has been defined. For the implementation of the impedance controller, $^l\mathbf{T}_{cf}$ is converted into the 6-vector $\mathbf{X} = {}^l[x, y, z, \phi_x, \phi_y, \phi_z]_{cf}^\top$ made up of the Cartesian coordinates and Euler angles (the ZYX convention is used which conveniently places the singularity at $\phi_y = \pm 90°$, an impossible case of the joint transportation task).

An intuitive description of the task is to "keep the object on the table from falling off". The control design can be defined to attract the object (o) towards an appropriate desired goal point (d) (refer to Fig. 3). To realize this task, direct force application on the object is not possible, since the priority is table transportation. Hence, only an indirect action can be applied by tilting the table to contrast gravity. To do this, a decoupled approach can be used, where the control of $^t y_o$ is done through ϕ_x, and that of $^t x_o$ through z. It is chosen to regulate $^t x_o$ with z and not with ϕ_y because the latter option would make the task uncomfortable for the human. In fact, controlling ϕ_y forces the human to actively move his/her z position. Instead, by controlling z and leaving ϕ_y compliant, the human at the other end just needs to be compliant in his/her ϕ_y, which is more comfortable.

To integrate the whole system, the important part is defining the admittance controller's desired trajectory $(\mathbf{X}_d, \dot{\mathbf{X}}_d, \ddot{\mathbf{X}}_d)$ for all 6 DOF. The vision-based control takes care of two DOF (z, ϕ_x). Three DOF (x, y, ϕ_z) of the pose are defined from the FSM of our group's earlier work [3], [4] (described briefly in Section II-B). Finally, the remaining DOF (ϕ_y) is made compliant by setting $\phi_{y,d} = 0$.

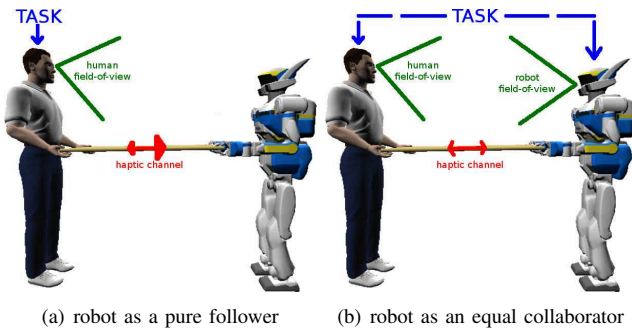(a) robot as a pure follower   (b) robot as an equal collaborator

Fig. 2.   Human-humanoid collaboration. (a) shows the passive case with the robot as a pure follower guided only by haptic information. (b) illustrates an equal collaboration approach where both human and robot have a complete knowledge of the task (represented by the blue arrows). Furthermore, each uses both vision (green) and haptic (red) information to achieve this task.

Visual servoing is a term used for controlling robots using visual information [16]. To create the visual servoing portion of the framework, two important components are needed: visual feature tracking and a controller based on this feature [16]. However, in the current state-of-the-art for both modules there is no "best" approach that fits all tasks and problems. Existing methods have important tradeoffs to consider for the whole system [16]. In our works, we take an analytical approach to building the visual servoing portion. In [12], the task is to keep the table horizontal (perpendicular to the gravity field). The goal is then to minimize the rotation between the normal of the plane and the gravity field vector and a visual servoing system was designed to do this.

III. TASK DESCRIPTION

As a test for the general framework described, the task of jointly transporting a surface while keeping a mobile object on top from falling off is used. Fig. 3 illustrates the task with the important reference frames and naming convention used in the rest of this paper. The vectors composing the Cartesian frames are color coded: Red-Green-Blue correspond to $(\vec{x}, \vec{y}, \vec{z})$ respectively.

