Robotica: (2020) volume 38, pp. 1867–1879. © Cambridge University Press 2019
doi:10.1017/S0263574719001449

Speed Adaptation in Learning from Demonstration through Latent Space Formulation

Maria Koskinopoulou†‡*, Michail Maniadakis†
and Panos Trahanias†‡

†*Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), Heraklion, Greece*
‡*Department of Computer Science, University of Crete, Heraklion, Crete, Greece*

(Accepted September 20, 2019. First published online: October 17, 2019)

SUMMARY

Performing actions in a timely manner is an indispensable aspect in everyday human activities. Accordingly, it has to be present in robotic systems if they are going to seamlessly interact with humans. The current work addresses the problem of learning both the spatial and temporal characteristics of human motions from observation. We formulate learning as a mapping between two worlds (the observed and the action ones). This mapping is realized via an abstract intermediate representation termed "Latent Space." Learned actions can be subsequently invoked in the context of more complex human–robot interaction (HRI) scenarios. Unlike previous learning from demonstration (LfD) methods that cope only with the spatial features of an action, the formulated scheme effectively encompasses spatial and temporal aspects. Learned actions are reproduced under the high-level control of a time-informed task planner. During the implementation of the studied scenarios, temporal and physical constraints may impose speed adaptations in the reproduced actions. The employed latent space representation readily supports such variations, giving rise to novel actions in the temporal domain. Experimental results demonstrate the effectiveness of the proposed scheme in the implementation of HRI scenarios. Finally, a set of well-defined evaluation metrics are introduced to assess the validity of the proposed approach considering the temporal and spatial consistency of the reproduced behaviors.

KEYWORDS: Human–robot interaction; Learning from demonstration; Artificial temporal cognition; Temporal planning; Latent representation.

1. Introduction

Contemporary learning from demonstration (LfD) methods have been widely employed for robotic imitation of human action behaviors,[1,2] with corresponding implementations in various application domains. In our recent work,[3] we developed and evaluated an LfD framework, termed IMitation Framework by Observation (IMFO), that is based on the compact, low-dimensional representation of both human and robot arm motions, which are properly associated with facilitate learning. Similar approaches have also been studied in refs. [4–6]. Interestingly, the compressed representation of actions, in the so-called latent space, preserves the significant properties of the actions' spatial trajectories and facilitates mapping between an observed human action and the reproduced robotic one.[7,8] Accordingly, it allows robots to learn human-like behavioral acts by observing human demonstrators, based on a compact association of the two acting worlds, the observed and the reproduced one.

[*] Corresponding author. E-mail: mkosk@ics.forth.gr

Despite the significant number of works exploring LfD, there are important parameters of action implementation that have not been sufficiently studied yet. In particular, the role of temporal information in the computational representation and reproduction of actions remains poorly understood.

The execution time of activities is directly linked to the speed of action performance. The ability to adjust speed is fundamental for humans, allowing to deal with cases where the physical properties of manipulated objects impose constraints (e.g., slow down to move a glass full of water), or emergency-like situations (e.g., speed up to accomplish a goal within certain time constraints). It is therefore important to study how spatio-temporal variations affect the representation of actions considered in LfD scenarios.

To this end, the main purpose of this work is to develop a novel LfD methodological framework that relies on the compact representation of the observed behaviors and greatly facilitates the learning of both spatial and temporal characteristics of the observed actions. Hence, it focuses on speed adaptation in the context of IMFO, considering how speed shapes the low-dimensional latent representation of the demonstrated actions. Specifically, the actions considered in the current study are arm motions, although the introduced formulation can be readily generalized to other human actions. In short, the proposed method regards augmenting the algorithm that implements the transformation from the full configuration space to the compact latent space, with temporal information that affects execution speed. The compressed representation of similar actions with different spatio-temporal characteristics shows that speed plays a major role in the derived latent representation, effectively separating similar arm motions that are executed at different speeds. Accordingly, the latter actions assume unambiguous latent space representations when only speed of execution varies, allowing thus the accurate reproduction of acts with different velocities.

