Parallel Guiding Virtual Fixtures: Control and Stability

Gennaro Raiola, Pedro Rodriguez-Ayerbe, Xavier Lamy, Sami Tliba, Freek
Stulp

Parallel Guiding Virtual Fixtures: Control and Stability

Gennaro Raiola[1,2]   Pedro Rodriguez-Ayerbe[3]   Xavier Lamy[4]   Sami Tliba[3]   Freek Stulp[1,2]

Abstract— **Guiding virtual fixtures have been proposed as a method for human-robot co-manipulation. They constrain the motion of the robot to task-relevant trajectories, which enables the human to execute the task more efficiently, accurately and/or ergonomically. When sequences of different tasks must be solved, multiple guiding virtual fixtures are required, and the appropriate guide for the current task must be detected automatically. To this end, we propose a novel control scheme for multiple guiding virtual fixtures that are active in parallel. Furthermore, we determine under which conditions using multiple fixtures is stable. Finally, we perform a pilot study for a real-world application with a humanoid robot.**

I. INTRODUCTION

A *guiding virtual fixture* constrains robot motion along task-relevant trajectories [1], [2]. Such fixtures make it easy to move the robot end-effector along a trajectory, but not in other directions away from the fixture. They 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 [3], where heavy parts must be transported, and surgical tasks [4], [5], where accuracy and stability of the end-effector are essential.

Joly [1] proposed to implement guiding virtual fixtures as *virtual mechanisms*, where a passive virtual mechanism is connected to the robot tooltip by a spring-damper system. This mechanism can be, for instance, another virtual robot, or a cart moving along a rail, as illustrated in Fig. 1. The robot end-effector and virtual "cart" mechanism are coupled by a spring-damper system.

The overall scheme is reminiscent of Victorian waterway transport, when horses pulled boats along canals with ropes. The horse (the robot end-effector) can pull the boat (the virtual mechanism) along the canal (the guide) with the rope (the spring-damper system). And the boat and rope constrain the horse so that it cannot walk away from the canal.

One issue in using virtual guides is addressing contexts in which multiple tasks are to be solved. For multiple tasks, multiple guides are required. And with multiple guides, it is necessary to have methods for recognizing and selecting the appropriate guide for a particular task. For safety reasons, it
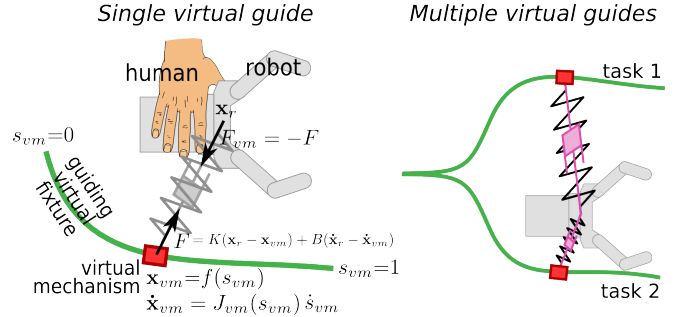
Fig. 1. Left: Schematic illustration of modelling a guiding virtual fixture with a virtual mechanism [1]. Here, the virtual mechanism is implemented as a virtual cart on a rail. It is connected to the robot end-effector with a spring-damper system. Main variables and equations of the virtual mechanism are included. Right: Multiple virtual mechanisms – one for each task – simultaneously connected to the robot end-effector.

must be shown that using multiple guides in parallel does not lead to instabilities. The contributions of this paper are:

- to propose a control framework for multiple virtual guides, based on virtual mechanisms.
- to prove the stability of multiple virtual mechanisms that are active in parallel[1].

To illustrate the practical relevance of these theoretical contributions to real-world applications, we include in this paper a pilot study with a humanoid robot, which has previously been reported in [6].

The rest of this paper is structured as follows. In the next section, we discuss related work. In Section III, we then present how guiding virtual fixtures are implemented as virtual mechanisms. The control framework and stability proof for multiple guides are presented in Section IV and Section V respectively. A pilot study is presented in Section VI, after which we conclude with Section VII

II. RELATED WORK

The "virtual fixture" method [7], [8], is a successful approach to joint human-robot tasks. This method has been used especially in the context of teleoperation, where virtual constraints are used to improve the precision of a teleoperated robot. Virtual fixtures can be also used to avoid sending wrong commands to a teleoperated robot, an example is a mobile robot that does not accept teleoperation commands that would make it fall into a pit. Such virtual fixtures are known also as "forbidden regions virtual fixtures" [9].

Guiding virtual fixtures on the other hand allow the operator to control the motion of the robot along a preferred direction (e.g. along a pre-specified trajectory) whilst constraining

[1]The passivity of single virtual mechanisms was shown in [1].

it in the non-preferred directions [8] (e.g. not allowing the robot to move too far away from the pre-specified trajectory). In guiding virtual fixtures, the compliance along the trajectory is high, whereas it is low in directions perpendicular to the trajectory. This direction guidance thus pulls the user towards the trajectory, which is what makes it easier to move along the trajectory. Pezzementi et al. [10] add speed guidance too, to achieve desired tool velocities during the movement.