Fig. 3 shows that the robot can control the table through its hands $\{rh\}$ and $\{lh\}$. The control design consists in driving
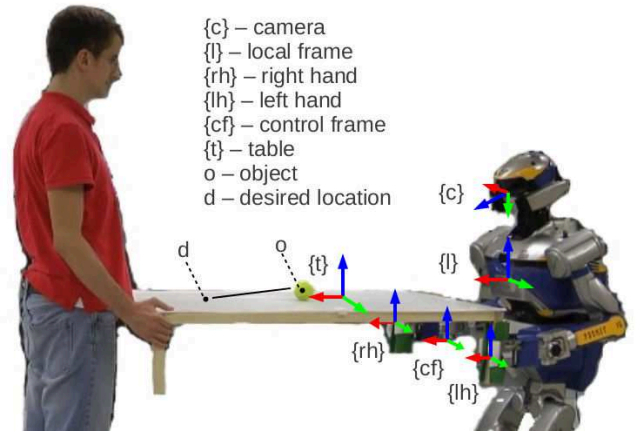
IV. IMPLEMENTATION DETAILS

As explained in Section II-B, the approach is to design a visual servoing controller for the task to make the robot proactive. Two main components are needed: visual tracking and the control design.

A. Vision Algorithm

In the HRP-2, RGB-D data is obtained from an embedded ASUS Xtion device located in the head. Fig. 4 shows typical data of the task. The aim of the vision algorithm is to process this raw data into visual features that can be used for control. An error signal can be defined by $^tx_o - {}^tx_d$ and $^ty_o - {}^ty_d$. For the example task here, z is irrelevant, since $^tz_d \equiv {}^tz_o$. Since the desired location $^t(x, y)_d$ is arbitrarily defined, the vision algorithm only needs to obtain $^t(x, y)_o$. A variety of vision algorithms that can do this may be used, with speed as another consideration. For example, given the object model and the table model, it is possible to use a model based tracker. Designing a novel vision algorithm is not the focus of this work, so we use well-known methods [17]–[19]. Nevertheless, it is briefly described here for completeness.
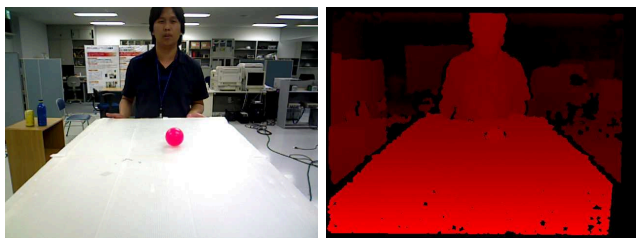

Fig. 4. Typical raw data (RGB + Depth images) during the task. Left: RGB image. Right: Depth image, where dark red→bright red corresponds to "far"→"near" and black pixels are regions without data.

The feature used here is the centroid of the object and the table. The first step is to segment these from the image. Color segmentation is used in our system. For example the pink object in Fig. 4 and yellow object in Fig. 3 can be easily characterized and thresholded by a specific hue range and a high saturation (from the HSV color space). To add robustness, morphological operations (opening and closing) are used to remove outliers. After this, sliding window detection (sped up using the image pyramids concept) finds the most probable location. The centroid of the detected blob is (u, v) in pixel coordinates. This is then converted into ^cx_o and ^cy_o by using the intrinsic camera calibration parameters (f_x, f_y, c_x, c_y) and the depth ^cz_o in the following equations:

$$^cx_o = \frac{^cz_o(u - c_x)}{f_x} \quad , \quad {}^cy_o = \frac{^cz_o(v - c_y)}{f_y}. \quad (2)$$

The next step is to segment the table in the image. A flood fill algorithm [19] is done in saturation-value-depth space. This algorithm starts with a "seed" point and grows the region based on a connectivity criteria between neighboring pixels. Here, the seed point is the bottom pixel of the ball. A low saturation and high value characterize well the "white" color

of the table. The addition of depth ensures connectivity in Cartesian space, simplifying for example the segmentation between table and floor pixels. Finally, some morphological operations (opening and closing) are done to remove outliers. From these segmented points, the Cartesian centroid is used as $^c\mathbf{t}_t$ (a translation vector). The Cartesian coordinates of the object in the table frame are then obtained by:

$$^t\mathbf{t}_o = {}^c\mathbf{T}_t^{-1} \, {}^c\mathbf{t}_o. \quad (3)$$

The homogeneous transformation matrix $^c\mathbf{T}_t$ is composed of the table centroid position $^c\mathbf{t}_t$ and the rotation matrix $^c\mathbf{R}_t$. A simple approximation consists in setting $^c\mathbf{R}_t$ equal to $^c\mathbf{R}_{cf}$, which is obtained from proprioception.

