



HAL
open science

Co-manipulation with Multiple Probabilistic Virtual Guides

Gennaro Raiola, Xavier Lamy, Freek Stulp

► **To cite this version:**

Gennaro Raiola, Xavier Lamy, Freek Stulp. Co-manipulation with Multiple Probabilistic Virtual Guides. IROS 2015 - International Conference on Intelligent Robots and Systems , Sep 2015, Hamburg, Germany. pp.7 - 13 10.1109/IROS.2015.7353107 . hal-01170974

HAL Id: hal-01170974

<https://hal.science/hal-01170974v1>

Submitted on 5 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Co-manipulation with Multiple Probabilistic Virtual Guides

Gennaro Raiola^{1,2}

Xavier Lamy³

Freek Stulp^{1,2}

Abstract—In co-manipulation, humans and robots solve manipulation tasks together. Virtual guides are important tools for co-manipulation, as they constrain the movement of the robot to avoid undesirable effects, such as collisions with the environment. Defining virtual guides is often a laborious task requiring expert knowledge. This restricts the usefulness of virtual guides in environments where new tasks may need to be solved, or where multiple tasks need to be solved sequentially, but in an unknown order.

To this end, we propose a framework for *multiple probabilistic virtual guides*, and demonstrate a concrete implementation of such guides using kinesthetic teaching and Gaussian mixture models. Our approach enables non-expert users to design virtual guides through demonstration. Also, they may demonstrate novel guides, even if already known guides are active. Finally, users are able to intuitively select the appropriate guide from a set of guides through physical interaction with the robot. We evaluate our approach in a pick-and-place task, where users are to place objects at one of several positions in a cupboard.

I. INTRODUCTION

Industrial robots have had a crucial role in achieving very efficient assembly of products with a consistent, high quality. However, due to limitations of robots in terms of their adaptivity and flexibility, and difficulties in tracking and manipulating non-rigid objects, many assembly lines still require human workers. To exploit the best of both worlds, *co-manipulation* enables robots and humans to solve manipulation/assembly tasks together. One successful approach to enabling joint human-robot tasks has been the “virtual guiding fixture” approach [1], which constrains the motion of the robot along only certain task-relevant trajectories. An example of a “guiding fixture” from everyday life is the ruler, which allows us to draw very straight lines by constraining the movement of the pen tip along a 1-D trajectory on the 2-D paper.

Virtual guiding fixtures are especially useful in contexts where human decision making is still required to perform the overall task, but where constraints on the accuracy or required forces of the motion preclude humans from efficiently or effectively performing such tasks without robot assistance. Examples include industrial tasks [2], where heavy parts must be transported, and surgical tasks [3], [4], where accuracy and stability of the end-effector are essential. In this paper, we consider the task of placing objects in a cupboard, where guides assist the human in avoiding collisions with the shelves, as illustrated in Fig. 1.

¹Robotics and Computer Vision, Unité d’Informatique et d’Ingénierie des Systèmes, ENSTA ParisTech, Université Paris-Saclay, 828 bd Maréchaux, 91762 Palaiseau cedex France

²FLOWERS Research Team, INRIA Bordeaux Sud-Ouest, 33405 Talence, France

³CEA-List, Gif-sur-Yvette, France

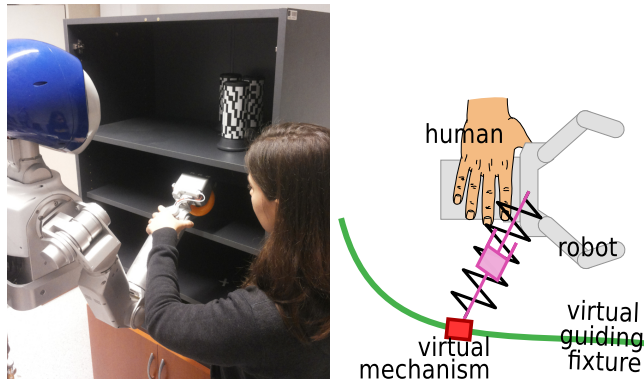


Fig. 1. Left: Using a virtual guiding fixture to facilitate the placement of objects in a cupboard with shelves. Right: Schematic illustration of modelling a virtual guiding fixture with a virtual mechanism [5]. Here, the virtual mechanism is implemented as a virtual cart on a rail. It is connected to the robot end-effector with a spring-damper system.

One issue in using virtual guides is addressing contexts in which multiple tasks are to be solved, or where new tasks may arise during operation. How can users intuitively demonstrate guides for novel tasks? And if a movements are constrained to a guide, how can users ‘escape’ the guide to demonstrate new trajectories? And if multiple tasks are to be solved, how can the user indicate which guide is relevant for the current task? And how should the robot detect this user choice?

We address these questions by making the following contributions: 1) Enabling users to *demonstrate virtual guides* through kinesthetic teaching, based on [6]. 2) Introducing *probabilistic virtual guides*, which enable users to ‘escape’ a virtual guide, and thus to demonstrate novel guides for novel tasks. 3) Using *multiple probabilistic virtual guides* for multiple tasks; the robot estimates the probability of each guide to interpret user intentions, and to choose the correct guide for the current task. 4) Evaluating these contributions with several users on a 7-DOF robotic arm, as illustrated in Fig. 1.

The rest of this paper is structured as follows. In the next section, we discuss related work. In III, we introduce a possible way to generate virtual guides by using virtual mechanisms as proposed in [5], which forms the basis of our work. We describe the general framework of multiple probabilistic virtual guides in Section IV, and a concrete implementation using Gaussian mixture models in Section V. We evaluate our approach with a humanoid robot in Section VI, and conclude with Section VII.