The proposed spatio-temporal formulation of IMFO readily lends itself to integration with time-informed planning approaches[9] to effectively address temporal constraint satisfaction in real-world scenarios. Following our previous work on latency estimation,[10] the composite system is capable to cope with cases where completion of certain behaviors is expected to be delayed. The elimination of latency is accomplished through the estimation of a requested, reduced time for action completion. Taking advantage of LfD actions at different speeds, we select the action implementation that best matches the requested completion time, therefore facilitating the realization of the composite behavior within the predefined time limits.

The rest of the paper is organized as follows. Section 2 presents a brief literature review, and then Section 3 describes the proposed LfD formulation and the time-informed planning. Experimental results obtained with a robotic arm are presented in Section 4, and the paper concludes with a summary and suggestions for future work in Section 5.

2. Related Work

LfD has become an important topic in robotics research with notable applications in relevant sectors, such as motion behaviors, human–robot interaction, artificial intelligence, and goal-based learning.[11–14] In the current section, we briefly review and highlight representative works from contemporary literature. Accordingly, Gupta et al.[15] proposed an algorithm for policy learning and generalization that allows complex dexterous manipulators to learn from multiple human demonstrations. Additionally, Li et al. in ref. [1] learn grasp adaptation through experience and tactile sensing, whereas Kaelbling in ref. [16] exploited hidden Markov models to plan and act in partially observable stochastic domains. Evrard et al.[17] adopted Gaussian mixture models to represent the variance over time in the demonstrated trajectories to teach a humanoid physical collaborative task.

Recently, we established an LfD methodological framework for robots to expand their repertoire of actions, by formulating a mapping between two worlds, the observed and the action one.[3] This mapping is realized via an abstract intermediate representation, termed *Latent Space*. Latent representations, commonly derived from dimensionality reduction methods, have been successfully applied in various application domains, such as computer vision for body pose tracking,[18] computer graphics for graphical model representation (character animation),[19,20] and machine learning for missing data formation.[21] Additionally, LfD via latent space representation has been successfully applied for human–robot collaborative task execution in ref. [22].

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

Interestingly, the topic of speed adaptation of executed actions has received rather limited attention. In ref. [5] the high-level temporal alignment of demonstrated actions is used to guide trajectory generation in the actor space. In addition, in ref. [6] actions learned through a slow demonstration procedure are gradually self-improved to accomplish speed adaptation in action execution. Furthermore, to the best of our knowledge, LfD and associated latent representations that compactly describe spatio-temporal features have not been extensively studied to-date in a temporal context. However, indicative works, such as those of Calinon et al. in refs. [5] and [23] have set the basis, by formulating the well-known dynamical movement primitives (DMPs), learning time, and space constraints during a task. Additional works[24–27] by Ijspeert et al., Rozo et al., and Ewertoneta et al. have also investigated the implementation of DMPs within multiple conditions to reproduce force-based manipulation tasks and learn reactive and proactive behaviors in human–robot collaboration.

Execution of learned actions is in many cases affected by temporal constraints that are externally imposed. Processing of such constraints in planning problems has been typically based on simple temporal networks, which are mapped on the equivalent distance graphs, to verify the existence of no negative cycles and thus prove the consistency and dispatchability of the plan.[28] Along this line, recent works have considered back propagation rules to dynamically preserve dispatchability of plans,[29] as the implementation of the plan proceeds and time constraints are updated. However, these works focus on specifying the start moment of a given action in relation to the others, without considering the role of speed adaptation on action synchronization and temporal constraint satisfaction. To address this issue, we have recently proposed an interval calculus approach to estimate the expected latency of actions which directs informed adaptations on action implementation.[10] The present work aims at combining the estimation of expected latency and the concise specification of a revised execution time, with the ability to learn by observation actions demonstrated at different speeds, in order to implement an enhanced composite system that is capable to effectively comply with dynamically changing temporal constraints.

3. Spatio-Temporal Representation of Actions

In this section, we formulate the methodological framework for action learning and representation in the spatio-temporal domain. The latter effectively facilitates action reproduction in a given scenario, and action execution at a speed different than the learned one. Action learning is accomplished via a latent representation of observed actions. Such a representation achieves to compactly depict pertinent action information and abstract from the actual kinematic configuration of a system, for example, human demonstrator or robotic platform. Herewith, we extent the IMFO methodological framework, introduced in ref. [3], in order to cope with the temporal aspects of human arm motions.