B. Vision-Based Control

The control design needs to drive $^t\mathbf{t}_o$ to $^t\mathbf{t}_d$. There are several methods to do this. Here, a simple PD controller is used such that:

$$C_i(s) = K_{p,i} + K_{d,i}s \qquad i = \{x, y\}. \quad (4)$$

This choice is justified by analyzing the task using a simple sliding model (i.e., neglecting friction and angular momentum). Fig. 5 illustrates the necessary variables for this analysis. Since a control with z rather than ϕ_y is desired, the trigonometric identity $z_r = l_t \sin \phi_y$ is used, where l_t is the length of the table and z_r is the differential height. z_r can be converted to z by a trivial change of frame.
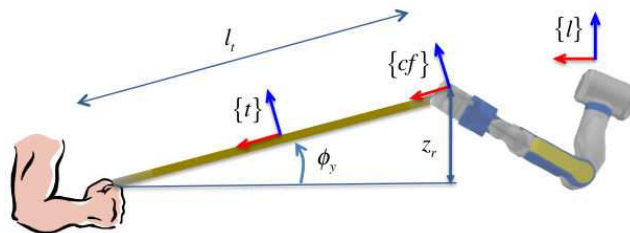

Fig. 5. A simplified "thin beam" model used to control the table height

The Lagrangian equation of motion along $^t\vec{x}$ is:

$$m\ddot{x} = mg \sin \phi_y = mgz_r/l_t. \quad (5)$$

Along y, linearization of the Lagrangian equation about $\phi_x = 0$ leads to:

$$m\ddot{y} = -mg\phi_x. \quad (6)$$

Taking the Laplace transforms of these two equations yields:

$$\begin{cases} s^2 X(s) = gZ_r(s)/l_t \\ s^2 Y(s) = -g\Phi(s). \end{cases} \quad (7)$$

Rearranging, the transfer functions describing the dynamics on the 2 DOF can be derived:

$$\begin{cases} P_x(s) = \frac{X(s)}{Z_r(s)} = \frac{g}{l_t s^2} \\ P_y(s) = \frac{Y(s)}{\Phi(s)} = -\frac{g}{s^2}. \end{cases} \quad (8)$$

It should be noted that both are double integrators. As such, they are only marginally stable when feedback controlled

with a Proportional gain. But the Proportional Derivative controller (PD) chosen can be used. The denominator of the closed loop system transfer function in the two cases is:

$$\begin{cases} D_x(s) = l_t s^2 + gK_{d,x}s + gK_{p,x} \\ D_y(s) = s^2 - gK_{d,y}s - gK_{p,y}. \end{cases} \tag{9}$$

The two systems are asymptotically stable if all the roots of these two polynomials have non-multiple negative real parts. This condition is verified, for a second order polynomial, if all the coefficients are strictly positive. In the case of the characteristic polynomials in (9), this is equivalent to:

$$K_{p,x} > 0 \qquad K_{d,x} > 0 \qquad K_{p,y} < 0 \qquad K_{d,y} < 0. \tag{10}$$

Finally, the applied controllers are:

$$\begin{cases} z = K_{p,x}(x_d - x) - K_{d,x}\dot{x} \\ \phi_x = K_{p,y}(y_d - y) - K_{d,y}\dot{y}. \end{cases} \tag{11}$$

By numerical differentiation $\dot{x}$ (and $\dot{y}$) is obtained as:

$$\dot{x}(t) = \frac{x(t) - x(t - \Delta t)}{\Delta t},$$

with Δt the sampling step. Tuning the gains in (11) according to (10) guarantees stability of the closed loop system, as long as the linear approximation is valid. This implies that ${}^t\mathbf{t}_o$ will converge to ${}^t\mathbf{t}_d$, as desired. The outputs of (11) are fed to the admittance controller (1) as desired values z_d and $\phi_{x,d}$. Numerical differentiation is used to obtain $\dot{z}, \dot{\phi}_x$ in $\dot{\mathbf{X}}_d$. However, for $\ddot{\mathbf{X}}_d$ a piece-wise constant velocity is assumed such that $\ddot{z} = \ddot{\phi}_x = 0$. This also prevents too much noise introduced by a second numerical differentiation.