II. RELATED WORK

Virtual fixtures [7], [8] enforce virtual constraints on the movements of robots. In teleoperation for instance, virtual fixtures have been used to avoid robots falling into pits (with “forbidden regions virtual fixtures” [8]), or to constrain the movement of a robot along a preferred direction or trajectory with “virtual guiding fixtures” [9], which we sometimes refer to as ‘virtual guides’ or simply ‘guides’ in this paper.

Virtual guides are especially useful in contexts where human decision making is still required to perform the overall task, but where constraints on the accuracy or required forces of the motion preclude humans from performing such tasks without robot assistance. Virtual guides have been used for instance in industrial [2] and surgical [3] tasks. Colgate et al. provide an early overview of the use of intelligent assistive devices and virtual guides in industrial applications [10]. Yoon et al. [11] use inverse optimal control (IOC) to accomplish a steering task with a mobile robot, while Ryden et al. [12] use virtual guides to teleoperate an underwater robot.

The particular implementation of virtual guides we use is based on the work by Joly [5], where a passive virtual mechanism is connected to the robot tooltip by a spring-damper system in a teleoperation context. Moreover, the passivity of the system is proven when using one virtual guide. We extended this proof to multiple probabilistic virtual guides in [20]. Virtual mechanisms have also been used by Pezzementi et al. [13], where they are called “proxies”. Virtual guides may also be implemented by using anisotropic admittances to attenuate the non-preferred user force component [9], [14]. These methods require the sensing of external inputs by the user, such as the force or the velocity applied by the user on the robot tooltip. This is not required with our control scheme.

Virtual guides have often been limited to pre-defined geometric shapes [9] or combinations of shapes [15], [16]. Generating guides from demonstrations has also been explored by Aarno et al. [15], who model demonstrations in a segmented sequence of straight lines. In our work, we train Gaussian mixture models with user demonstrations, as in [6], which ensures smooth movements, and explicitly models the variance in user demonstrations.

One novel aspect of our work is to define *probabilistic* virtual guides. A first advantage of the probabilistic approach is that it enables a guide to be activated/deactivated based on the probability of belonging to it, which leads to smooth transitions. This is preferable to switching the guide on/off as in [15], [17], and does not require the manual design of distance thresholds for activation, as in [18]. A second advantage is that the probabilistic approach allows us to simultaneously activate and recognize several guides, by assigning probabilities to each guide based on user behavior. Thus, our method enables the use of a *library of guides*, with one guide for each distinct tasks. Multiple guides have been previously used, but these (sub)guides are activated sequentially for one unique task, rather than in parallel for several

tasks. For instance, Kuang et al. [16] combine different shape primitives to facilitate maze navigation. Aarno et al. [15] use a Hidden Markov Model to probabilistically choose a guide in a sequence of linear guides to accomplish a pick and place task. Finally, unlike many of previous works, we use virtual guides in a co-manipulation framework instead of teleoperation; in this aspect our work can be compared to [19] where the user and the robot have to execute a learned task together.

III. VIRTUAL MECHANISMS AS VIRTUAL GUIDES

We model the virtual guiding fixture using a virtual mechanism (VM) [5]. In the context of this paper, the VM can best be thought as a cart moving along a rail. The robot end-effector and virtual “cart” mechanism are coupled by a spring-damper system. If the robot end-effector moves, it pulls the cart along the rail. On the other hand, the cart also pulls the robot towards the rail, because the connection pulls in both directions. The overall effect is that the robot end-effector can be moved easily along the virtual rail, but not away from the rail, which constrains the movement of the robot such that tasks are more easily achieved.

The position of the cart on the rail in Cartesian space is described by \mathbf{x}_{vm} . The distance it has travelled along the rail is function of the phase s_{vm} , with $s_{vm} = 0$ at the beginning and $s_{vm} = 1$ at the end of the rail, as illustrated in Fig. 2. The kinematics of the virtual mechanism is described by:

$$\mathbf{x}_{vm} = f(s_{vm}) \quad (1)$$

$$\dot{\mathbf{x}}_{vm} = \mathbf{J}_{vm}(s_{vm})\dot{s}_{vm} \quad (2)$$

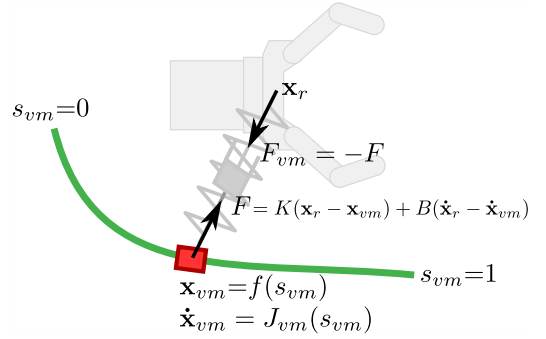


Fig. 2. The main variables and equations of the virtual mechanism.

