Robotica: (2020) volume 38, pp. 1824–1841. © Cambridge University Press 2019
doi:10.1017/S0263574719001437

Kinesthetic Guidance Utilizing DMP Synchronization and Assistive Virtual Fixtures for Progressive Automation

Dimitrios Papageorgiou, Fotios Dimeas*, Theodora Kastritsi and Zoe Doulgeri

Automation & Robotics Lab, Department of Electrical & Computer Engineering, Aristotle University of Thessaloniki, Greece. E-mails: dimpapag@eng.auth.gr, tkastrit@ece.auth.gr, doulgeri@eng.auth.gr

(Accepted September 20, 2019. First published online: October 14, 2019)

SUMMARY
The progressive automation framework allows the seamless transition of a robot from kinesthetic guidance to autonomous operation mode during programming by demonstration of discrete motion tasks. This is achieved by the synergetic action of dynamic movement primitives (DMPs), virtual fixtures, and variable impedance control. The proposed DMPs encode the demonstrated trajectory and synchronize with the current demonstration from the user so that the reference generated motion follows the human's demonstration. The proposed virtual fixtures assist the user in repeating the learned kinematic behavior but allow penetration so that the user can make modifications to the learned trajectory if needed. The tracking error in combination with the interaction forces and torques is used by a variable stiffness strategy to adjust the progressive automation level and transition the leading role between the human and the robot. An energy tank approach is utilized to apply the designed controller and to prove the passivity of the overall control method. An experimental evaluation of the proposed framework is presented for a pick and place task and results show that the transition to autonomous mode is achieved in few demonstrations.

KEYWORDS: Physical human–robot interaction; Programming by demonstration; Dynamic movement primitives; Virtual fixtures.

1. Introduction

Robot programming by demonstration (PbD) aims to reduce the programming time compared to conventional methods which involve time-consuming procedures from expert personnel and, therefore, significant financial burden. Utilizing kinesthetic guidance a human can quickly demonstrate the task to the robot, outperforming conventional offline programming techniques, that may require up to many hours for a simple task that lasts only a few seconds to execute.[1] The benefit of developing effective and practical methods for PbD can eventually render collaborative robots viable for small- and medium-sized enterprises, where changes in the production line are frequent.

Progressive automation is a method for seamless PbD, introduced in ref. [2]. It allows a human to easily program a repetitive robotic motion task kinesthetically and it actively assists the user while the robot learns to execute the task autonomously. The robot starts fully compliant and after few demonstrations, it gradually becomes autonomous, without requiring prior knowledge, pre-programming of the task sequence, or interaction of the user with graphical interfaces. During the demonstrations, the user is able to make adjustments and refinements to the learned trajectory. A variable stiffness controller transitions the leading role among the robot and the human partner, based on the forces

* Corresponding author. E-mail: dimeasf@ee.auth.gr

applied by the human and the correspondence between consecutive demonstrations. The time and effort required in the learning procedure from the human-teacher to the robot-learner depend on the quality of communication between them. To improve the quality of communication in progressive automation, the robot-learner provides the human-teacher with real-time feedback of its current knowledge to make the learning process transparent.[3] This feedback is provided by the robot through haptic means and in particular via the utilization of virtual fixtures that induce virtual forces around the learned kinematic behavior.

In this paper, we propose novel features in the progressive automation framework,[2] which are able to increase the speed and effectiveness of teaching by demonstration in repetitive tasks. Dynamic movement primitives (DMPs) are utilized for motion generation that are synchronized with the human in the current demonstration. Moreover, assistive virtual fixtures are imposed around the generated trajectory for providing the user with haptic feedback of the already learned kinematic behavior; virtual fixtures are penetrable to allow the human to modify the trajectory if needed. Specifically, in this work, we propose the following significant extensions compared to our previous work:[2,4,5]

- The conceptual framework originally presented in ref. [2] considered only the position of the robot during the demonstration, while the representation of the robot's orientation was introduced in ref. [4]. However, none of these works provided any theoretical justification on the stability and passivity of the control system under the external interaction forces. Moreover, synchronization of the DMP generated motion with the current demonstration was not addressed.
- The works in refs. [5, 6] achieved synchronization of the generated motion via the adaptation of the temporal scaling of the DMP based only on position error; here the synchronization is based on both the position and the orientation error. Furthermore, we introduce virtual fixtures that produce assistive forces and torques toward the learned trajectory. A rigorous proof of the passivity of the overall control system is provided.

The rest of the paper is structured as follows. In Section 2, an overview of the proposed progressive automation framework is initially provided. In Section 2.1, the proposed motion synchronization of the DMP and in Section 2.2 the assistive virtual fixtures are presented. The variable stiffness strategy for role allocation is presented in Section 2.3, while the controller and its stability proof are presented in Sections 2.4 and 2.5, respectively. An experimental evaluation of the proposed framework is presented in Section 3.

1.1. Related work

PbD lies within the concept of robot learning from humans[7] aiming to autonomous execution of a task by the robot. The learning procedure usually involves shared control, where the robot actively participates in the task through the exertion of forces in parallel with the human. The degree of the robot's leading role in shared control is defined as the level of automation.[8] In ref. [9], an adaptable control scheme was proposed for online allocation of the leading role between the two agents, the human and the robot, based solely on the measurements of interaction forces. The utilization of variable impedance control for implementing role allocation was examined in the literature.[10,11] In ref. [10], a complex human–robot cooperative transportation scenario was considered, in which the robot identified the intention and allowed the proactive behavior of the human when needed, by switching among predefined constant impedance parameters. In ref. [11], an incremental learning method was proposed, involving the fusion of data from consecutive demonstrations and the gradual increase of the robot's stiffness in each demonstration up to a maximum level, without, however, taking into account the tracking error or the interaction forces. Saveriano et al.[12] proposed an iterative learning method, in which the human could kinesthetically teach tasks related to the end-effector, as well as secondary tasks in the null-space. In ref. [13], Jarrasse et al. revealed the necessity of a continuous dynamic role assignment policy, as compared to the case of pre-defining a constant role assignment. Although employing a variable impedance controller can resolve the problem of role allocation, it can also lead to loss of passivity.[14] To resolve this problem, energy tanks can be utilized[15] by virtually storing the dissipated energy and reusing it only when it is necessary, guaranteeing in this way the passivity of the system.[16]

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

Dynamical systems (DSs) are widely used to encode kinematic behaviors demonstrated by a human-teacher either kinesthetically or by utilizing other sensorial inputs (e.g., robotic vision to capture the human's motion). The DSs can be adapted or modified (e.g., for obstacle avoidance) and are also robust to perturbations of the target and initial conditions. Two commonly used DSs for encoding kinematic behaviors are the autonomous DSs, for example, the ones utilizing Gaussian mixture models,[17] and the DMPs. The DMPs are appropriate for encoding a trajectory-based representation, for example, to encode both spatial and temporal properties of a single kinematic behavior. Despite their time dependency, DMP learning is possible with one demonstrated trajectory. In contrast, autonomous DSs are more appropriate for state-space representations of complex attractor landscapes and require many demonstrations to cover the whole state-space.[18]

For the online adaptation of the DMP evolution to external disturbances, both the adaptation of its temporal evolution, such as the "phase stopping," and the reshaping of its spatial evolution are proposed in the literature, utilizing coupling terms related either to the measurement of external forces or the tracking error.[19] In this direction, phase estimation was proposed to temporally align trajectories for tackling the problem of cooperative agent coordination,[20] as well as for learning from a set of demonstrations with different time scaling.[21] However, phase adjustment cannot tackle the problem of temporal synchronization (speeding up or slowing down of the motion), as it affects only the non-linear part of the DMP. A promising solution toward achieving synchronization with the human's motion is proposed for rhythmic kinematic behaviors, via the adaptation of the DMP's main frequency.[22] Similarly, for the synchronization of discrete kinematic behaviors, the adaptation of the temporal scaling factor is proposed in ref. [23], in the context of a moving target. However, the adaptation is based on a model for predicting the total motion duration, making the method not applicable in cases of arbitrary trajectory modifications for kinesthetic teaching. In the case of autonomous DS, even though they do not have a time dependency, they would require appropriate scaling of their output velocity in order to remain synchronized with the user's demonstration.

During the kinesthetic guidance, virtual fixtures can improve the performance of the human in following a desired path.[24] Particularly when multiple demonstrations are required, the enforcement of virtual fixtures around the learned trajectory can improve precision and reduce the mental workload of the human. Virtual fixtures were firstly introduced in ref. [25] for tele-robotics and have been utilized in many robotic tasks.[26,27] To discourage the motion close to constraints during kinesthetic guidance, Bettini et al.[28] proposed the introduction of an anisotropic damping in an admittance control law, which considers a parametric analytic expression of the reference path. The energy redirection was proposed in ref. [29] as a form of virtual fixtures for increased task accuracy, by introducing a dissipative control scheme. For kinesthetic teaching, the motion refinement tube approach was proposed in ref. [30], in which virtual fixtures were placed around the nominal path depending on the magnitude of the tracking error to prevent the human from guiding the robot too far from the nominal path during demonstrations. Restrepo et al.[31] programmed virtual fixtures iteratively by demonstration with the use of splines. Here we introduce assistive virtual fixtures with variable magnitude depending on the progressive automation level that can also be penetrated and re-entered by the user. The fixtures are computationally efficient since they are constructed as a hypersphere around the tracking error between the robot's pose and the synchronized motion from the DMP.

2. The Proposed Progressive Automation Framework

Progressive automation consists of a variable stiffness controller for role allocation, assistive penetrable virtual fixtures around the already learned kinematic behavior, and an online motion generation system based on DMP that is synchronized with the user, encodes, and reproduces the demonstrated trajectory, as shown in the block diagram of Fig. 1. A task cycle is demonstrated kinesthetically and consists of discrete motion segments between cues. The cues are provided by the user to signal start/stop of the demonstration as well as intermediate actions. In case a gripper is attached to the robot for object handling, the cues also determine actions of the gripper for opening/closing the fingers.