V. RESULTS

For the experiments, we chose a ball to be the moving object. This makes it similar to a well-studied problem/example in control theory: the "ball-and-plate" system, which is a 2-DOF generalization of the "textbook example" ball-on-beam system (used to study advanced control methods [20]). Although similar, significant differences exist - notably that collaboration is the main issue here.

Several experiments were performed and with 2 different balls - a yellow tennis ball which tends to move slower and a pink ball which moves quite fast. A few different users also tested this early system, but as the described experience was similar this is not discussed here. A more extensive and statistically based "usability study" is planned for future work. Some experiments are shown in the accompanying video and in Fig. 6. The video also shows some results of the vision algorithm detecting the ball and the table. In the initial experiments, both human and humanoid stand stationary and balance the ball on the table. Some disturbance is then introduced (e.g. the ball is pushed by another person) and the gains of the PD controller are tuned according to (10) in order to be able to handle such a disturbance.

After "light" gain tuning of the vision-based controller with such tests, the complete system is tested where the human-humanoid dyad transport the table with the ball on top. To show the performance of the framework, the results

for the complete experiment are shown in Fig. 7, 8 and 9. The results here are from the yellow ball, and the desired ball position is set to ${}^t\mathbf{t}_d = (0.15, 0, 0)$, which is 15cm closer to the human than the table centroid. This is done to avoid the minimum limit of the depth sensor. The robot starts out walking in place. At $time \approx 50s$, the human pulls the table, signaling the robot to help transport the table forward via the FSM developed in [3], [4]. In this task, walking introduces a significant disturbance that can move the ball. Fig. 7 shows the estimated trajectory (blue) of the ball in the perceived table frame. This data is derived from the visual estimate of the ball and goal location. The red border signifies a rough approximate of the table boundaries. The results show that although the ball moves a lot, it doesn't fall off the table during this transportation task.
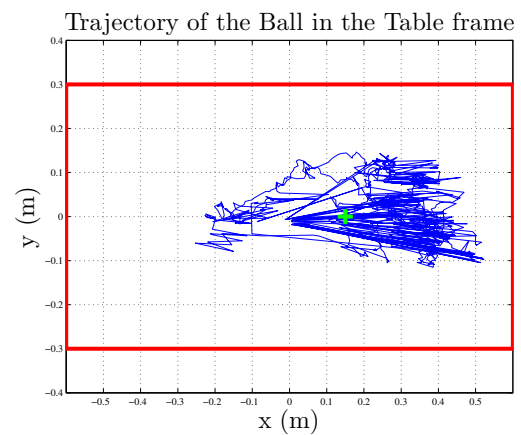

Fig. 7. Controlled ball trajectory (blue) during the experiment, computed from the visual data. The red border is a rough estimate of the table edges and the green + symbolizes the goal.

For verifying the controller, the evolution of the controlled variables is plotted in Fig. 8. From the plots, it shows that the robot is making minor adjustments to ${}^l(\phi_x)_{cf}$ and more adjustments to ${}^l z_{cf}$. This supports what is seen in Fig. 7, i.e., that there is more motion in the ${}^t\vec{x}$ than in the ${}^t\vec{y}$ direction.

Fig. 9 shows that τ_x averages to about $0Nm$ which means that the interaction force is regulated well. As for F_z, it averages to about $12N$. This shows that the robot carries part of the weight of the table and thus lightens the burden on the human. Although there are some small spikes, the overall result still shows that this interaction force is also well-regulated. Furthermore, in both signals a noticeable oscillation can be seen. This correlates to the frequency of the walking gait and the disturbance that it causes.