Colgate et al. [11] provide an early overview of the use of intelligent assistive devices and virtual guides in industrial applications.

Apart from the work of Joly [1] discussed in the introduction, virtual mechanisms have also been used by Pezzementi et al. [10], where they are called "proxies". Virtual guides may also be implemented by using anisotropic admittances to attenuate the non-preferred user force component [8], [12] 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 [8] or combinations of shapes [13], [14]. Generating guides from demonstrations has also been explored by Aarno et al. [13], who model demonstrations in a segmented sequence of straight lines.

One novel aspect of our work is to provide a control scheme in which the output of individual virtual guides are weighted within a probabilistic framework. 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 [13], [15], [16], and does not require the manual design of distance thresholds for activation, as in [17].

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 task. 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. [14] combine different shape primitives to facilitate maze navigation. Aarno et al. [13] use a Hidden Markov Model to probabilistically choose a guide in a sequence of linear guides to accomplish a pick and place task.

Another important aspect of our work is to prove the stability of the controller. In [1] the passivity for a single virtual mechanism is proven by studying its dissipated energy. Intuitively a virtual mechanism is passive because it is realized with passive mechanical components such as springs and dampers. In our case, due the non-linearity of the controller we use the Lyapunov theory for the stability analysis as explained in [18].

III. Virtual Mechanisms as Virtual Guides

We model the guiding virtual fixture using a virtual mechanism (VM) [1]. In the context of this paper, the VM can best be thought as a cart moving along a rail. The position of the cart on the rail in Euclidean space is described by $\mathbf{x}_{\text{vm}}$. The distance it has travelled along the rail is function of the phase s_{vm}, with $s_{\text{vm}} = 0$ at the beginning and $s_{\text{vm}} = 1$ at the end of the rail, as illustrated in Fig. 1. The kinematics of the virtual mechanism is described by:

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

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

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:

$$\mathbf{F}_{\text{r}} = k(\mathbf{x}_{\text{r}} - \mathbf{x}_{\text{vm}}) + b(\dot{\mathbf{x}}_{\text{r}} - \dot{\mathbf{x}}_{\text{vm}}). \tag{3}$$

Where $k, b > 0$ are respectively the stiffness and the damping gains and $\mathbf{x}_{\text{r}}$ represents the robot position in the workspace. The virtual mechanism is ideal, so the force exerted on the virtual mechanism tip is always orthogonal to its velocity:

$$\mathbf{J}_{\text{vm}}^{\mathsf{T}}\mathbf{F}_{\text{r}} = 0, \tag{4}$$

meaning that no power can be transmitted to it nor taken from it, this leads to:

$$\mathbf{J}_{\text{vm}}^{\mathsf{T}}(k(\mathbf{x}_{\text{r}} - \mathbf{x}_{\text{vm}}) + b(\dot{\mathbf{x}}_{\text{r}} - \mathbf{J}_{\text{vm}}\dot{s}_{\text{vm}})) = 0. \tag{5}$$

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

$$\dot{s}_{\text{vm}} = (\mathbf{J}_{\text{vm}}^{\mathsf{T}}b\mathbf{J}_{\text{vm}})^{-1}\mathbf{J}_{\text{vm}}^{\mathsf{T}}(k(\mathbf{x}_{\text{r}} - \mathbf{x}_{\text{vm}}) + b\dot{\mathbf{x}}_{\text{r}}). \tag{6}$$

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

To avoid singularities in (6) we add an adaptive damping term to the dynamical system:

$$\dot{s}_{\text{vm}} = (\mathbf{J}_{\text{vm}}^{\mathsf{T}}b\mathbf{J}_{\text{vm}} + b_{\text{vm}})^{-1}\mathbf{J}_{\text{vm}}^{\mathsf{T}}(k(\mathbf{x}_{\text{r}} - \mathbf{x}_{\text{vm}}) + b\dot{\mathbf{x}}_{\text{r}}). \tag{7}$$

Where b_{vm} is computed based on the proximity to the singularity:

$$b_{\text{vm}} = B_{\text{vm}_{\text{max}}} \exp\left(-\frac{4}{\gamma} \det(\mathbf{J}_{\text{vm}}^{\mathsf{T}}\mathbf{J}_{\text{vm}})\right), \tag{8}$$

$B_{\text{vm}_{\text{max}}}$ define the maximum damping value obtained when the determinant of $\mathbf{J}_{\text{vm}}^{\mathsf{T}}\mathbf{J}_{\text{vm}}$ is zero (the virtual mechanism is in a singular configuration). γ defines the value of $\det(\mathbf{J}_{\text{vm}}^{\mathsf{T}}\mathbf{J}_{\text{vm}})$ above of what the damping is practically zero (the virtual mechanism is away from a singular configuration).

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.

$$\mathbf{F}_{\text{vm}} = -\mathbf{F}_{\text{r}} = k(\mathbf{x}_{\text{vm}} - \mathbf{x}_{\text{r}}) + b(\dot{\mathbf{x}}_{\text{vm}} - \dot{\mathbf{x}}_{\text{r}}). \quad (9)$$

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 $\mathbf{J}_{\text{r}}$ to convert the forces into torque references for the motor controllers.