A. Force on the virtual mechanism

The virtual mechanism is connected to the robot end-effector with a virtual spring-damper system. The force applied to the virtual mechanism by the robot is:

$$\mathbf{F}_r = K(\mathbf{x}_r - \mathbf{x}_{vm}) + B(\dot{\mathbf{x}}_r - \dot{\mathbf{x}}_{vm}). \quad (3)$$

The virtual mechanism is ideal, so the efforts applied on it are null

$$\mathbf{J}_{vm}^\top \mathbf{F}_r = 0, \quad (4)$$

which leads leads to

$$\mathbf{J}_{vm}^\top (K(\mathbf{x}_r - \mathbf{x}_{vm}) + B(\dot{\mathbf{x}}_r - \mathbf{J}_{vm}\dot{s}_{vm})) = 0. \quad (5)$$

By solving (5) with respect to \dot{s}_{vm} , we obtain a first order dynamical system that expresses the evolution of the virtual cart along the virtual rail:

$$\dot{s}_{vm} = (\mathbf{J}_{vm}^\top B \mathbf{J}_{vm})^{-1} \mathbf{J}_{vm}^\top (K(\mathbf{x}_r - \mathbf{x}_{vm}) + B\dot{\mathbf{x}}_r). \quad (6)$$

Moving the robot end-effector away from the virtual cart ($\mathbf{x}_r \neq \mathbf{x}_{vm}$) will thus make it slide along the rail, with a velocity described by (6).

B. Force on the robot end-effector

Because the virtual mechanism and the robot end-effector are connected to *each other*, the virtual mechanism also applies a force on the robot end-effector, i.e.

$$\mathbf{F}_{vm} = -\mathbf{F}_r = K(\mathbf{x}_{vm} - \mathbf{x}_r) + B(\dot{\mathbf{x}}_{vm} - \dot{\mathbf{x}}_r). \quad (7)$$

This virtual force can be transformed into actual control commands for the robot, for instance with a compliance controller. In our implementation, we used the Jacobian of the robot \mathbf{J}_r to convert the forces into torque references for the motor controllers.

IV. MULTIPLE PROBABILISTIC VIRTUAL GUIDES

Consider a scenario in which multiple guides are active, i.e. the robot end-effector is connected to multiple virtual mechanisms, as illustrated in Fig. 3. This scenario applies when there are multiple tasks, i.e. transporting an object to one of multiple possible positions. In this section, we propose three modes of interaction when multiple guides are active, based on probabilistic virtual mechanisms.

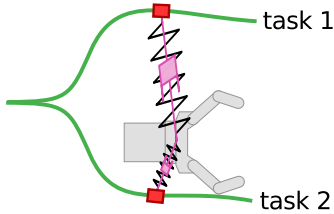


Fig. 3. Multiple virtual mechanisms – one for each task – simultaneously connected to the robot end-effector.

A. Probabilistic Virtual Mechanisms

We define a “probabilistic virtual mechanism” to be a VM for which the position of the cart is modeled by a multivariate normal distribution

$$\mathbf{x}_{vm} \sim \mathcal{N}(\boldsymbol{\mu}_{vm}, \boldsymbol{\Sigma}_{vm}) \quad (8)$$

$$\boldsymbol{\mu}_{vm} = f(s_{vm}) \quad \text{cf. (1)} \quad (9)$$

$$\boldsymbol{\Sigma}_{vm} = v(s_{vm}) \quad (10)$$

In Section V-B, we describe how the functions (9) and (10) are learned from demonstrated trajectories. This distribution has the probability density function

$$g(\mathbf{x}; \boldsymbol{\mu}_{vm}, \boldsymbol{\Sigma}_{vm}) = \frac{e^{(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{vm})^\top \boldsymbol{\Sigma}_{vm}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{vm}))}}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}_{vm}|}}. \quad (11)$$

Because $\boldsymbol{\mu}_{vm}$ and $\boldsymbol{\Sigma}_{vm}$ are completely specified by the phase through the functions in (9) and (10), we also define the abbreviation

$$g(\mathbf{x}; s_{vm}) = g(\mathbf{x}; f(s_{vm}), v(s_{vm})). \quad (12)$$

B. Three interaction modes

When using multiple virtual guides, an assignment problem arises: which guide is currently responsible for guiding the human? Probabilistic virtual guides enable us to address this question by weighting the contribution of each guide with the probability that this is the guide chosen by the human. We define three weighting schemes for three different modes of interaction:

- 1) Hard virtual guides. This weighting scheme enforces that the user is constrained to the guides, and cannot escape them. This is useful in scenarios where no novel tasks arise, and the user chooses to *exploit* only the current guides.
- 2) Soft virtual guides. With this weighting scheme, the user can ‘escape’ the virtual guides, for example to *train* novel guides. When the end-effector is not close to any of the guides, none of the guides is active. The robot then operates in zero-gravity mode.
- 3) Zero virtual guides. This turns off all of the virtual guides, and is equivalent to operating the robot in zero-gravity mode (a.k.a. gravity compensation mode). This mode may also be used to *train* novel guides.

Table I summarizes the most relevant differences between the three interaction modes, from the user’s perspective.

	Type of guide		
	Zero	Soft	Hard
User can demonstrate new guides?	Yes	Yes	No
Guides enabled when close to guide?	No	Yes	Yes
Guides enabled when far from guide?	No	No	Yes

TABLE I

MAIN FUNCTIONALITIES FOR THE THREE INTERACTION MODES.

1) *Hard virtual guides*: If we have N virtual mechanisms, there are N cart positions $\mathbf{x}_{vm}^{n=1:N}$, and N probabilities. The probability that the n^{th} cart is responsible for guiding the end-effector at position \mathbf{x}_r is

$$p(n; \mathbf{x}_r, s_{vm}^n) = \frac{g(\mathbf{x}_r; \boldsymbol{\mu}_{vm}^n, \boldsymbol{\Sigma}_{vm}^n)}{\sum_{i=1}^N g(\mathbf{x}_r; \boldsymbol{\mu}_{vm}^i, \boldsymbol{\Sigma}_{vm}^i)} = \frac{g(\mathbf{x}_r; s_{vm}^n)}{\sum_{i=1}^N g(\mathbf{x}_r; s_{vm}^i)}, \quad (13)$$

where the means and covariance matrices of the cart position are determined from the cart phase s_{vm} with (9) and (10) respectively.