VI. DISCUSSION ON EQUAL COLLABORATION

The results show that the complete system (Fig. 2b) can do the job well: the vision-based controller tries to keep the ball on the table while the impedance controller regulates interaction forces. A simple analysis of Fig. 2 shows that a disadvantage of the pure follower (Fig. 2a) is that the success/failure of the vision task depends solely on the human partner. Specifically, the human needs to use his/her
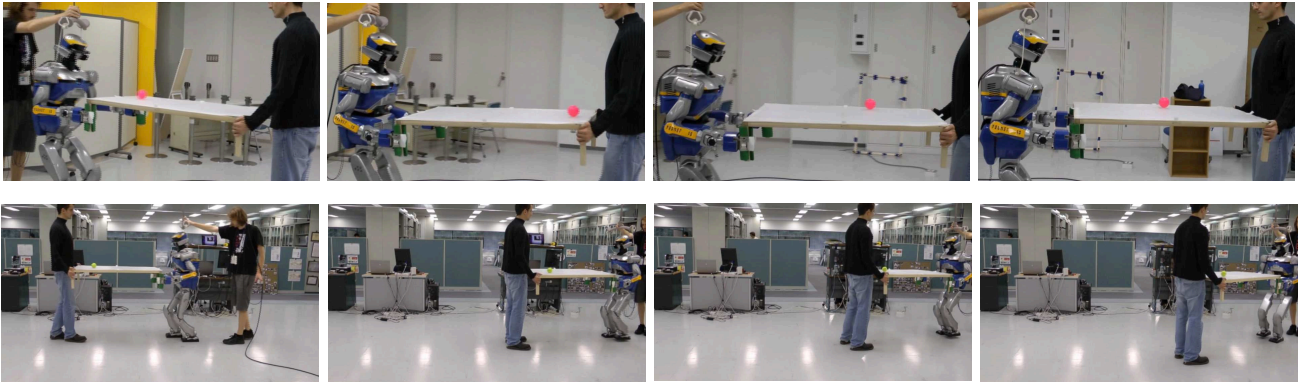
Fig. 6.   Snapshots of two experiments where the human-humanoid dyad transports a table with a ball (fast in the top sequence, slow in the bottom one).
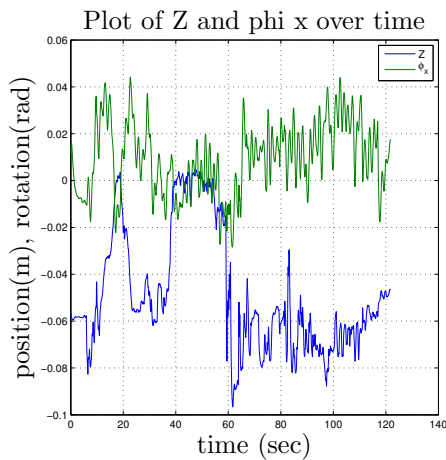

Fig. 8.   Plot of the 2 vision-controlled positions executed by the robot. Blue: $^{l}z_{cf}$, in green: $^{l}(\phi_x)_{cf}$
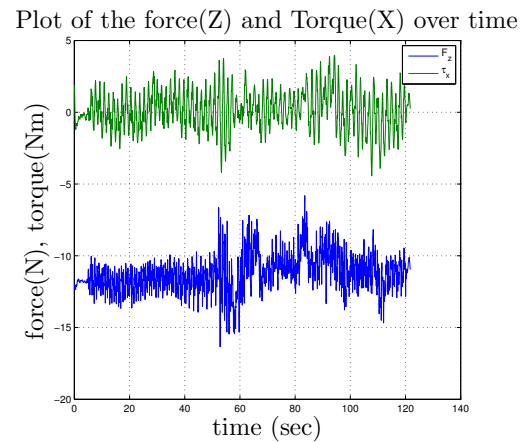