IV. MULTIPLE WEIGHTED VIRTUAL GUIDES

We now consider a control structure in which N guides are active in parallel, i.e. the robot end-effector is connected to multiple virtual mechanisms, as illustrated in Fig. 1. This scenario applies when there are multiple tasks (e.g. transporting an object to one of multiple possible positions), and the robot does not know initially which task will be executed.

Each of the N virtual mechanisms applies a force $\mathbf{F}_{\text{vm}}^{i}$ to the end-effector. The relative influence of each VM is scaled with a weight p_i, so that the resultant force on the end-effector is:

$$\mathbf{F}_{\text{res}} = \sum_{i=1}^{N} p_i \cdot \mathbf{F}_{\text{vm}}^{i}. \quad (10)$$

Thus, the final force applied to the end-effector is a weighted sum of the forces from each guide, where the weights are determined by their probability, as shown in the control diagram in Fig. 2
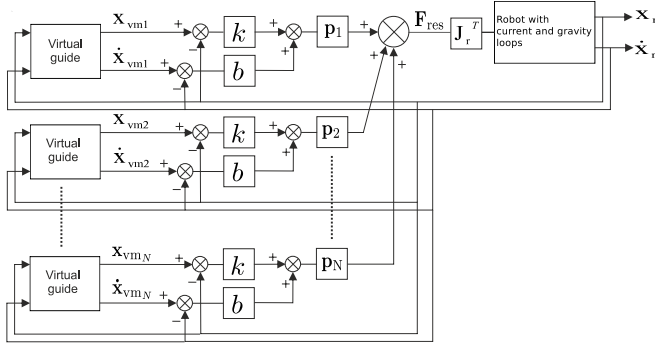

Fig. 2. Control structure for multiple virtual fixtures.

The main question now is under which conditions on p_i this system is stable. This is studied in the next section. As we shall see, p_i must behave as a probability, that is $\sum_{i=1}^{N} p_i = 1$ and $\forall i, p_i \geq 0$.

V. STABILITY ANALYSIS

Our aim is now to prove the stability of using multiple guides in parallel. We do so with the Lyapunov direct method. This method has extensively used in robotics applications due the intrinsic nonlinearities of a high-dofs robots [18].

The robot non linear dynamic model in joint space can be written as:

$$M(q_{\text{r}})\ddot{q}_{\text{r}} + C(q_{\text{r}}, \dot{q}_{\text{r}})\dot{q}_{\text{r}} + F\dot{q}_{\text{r}} + g(q_{\text{r}}) = u, \quad (11)$$

where u represents the command torques, M the inertia matrix, C the Coriolis matrix, F is the viscous attrition, and g is the gravity term. We consider the Lyapunov function [18]

$$V = \tfrac{1}{2}\dot{q}_{\text{r}}^{\mathsf{T}} M(q_{\text{r}})\dot{q}_{\text{r}} + \tfrac{1}{2}k\sum_{i=1}^{N} \epsilon_i^{\mathsf{T}} p_i \epsilon_i \; > \; 0, \; \forall \, \dot{q}_{\text{r}}, \epsilon \neq 0, \quad (12)$$

where ϵ_i is the error $\epsilon_i = \mathbf{x}_{\text{vm}i} - \mathbf{x}_{\text{r}}$, and p_i is the weight in (10). By differentiating (12) with respect to time we obtain:

$$\dot{V} = \dot{q}_{\text{r}}^{\mathsf{T}} M(q_{\text{r}})\ddot{q}_{\text{r}} + \tfrac{1}{2}\dot{q}_{\text{r}}^{\mathsf{T}} \dot{M}(q_{\text{r}})\dot{q}_{\text{r}} + k\sum_{i=1}^{N} \dot{\epsilon}_i^{\mathsf{T}} p_i \epsilon_i + D, \quad (13)$$

$$\text{with} \quad D = \tfrac{1}{2}k\sum_{i=1}^{N} \epsilon_i^{\mathsf{T}} \dot{p}_i \epsilon_i. \quad (14)$$

By substituting $M(q_{\text{r}})\ddot{q}_{\text{r}}$ with the robot model, omitting the dependencies on $q_{\text{r}}, \dot{q}_{\text{r}}$ and rearranging the terms we acquire

$$\dot{V} = \tfrac{1}{2}\dot{q}_{\text{r}}^{\mathsf{T}} \left[\dot{M} - 2C\right] \dot{q}_{\text{r}} - \dot{q}_{\text{r}}^{\mathsf{T}} F\dot{q}_{\text{r}} + \dot{q}_{\text{r}}^{\mathsf{T}} \left[u - g\right]$$
$$+ k\sum_{i=1}^{N} \dot{\epsilon}_i^{\mathsf{T}} p_i \epsilon_i + D. \quad (15)$$

Due to the skew-symmetry property of the matrix $\dot{M} - 2C$ the term $\dot{q}_{\text{r}}^{\mathsf{T}} \left[\dot{M} - 2C\right] \dot{q}_{\text{r}}$ is null, and we simplify to

$$\dot{V} = -\dot{q}_{\text{r}}^{\mathsf{T}} F\dot{q}_{\text{r}} + \dot{q}_{\text{r}}^{\mathsf{T}} \left[u - g\right] + k\sum_{i=1}^{N} \dot{\epsilon}_i^{\mathsf{T}} p_i \epsilon_i + D. \quad (16)$$