Our aim is to design a controller for progressive automation which provides the user with haptic feedback of the currently learned kinematic behavior and smoothly transition into autonomous

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

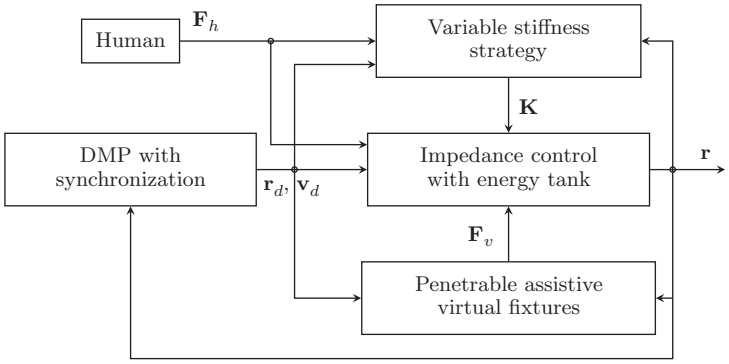

Fig. 1. Block diagram of the overall proposed system.

execution. The role of the robot is modified online by utilizing a variable stiffness strategy, based on the tracking error between demonstrations and the external forces and torques. The haptic feedback is provided by assistive penetrable virtual fixtures, which smoothly vanish as the role of the robot becomes more active. The virtual fixtures aim at the operator's assistance during demonstration and act in a complementary way to the variable stiffness of the impedance controller. When the robot is guided kinesthetically, the target stiffness of the impedance increases slowly during the course of multiple demonstrations. While the stiffness is low, the advantage of using virtual fixtures is in assisting the user track accurately the learned behavior and thus achieve faster progressive automation. A drawback is that the fixtures can increase the force required by the user to initiate modifications because of the fixture's potential.

At the initial demonstration of a task, the robot has zero stiffness and the human is the leader of the movement. Since there is no currently learned kinematic behavior, no assistive feedback is provided. After each motion segment is demonstrated, it is encoded with DMP. Let $\mathbf{r}^i(t) = [\mathbf{p}^T \ \mathbf{Q}^T]^T \in \mathcal{T}$ be the task coordinates of the robot that are recorded in the i^{th} demonstration of a complete task cycle. $\mathcal{T} \triangleq \mathbb{R}^3 \times \mathbb{S}^3$ is the special Euclidean space $SE(3)$ expressed as the combination of position $\mathbf{p} \in \mathbb{R}^3$ and unit quaternion $\mathbf{Q} = [\eta \ \boldsymbol{\epsilon}^T]^T \in \mathbb{S}^3$ ($\mathbb{S}^3$ is a unit sphere in $\mathbb{R}^4$), with $\eta \in \mathbb{R}$ being the scalar part and $\boldsymbol{\epsilon} = \text{vect}(\mathbf{Q}) \in \mathbb{R}^3$ being the vector part of $\mathbf{Q}$. On completion of the first demonstration, the second one starts and, simultaneously, the desired trajectory $\mathbf{r}_d(t) = [\mathbf{p}_d^T \ \mathbf{Q}_d^T]^T \in \mathcal{T}$, $\mathbf{v}_d(t) = [\dot{\mathbf{p}}_d^T \ \boldsymbol{\omega}_d^T]^T \in \mathbb{R}^6$, starts being produced by the DMP and is provided to a variable impedance controller. During the learning procedure, the DMP evolution is synchronized with the user, while penetrable virtual fixtures are utilized to provide haptic feedback to the user to remain close to the DMP evolution; if no corrections are desired by the user, the stiffness increases while the effect of virtual fixtures smoothly decreases to zero. The user can apply forces to penetrate the virtual fixtures and re-demonstrate or modify a segment of the task. In this case, the stiffness decreases and the effect of the fixtures increases, to provide the user with the ability to easily return to the previously learned path if it is desired. When target stiffness reaches the maximum value, the user can stop interacting with the robot and the latter can continue moving autonomously according to the learned trajectory $\mathbf{r}_d$. In the following subsections, each block of the proposed method in Fig. 1 is presented in detail.

2.1. Dynamical movement primitives with motion synchronization

The use of DMP[32] for encoding the demonstrated task coordinates in the progressive automation framework offers the benefit of a compact and computationally efficient representation. It also offers temporal and spatial scaling of the reproduced motion that is useful for adaptability and generalization. For a segment of a discrete motion task, a DMP is required which consists of separate transformation systems for each coordinate and a common canonical system. Let $\mathbf{g} \in \mathbb{R}^3$ be the position and $\mathbf{Q}_g \in \mathbb{S}^3$ the orientation goal of a task segment expressed as a unit quaternion. The transformation systems for the generation of the reference position are:[32]

$$\tau \dot{\mathbf{z}} = \alpha_y \left(\beta_y \left(\mathbf{g} - \mathbf{p}_d \right) - \mathbf{z} \right) + \mathbf{f}_p(x), \tag{1}$$

$$\tau \dot{\mathbf{p}}_d = \mathbf{z}, \tag{2}$$

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

and for the orientation are:[33]

$$\tau\dot{\boldsymbol{\omega}}_s = \alpha_y\left(\beta_y 2\log\left(\mathbf{Q}_g * \mathbf{Q}_d^{-1}\right) - \boldsymbol{\omega}_s\right) + \mathbf{f}_o(x), \tag{3}$$

$$\tau\dot{\mathbf{Q}}_d = \frac{1}{2}\mathbf{J}_Q(\mathbf{Q}_d)\boldsymbol{\omega}_s, \tag{4}$$

where "$*$" denotes the quaternion product, $\boldsymbol{\omega}_s = \tau\boldsymbol{\omega}_d$ is the scaled angular velocity reference and:

$$\mathbf{J}_Q(\mathbf{Q}_d) = \begin{bmatrix} -\boldsymbol{\epsilon}_d^T \\ \eta_d\mathbf{I}_3 - \mathbf{S}(\boldsymbol{\epsilon}_d) \end{bmatrix} \in \mathbb{R}^{4\times 3}, \tag{5}$$

with $\mathbf{S}(\boldsymbol{\epsilon}_d) : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ being the skew-symmetric matrix corresponding to vector $\boldsymbol{\epsilon}_d$. Constants α_y, β_y are design parameters while the forcing terms $\mathbf{f}_p$, $\mathbf{f}_o$ are given in the Appendix A. The common canonical system is:

$$\tau\dot{x} = -\alpha_x x, \qquad x(0) = 1 \tag{6}$$

where α_x is a time constant and τ is the temporal scaling factor.

By adapting τ, the DMP evolution can speed up or slow down appropriately to remain synchronized with the current demonstration. To this aim, the temporal scaling term τ was adapted in ref. [5] based only on the position error. In this work, the proposed temporal synchronization method considers the aggregated position and orientation error via the following temporal scaling adaptation law:

$$\dot{\tau} = \begin{cases} a_\tau\left(\frac{\tau_s}{\xi+\delta} - \tau\right), & \text{if } \xi > 0 \\ a_\tau\left(\frac{\tau_s}{\delta} - \tau\right), & \text{if } \xi \le 0 \end{cases}, \quad \tau(0) = \tau_s, \tag{7}$$

$$\xi = 1 + n_y\widetilde{\mathbf{x}}^T\mathbf{M}\mathbf{v}_d, \tag{8}$$

where n_y is a positive control parameter to scale the intensity of synchronization, a_τ is a positive control constant, δ is a positive small number to avoid dividing by zero when $\xi = 0$, $\tau_s \in \mathbb{R}^+$ is the duration of the demonstrated segment used for DMP training, $\mathbf{M} \in \mathbb{R}^6$ a positive definite diagonal constant matrix, and $\widetilde{\mathbf{x}}$ is the tracking error defined as follows:

$$\widetilde{\mathbf{x}}(\mathbf{r}, \mathbf{r}_d) = \begin{bmatrix} \widetilde{\mathbf{p}} \\ \boldsymbol{\epsilon}_e \end{bmatrix} \in \mathbb{R}^6, \tag{9}$$

with $\widetilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_d$ and $\boldsymbol{\epsilon}_e$ the vector part of the unit quaternion difference[34] $\mathbf{Q}_e \in \mathbb{S}^3$ given by:

$$\mathbf{Q}_e \triangleq \mathbf{Q} * \mathbf{Q}_d^{-1} = \begin{bmatrix} \eta_e \\ \boldsymbol{\epsilon}_e \end{bmatrix}, \tag{10}$$

where $\eta_e \in \mathbb{R}$ is the scalar part of $\mathbf{Q}_e$. The values of η_e and $\boldsymbol{\epsilon}_e$ can be computed as follows:

$$\begin{aligned} \eta_e &= \mathbf{Q}^T\mathbf{Q}_d, \\ \boldsymbol{\epsilon}_e &= \mathbf{J}_Q^T(\mathbf{Q}_d)\mathbf{Q}. \end{aligned} \tag{11}$$

The adaptation law of Eq. (7) ensures that τ tracks the value of $\tau_s/(\xi + \delta)$, which reflects the temporal scaling corresponding to the current speed of the demonstration. The second term of (8) is the inner product of the normalized tracking error $\mathbf{M}\widetilde{\mathbf{x}}$ with the velocity $\mathbf{v}_d$ produced by the DMP, which implicitly combines the position and the orientation dot products according to the matrix $\mathbf{M}$ and is proportional to the cosine of angle ϕ between them, as shown in Fig. 2. A positive product indicates that the robot is ahead of the DMP and a negative one indicates that the robot is behind. When the product is zero, the addition of one as the first term in (8) ensures that $\xi \to 1$ and the equilibrium point for (7) equals to τ_s. With the proposed temporal scaling, the DMP synchronizes with the position and orientation of the robot while it is guided by the human, even in the case when a modification of the learned trajectory is demonstrated, as shown in Fig. 2. Combining position and orientation errors is necessary to correctly synchronize in tasks where the position and orientation

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

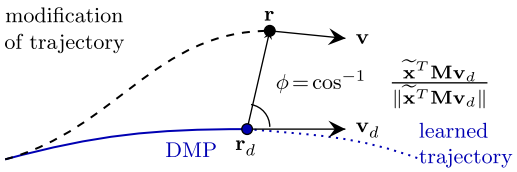

Fig. 2. Illustration of the variables for temporal synchronization during a modification of the learned trajectory.

are coupled, like the case of an oriented path. Notice that the tunable values of the weight matrix $\mathbf{M}$ in (8) allow us to modify the importance of translation and orientation errors for synchronization in such tasks. For the combined synchronization proposed here, two main situations may occur during teaching: (i) the dot products of the position and orientation errors with the respective desired velocities have the same sign. Hence both the position and orientation are either ahead of the DMP (positive signs) or behind (negative signs) and synchronization is correctly performed by speeding up or slowing down, respectively, (ii) the dot products have opposite signs. The worst case in this situation is when the two dot products cancel out each other; then, $\xi = 1$ and the DMP is executed with τ_s, that is, there is no synchronization. This is a case that may appear at some instant but cannot persist during the demonstration. When the human guides the robot in the same direction with the DMP ($\phi \leq \pi/2$), the desired pose $\mathbf{r}_d$ in Fig. 2 remains synchronized to the current robot pose $\mathbf{r}$. However, if the human guides the robot normal to or in the opposite direction of the velocity vector $\mathbf{v}_d$, then the DMP evolution is paused until the robot is guided in the same direction again. When either the desired translational ($\dot{\mathbf{p}}_d$) or desired rotational ($\boldsymbol{\omega}_d$) velocity component equals $\mathbf{0}$, then the dot product of this component will be 0 and will not affect the temporal scaling. In such case, synchronization will be based solely on the error of the component with the non-zero desired velocity. If both are zero, that is, $\|\mathbf{v}_d\| = 0$, for a time interval, the DMP will evolve with the nominal value of time scaling (τ_s) and the synchronization will be deactivated during this interval. Nevertheless, any error in the position or orientation that develops while the corresponding desired velocity is zero will affect the stiffness variation and will be taken into consideration in the temporal scaling as soon as the respective desired velocity takes non-zero values.