3.1. Learning phase

A conceptual representation of the employed methodology is depicted in Fig. 1. Henceforth, and without loss of generality, we consider only arm motions. The spatial representation of such actions is readily available from the trajectory of the parameters in the arm's configuration space. The latter parameters that convey spatial information of the executed action are augmented with timestamps that signify time of occurrence of the particular snapshot in the action trajectory. Effective transformation of the spatial (configuration space) and temporal (timestamp) parameters into a latent (compact) space establishes a unified spatio-temporal representation of the considered action, which enables robots to execute actions at different speeds. With the above representation in place, learning is accomplished by establishing an association across the two latent representations that correspond to the human action and the robotic one. This is presented in detail in the following.

3.1.1. Data acquisition. Let an arm motion trajectory performed by a human actor and the corresponding one by the robotic system. The former describes a trajectory in a 11D configuration space, that is, three joints of three coordinates each, one grasp parameter, and one timestamp value. This is termed as human action space (HAS). The coordinates for the three joints in HAS are computed as the center points of relevant joints. For that a model-based approach has been adopted, analogous to the approach followed in ref. [30]. Similarly, we obtain an 8D representation for robot arm actions

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

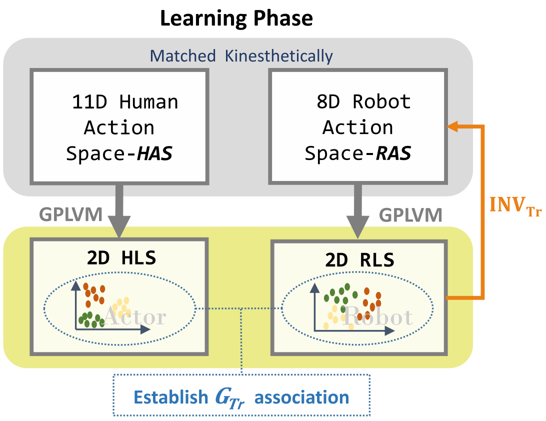
Learning Phase
Matched Kinesthetically
11D Human Action Space-HAS
8D Robot Action Space-RAS
GPLVM
GPLVM
INV$_{Tr}$
2D HLS
2D RLS
Actor
Robot
Establish G_{Tr} association

Fig. 1. Schematic overview of the learning process.

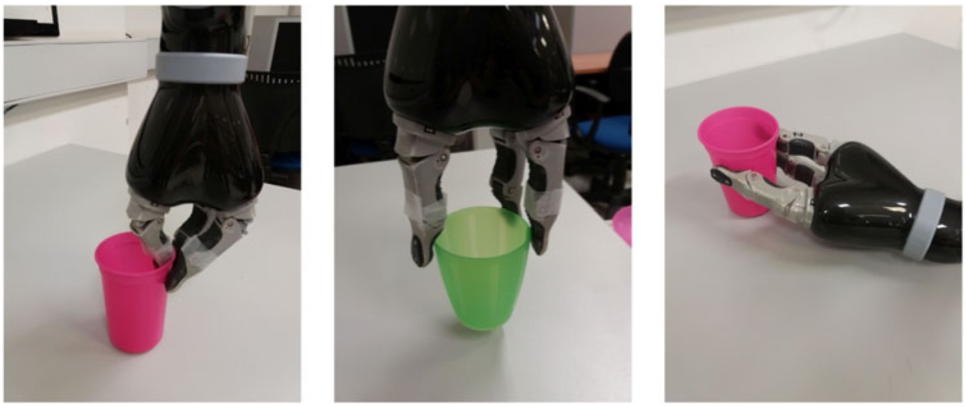

Fig. 2. Grasping configuration poses, that is, grasping handle encoded as “0” (Left), “1” (Center), and “2” (Right).

(six arm joints, one grasp parameter, one timestamp value), termed as robot action space (RAS). The latter representation arises since we employ in this work the six-joint-arm JACO, by Kinova Robotics. The grasp parameter introduced above assumes values 0, 1, or 2 to represent the grasping configuration pose of the hand, as illustrated in Fig. 2. The correspondence of human and robot arm motions is achieved by adopting kinesthetic teaching. In other words, the human demonstrator performs several demonstrations that form trajectories in a rather high-dimensional actor space (HAS). The same acts are performed by the robotic platform, physically steered by the human, formulating an equivalent high robots space (RAS).