A. Virtual mechanisms with fixed positions.

We first study the case where all virtual mechanisms have a fixed position in the robot workspace, i.e. $\forall i, \dot{\mathbf{x}}_{\text{vm},i} = 0$. This allows the simplification $\dot{\epsilon}_i = -\dot{\mathbf{x}}_{\text{r}} = -\mathbf{J}_{\text{r}}\dot{q}_{\text{r}}$, which leads to:

$$\dot{V} = -\dot{q}_{\text{r}}^{\mathsf{T}} F\dot{q}_{\text{r}} + \dot{q}_{\text{r}}^{\mathsf{T}} \left[u - g\right] - k\sum_{i=1}^{N} \dot{q}_{\text{r}}^{\mathsf{T}} \mathbf{J}_{\text{r}}^{\mathsf{T}} p_i \epsilon_i + D. \quad (17)$$

We choose a control input u that compensates the gravity term and introduces a proportional-derivative control: $u = g + \mathbf{J}_{\text{r}}^{\mathsf{T}} \sum_{i=1}^{N} p_i(k\epsilon_i - b\dot{\mathbf{x}}_{\text{r}})$, where each term in the controller is weighted with its own probability p_i. By substituting the chosen control in (17) we obtain:

$$\dot{V} = -\dot{q}_{\text{r}}^{\mathsf{T}}(F + b)\dot{q}_{\text{r}} + D \quad (18)$$

The term $-\dot{q}_{\text{r}}^{\mathsf{T}}(F + b)\dot{q}_{\text{r}}$ is negative. Thus, the system is stable if $D = \tfrac{1}{2}k\sum_{i=1}^{N} \epsilon_i^{\mathsf{T}} \dot{p}_i \epsilon_i$ is also negative. We now study which conditions on p_i ensure this is the case.

B. Constraints on the weights p_i.

First of all, we assume that p_i behaves as a probability, that is $\sum_{i=1}^{N} p_i = 1$, $\sum_{i=1}^{N} \dot{p}_i = 0$ and $\forall i, p_i \geq 0$. To simplify the study we consider the case in which only two virtual mechanisms are active, and thus $\dot{p}_2 = -\dot{p}_1$. By simplifying (18) we get:

$$\dot{V} = -\dot{q}_r^\mathsf{T}(F + b)\dot{q}_r + \tfrac{1}{2}k\dot{p}_1\left[\epsilon_1^\mathsf{T}\epsilon_1 - \epsilon_2^\mathsf{T}\epsilon_2\right]. \qquad (19)$$

Because $k > 0$, the system is asymptotically stable *iff* $\dot{p}_1\left[\epsilon_1^\mathsf{T}\epsilon_1 - \epsilon_2^\mathsf{T}\epsilon_2\right] \leq 0$. This leads to the construction of a probability function that satisfy the following conditions:

$$\begin{cases} \dot{p}_1 > 0 \text{ when } \|\epsilon_1\| < \|\epsilon_2\|, \\ \dot{p}_1 < 0 \text{ when } \|\epsilon_2\| < \|\epsilon_1\|. \end{cases} \qquad (20)$$

and vice-versa $\dot{p}_2$.

The conditions above show that the probability function p_i has to be dependent on the errors ϵ_i, in particular, the inequalities provide the intuition that if the robot is closer to one guide the probability of belonging to that guide should increase over time, and consequently its error should decrease. For instance, we can define a probability function that increases over time when the square error is decreasing: $\dot{p}_i > 0 \iff \frac{d(\epsilon_i^\mathsf{T}\epsilon_i)}{dt} = 2\dot{\epsilon}_i^\mathsf{T}\epsilon_i < 0$. We first model this function as a probability over all the guides, given an activation weight over individual guides. The probability p_i that the ith cart is responsible for guiding the end-effector at position $\mathbf{x}$ becomes:

$$p_i = p(i; \mathbf{x}_r, s_{\text{vm}}^i) = \frac{g(\mathbf{x}_r; s_{\text{vm}}^i)}{\sum_{j=1}^{N} g(\mathbf{x}_r; s_{\text{vm}}^j)}. \qquad (21)$$

We implement the activation weight $g(\mathbf{x}_r; s_{\text{vm}}^i)$ as a Gaussian function:

$$g(\mathbf{x}_r; \mathbf{x}_{\text{vm}}, \mathbf{\Sigma}_{\text{vm}}) = e^{\left(-\frac{1}{2}(\mathbf{x}_{\text{vm}} - \mathbf{x}_r)^\mathsf{T}\mathbf{\Sigma}_{\text{vm}}^{-1}(\mathbf{x}_{\text{vm}} - \mathbf{x}_r)\right)}. \qquad (22)$$