Fig. 9.   Plot of the force and torque in the control frame during the experiment. The 2 correspond to what is used by the impedance controller for the 2 degrees of freedom in the task

vision to observe the state of the task and then apply a sufficient force to haptically communicate to the robot what s/he wants to do. While in Fig. 2b the cognitive load of the task is shared in some capacity - both human and robot are able to observe the state of the task and act accordingly. However, this sharing can become a disadvantage when the human and robot disagree on the state of the task and the action to take [21]. Experimentally, this is handled in our system by making the robot more compliant and less stiff (impedance parameter tuning). This ensures that the human can always safely impose his intention through the haptic channel. This also shows a possible extension of the system which is to dynamically change the impedance parameters: making it more stiff when the robot is more certain of his observations and more compliant when there is more uncertainty. In effect, this makes the impedance parameters a method to weigh the importance between vision (task knowledge) and haptic (human intention) information channels. But, it is important to note that this disadvantage of equal collaboration also applies to human-human pairs and more generally in teams - "teamwork" (or the lack of it). Some preliminary experiments have been made with both

the passive follower and the approach of equal collaboration and the advantages/disadvantages briefly described here can be observed by the human collaborator. One difficulty in presenting these results is in the use of proper evaluation methods since the most important aspect - the comfort of the human collaborator- is very subjective. Another difficulty is to separate the contribution of the human and robot. Although in the results presented here the human is told to be more "passive" (does not try that hard to keep the ball on the table) he also does not try to make the ball fall off, since teamwork is a factor in the overall result. The resolution of these issues is left for future work.

VII. CONCLUSION AND FUTURE WORK

In this paper, a general framework for human-robot joint collaborative tasks was presented. It uses a visual servoing controller to realize the task and a haptic channel to recognize human intention. Both vision and force control are combined in the impedance control framework. This is implemented and tested on a joint transportation task where a human and humanoid robot carry a table with a freely moving ball on top. The objective is to transport the table while keeping

the ball from falling off. This task is used to explore some important issues in robotics: the combination of vision and force information and the issues concerning collaboration - safety and effective human-robot collaboration strategies.

To continue the work here, it is planned to further investigate the combination of vision and force information. Another major area for continued study is in collaboration, such as the idea of dynamically changing the impedance parameters described in Section VI. Further works are to utilize good statistical methodology and experiments with different users to better analyze the qualitative results.

VIII. ACKNOWLEDGEMENT

REFERENCES

[1] K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, "Cooperative works by a human and a humanoid robot," in *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2985–2991, IEEE, 2003.

[2] P. Evrard and A. Kheddar, "Homotopy switching model for dyad haptic interaction in physical collaborative tasks," in *EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.*, pp. 45–50, IEEE, 2009.

[3] A. Bussy, A. Kheddar, A. Crosnier, and F. Keith, "Human-humanoid haptic joint object transportation case study," in *IEEE/RSJ International Conference on Robots and Intelligent Systems*, pp. 3633–3638, IEEE, 2012.

[4] A. Bussy, P. Gergondet, A. Kheddar, F. Keith, and A. Crosnier, "Proactive behavior of a humanoid robot in a haptic transportation task with a human partner," in *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 962–967, IEEE, 2012.

[5] B. J. Nelson, J. D. Morrow, and P. K. Khosla, "Improved force control through visual servoing," in *Proc. of the American Control Conference*, vol. 1, pp. 380–386, IEEE, 1995.

[6] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[7] J. Baeten, H. Bruyninckx, and J. De Schutter, "Integrated vision/force robotic servoing in the task frame formalism," *The International Journal of Robotics Research*, vol. 22, no. 10-11, pp. 941–954, 2003.

[8] M. Prats, P. Martinet, A. P. del Pobil, and S. Lee, "Vision force control in task-oriented grasping and manipulation," in *IEEE/RSJ International Conference on Robots and Intelligent Systems*, pp. 1320–1325, IEEE, 2007.

[9] N. Hogan, "Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications," *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, Mar. 1985.

[10] A. De Santis, V. Lippiello, B. Siciliano, and L. Villani, "Human-robot interaction control using force and vision," *Advances in Control Theory and Applications*, pp. 51–70, 2007.

[11] G. Morel, E. Malis, and S. Boudet, "Impedance based combination of visual and force control," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1743–1748, IEEE, 1998.