3.1.2. Latent space representation. In order to facilitate meaningful association across HAS and RAS, both action spaces are transformed to analogous latent representations. To this end, we employ the Gaussian process latent variable model algorithm, GPLVM in short,[31] as formulated in ref. [3]; an algorithmic depiction of GPLVM is outlined in Algorithm 1 template.

GPLVM effectively provides an accurate representation of an abstract multidimensional space projected to a lower dimensional one, by performing a non-linear dimensionality reduction in the context of Gaussian processes. Hence, two latent space representations are established. The “Human Latent Space” (HLS) is derived as a set of points $X_{\text{HLS}} \in \mathbb{R}^q$ and the “Robot Latent Space” (RLS) as $X_{\text{RLS}} \in \mathbb{R}^q$. Using maximum-likelihood estimation methodology,[32] the optimal dimension of HLS and RLS is determined as $q = 2$. The latter has also been experimentally verified.

3.1.3. Space association. Having established the two latent spaces, an appropriate geometric transformation G_{Tr} that matches the corresponding points across the derived latent representations HLS and RLS is defined. Given two sets of corresponding 2D latent variables $\mathbf{X}_{\text{HLS}} = \{X_{H1}, \ldots X_{Hn}\} \in \mathbb{R}^2$

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

Algorithm 1 Kernel GPLVM

```
 1: Given a set of observations $\mathbf{O} \in \mathbb{R}^D$, $\mathbf{N}$: the number of iterations.
 2: Initialize latent variables $\mathbf{X} \in \mathbb{R}^2$ through PCA
 3: for T Iterations do
 4:     Randomly Select a subset of $\mathbf{X}$, $\mathbf{x_s} \subset \mathbf{X}$ and find the neighbors around $\mathbf{x_s} : \mathbf{X_S} = \mathbf{X} \in \mathbb{R}$
 5:     Compute the gradients: $\frac{\partial L}{\partial \mathbf{X_S}}$ and $\frac{\partial L}{\partial \beta_S}$ (Optimization)
 6:     Update $\mathbf{X}$ and $\beta$ by:
 7:     $\Delta \mathbf{X_t} = \mu_x \Delta \mathbf{X}_{t-1} + n_x \frac{\partial L}{\partial \mathbf{X_S}}$
 8:     $\mathbf{X_t} \leftarrow \mathbf{X}_{t-1} + \Delta \mathbf{X_t}$
 9:     $\Delta \beta_\mathbf{t} = \mu_\beta \Delta \beta_{t-1} + n_\beta \frac{\partial L}{\partial \beta_S}$
10:     $\beta_\mathbf{t} \leftarrow \beta_{t-1} + \Delta \beta_\mathbf{t}$
11: end for
    where, $\beta$:Kernel's parameter and $\mu$, $n$: parameters of the gradient descent process (stps 5-10)
```

and $\mathbf{X}_{\mathrm{RLS}} = \{X_{R1}, \ldots X_{Rn}\} \in \mathbb{R}^2$, the transformation is derived as a pair of translation t and rotation r that minimizes the sum of squared error:

$$E(t, r) = \frac{1}{T} \sum_{i=1}^{n} \|X_{Hi} - rX_{Ri} - t\|^2 \tag{1}$$

where X_{Hi}, X_{Ri} are corresponding points. The correspondence between the pairings is ensured during the data acquisition, where the trajectories are captured by kinesthetic teaching. The iterative closest point algorithm[33] is employed to minimize Eq. (1). The resulting transformation pair $G_{Tr}(t, r)$ can subsequently be used to map any point from HLS to the corresponding point in RLS.