Each of the N virtual mechanisms applies a force \mathbf{F}_{vm}^n to the end-effector. The relative influence of each VM is scaled with the probability $p(n; \mathbf{x}_r, s_{vm}^n)$, so that the resultant force on the end-effector is:

$$\mathbf{F}_{res} = \sum_{n=1}^N p(n; \mathbf{x}_r, s_{vm}^n) \cdot \mathbf{F}_{vm}^n. \quad (14)$$

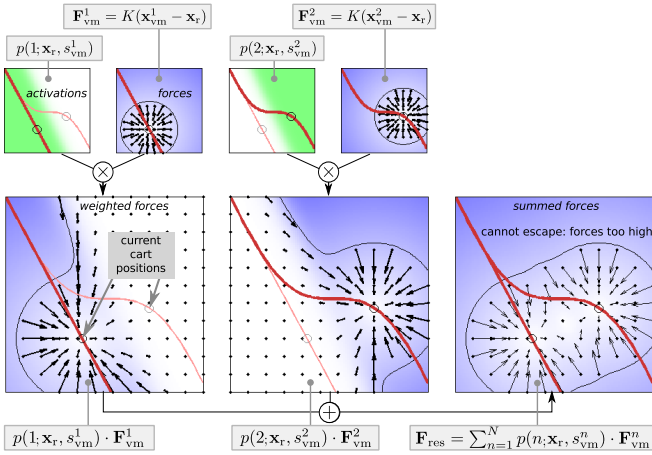


Fig. 4. Scaling the forces of multiple virtual guides with probabilistic virtual mechanisms, using the first interaction mode (hard virtual guides). Each graph represents the value of one equation, depicted above/below the graphs. These illustrations are based on synthetic data. Since they represent a static snapshot, damping terms are omitted for simplicity.

The relevant probabilities and forces in this control mode are visualized in Fig. 4. The stability of hard virtual guides is analyzed in [20].

2) *Soft virtual guides*: The underlying assumption in using the hard virtual guides is that \mathbf{x}_r *must* belong to one of the VMs. Another approach is to assume that if \mathbf{x}_r is too far from the VMs, it does not belong to any of the VMs. To do so, we use a Gaussian function $h(\mathbf{x}_r; \boldsymbol{\mu}_{vm}, \boldsymbol{\Sigma}_{vm})$ with mode 1, i.e. a probability density function as in (11), but without the normalization factor $\sqrt{(2\pi)^k |\boldsymbol{\Sigma}_{vm}|}$. By using these weights (to determine if an *individual* virtual guide is active in the first place), as well as the probability $p(n; \mathbf{x}_r, s^n_{vm})$ (to determine the relative weighting *between all the guides*), the resultant force becomes

$$\mathbf{F}_{res} = \sum_{n=1}^N h(\mathbf{x}_r; s^n_{vm}) \cdot p(n; \mathbf{x}_r, s^n_{vm}) \cdot \mathbf{F}_{vm}^n. \quad (15)$$

The main difference between these methods arises when \mathbf{x}_r is not close to any \mathbf{x}^n_{vm} , as illustrated in Fig. 5. With the hard virtual guides, \mathbf{x}_r is always pulled towards the closest \mathbf{x}^n_{vm} ; the further you are, the stronger the force (standard PD control). With the soft virtual guides, \mathbf{x}_r does not belong to any \mathbf{x}^n_{vm} , and $\mathbf{F}_{vm}^k = 0$ for all n . The resultant force is therefore also zero.

3) *Zero virtual guides*: Finally, a second training mode in which only gravity-compensation is active is provided. Within our probabilistic framework, this training mode is interpreted as all virtual guides (if any) having a probability 0 of being responsible, i.e. $\forall n, p(n; s^n_{vm}) = 0$

V. IMPLEMENTATION

To enable the use of probabilistic virtual guides, the functions $\boldsymbol{\mu}_{vm} = f(s_{vm})$ (9) and $\boldsymbol{\Sigma}_{vm} = v(s_{vm})$ (10) must be implemented. In this section, we describe one possible implementation, based on kinesthetic teaching of demonstration trajectories, trajectory clustering, and the modelling of trajectory clusters with Gaussian mixture models [6].

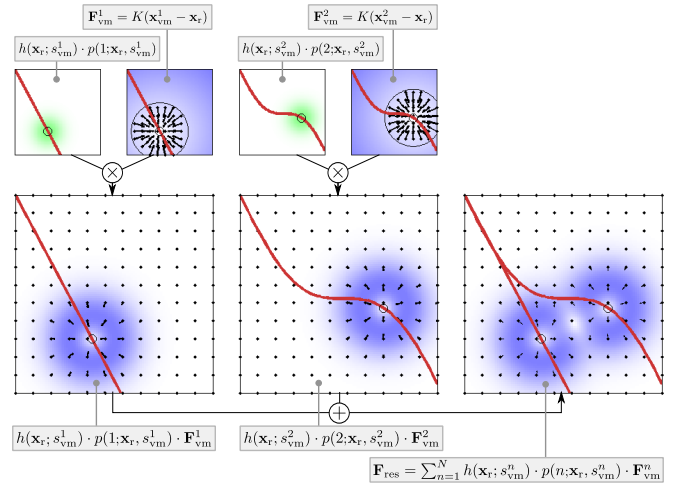


Fig. 5. Scaling the forces of multiple virtual guides with probabilistic virtual mechanisms, using the second interaction mode (soft virtual guides).