In this paper, the latest demonstration is used for training the DMP that generates the reference pose trajectory in the next demonstration, while the joining of consecutive DMP is used to reproduce the entire task cycle. The completion of a motion segment —and therefore the start of the next one— can be assessed by the DMP convergence to the goal. The velocity of the robot when the goal of a segment is reached is assumed to be zero.

2.2. *Penetrable assistive virtual fixtures*

To assist the user in following the already learned path that is generated by the DMP and at the same time allow modifications during a new demonstration, we propose the use of spherical penetrable virtual fixtures for the end-effector pose $\mathbf{r}$ around the synchronized DMP evolution $\mathbf{r}_d$ in $SE(3)$. Let $\bar{r} > 0$ and $\bar{\vartheta} \in (0, \pi)$ be constant parameters representing the pre-specified (user defined) bounds of fixture's effective area in position and orientation, respectively. For the construction of the virtual fixture spheres, the metric of the weighted generalized distance in $SE(3)$ $\zeta(\mathbf{r}, \mathbf{r}_d)$ is

$$\zeta(\mathbf{r}, \mathbf{r}_d) = \frac{\|\widetilde{\mathbf{p}}\|^2}{\bar{r}^2} + \frac{1 - \eta_e}{1 - \cos\frac{\bar{\vartheta}}{2}} \tag{12}$$

given by (A2) in Appendix B, utilizing

$$\begin{aligned} f(\mathbf{p}, \mathbf{p}_d) &\triangleq \|\widetilde{\mathbf{p}}\|^2 \\ g(\mathbf{Q}, \mathbf{Q}_d) &\triangleq 1 - \cos\left(\frac{\theta_e(\mathbf{Q}, \mathbf{Q}_d)}{2}\right) = 1 - \eta_e, \\ \alpha_\zeta &\triangleq \frac{1}{\bar{r}^2}, \\ \beta_\zeta &\triangleq \frac{1}{1 - \cos\frac{\bar{\vartheta}}{2}}, \end{aligned} \tag{13}$$

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

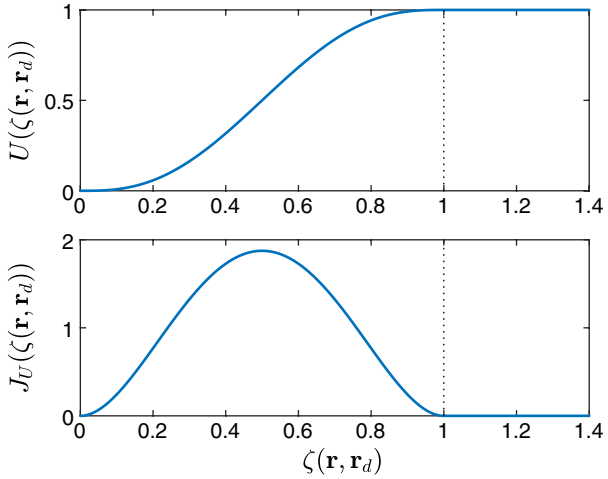

Fig. 3. The proposed artificial potential and its gradient.

with $\theta_e(\mathbf{Q}, \mathbf{Q}_d)$ being the angle between the orientations corresponding to $\mathbf{Q}$ and $\mathbf{Q}_d$, as defined in Appendix B. Let us now define the ball centered at $\mathbf{r}_d$, utilizing the generalized distance $\zeta(\mathbf{r}, \mathbf{r}_d)$, as

$$\Omega(\mathbf{r}_d) = \{\mathbf{r} \in \mathcal{T} : \zeta(\mathbf{r}, \mathbf{r}_d) < 1\}, \tag{14}$$

and its boundary $\partial\Omega(\mathbf{r}_d) = \{\mathbf{r} \in \mathcal{T} : \zeta(\mathbf{r}, \mathbf{r}_d) = 1\}$. Notice that for all $\mathbf{r} \in \Omega(\mathbf{r}_d)$, it holds

$$\begin{aligned} 0 \leq \tfrac{\|\tilde{\mathbf{p}}\|^2}{\bar{r}^2} < 1, \\ 0 \leq \tfrac{1-\cos(\frac{\theta_e}{2})}{1-\cos(\frac{\bar{\vartheta}}{2})} < 1, \end{aligned} \tag{15}$$

which implies

$$\begin{aligned} \|\tilde{\mathbf{p}}\| < \bar{r}, \\ |\theta_e(\mathbf{Q}, \mathbf{Q}_d)| < \bar{\vartheta}. \end{aligned} \tag{16}$$

Let us further denote the complement of Ω as $\Omega^{\complement} = \mathcal{T} \setminus \Omega$, which is the region outside the sphere. Notice that $\partial\Omega \subset \Omega^{\complement}$. Further notice that setting the initial state of the DMP to that of the actual robot then $\mathbf{r}_d$ and $\mathbf{r}$ coincide at $t = 0$.

The control signal drives the user toward the DMP evolution $\mathbf{r}_d(t)$ when the end-effector's pose lies within $\Omega(\mathbf{r}_d)$, while allowing to penetrate the boundary $\partial\Omega(\mathbf{r}_d)$ and freely guide the robot during the corrections by exerting, initially, a relatively higher force. In that direction, the following artificial potential $U(\zeta(\mathbf{r}, \mathbf{r}_d)) \in \mathbb{R}_{\geq 0}$ is proposed:

$$U(\zeta(\mathbf{r}, \mathbf{r}_d)) \triangleq \begin{cases} 10\zeta^3 - 15\zeta^4 + 6\zeta^5, & 0 \leq \zeta(\mathbf{r}, \mathbf{r}_d) \leq 1 \\ 1, & \text{otherwise} \end{cases}. \tag{17}$$

Let us also define:

$$J_U(\zeta(\mathbf{r}, \mathbf{r}_d)) \triangleq \frac{\partial U(\zeta)}{\partial \zeta} \in \mathbb{R}_{\geq 0}. \tag{18}$$

The potential (17) and its derivative $J_U(\zeta)$ are shown in Fig. 3.

Notice that U and J_U possess the following properties:

1. $U(\zeta(\mathbf{r}, \mathbf{r}_d))$ is increasing w.r.t. ζ within Ω,
2. $U(\zeta(\mathbf{r}, \mathbf{r}_d)) \geq 0$, $\forall \mathbf{r} \in \mathcal{T}$ and $U(\zeta(\mathbf{r}, \mathbf{r}_d)) = 0$ only at $\mathbf{r} = \mathbf{r}_d$,
3. $U(\zeta(\mathbf{r}, \mathbf{r}_d)) < 1$, $\forall \mathbf{r} \in \Omega$ and $U(\zeta(\mathbf{r}, \mathbf{r}_d)) = 1$, $\forall \mathbf{r} \in \Omega^{\complement}$,
4. $J_U(\zeta(\mathbf{r}, \mathbf{r}_d)) \geq 0$, $\forall \mathbf{r} \in \mathcal{T}$ and $J_U(\zeta(\mathbf{r}, \mathbf{r}_d)) = 0$ only for all $\mathbf{r} \in \Omega^{\complement} \cup \{\mathbf{r}_d\}$.

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

The following control signal is then proposed to provide haptic feedback during kinesthetic guidance of a learned kinematic behavior:

$$\mathbf{F}_v \triangleq -k_{vf}\mathbf{A}^T\frac{\partial U}{\partial \widetilde{\mathbf{x}}}, \tag{19}$$

with $k_{vf} \in \mathbb{R}_{>0}$ a tunable control gain and $\mathbf{A}$ being the following matrix:

$$\mathbf{A}(\widetilde{x}) \triangleq \text{diag}\left(\mathbf{I}_3,\ \frac{1}{2}\left(\eta_e\mathbf{I}_3 - \mathbf{S}(\boldsymbol{\epsilon}_e)\right)\right). \tag{20}$$

which relates the time derivative of $\widetilde{\mathbf{x}}$ with the velocity error $\widetilde{\mathbf{v}} = \mathbf{v} - \mathbf{v}'_d$, with $\mathbf{v} = [\dot{\mathbf{p}}^T\ \boldsymbol{\omega}^T]^T$ being the generalized velocity of the end-effector and $\mathbf{v}'_d = [\mathbf{p}_d'^T\ \boldsymbol{\omega}_d'^T]^T$, with $[0\ \boldsymbol{\omega}_d'^T]^T = \mathbf{Q}_e * [0\ \boldsymbol{\omega}_d^T]^T * \mathbf{Q}_e^{-1}$ (notice that when there is no orientation error, $\mathbf{Q}_e = [1\ 0\ 0\ 0]^T$ and $\boldsymbol{\omega}_d = \boldsymbol{\omega}'_d$):

$$\dot{\widetilde{\mathbf{x}}} = \mathbf{A}(\widetilde{\mathbf{x}})\widetilde{\mathbf{v}}. \tag{21}$$

By differentiating $U(\zeta(\widetilde{\mathbf{x}}))$ with respect to $\widetilde{\mathbf{x}}$, we get $\frac{\partial U}{\partial \widetilde{\mathbf{x}}} = \frac{\partial U}{\partial \zeta}\frac{\partial \zeta}{\partial \widetilde{\mathbf{x}}} = J_U\frac{\partial \zeta}{\partial \widetilde{\mathbf{x}}}$. The partial derivative of ζ for the selection (13) yields:

$$\frac{\partial \zeta}{\partial \widetilde{\mathbf{x}}} = \begin{bmatrix} \frac{2\widetilde{\mathbf{p}}}{m_p} \\ \frac{\boldsymbol{\epsilon}_e}{\eta_e m_o} \end{bmatrix}, \tag{22}$$

where $m_p = \bar{r}^2$, $m_o = 1 - \cos(\frac{\bar{\vartheta}}{2})$. Substituting (22) to (19), the control signal can be written as follows:

$$\mathbf{F}_v = -k_{vf}J_U\mathbf{M}_6\widetilde{\mathbf{x}}(\mathbf{r}, \mathbf{r}_d), \tag{23}$$

with $\mathbf{M}_6 = \text{diag}(\frac{2}{m_p}\mathbf{I}_3,\ \frac{1}{2m_0}\mathbf{I}_3)$.