3.1.4. Inverse transformation. In order to enable the robotic arm to reproduce a demonstrated human action, it is necessary to transform the RLS representation to the robotic physical configuration space, namely RAS. Formally speaking, this association should be implemented as the inverse of GPLVM, which is not analytically available. To approximate this inverse transformation (marked as INV_{Tr}), a high-order interpolation of the training learned data is formulated, using radial basis function as proposed in refs. [34, 35]. Here, this is addressed by computing off-line the latent representations (RLS) of a sufficiently large population of physical configurations (RAS) of the robotic arm. In practice, we iterate overall arm-DoFs, and for each arm configuration in RAS, the representation in latent space (RLS) is obtained via the GPLVM.

The above iterations are performed by (a) respecting the physical limits of each arm-joint, and (b) employing an appropriate iteration step. A small step value results in a denser representation of the RAS–RLS pairings. For points in RLS that are not included in the above pre-computed pairs, the corresponding points in RAS are derived by interpolation. Experimentally, it has been established that neither the step value nor the actual method of interpolation was critical. This is due to the fact that the employed inverse transformation should not render an accurate (exact) replica of a demonstrated act, but rather a reproduced robotic behavior sufficiently similar to the latter.

Consequently, learning concludes with the two latent space representations (HAS → HLS and RAS → RLS) along with the geometric transformation G_{Tr} that associates them (HLS ↔ RLS) and the established inverse mapping INV_{Tr} (RLS → RAS). As already mentioned above, Fig. 1 schematically presents the learning process.

3.2. Speed inference based on temporal information

A variety of issues may affect the temporal aspects of task execution in real-world applications. These may regard, for example, the physical properties of interacting objects (e.g., slow down to move a glass of water) or the need to synchronize with real-world temporal constraints or ongoing procedures (e.g., speed up to open the door after a bell-ring). However, time is a parameter that, so far, has been rather rarely considered in robot action planning.

Recently, we introduced the Daisy Planner, in an attempt to address time-informed planning in multi-agent setups.[9] Below, we summarize the most relevant issues which are essential for the completeness of the present work. The planner uses a fuzzy number representation of time to enable the processing of temporal information and develop time-informed multi-criteria optimized plans.

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

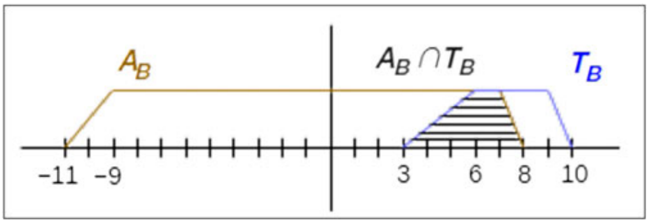

Fig. 3. Graphical illustration of the intersection of two fuzzy intervals.

More specifically, each time interval is mapped on a trapezoidal fuzzy number that is represented by the quadruple (p, m, n, q). For example, following this formulation, Fig. 3 depicts the time interval "approximately 6–9 moments" represented as the fuzzy number $T_B = (3, 6, 9, 10)$. In the same figure, negative values represent past time moments.

The present work exploits the fuzzy number representation of time intervals in order to (i) monitor the temporal aspects of composite task implementation, (ii) adapt the execution of individual primitive actions, and (iii) enforce the timely implementation of the overall task. In particular, using the ordinary fuzzy calculus, it is possible to associate the temporal properties of individual actions, predict delays, and then take corrective measures that account for speed adaptations, in order to ensure that composite behaviors are implemented on time.

As an illustrative example, we consider the case of a composite task that is based on the sequential implementation of actions A, B, and C. The prior knowledge of the system on the completion time of the three actions is as follows: $T_A = (3, 4, 7, 9)$, $T_B = (3, 6, 9, 10)$, $T_C = (5, 6, 9, 11)$. Using the well-known L-R calculus,[36] the total implementation time for the three actions is estimated as: $T_{\text{total}} = (11, 16, 25, 30)$. To clarify the concept, let a temporal constraint in the problem formulation, which requires that the three actions should complete at a specific maximum time of $C_{\max} = 20$. The system monitors the sequential execution of actions to ensure the timely accomplishment of the composite task. Assuming that seven moments have been devoted to the implementation of action A, then B and C should be implemented at a maximum time of $C_{\max} = 13$ moments. This is represented by the fuzzy time interval (0, 0, 13, 13).