A. Training the Gaussian Mixture Model

Each task is performed multiple times by the user through kinesthetic teaching, i.e. the user holds the end-effector of the robot, and demonstrated by guiding it along the desired trajectory. The demonstrations are gathered without a specific order, in this way the number of tasks is not known a priori. Then dynamic time warping [6] is used to compute distances between trajectories that are independent of time. These distances are used to perform hierarchical clustering of the trajectories. An example of clustering on data from the placement tasks (cf. Fig. 1) is shown in Fig. 6.

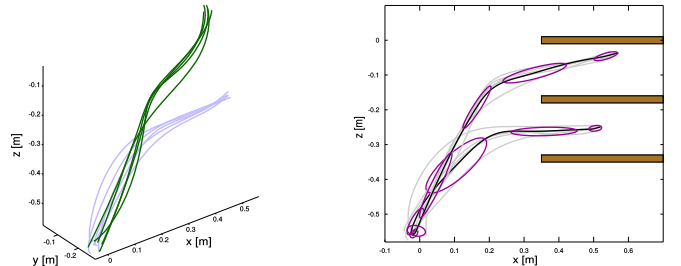


Fig. 6. Left: Clusters representing two different movements for placing objects in shelves. Right: Gaussian mixture models for the clustered data. Training trajectories are light gray, the mean of the GMM is black. For visualization purposes, the GMM is projected on the xz -plane.

Each sample $[x(t_i), y(t_i), z(t_i)]_{i=1:N}$ in a trajectory is associated with a phase value $s(t_i) = (t_i - t_1)/(t_N - t_1)$, i.e. $s(t_1) = 0$ at the beginning of the trajectory, and $s(t_N) = 1$ at the end. The last step is to fit a Gaussian mixture model (GMM) to all the 4-D data points (3D position *and* phase) in all the trajectories in each of the clusters. In a GMM, data is represented by a set of E multivariate Gaussians

$$p(\mathbf{r}) = \sum_{e=1}^E \pi_e \mathcal{N}(\mathbf{r}; \boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e), \text{ with } \sum_{e=1}^E \pi_e = 1. \quad (16)$$

The Expectation-Maximization algorithm adjusts the priors π_e and the parameters $\boldsymbol{\mu}_e$ and $\boldsymbol{\Sigma}_e$ of the Gaussian functions

that define this model [21]. Fig. 6 depicts the result of fitting a GMM to the clusters, where the number of Gaussians is manually set to 5 per cluster. In future work, we will consider methods to set this parameter automatically [22].

B. GMMs as Probabilistic Virtual Mechanisms

Probabilistic virtual mechanisms require implementations of the functions $\mathbf{x}_{vm} = f(s_{vm})$ (1), $\dot{\mathbf{x}}_{vm} = \mathbf{J}_{vm}(s_{vm})$ (2), and $\Sigma_{vm} = v(s_{vm})$ (10). We implement these functions through Gaussian mixture regression (GMR) [6], based on the Gaussian mixture model (GMM) trained in the previous section.

In the context of a virtual mechanism, the input space is S , and the output space is X , corresponding to the phase s_{vm} and virtual mechanism position \mathbf{x}_{vm} respectively. Given this partition, the mean and covariance matrix¹ are decomposed as

$$\boldsymbol{\mu}_e = [\boldsymbol{\mu}_{e,S}^\top, \boldsymbol{\mu}_{e,X}^\top]^\top \text{ and } \boldsymbol{\Sigma}_e = \begin{bmatrix} \boldsymbol{\Sigma}_{e,S} & \boldsymbol{\Sigma}_{e,SX} \\ \boldsymbol{\Sigma}_{e,XS} & \boldsymbol{\Sigma}_{e,X} \end{bmatrix}, \quad (17)$$

The implementation of $\mathbf{x}_{vm} = f(s_{vm})$ in (1) corresponds to computing $\bar{\mathbf{x}}_{vm} = E(\mathbf{x}_{vm}|s_{vm})$, i.e. the expectation of \mathbf{x}_{vm} given the input s_{vm} :

$$\bar{\mathbf{x}}_{vm} = \sum_{e=1}^E \beta_e(s_{vm})(\boldsymbol{\mu}_{e,X} + \boldsymbol{\Sigma}_{e,XS} \boldsymbol{\Sigma}_{e,S}^{-1}(s_{vm} - \boldsymbol{\mu}_{e,S})), \quad (18)$$

with:

$$\beta_e(s_{vm}) = \frac{\pi_e g(\mathbf{x}; \boldsymbol{\mu}_{e,S}, \boldsymbol{\Sigma}_{e,S})}{\sum_{l=1}^E \pi_l g(\mathbf{x}; \boldsymbol{\mu}_{l,S}, \boldsymbol{\Sigma}_{l,S})} = \frac{\pi_e g(\mathbf{x}; s_{vm}^e)}{\sum_{l=1}^E \pi_l g(\mathbf{x}; s_{vm}^l)}. \quad (19)$$

The function $\mathbf{J}_{vm}(s_{vm})$ in (2) is implemented with the analytical derivative of (18). Finally, Σ_{vm} in (2) is implemented by computing the variance in the estimate $\text{var}(\bar{\mathbf{x}}_{vm})$ with

$$\Sigma_{vm} = \sum_{e=1}^E \beta_e(s_{vm})^2 \left(\boldsymbol{\Sigma}_{e,X} - \boldsymbol{\Sigma}_{e,XS} \boldsymbol{\Sigma}_{e,S}^{-1} \boldsymbol{\Sigma}_{e,XS}^\top \right). \quad (20)$$

VI. PILOT STUDY

Our application task, illustrated in Fig. 1, is to place objects on different shelves in a cupboard. During one episode, users take the robot at the wrist, and guide it to one of two positions in the cupboard. At the desired position, the robot releases the object for placement on the shelf (when the phase of the movement > 0.9). Although using heavy objects would be better demonstrate the advantages of using virtual guides, we cannot do so due to the limited payload of our robot.