2.3. The variable stiffness strategy

The progressive automation level is based on a variable gain to transition the role of the robot from passively following the user's demonstrations to accurately following the trajectory produced by the DMP. The desired Cartesian stiffness matrix $\mathbf{K} \in \mathbb{R}^{6\times 6}$ of the robot is constructed as:

$$\mathbf{K} = \kappa(t)\begin{bmatrix} k_T\mathbf{I_3} & \mathbf{0} \\ \mathbf{0} & k_R\mathbf{I_3} \end{bmatrix}, \tag{24}$$

with $k_T, k_R \in \mathbb{R}^+$ being the maximum desired translational and torsional stiffness, respectively, and $\kappa \in [0, 1]$ being a variable gain whose rate of change κ_r depends on the external interaction force and on the deviation from the reference path as well as the current value of $\kappa(t)$:

$$\dot{\kappa} = \begin{cases} \max\{\kappa_r, 0\}, & \kappa = 0 \\ \kappa_r, & 0 < \kappa < 1 \\ \min\{\kappa_r, 0\}, & \kappa = 1 \end{cases}, \qquad \text{with } \kappa(0) = 0, \tag{25}$$

$$\kappa_r = \left(\frac{\kappa}{f_r} + f_{min}\right)\left(1 - \frac{1}{\lambda_2}(||\mathbf{F}_P|| + \gamma_1||\mathbf{F}_R||)^3 - \frac{1}{\lambda_1}(||\widetilde{\mathbf{p}}|| + \gamma_2||\boldsymbol{\epsilon}_e||)\right). \tag{26}$$

where $\mathbf{F}_P, \mathbf{F}_R$ are the external forces and torques, respectively, applied by the human (i.e., $\mathbf{F}_h = [\mathbf{F}_P^T\ \mathbf{F}_R^T]^T \in \mathbb{R}^6$). The design parameters λ_1, λ_2 affect the impact of the interaction force and the tracking error on the stiffness rate of change, f_{min} is a positive constant to induce a gain increase when $\kappa = 0$, and f_r, γ_1, γ_2 are scaling terms.

The variable gain κ increases after the first demonstration and allocates an increasingly leading role to the robot, while the human demonstrates a path with small tracking error and the applied forces/torques are relatively low. The transition rate κ_r also depends on the current value of κ. This rate is initially slow, requiring the human to conduct a few demonstrations and allow fine-tuning of the reference trajectory. The rate increases with the increase of κ. When the robot moves

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

autonomously ($\kappa = 1$), a high interaction force or torque quickly reverses the leading role back to the human in order to allow modifications of the path. The higher the level of automation κ, the slightly higher are the forces required by the user to initiate an adjustment, but the bigger is the assistance to the user. However, while $\kappa = 1$, the force required to make an adjustment at any location along the segment would be equal.

2.4. The proposed robot controller

Consider the dynamic model of a n-dof non-redundant manipulator in the six-dimensional operational space with gravity compensation under the kinesthetic guidance of a human force $\mathbf{F}_h$ and the action of a control input $\mathbf{F}_\tau \in \mathbb{R}^6$ that is designed to achieve the control objective for the progressive automation:

$$\mathbf{\Lambda}_x(\mathbf{q})\dot{\mathbf{v}} + \mathbf{C}_x(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v} = \mathbf{F}_h + \mathbf{F}_\tau \tag{27}$$

where

$$\mathbf{\Lambda}_x(\mathbf{q}) = [\mathbf{J}(\mathbf{q})\mathbf{\Lambda}(\mathbf{q})^{-1}\mathbf{J}(\mathbf{q})^T]^{-1} \tag{28}$$

$$\mathbf{C}_x(\mathbf{q}, \dot{\mathbf{q}})v = \mathbf{J}(\mathbf{q})^{-T}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{q} - \mathbf{\Lambda}_x(\mathbf{q})\dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \tag{29}$$

with $\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n$ being the robot joint position and velocity, $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6\times n}$ the robot Jacobian, $\mathbf{\Lambda}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ the manipulator's inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ the Coriolis and centripetal matrix. Notice that the task space inertia matrix $\mathbf{\Lambda}_x$ is positive definite and the matrix $\dot{\mathbf{\Lambda}}_x - 2\mathbf{C}_x$ is skew symmetric.

To achieve the target dynamic behavior with the assistive virtual fixtures, the following control signal is utilized that imposes a desired variable Cartesian stiffness and damping without shaping the inertia:

$$\mathbf{F}_\tau = \mathbf{\Lambda}_x(\mathbf{q})\dot{\mathbf{v}}'_d + \mathbf{C}_x(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v}'_d - \mathbf{D}\tilde{\mathbf{v}} - \beta_r\mathbf{K}\tilde{\mathbf{x}} + (1 - \kappa\beta_r)\mathbf{F}_v, \tag{30}$$

where $\mathbf{D} \in \mathbb{R}^{6\times 6}$ is the positive definite damping matrix that is selected according to $\mathbf{K}$, $\mathbf{\Lambda}_x$ for critically damped behavior, as described in ref. [35]. The last term of (30) is designed such that the virtual fixtures are gradually vanished, as the system approaches complete autonomy ($\kappa \to 1$), since in complete autonomy their assistance is not further required. The variable $\beta_r \in [0, 1]$ implements the energy tank to keep the system passive, by further introducing a new auxiliary state $x_t \in \mathbb{R}_{\geq 0}$ representing the state of energy tank (i.e., the tank's level). Hence, β_r is designed such that $\beta_r = 1$ for values of x_t less than $\bar{x}_t$ and $\beta_r = 0$ when $\bar{x}_t$ is reached. The smooth transition of β_r between 1 and 0 is predefined by the designer (see ref. [5]). More specifically, let the following equation express the dynamic behavior of x_t:

$$\dot{x}_t = \frac{\mu(x_t)}{\mu_c}\tilde{\mathbf{v}}^T\mathbf{D}\tilde{\mathbf{v}} - \beta(x_t, v)v \tag{31}$$

where $\mu_c > 1$ is a constant parameter and $\mu(x_t)$, $\beta(x_t, v)$ are scalar functions which control the flow of energy between the virtual storage x_t and the robot:

$$\mu(x_t) = \begin{cases} 0, & \text{if } x_t \geq \bar{x}_t \\ h(x_t), & \text{otherwise} \end{cases} \tag{32}$$

$$\beta(x_t, v) = \begin{cases} 0, & \text{if } x_t \geq \bar{x}_t \wedge v \leq 0 \\ & \vee x_t = 0 \wedge v \geq 0 \\ g(x_t, v) & \text{otherwise} \end{cases}, \tag{33}$$

with $h(x_t)$, $g(x_t, v) \in [0, 1]$ being selected so that $\mu(x_t)$, $\beta(x_t, v)$ will be differentiable and

$$v = -\tilde{\mathbf{v}}^T\mathbf{K}\tilde{\mathbf{x}} - \kappa\,\tilde{\mathbf{v}}^T\mathbf{F}_v \tag{34}$$

being the power related to the variable stiffness gain $\kappa(t)$.

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

We are now ready to define β_r:

$$\beta_r(x_t, v) : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1], \text{ such that: } \begin{cases} \beta_r(x_t, v) = \beta(x_t, v), & \text{if } v \geq 0 \\ \beta_r(x_t, v) \geq \beta(x_t, v) & \text{otherwise} \end{cases} \tag{35}$$

Notice that by selecting $0 \leq x_t(0) \leq \bar{x}_t$, equations (31), (32), and (33) imply $x_t \in \mathcal{C}, \ \forall t > 0$ with $\mathcal{C} = [0, \bar{x}_t]$; thus, $\mathcal{C}$ is positively invariant with respect to (31). A specific selection of $h(x_t)$, $g(x_t, v)$, $\beta_r(x_t, v)$ can be found in the Appendix C.

2.5. Stability analysis

Substituting $\mathbf{F}_\tau$ from (30) in (27) yields the following closed loop robot dynamics:

$$\mathbf{\Lambda}_x(\mathbf{q})\dot{\widetilde{\mathbf{v}}} + (\mathbf{C}_x(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{D})\,\widetilde{\mathbf{v}} + \beta_r(x_t, v)\mathbf{K}\widetilde{\mathbf{x}} = \mathbf{F}_h + (1 - \kappa\beta_r(x_t, v))\mathbf{F}_v. \tag{36}$$

To proceed with the stability analysis, we initially write the closed loop system in state-space. We define the state vector $\mathbf{s} = [\widetilde{\mathbf{v}}^T \ \widetilde{\mathbf{x}}^T \ x_t]^T \in \mathcal{Z}$ with $\mathcal{Z} = \mathbb{R}^6 \times \mathbb{R}^6 \times \mathcal{C}$ and $\mathbf{F}_h$ is the input. The closed loop system can be written in the following compact form:

$$\dot{\mathbf{s}} = \mathbf{H}(\mathbf{s}, \mathbf{F}_h), \mathbf{s}_0 = \mathbf{s}(0) \in \mathcal{Z} \tag{37}$$

and

$$\mathbf{H}(\mathbf{s}, \mathbf{F}_h) = \begin{bmatrix} \mathbf{\Lambda}_x^{-1}\,(-\,(\mathbf{C}_x + \mathbf{D})\,\widetilde{\mathbf{v}} + \mathbf{F}_h - \beta_r(x_t, v)\mathbf{K}\widetilde{\mathbf{x}} + (1 - \kappa\beta_r(x_t, v))\mathbf{F}_v) \\ \mathbf{A}(\widetilde{\mathbf{x}})\widetilde{\mathbf{v}} \\ \dfrac{\mu(x_t)}{\mu_c}\widetilde{\mathbf{v}}^T\mathbf{D}\widetilde{\mathbf{v}} - \beta(x_t, v)v \end{bmatrix}.$$

Theorem 1. *The system dynamics* (37) *is strictly output passive under the exertion of a human force* $\mathbf{F}_h$ *with respect to the output* $\widetilde{\mathbf{v}}$, $\forall t \in [t_0, \infty)$.