In order to find a safe completion time for action B, the time to be spend on C is subtracted from the maximum available time. According to the L-R fuzzy calculus, this results into $A_B = (0, 0, 13, 13) - (5, 6, 9, 11) = (-11, -9, 7, 8)$, which is the time available for B. We take the intersection of the estimated available time A_B and the actual, known by experience, implementation time T_B to estimate the available temporal flexibility for the execution of B. The intersection of A_B and T_B is calculated as shown in Fig. 3. The defuzzification of this interval (implemented by the classic graded mean integration representation[37]) results into the requested action B implementation time, that is, $t = 6.2$.

3.3. Time-informed robotic action execution

The present work considers tasks implemented as a sequence of three actions, namely reach, grasp, and move/place. To ensure that the composite task is implemented on time, the system monitors the execution of each action, estimates possible delays as described above, and suggests a completion time for the action to be implemented next. The requested time is used to create an implementation of a known act within the provided time interval. The obtained, time-modulated human action in HAS is represented through GPLVM compression to HLS. The latter is in turn transferred to the robot's space by means of the learned HLS–RLS mapping, and subsequently unfolded to RAS. The robot implements the action within the requested time limits by implementing the sequence of configurations encoded in RAS.

In particular, the framework comprises off-line (action learning) as well as on-line (action execution) modules. Accordingly, the following steps, summarized also in Fig. 4, are employed to effectively reproduce an arm motion by the robotic system:

(a) A demonstrated action, presented as a sequence of points in HAS, namely $X^{t_n}_{\text{HAS}} = \{x_1, x_2, \ldots x_n\}$, is mapped to the corresponding one in HLS; GPLVM transforms $X^{t_n}_{\text{HAS}}$ to $X^{t_n}_{\text{HLS}}$.
(b) Mapping to robot latent space: the $G_{Tr}(t, r)$ mapping is used to obtain the relevant RLS representation, X_{RLS}.

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

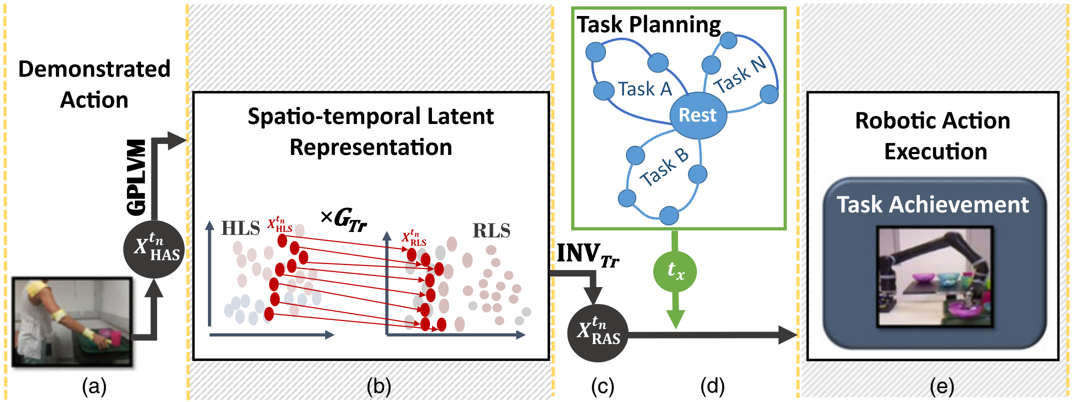

Fig. 4. Block diagram representation of time-informed robotic action execution. (a) Demonstrated action. (b) Spatio-temporal latent representation. (c) Inverse transformation INVTr. (d) Task planning (e) Spatio-temporal robotic action execution.

(c) Action reproduction: the formulated inverse transformation INV_{Tr} maps the X_{RLS} to RAS, formulating a set of points $X_{\mathrm{RAS}} \in \mathbb{R}^8$.

(d) The planer's output specifies the action to be implemented (i.e., reach, grasp, move), and the requested execution times. The latter determines the speed of action execution that is low, normal, high.

(e) Action is reproduced by the robot at the appropriate speed. Successful completion of the action is visually verified by a human observer (e.g., successful grasp of an object). In case of unsuccessful robotic action, the session is dropped.