Defined the probability function (21) we can compute the derivatives for the two guides[2]:

$$\dot{p}_1 = p_1 p_2 (\dot{\epsilon}_2^\mathsf{T}\mathbf{\Sigma}_{\text{vm},2}^{-1}\epsilon_2 - \dot{\epsilon}_1^\mathsf{T}\mathbf{\Sigma}_{\text{vm},1}^{-1}\epsilon_1),$$
$$\dot{p}_2 = p_1 p_2 (\dot{\epsilon}_1^\mathsf{T}\mathbf{\Sigma}_{\text{vm},1}^{-1}\epsilon_1 - \dot{\epsilon}_2^\mathsf{T}\mathbf{\Sigma}_{\text{vm},2}^{-1}\epsilon_2). \qquad (23)$$

By considering the virtual mechanisms to have fixed positions i.e. $\dot{\epsilon}_i = -\dot{\mathbf{x}}_r$ and by changing system reference in order to have $\mathbf{x}_r = 0$, the expressions above can be simplified to:

$$\dot{p}_1 = p_1 p_2 \dot{\mathbf{x}}_r^\mathsf{T}(\mathbf{\Sigma}_{\text{vm},1}^{-1}\mathbf{x}_{\text{vm},1} - \mathbf{\Sigma}_{\text{vm},2}^{-1}\mathbf{x}_{\text{vm},2}),$$
$$\dot{p}_2 = p_1 p_2 \dot{\mathbf{x}}_r^\mathsf{T}(\mathbf{\Sigma}_{\text{vm},2}^{-1}\mathbf{x}_{\text{vm},2} - \mathbf{\Sigma}_{\text{vm},1}^{-1}\mathbf{x}_{\text{vm},1}). \qquad (24)$$

[2]To compute these derivatives we consider the covariance matrices to be constant in respect of time, this is practically true if we consider that the covariance changes slowly due the fact that is extracted by human demonstrations, and by considering the virtual mechanisms with fixed position.

We are interested in studying the inequality $\dot{p}_i > 0$, to do so we can simplify (24) in the following:

$$\dot{p}_1 \sim \dot{\mathbf{x}}_r^\mathsf{T}(\mathbf{x}_{\text{vm},1} - \mathbf{x}_{\text{vm},2}) > 0,$$
$$\dot{p}_2 \sim \dot{\mathbf{x}}_r^\mathsf{T}(\mathbf{x}_{\text{vm},2} - \mathbf{x}_{\text{vm},1}) > 0. \qquad (25)$$

Note that this simplification is possible because $p_1, p_2 > 0$ and the covariance matrices are definite positive. By construction these scalar products are positive if the robot moves toward one of the point $\mathbf{x}_{\text{vm},i}$, so for example $\dot{p}_1 > 0$ if the robot moves toward the point $\mathbf{x}_{\text{vm},1}$. The inequality (25) combined with (20) shows that the robot is attracted by the closest guide, that the probabilities of that guide thus increase, and the system is thus stable.

C. Virtual mechanisms with moving positions.

We have shown under which conditions on p_i the system is stable for guides that do not move. We now consider the more general case where they do move, i.e. $\dot{\mathbf{x}}_{\text{vm},i} \neq 0, \ i = 1..N$.

In this situation we can add to the control input the tracking of the velocities generated by the virtual mechanisms: $u = g + \mathbf{J}_r^\mathsf{T}\sum_{i=1}^{N} p_i(k\epsilon_i + b\dot{\epsilon}_i)$. By substituting in (16) we obtain:

$$\dot{V} = -\dot{q}_r^\mathsf{T}F\dot{q}_r + E + D. \qquad (26)$$

with

$$E = \sum_{i=1}^{N}(\dot{q}_r^\mathsf{T}\mathbf{J}_r^\mathsf{T}p_i k\epsilon_i + \dot{q}_r^\mathsf{T}\mathbf{J}_r^\mathsf{T}p_i b\dot{\epsilon}_i + \dot{\epsilon}_i^\mathsf{T}p_i k\epsilon_i) \qquad (27)$$

The first term in (26) is always negative. We now focus on E. By substituting the error derivate $\dot{\epsilon}_i = \mathbf{J}_{\text{vm},i}\dot{s}_{\text{vm},i} - \mathbf{J}_r\dot{q}_r$:

$$E = \sum_{i=1}^{N}(-\dot{q}_r^\mathsf{T}\mathbf{J}_r^\mathsf{T}p_i b\mathbf{J}_r\dot{q}_r + \dot{s}_{\text{vm},i}^\mathsf{T}\mathbf{J}_{\text{vm},i}^\mathsf{T}p_i k\epsilon_i$$
$$+ \dot{q}_r^\mathsf{T}\mathbf{J}_r^\mathsf{T}p_i b\mathbf{J}_{\text{vm},i}\dot{s}_{\text{vm},i}). \quad (28)$$

$$= \sum_{i=1}^{N}(\underbrace{-\dot{q}_r^\mathsf{T}\mathbf{J}_r^\mathsf{T}p_i b\mathbf{J}_r\dot{q}_r}_{E.1} + \underbrace{p_i(\epsilon_i^\mathsf{T}k + \dot{q}_r^\mathsf{T}\mathbf{J}_r^\mathsf{T}b)\mathbf{J}_{\text{vm},i}\dot{s}_{\text{vm},i}}_{E.2}).$$
$$(29)$$

The latter simplification can be made because $\dot{s}_{\text{vm},i}^\mathsf{T}\mathbf{J}_{\text{vm},i}^\mathsf{T}p_i k\epsilon_i$ is a scalar, we can transpose it.