Proof. Consider the continuously differentiable function:

$$V = \frac{\widetilde{\mathbf{v}}^T\mathbf{\Lambda}_x\widetilde{\mathbf{v}}}{2} + k_{vf}U + x_t \tag{38}$$

Taking its time derivative while utilizing $\mathbf{\Lambda}_x\dot{\widetilde{\mathbf{v}}}$ from (36) and employing the skew-symmetric property of $\dot{\mathbf{\Lambda}}_x - 2\mathbf{C}_x$, $\dot{V}$ becomes:

$$\dot{V} = -\widetilde{\mathbf{v}}^T\mathbf{D}\widetilde{\mathbf{v}} + \widetilde{\mathbf{v}}^T\mathbf{F}_v + \widetilde{\mathbf{v}}^T\mathbf{F}_h - \widetilde{\mathbf{v}}^T\beta_r\mathbf{K}\widetilde{\mathbf{x}} - \widetilde{\mathbf{v}}^T\beta_r\,\kappa\mathbf{F}_v + k_{vf}\left(\frac{\partial U}{\partial \widetilde{\mathbf{x}}}\right)^T\dot{\widetilde{\mathbf{x}}} + \dot{x}_t \tag{39}$$

Using (19) and the mapping of (21), $\dot{V}$ becomes:

$$\dot{V} = -\widetilde{\mathbf{v}}^T\mathbf{D}\widetilde{\mathbf{v}} + \widetilde{\mathbf{v}}^T\mathbf{F}_h - \widetilde{\mathbf{v}}^T\beta_r\mathbf{K}\widetilde{\mathbf{x}} - \widetilde{\mathbf{v}}^T\beta_r\,\kappa\mathbf{F}_v + \dot{x}_t \tag{40}$$

Utilizing $\dot{x}_t$ from (31), $\dot{V}$ becomes:

$$\dot{V} = -(1 - \frac{\mu}{\mu_c})\widetilde{\mathbf{v}}^T\mathbf{D}\widetilde{\mathbf{v}} + \widetilde{\mathbf{v}}^T\mathbf{F}_h - \beta_r(\widetilde{\mathbf{v}}^T\mathbf{K}\widetilde{\mathbf{x}} + \widetilde{\mathbf{v}}^T\kappa\mathbf{F}_v) - \beta v. \tag{41}$$

From (34) $-\widetilde{\mathbf{v}}^T\mathbf{K}\widetilde{\mathbf{x}} - \widetilde{\mathbf{v}}^T\kappa\mathbf{F}_v = v$. Since $(\beta_r - \beta)v \leq 0$ from (35) and $\mu \in [0, 1]$

$$\dot{V} \leq -\frac{\mu_c - 1}{\mu_c}\widetilde{\mathbf{v}}^T\mathbf{D}\widetilde{\mathbf{v}} + \widetilde{\mathbf{v}}^T\mathbf{F}_h \leq -\lambda_{min}(\mathbf{D})\frac{\mu_c - 1}{\mu_c}\widetilde{\mathbf{v}}^T\widetilde{\mathbf{v}} + \widetilde{\mathbf{v}}^T\mathbf{F}_h \tag{42}$$

where $\lambda_{min}(.)$ is the minimum eigenvalue of a matrix. Note that $\mu_c > 1$. Hence, system (37) is strictly output passive (see Definition 6.3 of ref. [36]). □

3. Experimental Evaluation

The proposed method is evaluated experimentally using a KUKA LWR4+ manipulator with a BH8 hand attached as an end-effector. A human is asked to teach a pick and place task to a robot by

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

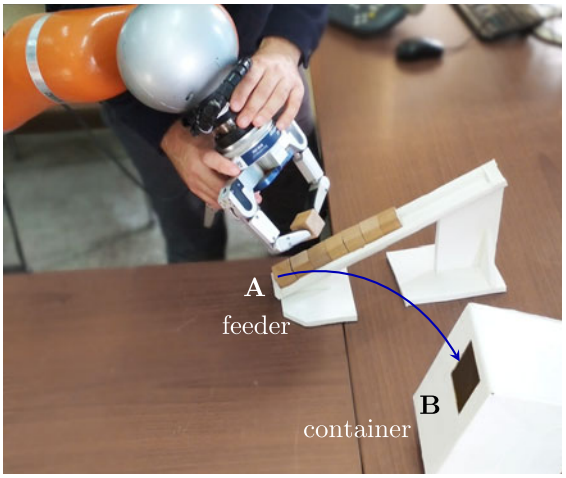

Fig. 4. The experimental setup showing the repetitive pick and place task of objects from point A (feeder) to point B (container).

repetitively moving small objects from a feeder (point A) to a container (point B), as it is shown in Fig. 4. The objective of this task is to seamlessly transition from kinesthetic teaching into autonomous operation. The task requires placing of objects into a container with specific orientation and high accuracy. To achieve it, two segments are required which are divided by the user tapping to an unused finger of the hand. Apart from segmenting the task, the tapping also changes the status of the hand (grasp/release). When a demonstration is completed by the user, the second demonstration starts immediately. Each segment is encoded with DMP right after the end of the demonstration. The encoding of the latest demonstrated task cycle is used as a reference for motion generation. To obtain the user's applied wrench $\mathbf{F}_h$, we use the estimation provided by the robot.

The control parameters are selected to be $\mathbf{M} = \mathbf{M}_6$, $n_y = \alpha_\tau = 100$ and $\delta = 10^{-4}$ for the synchronization, $\alpha_y = 10$, $\beta_y = \frac{\alpha_y}{4}$, $\alpha_x = 1$ for the DMP yielding a critically damped linear part response, $\bar{r} = 0.02$m, $\bar{\theta} = 10^o$ and $k_{vf} = 0.1$ for the virtual fixtures, $\lambda_1 = 0.02$, $\lambda_2 = 10$, $\gamma_1 = 3$, $\gamma_2 = 0.1$, $f_r = 4$, $f_{min} = 0.01$, $k_R = 100$ Nm/rad and $k_T = 2000$ N/m for the maximum stiffness values, and $\mu_c = 20$, $\bar{x}_t = 10$, $x_t(0) = 0.2$, $\delta_t = 0.1$, $\delta_v = 0.5$ for the energy tank. In the DMP temporal scaling, a continuous dead-band function of 8 mm and 5^o is utilized for rejecting small values of translation $\widetilde{\mathbf{p}}$ and orientation θ_e errors, respectively. A segment is finished and the next one begins when the goal position is within an accuracy of 10^{-4} m and the orientation error from the target is below 10^{-3} rad. The operation of the gripper switches to autonomous mode and re-encoding of demonstrations is stopped whenever $\kappa = 1$, that is, max stiffness is reached.

The experimental results of the pick and place task are illustrated in Fig. 5, where the time evolution of end-effector position $p(t)$, $p_d(t) \in \mathbb{R}$ in the x-axis with the corresponding evolution of the phase variable $x(t)$, the time scaling parameter $\tau(t)$, the state of energy tank $x_t(t)$, the variable stiffness parameter $\kappa(t)$, the interaction forces/torques $\|\mathbf{F}_P\|$, $\|\mathbf{F}_R\|$, and the control forces/torques $\|\mathbf{F}_{v,P}\|$, $\|\mathbf{F}_{v,R}\|$ induced by the virtual fixtures ($\mathbf{F}_v = [\mathbf{F}_{v,P}^T \, \mathbf{F}_{v,R}^T]^T$) are presented. During the initial demonstration, until $t = t_1 \approx 12$ s, the user freely moves the end-effector, demonstrating the task to the robot. After the end of the first task cycle ($t > t_1$) the DMP produces the encoded previous path, and the virtual fixtures assist the user in repeating the demonstration until the robot becomes able to execute the task autonomously, when κ reaches the value of 1. Full autonomy is achieved in three demonstrations, as it is shown in Fig. 5(e). The relatively increased forces (Fig. 5(f) and (g)) that appear approximately at $t = 17$ s and $t = 30$ s do not induce tracking errors (Fig. 5(a)) due to the proposed synchronization mechanism (Fig. 5(c)). The effect of synchronization is depicted in the temporal scaling factor that is displayed normalized to the duration of the previous demonstration τ_s (Fig. 5(c)), so that values greater than one means that the DMP is slowing down and lower than one that is speeding up. The demonstrations and the final learned path are presented in Fig. 5(h), showing that a small variance between the demonstrations is achieved owing to the assistance of the virtual fixtures, which accelerate the learning procedure.