4. Experimental Results

The methodological framework presented in the previous sections has been implemented and exhaustively tested in order to assess its performance in realistic cases. In this section, we report relevant results that quantify the effectiveness of action reproduction at the learned or different execution speeds.

In the current study, the relevant experimental setup regarding the learning phase involved kinesthetic teaching in order to demonstrate certain action behaviors to the robotic arm. In all our experiments, a six-joint-arm manipulator was used, namely the JACO robotic arm, by Kinova Robotics. Its joints can be controlled independently either with position-control or with torque-control. Accordingly, the compliant mode of the JACO arm was used in order to physically steer the arm to execute various motion trajectories.

The learning set is composed by sequences of paired poses executed by the two agents, that is, actor human arm and JACO robot arm, respectively. More specifically, we recorded a set of 30 primitive right-arm-movements of reaching, grasping, and placing an object of 100 frames each. Each demonstrated action is performed either at low, normal, or high speed. Given that for a human actor it is rather unrealistic to exactly perform actions at certain speeds, we employed the following convention in our experiments. At first, in an off-line session, the human demonstrator performed repeatedly the 30 actions in order to train with respect to the execution speeds. This resulted at a consistency in the execution times within a 20% margin. In other words, a sample reach/grasp/move action at normal speed, after training, would have a duration of 10 ± 2 s. Similarly, durations at high-to-low speed were performed on the average at 5-to-16 s, respectively. Subsequently, we used a threshold-based categorization to assign learned actions as low, normal, or high speed ones. The corresponding thresholds were set at durations of (a) 7 s to discriminate between high and normal speed actions, and (b) 14 s to distinguish among normal and low speed actions. In order to end up with an un-biased population of the demonstrated actions, out of the 30 recorded actions $1/3$ were executed at each of the three specified execution speeds.

Demonstration of the above-outlined 30 actions, and concurrent kinesthetic teaching to the JACO arm, gave rise to the establishment of the HAS and RAS spaces that were instantiated with the recorded samples from the actions' configuration spaces. More specifically, both HAS and RAS assumed 3000 configuration poses (samples) at the end of the training phase (30 actions $\times 100$

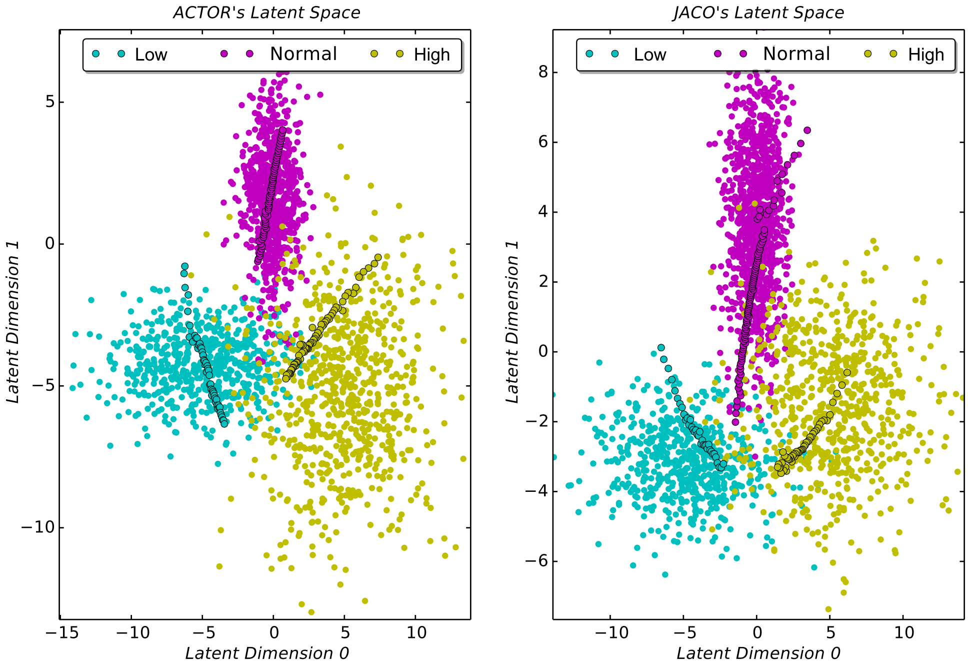

Fig. 5. Formulated latent space representations for human actor (left) and JACO arm (right), respectively. Actions performed at normal speed are represented by magenta circles, whereas slow and fast actions (in low and high speed, respectively) are depicted with cyan and yellow circles, respectively.

frames each). Following the learning phase, as described above, GPLVM was employed in order to compress the high-dimensional acting spaces to the latent ones. As a result, HAS and RAS point clouds have been transformed to HLS and RLS representations, respectively.