[12] D. J. Agravante, A. Cherubini, A. Bussy, and A. Kheddar, "Human-humanoid joint haptic table carrying task with height stabilization using vision," in *IEEE/RSJ International Conference on Robots and Intelligent Systems*, IEEE, 2013. to be published.

[13] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *International Conference on Advanced Robotics*, pp. 1–6, IEEE, 2009.

[14] A. Herdt, N. Perrin, P.-B. Wieber, *et al.*, "Walking without thinking about it," in *IEEE/RSJ International Conference on Robots and Intelligent Systems*, pp. 190–195, 2010.

[15] Y. Maeda, T. Hara, and T. Arai, "Human-robot cooperative manipulation with motion estimation," in *IEEE/RSJ International Conference on Robots and Intelligent Systems*, vol. 4, pp. 2240–2245, IEEE, 2001.

[16] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

[17] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*, vol. 2. Gatesmark Publishing Tennessee, 2009.

[18] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[19] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Incorporated, 2008.

[20] J. Hauser, S. Sastry, and P. Kokotovic, "Nonlinear control via approximate input-output linearization: The ball and beam example," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 392–398, 1992.

[21] A. Kheddar, "Human-robot haptic joint actions is an equal control-sharing approach possible?," in *4th International Conference on Human System Interactions (HSI)*, pp. 268–273, IEEE, 2011.

Bibliography

Agravante DJ, Cherubini A, Bussy A, Kheddar A (2013) Human-Humanoid Joint Haptic Table Carrying Task with Height Stabilization using Vision. In: IROS'13: International Conference on Robots and Intelligent Systems, Tokyo, Japan, p N/A, URL http://hal-lirmm.ccsd.cnrs.fr/lirmm-00857659

Agravante DJ, Cherubini A, Bussy A, Gergondet P, Kheddar A (2014) Collaborative Human-Humanoid Carrying Using Vision and Haptic Sensing. In: International Conference on Robotics and Automation (ICRA), IEEE, accepted for publication.

Bussy A, Gergondet P, Kheddar A, Keith F, Crosnier A (2012a) Proactive behavior of a humanoid robot in a haptic transportation task with a human partner. In: Ro-Man'2012: International Symposium on Robot and Human InteractiveCommunication, IEEE/RSJ, Université de Versaille, France, pp 962–967

Bussy A, Kheddar A, Crosnier A, Keith F (2012b) Human-humanoid haptic joint object transportation case study. In: IROS'12: International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal, vol 1, pp 3633–3638

Fox EB, Sudderth EB, Jordan MI, Willsky AS (2008) Nonparametric bayesian learning of switching linear dynamical systems. In: Advances in Neural Information Processing Systems, vol 21, pp 457–464

Fox EB, Sudderth EB, Jordan MI, Willsky AS (2009) Sharing features among dynamical systems with beta processes. In: Advances in Neural Information Processing Systems, vol 22, pp 549–557

Huang A, Wand MP (2013) Simple marginally noninformative prior distributions for covariance matrices. Bayesian Analysis 8(2):439–452

Hughes MC, Fox EB, Sudderth EB (2012) Effective split-merge monte carlo methods for nonparametric models of sequential data. In: Advances in Neural Information Processing Systems, vol 25, pp 1304–1312

Johansson RS, Theorin A, Westling G, Andersson M, Ohki Y, Nyberg L (2006) How a lateralized brain supports symmetrical bimanual tasks. PLoS Biol 4(6):e158

Li M, Yin H, Tahara K, Billard A (2014) Learning object-level impedance control for robust grasping and dexterous manipulation. In: Proceedings of the International Conference on Robotics and Automation (ICRA), accepted for publication.

Pais AL, Umezawa K, Nakamura Y, Billard A (2013) Task parametrization using continuous constraints extracted from human demonstrations. Submitted to IEEE Transactions on Robotics Under review.

Rosenberg A, Hirschberg J (2007) V-measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp 410–420

Shukla A, Billard A (2012) Coupled dynamical system based arm-hand grasping model for learning fast adaptation strategies. Robotics and Autonomous Systems 60(3):424 − 440