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

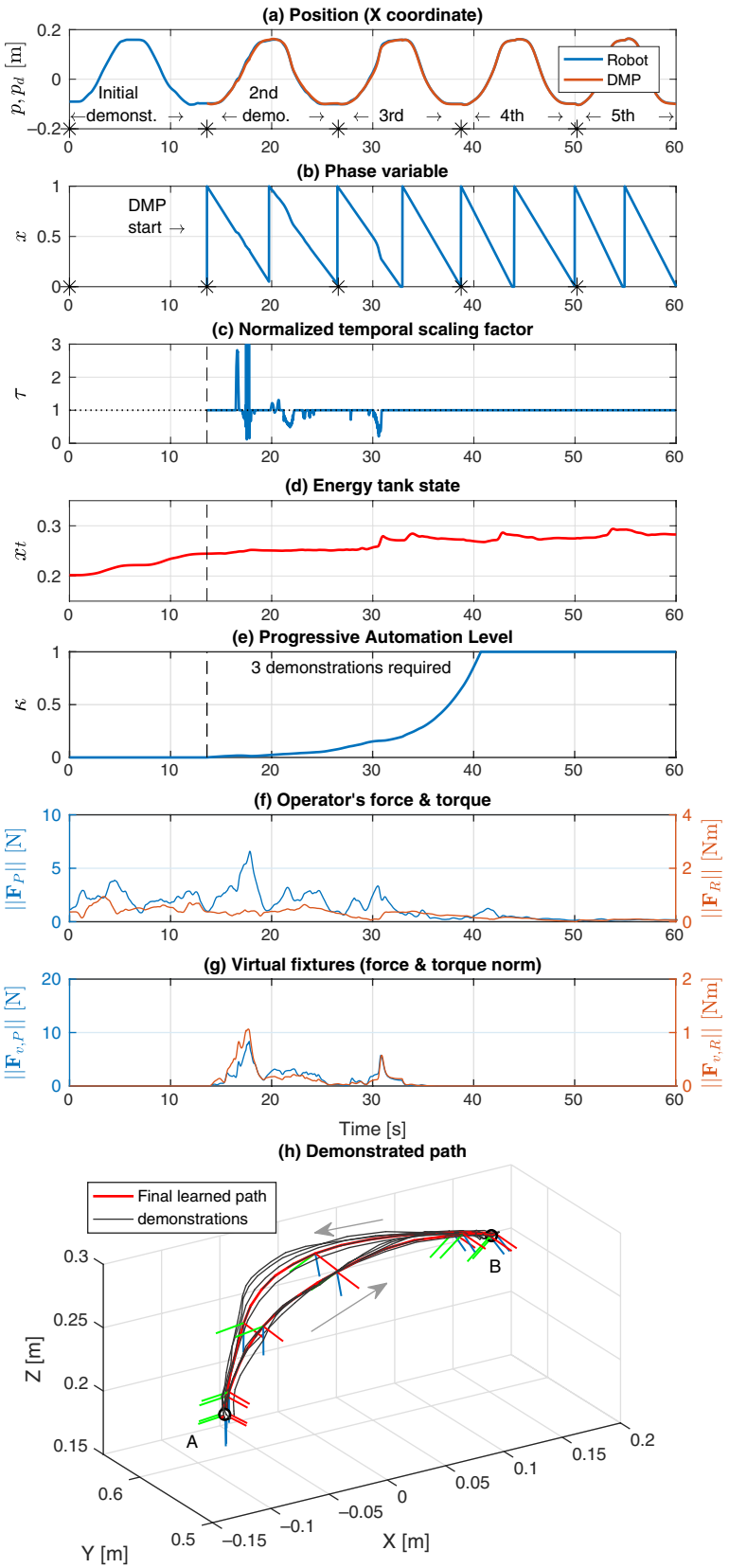

Fig. 5. Demonstration of a pick and place task with DMP synchronization and assistive virtual fixtures that lead to the learning of the task within three demonstrations.

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

The modification of a part of a motion segment is presented in Fig. 6(a), where the user interrupts the autonomous execution of the robot at $t \approx 74$ s by applying relatively high forces/torques (Fig. 6(f)). Notice the fast stiffness reduction (Fig. 6(e)) which allows the virtual fixture's introduction, although they are shortly penetrated as shown by the peak at $t \approx 74$ s in Fig. 6(g) and the subsequent zero value during the modification. The drop of the stiffness causes an increase of the energy tank level (Fig. 6(d)), which however remains within the boundaries. At $t \approx 81$ s, the user returns to the previous trajectory and completes the demonstration of the task. A second peak at the virtual fixtures at $t \approx 81$ s (Fig. 6(g)) signals the re-entrance of the fixtures. The magnitude of the control signal $\mathbf{F}_v$ in Fig. 6(g) is representative of the guidance provided to the user as a haptic feedback during the rest of the demonstration. In Fig. 6(b) and (c), the phase variable x and the temporal scaling factor τ are depicted, respectively, where one can see the adaptation of τ that directly affects the rate of change of x. Due to the adaptation of τ, the synchronization of the robot to the user's motion is achieved even when the virtual fixtures are penetrated, which allows the user to re-enter the fixture sphere at $t \approx 81$ s. The fixtures assist the user in repeating the rest of the previously learned motion, while no further modifications are applied. After the re-entrance into the fixture sphere, the progressive automation level starts increasing again (Fig. 6(e)), due to the relatively small tracking error and the low forces/torques. On completion of the task cycle at $t = 100$ s, the DMP is retrained to the modified demonstration.

The effectiveness of the proposed virtual fixtures in assisting the user to quickly achieve progressive automation is particularly evident in the experimental comparison of Fig. 7, where the user demonstrates the same pick and place task but without the feedback of virtual fixtures. In this case, even though the DMP remains synchronized during learning (Fig. 7(a)), the user cannot easily follow the previously demonstrated path, leading to relatively higher tracking error, as compared to the case in which a virtual fixture is utilized, and slower increase of the stiffness (Fig. 7(b)). Without using virtual fixtures, the robot is able to execute the task autonomously after six demonstrations (70 s) compared to three demonstrations (40 s) that are required with the assistance of fixtures (Fig. 5(e)). Moreover, without using virtual fixtures the external forces/torques over time in Fig. 7(c) are significantly increased compared to Fig. 5(f), meaning that more human effort is required to demonstrate the task.

To illustrate the benefits of the proposed synchronization method, we present in Fig. 8 an additional experiment comparing the proposed synchronization that considers both position and orientation with synchronization that is based only in the position. The task involves a translational movement, followed by a pure rotational movement and then followed by a translational again. In Fig. 8, an indicative translational and rotational coordinate is presented along with the corresponding position and orientation errors, the progressive automation level k, and the phase variable x. If the orientation is not considered for synchronization, the orientation of the DMP evolves according to the learned demonstration and is not synchronized with the current demonstration which might be faster or slower, leading to higher orientation errors (Fig. 8(f) and (g)) and slower progressive automation (Fig. 8(h)). By considering both the position and the orientation, the DMP is synchronized to the demonstration from the human and a small tracking error is maintained (Fig. 8(b) and (c)), achieving faster progressive automation (Fig. 8(d)).

4. Conclusions

In this paper, we built upon the progressive automation framework by introducing temporal synchronization of the DMP with the current demonstration, considering both the translation and the orientation of the robot, and by imposing virtual fixtures to assist the user in following the reference trajectory. In case a modification to the trajectory is desired, the fixtures can be penetrated but the user can re-enter into them after the modification. A variable stiffness strategy gradually increases the target stiffness according to the synchronized tracking error and interaction forces and torques, so that when maximum stiffness is reached, the robot can autonomously execute the learned trajectory. To retain passivity of the overall system we use an energy tank approach. The effectiveness of the framework is demonstrated in an experimental evaluation where a human demonstrates to the robot a repetitive pick and place task. With the proposed approach, we found that the robot is able to learn the task after three demonstrations which take less than a minute. The importance of incorporating the virtual fixtures in the methodology is revealed by the fact that for the same task and in the

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

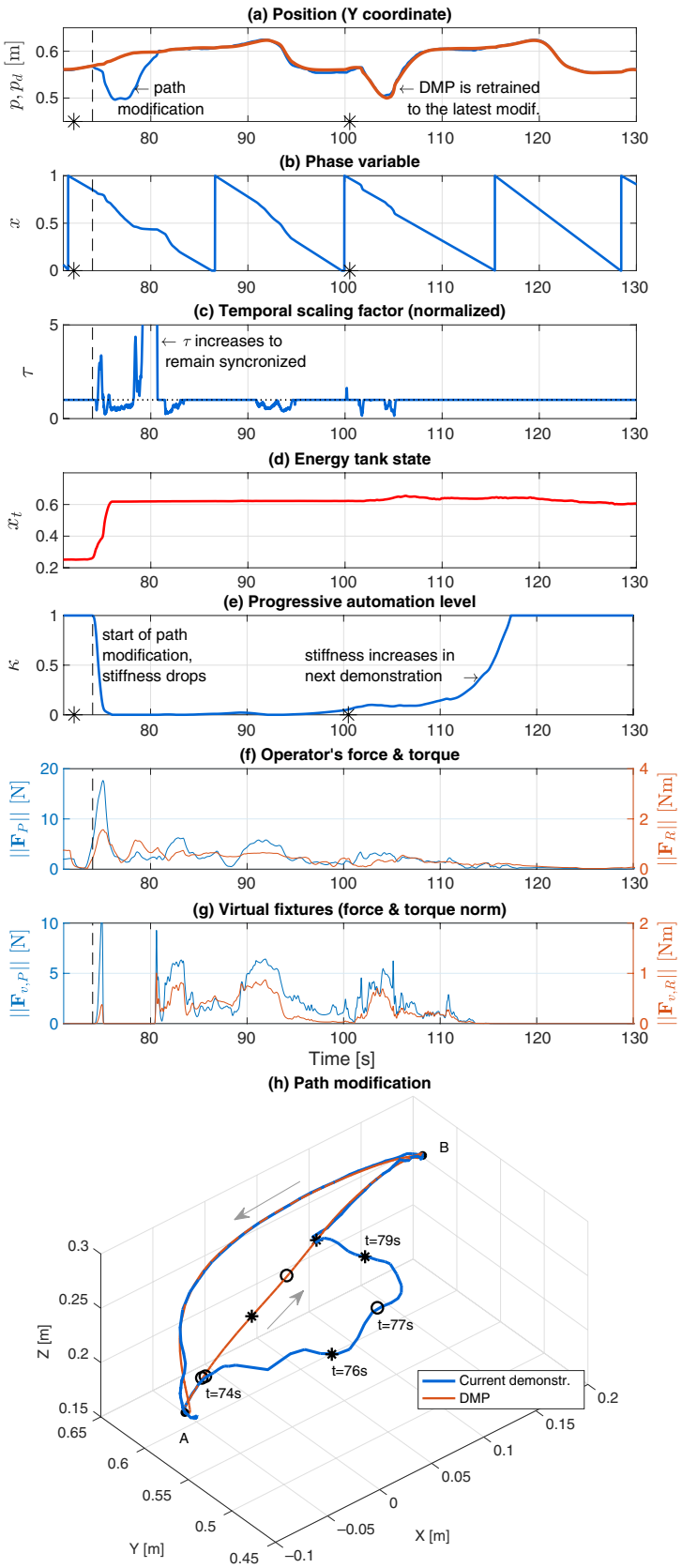

Fig. 6. Modification of the learned path in the segment from A to B while the DMP remains synchronized.