To study $(E.2)$ in (29) we can refer to the virtual mechanism equation in (6). By substituting (6) in $(E.2)$ we have:

$$p_i(k\epsilon_i + b\mathbf{J}_r\dot{q}_r)^\mathsf{T}\mathbf{J}_{\text{vm},i}(\mathbf{J}_{\text{vm},i}^\mathsf{T}b\mathbf{J}_{\text{vm},i})^{-1}\mathbf{J}_{\text{vm},i}^\mathsf{T}(-k\epsilon_i + b\mathbf{J}_r\dot{q}_r). \qquad (30)$$

By defining $A_i = \mathbf{J}_{\text{vm},i}(\mathbf{J}_{\text{vm},i}^\mathsf{T}b\mathbf{J}_{\text{vm},i})^{-1}\mathbf{J}_{\text{vm},i}^\mathsf{T}$, $(E.2)$ becomes:

$$-p_i k^2 \epsilon_i^\mathsf{T}A_i\epsilon_i + p_i\dot{q}_r^\mathsf{T}\mathbf{J}_r^\mathsf{T}b^2 A_i\mathbf{J}_r\dot{q}_r. \qquad (31)$$

We can use (31) in the expression (29), therefore:

$$E = \sum_{i=1}^{N} (-\dot{q}_\mathrm{r}^\intercal \mathbf{J}_\mathrm{r}^\intercal p_i b \mathbf{J}_\mathrm{r} \dot{q}_\mathrm{r} - p_i k^2 \epsilon_i{}^\intercal A_i \epsilon_i + p_i \dot{q}_\mathrm{r}^\intercal \mathbf{J}_\mathrm{r}^\intercal b^2 \epsilon_i A_i \mathbf{J}_\mathrm{r} \dot{q}_\mathrm{r}),$$

$$= \sum_{i=1}^{N} (-p_i k^2 \epsilon_i{}^\intercal A_i \epsilon_i - p_i b \dot{q}_\mathrm{r}^\intercal \mathbf{J}_\mathrm{r}^\intercal (I - bA_i) \mathbf{J}_\mathrm{r} \dot{q}_\mathrm{r}). \quad (32)$$

To have E definite negative we have to proof that the matrix $(I - bA_i)$ is semi-definite positive. Since $\mathbf{J}_{\mathrm{vm},i}$ is a column vector, the following inequality holds:

$$(I - \mathbf{J}_{\mathrm{vm},i}(\mathbf{J}_{\mathrm{vm},i}^\intercal \mathbf{J}_{\mathrm{vm},i})^{-1}\mathbf{J}_{\mathrm{vm},i}^\intercal) \succeq$$
$$(I - I\lambda_{max}(\mathbf{J}_{\mathrm{vm},i}(\mathbf{J}_{\mathrm{vm},i}^T \mathbf{J}_{\mathrm{vm},i})^{-1}\mathbf{J}_{\mathrm{vm},i}^T)) = 0, \quad (33)$$

as the maximal eigenvalue of matrix $\mathbf{J}_{\mathrm{vm},i}(\mathbf{J}_{\mathrm{vm},i}^\intercal \mathbf{J}_{\mathrm{vm},i})^{-1}\mathbf{J}_{\mathrm{vm},i}^\intercal$ is 1 by construction i.e.

$$\lambda_{max}(\mathbf{J}_{\mathrm{vm},i}(\mathbf{J}_{\mathrm{vm},i}^\intercal \mathbf{J}_{\mathrm{vm},i})^{-1}\mathbf{J}_{\mathrm{vm},i}^\intercal) = 1. \quad (34)$$

Due the presence of numeric issues, the scalar inverse $(\mathbf{J}_{\mathrm{vm},i}^\intercal \mathbf{J}_{\mathrm{vm},i})^{-1}$, is calculated as $(\mathbf{J}_{\mathrm{vm},i}^\intercal \mathbf{J}_{\mathrm{vm},i} + b_{\mathrm{vm}})^{-1}$ to avoid infinite value for certain configurations. In this case the inequality holds as:

$$\lambda_{max}(\mathbf{J}_{\mathrm{vm},i}(\mathbf{J}_{\mathrm{vm},i}^\intercal \mathbf{J}_{\mathrm{vm},i} + b_{\mathrm{vm}}))^{-1}\mathbf{J}_{\mathrm{vm},i}^\intercal) <$$
$$\lambda_{max}(\mathbf{J}_{\mathrm{vm},i}(\mathbf{J}_{\mathrm{vm},i}^\intercal \mathbf{J}_{\mathrm{vm},i})^{-1}\mathbf{J}_{\mathrm{vm},i}^\intercal). \quad (35)$$

Finally we proved that E is semi-definite negative.