In this pilot study, we first illustrate two advantages of probabilistic virtual guides over standard guides. We then show a typical use case for multiple probabilistic guides. With four human subjects, we finally compare single and multiple virtual guides with respect to 3 evaluation criteria.

¹The covariance matrix $\boldsymbol{\Sigma}_{e,S}$ is actually a scalar, because the phase is always 1-dimensional. For consistency, we nevertheless use the bold symbol $\boldsymbol{\Sigma}$ rather than σ^2 .

A. Advantage: Switching between multiple guides

In Fig. 7, we illustrate switching between multiple virtual guides by using the first interaction mode (hard virtual guides) to scale the forces, as described in Section IV-B.1. In the left graph, we see that at the beginning of the trajectory, when the end-effector is close to both of the guides, the probabilities of both guides ≈ 0.5 . Because the guides are close to each other, it is so easy to switch between them that it is difficult to feel the transition.

In the middle plot, a switch is attempted half-way through the movement, when the probability that is the upper trajectory is 1. But with enough force, the user can still change to the other guide. The sensation is that the robot “gives way”, and then locks into the other guide. At this point, we recommend viewing the video supplement. In it, we show a user reproducing the cases explained in Fig. 7 and also those in Fig. 8. Although sequential switching between multiple virtual (sub)guides for one task has been demonstrated [15], [16], the switching between multiple virtual guides (that are active in parallel) for multiple tasks is a novel feature of our approach.

Even further along the movements however (right plot), the distance between the guides is too large. Pushing the end-effector downwards hardly influences the probabilities, and the end-effector is locked to the upper guide.

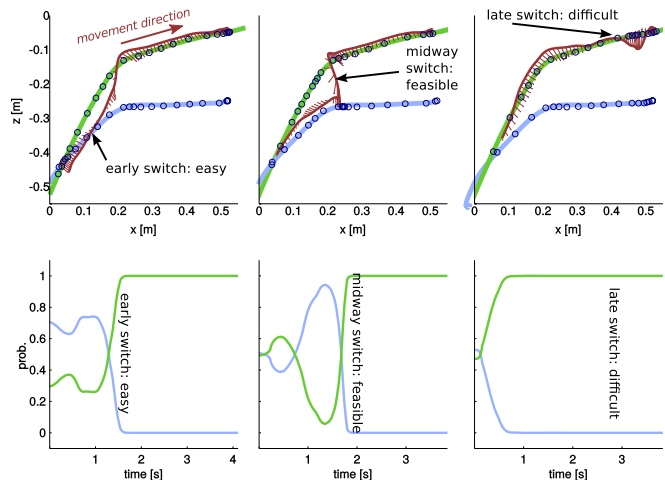


Fig. 7. Illustration of switching between multiple probabilistic virtual guides. Top: Side view of virtual guides and end-effector movement. Bottom: probabilities of the two virtual guides. The dark red line represents the user movement the thin short lines represent the direction and magnitude of the resultant force applied by the virtual mechanisms.

B. Advantage: Escaping an active guide

In Fig. 8, we illustrate how probabilistic virtual guides enable a user to escape the guide in a transparent way by using the second interaction mode (Soft virtual guides), as described in Section IV-B.2. Here, only the virtual guide for positioning the object on the lower shelf is active. With a standard virtual guide (left graph), the user cannot escape the virtual guide, as the spring will exert high forces to pull the end-effector back to the guide. When using soft virtual

guides (as in Fig. 5), forces decrease if the distance to the guide becomes large. Once the the user has ‘escaped’ the virtual guide, demonstrations for novel tasks can be given, as illustrated in the video supplement. The advantage of using probabilistic virtual guides is that it leads to smooth transitions (rather than on/off switching as in [15], [17]), and does not require the manual design of distance thresholds (as in [18]).

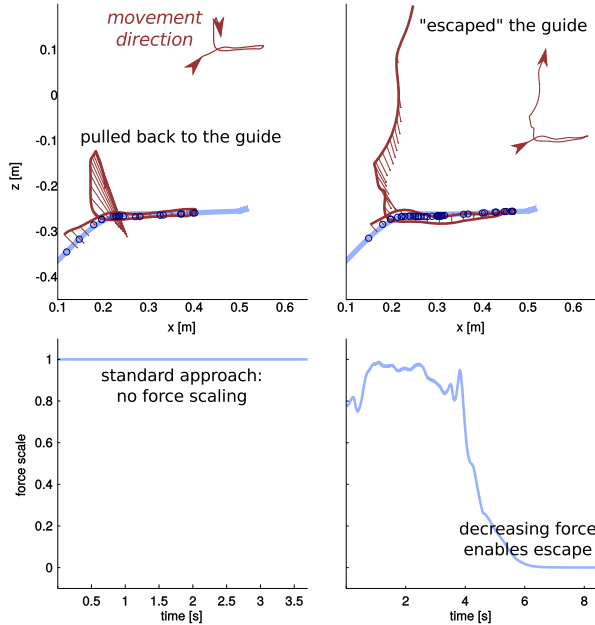


Fig. 8. Illustration of escaping a virtual guide. Top: Side view of virtual guide and end-effector movement. Bottom: Probabilities of the virtual guides.