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

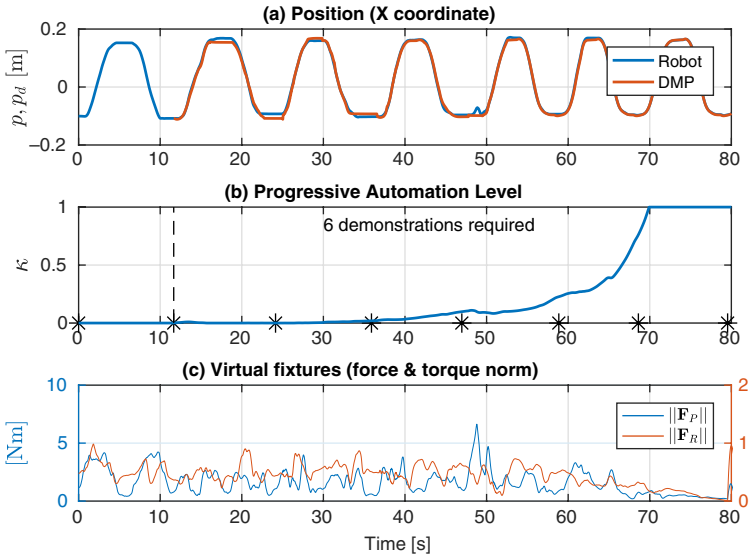

Fig. 7. Demonstration of a pick and place task without virtual fixtures in comparison with Fig. 5. The robot's position in the X coordinate, the progressive automation level, and user's applied forces/torques are illustrated.

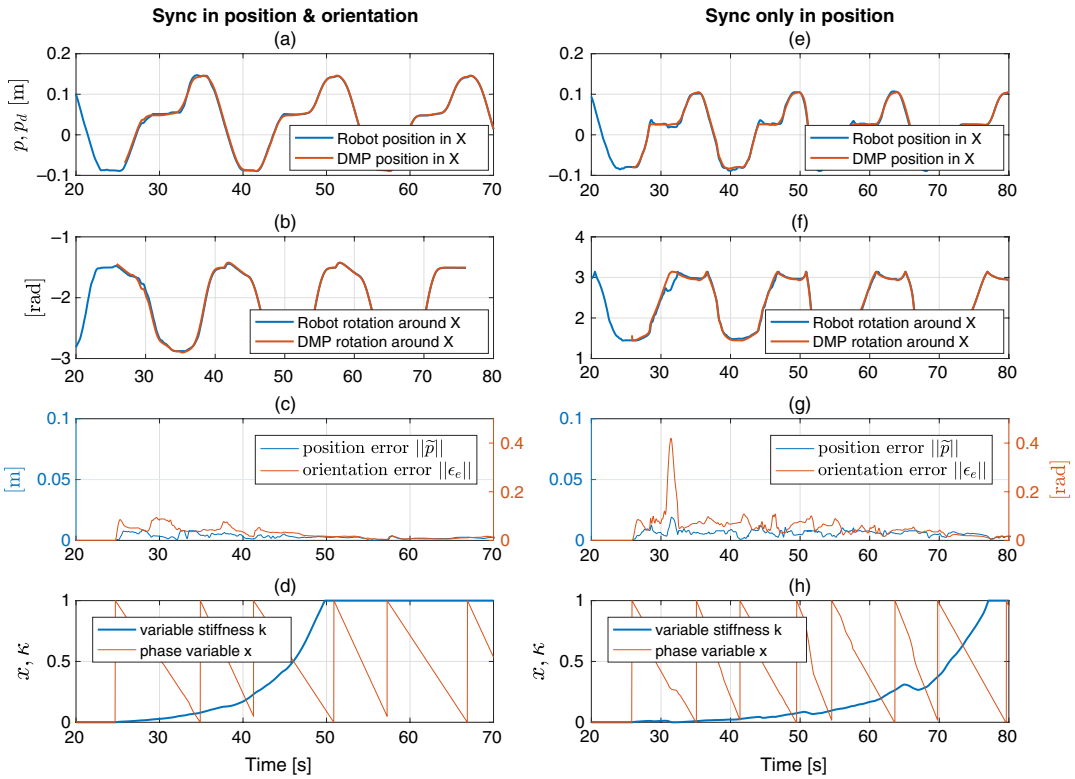

Fig. 8. Comparison between the proposed synchronization method with a synchronization that considers only the position.

absence of fixtures, the same user needs six demonstrations. Moreover, in an additional experiment a segment of the task is significantly modified by the user, while the robot is autonomously executing a learned task, demonstrating the penetration of the virtual fixtures, the successful DMP synchronization, and the quick role inversion through the variable stiffness strategy. Hence, it is suggested that the synergic action of these proposed methods results in an easy to use technique for quick programming of repetitive motion tasks.

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

Acknowledgements
This research is implemented through the Operational Program "Human Resources Development, Education and Lifelong Learning" and is co-financed by the European Union (European Social Fund) and Greek national funds.

References

1. Z. Pan, J. Polden, N. Larkin, S. Van Duin and J. Norrish, "Recent progress on programming methods for industrial robots," *Robot Cim.-Int. Manuf.* **28**(2), 87–94 (2012).
2. F. Dimeas, F. Fotiadis, D. Papageorgiou, A. Sidiropoulos and Z. Doulgeri, "Towards Progressive Automation of Repetitive Tasks Through Physical Human-Robot Interaction," **In:** *Human Friendly Robotics. Springer Proceedings in Advanced Robotics* (F. Ficuciello, F. Ruggiero, and A. Finzi, eds.) (Springer, Cham, 2019) pp. 151–163.
3. A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artif. Intell.* **172**(6–7), 716–737 (2008).
4. F. Dimeas and Z. Doulgeri, "Progressive Automation of Repetitive Tasks Involving both Translation and Rotation," **In:** *Advances in Service and Industrial Robotics. RAAD 2018. Mechanisms and Machine Science* (N. Aspragathos, P. Koustoumpardis and V. Moulianitis, eds.), vol. 67 (Springer, Cham, 2019) pp. 53–62.
5. T. Kastritsi, F. Dimeas and Z. Doulgeri, "Progressive automation with DMP synchronization and variable stiffness control," *IEEE Robot. Autom. Lett.* **3**(4), 3789–3796 (2018).
6. T. Kastritsi, A. Sidiropoulos, and Z. Doulgeri, "A pHRI Framework for Modifying a Robot's Kinematic Behaviour via Varying Stiffness and Dynamical System Synchronization," *26th Mediterranean Conference on Control and Automation (MED)*, Zadar, Croatia (2018) pp. 33–38.
7. A. G. Billard, S. Calinon and R. Dillmann, "Learning from Humans," **In:** *Springer Handbook of Robotics* (Springer International Publishing, Cham, 2016) pp. 1995–2014.
8. D. A. Abbink, M. Mulder and E. R. Boer, "Haptic shared control: Smoothly shifting control authority?" *Cognit. Technol. Work* **14**(1), 19–28 (2012).
9. Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua and D. K. Limbu, "Continuous role adaptation for human-robot shared control," *IEEE Trans. Robot.* **31**(3), 672–681 (2015).
10. 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," *IEEE International Workshop on Robot and Human Interactive Communication*, Paris (2012) pp. 962–967.
11. M. Tykal, A. Montebelli and V. Kyrki, "Incrementally Assisted Kinesthetic Teaching for Programming by Demonstration," *ACM/IEEE International Conference on Human-Robot Interaction*, Christchurch (2016) pp. 205–212.
12. M. Saveriano, S. I. An and D. Lee, "Incremental Kinesthetic Teaching of End-Effector and Null-Space Motion Primitives," *IEEE International Conference on Robotics and Automation*, Seattle, WA (2015) pp. 3570–3575.
13. N. Jarrasse, V. Sanguineti and E. Burdet, "Slaves no longer: Review on role assignment for human-robot joint motor action," *Adapt. Behav.* **22**(1), 70–82 (2013).
14. K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Trans. Robot.* **32**(5), 1298–1305 (2016).
15. F. Ferraguti, C. Secchi and C. Fantuzzi, "A Tank-Based Approach to Impedance Control with Variable Stiffness," *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany (2013) pp. 4948–4953.
16. K. Kronander and A. Billard, "Passive interaction control with dynamical systems," *IEEE Robot. Autom. Lett.* **1**(1), 106–113 (2016).
17. S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Trans. Robot.* **27**(5), 943–957 (2011).
18. P. Pastor, M. Kalakrishnan, F. Meier, F. Stulp, J. Buchli, E. Theodorou and S. Schaal, "From dynamic movement primitives to associative skill memories," *Robot. Auton. Syst.* **61**(4), 351–361 (2013).
19. A. J. Ijspeert, J. Nakanishi and S. Schaal, "Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots," **In:** *Proceedings 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, USA (2002) pp. 1398–1403.
20. H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer and J. Peters, "Interaction primitives for human-robot cooperation tasks," *IEEE International Conference on Robotics and Automation*, Hong Kong, China (2014) pp. 2831–2837.
21. A. Pervez, A. Ali, J. H. Ryu and D. Lee, "Novel Learning from Demonstration Approach for Repetitive Teleoperation Tasks," *2017 IEEE World Haptics Conference, WHC 2017*, Munich, Germany (2017) pp. 60–65.
22. J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robot. Auton. Syst.* **47**(2–3), 79–91 (2004).
23. S. Kim, E. Gribovskaya and A. Billard, "Learning Motion Dynamics to Catch a Moving Object," **In:** *2010 10th IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN (2010) pp. 106–111.
24. J. Abbott, P. Marayong and A. Okamura, "Haptic Virtual Fixtures for Robot-Assisted Manipulation," **In:** *Springer Tracts in Advanced Robotics* (S. Thrun, R. Brooks and H. Durrant-Whyte, eds.) (2007) pp. 49–64.
25. L. B. Rosenberg, "The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments.," *Technical Report*, DTIC Document (1992).