To proof that the Lyapunov function (26) is definite negative, we have to study D. Before we studied this term when the virtual mechanisms don't move, now we consider $\dot{\mathbf{x}}_{\mathrm{vm},i} \neq 0$, therefore we can write the derivative of the probability (23) for the first virtual mechanism as:

$$\dot{p}_1 = p_1 p_2 \Big(\dot{\mathbf{x}}_\mathrm{r}^\intercal \big(\boldsymbol{\Sigma}_{\mathrm{vm},1}{}^{-1} \mathbf{x}_{\mathrm{vm},1} - \boldsymbol{\Sigma}_{\mathrm{vm},2}{}^{-1} \mathbf{x}_{\mathrm{vm},2} \big)$$
$$+ \dot{\mathbf{x}}_{\mathrm{vm},2}^\intercal \boldsymbol{\Sigma}_{\mathrm{vm},2}{}^{-1} \mathbf{x}_{\mathrm{vm},2} - \dot{\mathbf{x}}_{\mathrm{vm},1}^\intercal \boldsymbol{\Sigma}_{\mathrm{vm},1}{}^{-1} \mathbf{x}_{\mathrm{vm},1} \Big). \quad (36)$$

Considering (4) we have that the velocities $\dot{\mathbf{x}}_{\mathrm{vm},i}$ are orthogonal to the forces applied on the virtual mechanisms, so, by plugging (3) into (4) and choosing $k >> b$ we obtain the following simplification

$$\mathbf{J}_{\mathrm{vm}}{}^\intercal [k(-\mathbf{x}_{\mathrm{vm}}) + b(\dot{\mathbf{x}}_\mathrm{r} - \dot{\mathbf{x}}_{\mathrm{vm}})] \approx \mathbf{J}_{\mathrm{vm}}{}^\intercal \mathbf{x}_{\mathrm{vm}} \approx 0, \quad (37)$$

meaning that the velocities of the virtual mechanism are orthogonal to its position vector. By substituting the virtual mechanism equation (6) in (36) and considering the simplification done before (37), we get

$$\dot{p}_1 = p_1 p_2 \Big(\dot{\mathbf{x}}_\mathrm{r}^\intercal \big(\boldsymbol{\Sigma}_{\mathrm{vm},1}{}^{-1} \mathbf{x}_{\mathrm{vm},1} - \boldsymbol{\Sigma}_{\mathrm{vm},2}{}^{-1} \mathbf{x}_{\mathrm{vm},2} \big)$$
$$+ k\mathbf{x}_{\mathrm{vm},1}{}^\intercal A_1{}^\intercal \boldsymbol{\Sigma}_{\mathrm{vm},1}{}^{-1} \mathbf{x}_{\mathrm{vm},1} - k\mathbf{x}_{\mathrm{vm},2}{}^\intercal A_2{}^\intercal \boldsymbol{\Sigma}_{\mathrm{vm},2}{}^{-1} \mathbf{x}_{\mathrm{vm},2} \Big). \quad (38)$$

The terms with A_i in the equation above can be proven to be null using (37):

$$k(\mathbf{J}_{\mathrm{vm},i}^\intercal b \mathbf{J}_{\mathrm{vm},i})^{-1} \underbrace{\mathbf{x}_{\mathrm{vm}i}^\intercal \mathbf{J}_{\mathrm{vm},i}}_{\approx 0} \mathbf{J}_{\mathrm{vm},i}^\intercal \boldsymbol{\Sigma}_{\mathrm{vm},i}{}^{-1} \mathbf{x}_{\mathrm{vm}i} \approx 0, \quad (39)$$

so equation (38) can be written as done before for the case with fixed virtual mechanisms (24), concluding with the same results. Finally the system with the chosen control converges asymptotically to the equilibrium point. The domain of attraction of the equilibrium point for each guide can be computed using (20).

D. Summary

We investigated the stability of the control system in Fig. 2. Using the direct Lyapunov method, we derived (18). By assuming that p_i behaves as a probability, we could derive the conditions (20). We showed that these conditions are met, and the system is thus stable, when a Gaussian function is used to relate probabilities p_i to errors ϵ_i. We then generalized to the case when the virtual mechanism is moving.

VI. PILOT STUDY

We perform a pilot study with four users, and the set-up in Fig. 3, where the task 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 final position, the robot automatically releases the object for placement on the shelf. Details of the experimental set-up are in [6].

Fig. 3. Pilot study task.

A. Implementation of the Guides

Implementing the weighting of multiple guides that are active in parallel with (10) depends on knowing the probabilities in (21), which itself depends on knowing the activation weight $g(\mathbf{x}_\mathrm{r}; s_{\mathrm{vm}}^n)$ related to the Mahalanobis distance of $\mathbf{x}_\mathrm{r}$ to the guide n with current phase s_{vm}^n. This function is acquired through Gaussian mixture regression (GMR) [19], based on a trained Gaussian mixture model (GMM).