C. Typical Use Case Illustration

We now illustrate how escaping active guides and switching between multiple guides combine in a typical use case where novel tasks arise during operation. Fig. 9 shows the force scaling of virtual guides during 30 task executions with the same subject. The use case is described in detail in the caption of Fig. 9 to allow easy switching between the graph and its description.

D. Comparing Safety and Efficiency.

We perform a pilot study with four users, and the set-up in Fig. 1. We compare three assistance modes: gravity compensation only (no virtual guides), a single virtual guide (for the current task), or multiple virtual guides (by using hard virtual guides). With these 3 modes, each user executes 10 episodes for each of the 2 tasks (object goal positions), i.e. a total of $60 = 3 \times 10 \times 2$ episodes per user. Each user thus tries all modes and positions, but were presented in a randomized order to the users to avoid training effects.

The measures that we are interested in are: • execution time, to measure efficiency. • accuracy of tracking • actually observed collisions, to measure safety. These results are summarized in Fig. 10

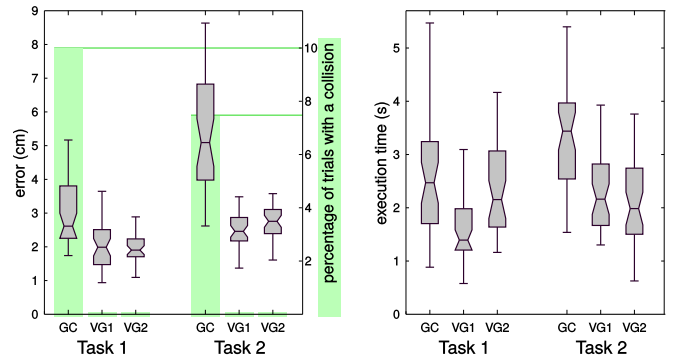


Fig. 10. Comparison of the three assistance modes (gravity compensation only, only one virtual guide, multiple virtual guides), for both guides (for the upper and lower shelf), and three measures (execution time, position error, number of collisions)

From these results, we draw the following conclusions: A) Tracking errors substantially (from 3.0cm to 2.0/2.0 for Task 1 and from 5.2 to 2.5/2.7 for Task 2) and significantly decrease when using virtual guides. B) This enables virtual guides reduce the percentages of trials in which a collision occurs to 0. C) Execution times also decrease (from 2.7s to 1.7/2.3 for Task 1 and from 3.3 to 2.3/2.1 for Task 2). D) Using multiple guides in parallel (instead of activating only the appropriate guide for the task at hand) does not lead to a significant deterioration in performance.

Some of the spontaneous user comments about the virtual guides, although anecdotal, are nevertheless interesting to report. One user reported “I feel more confident in moving the robot”; this confidence is verified by the fact that users are able to execute tasks more quickly, because they are more confident that the robot will not collide with the cupboard. Another comment includes “It feels like there are virtual shelves”. Thus the user interprets the robot as avoiding the shelves, rather than following a guide, which corresponds to what the demonstrator considers when demonstrating the trajectories.

VII. CONCLUSION

We propose probabilistic virtual guides for co-manipulation. These guides enable smooth deactivation of guides without manual thresholds (experiment VI-B). Furthermore, they enable the recognition of user intentions for multiple tasks (experiment VI-C), and transparent switching between multiple guides that are active in parallel during movement execution (experiment VI-A). Finally, we show that probabilistic guides improve the safety and efficiency of task completion, and that multiple guides do not deteriorate this performance (experiment VI-D). Furthermore, we define different interaction modes for multiple guides, based on using hard, soft, or zero virtual guides. Our implementation of probabilistic virtual guides is based on proven methods for kinesthetic teaching, clustering and probabilistic modelling with Gaussian mixture models [6].

Adding more and more virtual guides for an abundance of tasks has diminishing returns. For example, with 100