26. J. E. Colgate, M. Peshkin and S. H. Klostermeyer, "Intelligent Assist Devices in Industrial Applications: A Review," **In:** *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, Las Vegas, NV (2003) pp. 2516–2521.
27. H. C. Lin, K. Mills, P. Kazanzides, G. D. Hager, P. Marayong, A. M. Okamura and R. Karam, "Portability and applicability of virtual fixtures across medical and manufacturing tasks," **In:** *Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006*, Orlando, FL (2006) pp. 225–230.
28. A. Bettini, P. Marayong, S. Lang, A. M. Okamura and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *IEEE Trans. Robot.* **20**(6), 953–966 (2004).
29. S. A. Bowyer and F. R. y Baena, "Dissipative control for physical human–robot interaction," *IEEE Trans. Robot.* **31**(6), 1281–1293 (2015).
30. D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Auton. Robots* **31**(2–3), 115–131 (2011).
31. S. S. Restrepo, G. Raiola, P. Chevalier, X. Lamy and D. Sidobre, "Iterative Virtual Guides Programming for Human-Robot Comanipulation," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, Munich, Germany (2017) pp. 219–226.
32. A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor and S. Schaal, "Dynamical Movement Primitives: Learning attractor models for motor behaviors," *Neural Computat.* **25**(2), 328–373 (2013).
33. A. Ude, B. Nemec, T. Petrič and J. Morimoto, "Orientation in Cartesian Space Dynamic Movement Primitives," *IEEE International Conference on Robotics and Automation*, Hong Kong, China (2014) pp. 2997–3004.
34. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics: Modelling, Planning and Control* (Springer, Heidelberg, 2009).
35. C. Ott, "Cartesian Impedance Control of Redundant and Flexible-Joint Robots," **In:** *Springer Tracts in Advanced Robotics* (B. Siciliano, O. Khatib, and F. Groen, eds.), vol. 49 (2008).
36. H. Khalil, *Nonlinear Systems* (Prentice Hall, Upper Saddle River, 2002).
37. S. Schaal and C. Atkeson, "Constructive incremental learning from only local information," *Neural Comput.* **10**(8), 2047–2084 (1998).
38. J. J. Kuffner, "Effective Sampling and Distance Metrics for 3D Rigid Body Path Planning," **In:** *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004*, New Orleans, LA (2004) pp. 3993–3998.

Appendix

A. DMP Details

The forcing terms $\mathbf{f}_p$, $\mathbf{f}_o$ of the DMP are given by the weighted sum of N Gaussian kernels:

$$\mathbf{f}_p(x) = \mathbf{D}_p \frac{\sum_{i=1}^{N} \mathbf{w}_p^i \psi_i(x)}{\sum_{i=1}^{N} \psi_i(x)} x, \;\; \mathbf{f}_o(x) = \mathbf{D}_o \frac{\sum_{i=1}^{N} \mathbf{w}_o^i \psi_i(x)}{\sum_{i=1}^{N} \psi_i(x)} x,$$

where $\psi_i = e^{(-h_i(x-c_i)^2)}$ with h_i being the inverse width and c_i the center of the i_{th} Gaussian kernel. The position and orientation scaling factor are defined as $\mathbf{D}_p = \text{diag}(\mathbf{g} - \mathbf{p}_{d^0})$ and $\mathbf{D}_o = \text{diag}(2\log(\mathbf{Q}_g * \mathbf{Q}_{d^0}^{-1}))$, respectively. The terms $\mathbf{p}_{d^0}$, $\mathbf{Q}_{d^0}$ represent the starting position and orientation of the DMP and $\mathbf{w}_p$, $\mathbf{w}_o \in \mathbb{R}^{3\times N}$ are the weight matrices of the kernels, which are calculated with locally weighted regression.[37]

The logarithmic map $\log(\mathbf{Q})$ of a quaternion $\mathbf{Q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix}$ is defined as:

$$\log(\mathbf{Q}) = \begin{cases} \arccos(\eta) \frac{\boldsymbol{\epsilon}}{||\boldsymbol{\epsilon}||}, & \boldsymbol{\epsilon} \neq 0 \\ [0, 0, 0]^T, & \boldsymbol{\epsilon} = 0 \end{cases}.$$

Integration of (4) is performed using the exponential map that maintains the unit norm during the integration with step Δt:

$$\mathbf{Q}_d(t + \Delta t) = \exp\left(\frac{\Delta t}{2} \frac{\boldsymbol{\omega}_s(t)}{\tau}\right) * \mathbf{Q}_d(t). \tag{A1}$$

B. Distance Metric in $SE(3)$

Let $\mathbf{r}_1 = [\mathbf{p}_1^T \; \mathbf{Q}_1^T]^T$, $\mathbf{r}_2 = [\mathbf{p}_2^T \; \mathbf{Q}_2^T]^T \in \mathcal{T}$ be two poses, where $\mathcal{T} \triangleq \mathbb{R}^3 \times \mathbb{S}^3$ the special Euclidean space $SE(3)$ expressed as a combination of Cartesian coordinates $\mathbf{p} \in \mathbb{R}^3$ (for translation) and unit

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press

quaternions $\mathbf{Q} \in \mathbb{S}^3$ (for orientation). The following metric $\zeta(\mathbf{r}_1, \mathbf{r}_2) : \mathcal{T} \times \mathcal{T} \to \mathbb{R}_{\geq 0}$ is proposed:

$$\zeta(\mathbf{r}_1, \mathbf{r}_2) = \alpha_\zeta f(\mathbf{p}_1, \mathbf{p}_2) + \beta_\zeta g(\mathbf{Q}_1, \mathbf{Q}_2), \tag{A2}$$

representing the weighted generalized distance in $SE(3)$, with $\alpha_\zeta, \beta_\zeta \in \mathbb{R}_{>0}$ being the weights and $f : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}_{\geq 0}$, $g : \mathbb{S}^3 \times \mathbb{S}^3 \to \mathbb{R}_{\geq 0}$ being metrics in $\mathbb{R}^3$ and $\mathbb{S}^3$, respectively. Hence, f, g should have the following properties:

- $f(\mathbf{p}_1, \mathbf{p}_2), g(\mathbf{Q}_1, \mathbf{Q}_2) \geq 0$ for all $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{T}$,
- $f(\mathbf{p}_1, \mathbf{p}_2) = 0 \Leftrightarrow \mathbf{p}_1 = \mathbf{p}_2$ and $g(\mathbf{Q}_1, \mathbf{Q}_2) = 0 \Leftrightarrow \mathbf{Q}_1 = \mathbf{Q}_2$,
- $f(\mathbf{p}_1, \mathbf{p}_2) = f(\mathbf{p}_2, \mathbf{p}_1)$ and $g(\mathbf{Q}_1, \mathbf{Q}_2) = g(\mathbf{Q}_2, \mathbf{Q}_1)$,
- $f(\mathbf{p}_1, \mathbf{p}_2) \leq f(\mathbf{p}_1, \mathbf{p}_3) + f(\mathbf{p}_2, \mathbf{p}_3)$ and $g(\mathbf{Q}_1, \mathbf{Q}_2) \leq g(\mathbf{Q}_1, \mathbf{Q}_3) + g(\mathbf{Q}_2, \mathbf{Q}_3)$, for any $\mathbf{r}_3 = [\mathbf{p}_3^T \ \mathbf{Q}_3^T]^T \in \mathcal{T}$.

Common eligible metrics in translation are the Euclidean norm of the difference $f(\mathbf{p}_1, \mathbf{p}_2) \triangleq \|\mathbf{p}_1 - \mathbf{p}_2\|$, or the squared Euclidean distance $f(\mathbf{p}_1, \mathbf{p}_2) \triangleq \|\mathbf{p}_1 - \mathbf{p}_2\|^2$, while in orientation some eligible metrics are the geodesic distance $g(\mathbf{Q}_1, \mathbf{Q}_2) \triangleq \theta_e(\mathbf{Q}_1, \mathbf{Q}_2) = 2\cos^{-1}(\mathbf{Q}_1^T\mathbf{Q}_2)$, or its transformations $g(\mathbf{Q}_1, \mathbf{Q}_2) \triangleq 1 - \cos(\frac{\theta_e}{2})$ or $g(\mathbf{Q}_1, \mathbf{Q}_2) \triangleq \sin(\frac{\theta_e}{2})$, considering $\theta_e \in [0, \pi]$ [38].

C. Smooth Functions

In ref. [16], one possible choice of the scalar functions $h(x_t)$, $g(x_t, v)$, and $\beta_r(x_t, v)$ is provided that satisfy the equations (32), (33), and (35). For defining these scalar functions, the following smooth step function is used:

$$z(y; \overline{n}, \underline{n}) = \begin{cases} 0 & \text{if } y < \underline{n} \\ 6c(y)^5 - 15c(y)^4 + 10c(y)^3 & \text{if } \underline{n} \leq y \leq \overline{n} \\ 1 & \text{if } y > \overline{n} \end{cases}, \tag{A3}$$

where $c(y) = \frac{y - \underline{n}}{\overline{n} - \underline{n}}$. The complement of the smooth step function defines the function $h(x_t)$:

$$h(x_t) = 1 - z(x_t; (\overline{x}_t - \delta_t), \overline{x}_t), \tag{A4}$$

The function $g(x_t, v)$ is given by:

$$\begin{aligned} g(x_t, v) =& 1 - z(v; -\delta_v, 0)(1 - z(x_t; 0, \delta_t)) \\ & - (1 - z(v; 0, \delta_v))z(x_t; (\overline{x}_t - \delta_t), \overline{x}_t) \end{aligned} \tag{A5}$$

and the function $\beta_r(x_t, v)$ by:

$$\begin{aligned} \beta_r(x_t, v) =& (1 - z(v; -\delta_v, 0)(1 - z(x_t; 0, \delta_t)))(1 \\ & - z(v; -\delta_v, 0)(1 - z(v; 0, \delta_v))z(x_t; (\overline{x}_t - \delta_t), \overline{x}_t) \end{aligned} \tag{A6}$$

where $\delta_t > 0$ and $\delta_v > 0$ are parameters regulating the smooth transition.

https://doi.org/10.1017/S0263574719001437 Published online by Cambridge University Press