The training procedure is taken from [19]. First, each task is performed multiple times by the user through kinesthetic teaching, i.e. the user holds the end-effector of the robot and guides it in zero-gravity mode. The end-effector trajectories are then clustered with hierarchical clustering, based on the distances acquired through dynamic time warping. The Gaussian Mixture Model (GMM) is then trained on the data in the clusters through Expectation-Maximization; hence the GMM contains the relevant informations about the virtual guides such as position and covariance matrices. On-line when the guide is active, GMR is used to compute $\overline{\mathbf{x}}_{\mathrm{vm}}^n$, $\dot{\overline{\mathbf{x}}}_{\mathrm{vm}}^n$ and $\boldsymbol{\Sigma}_{\mathrm{vm}}^n$, by conditioning on the phase s_{vm}^n for each guide [6]. This allows us to compute the kinematics of the virtual mechanism (1), (2) and the activation weights $g(\mathbf{x}_\mathrm{r}; \boldsymbol{\mu}_{\mathrm{vm}}^n, \boldsymbol{\Sigma}_{\mathrm{vm}}^n)$ as a proxy for $g(\mathbf{x}_\mathrm{r}; s_{\mathrm{vm}}^n)$. These weights are

then used to compute the probabilities $p(n; \mathbf{x}_r, s^n_{vm})$ in (21), which are used to scale the controls in (10).

B. Comparing Safety and Efficiency.

In the pilot study, each of the four users executes in a randomized order 10 episodes for each of the 2 tasks (object goal positions), with the controller gains setted experimentally at the values $k = 700$ and $b = 50$. The measures that we are interested in are:

- execution time, to measure efficiency.
- accuracy of tracking.
- actually observed collisions, to measure safety.

These results of these measurements over all users are summarized in Fig. 4
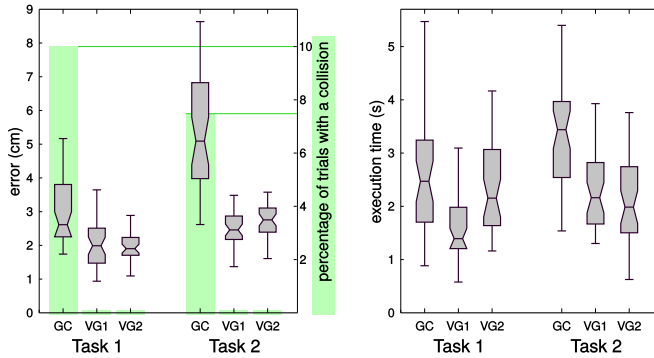

Fig. 4. Comparison of the three assistance modes (gravity compensation only, only one virtual guide, multiple virtual guides), for both guides (for the upper and lower shelf), and three measures (execution time, position error, number of collisions)

From these results, we draw the following conclusions:

- When using virtual guides, the tracking errors decrease substantially (from 3.0cm to 2.0/2.0 for Task 1 and from 5.2 to 2.5/2.7 for Task 2).
- This enables virtual guides reduce the percentages of trials in which a collision occurs to 0.
- 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.
- 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.

VII. Conclusion

In applications of co-manipulation where users must be able to sequentially switch between multiple tasks, it is necessary to have multiple guides i.e. one for each task. We propose a control framework where multiple virtual guides are active in parallel, and the appropriate guide is recognized on-line during the execution of the movement. By weighting the controls of the individual guides with their probabilities, we are able to show that stability of the overall control system is ensured. Our pilot study demonstrates the practical use and advantages of multiple virtual guides. In our future work, we will investigate the exact expression of the equilibrium points and their basin of attraction.

References

[1] L. Joly and C. Andriot, "Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE Int'l Conference on*, vol. 1, May 1995, pp. 357–362 vol.1.

[2] 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 *IEEE Int'l Conference on Robotics and Automation*, 2006, pp. 225–340.

[3] Q. Li, "Virtual reality for fixture design and assembly," Ph.D. dissertation, University of Nottingham, 2008.

[4] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual fixtures for robotic cardiac surgery," in *Proceedings of the 4th Int'l Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI '01. London, UK, UK: Springer-Verlag, 2001, pp. 1419–1420.

[5] 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*, vol. 29, no. 3, pp. 674–683, 2013.

[6] G. Raiola, X. Lamy, and F. Stulp, "Co-manipulation with multiple probabilistic virtual guides," in *IROS*, 2015.

[7] L. Rosenberg, "Virtual fixtures: perceptual tools for telerobotic manipulation," in *Proc. IEEE Virtual Reality Int'l Sympsoium*, 1993.

[8] 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.

[9] J. J. Abbott and A. M. Okamura, "Virtual fixture architectures for telemanipulation," 2003.

[10] Z. Pezzementi, G. D. Hager, and A. M. Okamura, "Dynamic guidance with pseudoadmittance virtual fixtures," in *IEEE Int'l Conference on Robotics and Automation*, 2007, pp. 1761–1767.

[11] J. E. Colgate, M. A. Peshkin, and S. H. Klostermeyer, "Intelligent assist devices in industrial applications: a review," in *IROS*, 2003, pp. 2516–2521.

[12] 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.

[13] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *ICRA*, 2005, pp. 897–903.

[14] A. Kuang, S. Payandeh, B. Zheng, F. Henigman, and C. MacKenzie, "Assembling virtual fixtures for guidance in training environments," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings. 12th Int'l Symposium on*, March 2004, pp. 367–374.

[15] M. Li and A. M. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *In Proc. 11th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2003, pp. 125–131.

[16] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete, "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.

[17] J. T. Nolin, P. M. Stemniski, and A. M. Okamura, "Activation cues and force scaling methods for virtual fixtures," in *in Proc. 11th Int. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, pp. 404–409.

[18] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer, 2009.

[19] 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.