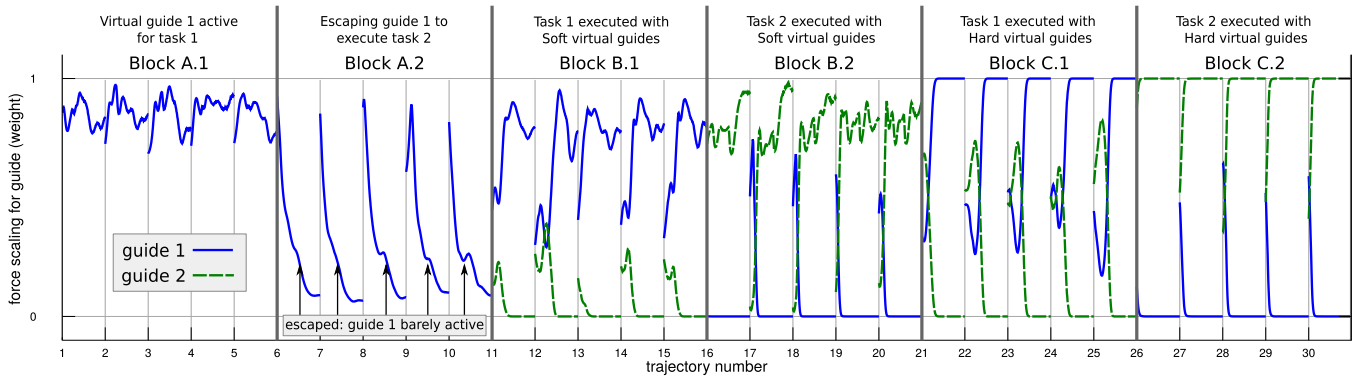



Fig. 9. Use case illustrating how escaping virtual guides enables on-the-fly generation of novel guides. In block A.1 the user executes task 1 (place the object on the lower shelf) using the previously trained guide 1. In block A.2 the user is able to execute task 2 – for which there is not yet a guide – by escaping guide 1. The trajectories from block A.2 are not considered to belong to guide 1 (see the low weights in block A.2), and are stored to train a new guide (guide 2), using the clustering and training methods from Section V. From then onwards, there are two guides. In block B.1 and B.2 (where task 1 and 2 are executed, respectively), we see that the correct guide is recognized during the movement, and used to guide the human. Escaping the guides would still be possible, but not necessary since no novel task has to be solved. Finally, in block C.1 and C.2, it is assumed that no further tasks will arise, and the interaction mode is (manually) switched to provide hard virtual guides, which leads to earlier and even more accurate recognition of the appropriate guide for the task.

different object positions on the shelves and 100 guides, there would be so many guides active at the same time that the resultant forces would be zero, corresponding to zero-gravity mode. Increasing the overall stiffness of the robot could alleviate this effect and allow more guides to be introduced, but may make switching between guides more difficult. Our future work is aimed at investigating these diminishing returns, and there relation to the task domain and stiffness of the robot.

Acknowledgements

This project has received funding from DIGITEO  (www.digiteo.fr).

REFERENCES

- [1] H. C. Lin, P. Marayong, K. Mills, R. Karam, P. Kazanzides, A. M. Okamura, and G. D. Hager, "Portability and applicability of virtual fixtures across medical and manufacturing tasks," in *ICRA*, 2006.
- [2] Q. Li, "Virtual reality for fixture design and assembly," Ph.D. dissertation, University of Nottingham, 2008.
- [3] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *Int'l Conference on Medical Image Computing and Computer-Assisted Intervention*, 2001, pp. 1419–1420.
- [4] B. C. Becker, R. A. Maclachlan, L. A. Lobes, G. D. Hager, and C. N. Riviere, "Vision-based control of a handheld surgical micromanipulator with virtual fixtures," *IEEE Trans Robot*, 29(3), 674–683, 2013.
- [5] L. Joly and C. Andriot, "Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism," in *ICRA*, 1995.
- [6] 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*, vol. 37, no. 2, pp. 286–298, 2007.
- [7] L. Rosenberg, "Virtual fixtures: perceptual tools for telerobotic manipulation," in *Proc. IEEE Virtual Reality Int'l Symposium*, 1993.
- [8] J. J. Abbott and A. M. Okamura, "Virtual fixture architectures for telemanipulation," 2003.
- [9] P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures," in *ICRA*. IEEE, 2003, pp. 1954–1959.
- [10] J. E. Colgate, M. A. Peshkin, and S. H. Klostermeyer, "Intelligent assist devices in industrial applications: a review," in *IROS*, 2003.
- [11] H. Yoon, R. Wang, and S. Hutchinson, "Modeling user's driving-characteristics in a steering task to customize a virtual fixture based on task-performance," in *Robotics and Automation (ICRA), 2014 IEEE Int'l Conference on*, May 2014, pp. 625–630.
- [12] F. Ryden, A. Stewart, and H. Chizeck, "Advanced telerobotic underwater manipulation using virtual fixtures and haptic rendering," in *Oceans - San Diego, 2013*, Sept 2013, pp. 1–8.
- [13] Z. Pezzementi, G. D. Hager, and A. M. Okamura, "Dynamic guidance with pseudo-admittance virtual fixtures," in *IEEE Int'l Conference on Robotics and Automation*, 2007, pp. 1761–1767.
- [14] A. Bettini, P. Marayong, S. Member, S. Lang, A. M. Okamura, and G. D. Hager, "Vision assisted control for manipulation using virtual fixtures," in *IROS*, 2004, pp. 1171–1176.
- [15] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *ICRA*, 2005, pp. 897–903.
- [16] A. Kuang, S. Payandeh, B. Zheng, F. Henigman, and C. MacKenzie, "Assembling virtual fixtures for guidance in training environments," in *HAPTICS*, 2004.
- [17] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalet, "Telemanipulation assistance based on motion intention recognition," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE Int'l Conference on*, April 2005, pp. 1121–1126.
- [18] J. T. Nolin, P. M. Steniski, and A. M. Okamura, "Activation cues and force scaling methods for virtual fixtures," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003.
- [19] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *Proceedings of 2014 IEEE Int'l Conference on Robotics and Automation (ICRA)*, 2014.
- [20] G. Raiola, P. Rodriguez-Ayerbe, X. Lamy, S. Tliba, and F. Stulp, "Parallel guiding virtual fixtures: Control and stability," in *IEEE Multi-Conference on Systems and Control (MSC)*, 2015.
- [21] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems 6*, 1993, pp. 120–127.
- [22] J. W. Yoon, "An efficient model selection for gaussian mixture model in a bayesian framework," *CoRR*, vol. abs/1307.0995, 2013.