Figure 5 illustrates the resulting latent spaces. It is of utmost importance that the derived representations in HLS and RLS reveal well-separated sectors that correspond to the three implemented speeds. In other words, the sole information of action execution speed seems adequate to classify the action representations in the latent space in practically non-overlapping clusters. Interestingly, the latter holds true for both derived latent representations, namely HLS and RLS. This result should come as no surprise since the information conveyed by speed is quite discriminative, at least as opposed to spatial information that for the studied human actions designates small differences in the recorded trajectories. Furthermore, each point in the latent space represents an 11D vector in the acting space of human and an 8D vector for the robot case. In other words, every latent point includes the information of the human or the robot state, the grasping configuration, and the timestamp. To this end, it is worth noting that the formulated latent spaces feature important and at the same time useful properties. Different actions result in well-separated trajectories in the latent space, and also the neighboring property of a 3D trajectory is maintained given the continuous curves that are derived in the latent space. Furthermore, actions at different speeds are depicted in distinct point clouds in the latent space.

Besides the rather straightforward classification of actions in the latent spaces as exemplified above, additional noteworthy remarks can also be drawn from Fig. 5. More specifically, for each action in HAS (and its corresponding in RAS) recorded at a certain speed, we have “mentally” derived the two analogous actions in the other two implemented speeds. As a result, the three actions instantiated the same spatial actions at the three speeds, that is, low, normal, and high. Subsequent derivation of the latent representations gave rise to motion paths as the ones illustrated in Fig. 5 with the marked trajectories.

As can be observed, actions in the latent spaces are characterized by spatial continuity, that is, neighboring points in the acting space result in neighboring points in the latent space. In other words, the physical continuity of an action’s 3D trajectory is also maintained in the derived latent representation. This is in full accordance with the findings in our previous work,[3] where we considered the latent representations of actions at a single execution speed.

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

4.1. Performance metrics

In order to quantitatively assess the proposed spatio-temporal LfD methodological framework, we adopt in this work similar evaluation metrics as in ref. [3], that is, HLS and RLS consistency and repeatability (E_{HLS}, E_{RLS}), robotic end-effector trajectory (E_{EF}), and temporal consistency (E_t). To this end, let a novel action X^N performed from the human demonstrator at normal speed. From X^N, the spatially equivalent actions at low (X^L) and high (X^H) speed are artificially generated. Additionally, from each of the three actions X^N, X^L, and X^H, a set of M actions $\mathbf{X}^{i,N}$, $\mathbf{X}^{i,L}$, $\mathbf{X}^{i,H}$, $i = 1, .., M$ is produced by keeping the temporal components of the actions unaltered and randomly perturbing the spatial ones. Accordingly, M action trajectories are produced for each speed category that are variations of the same initial human action; a value of $M = 50$ has been used in our experiments. Subsequently, each one is projected via GPLVM to HLS, and G_{Tr} is then employed to map the latter representations to RLS. Finally, INV_{Tr} is invoked to inversely transform the trajectories to RAS, effectively resulting to the motions being reproduced by the robotic arm along with their execution times.

4.1.1. HLS consistency and repeatability—E_{HLS}. Let the three sets of actions in HLS depicted from sets of low, normal, and high speed, respectively, $\mathbf{X}^{i,L}_{\mathrm{HLS}}$, $\mathbf{X}^{i,N}_{\mathrm{HLS}}$, $\mathbf{X}^{i,H}_{\mathrm{HLS}}$, $i = 1, .., M$. Let also $\bar{\mathbf{X}}^{i,L}_{\mathrm{HLS}}$, $\bar{\mathbf{X}}^{i,N}_{\mathrm{HLS}}$, $\bar{\mathbf{X}}^{i,H}_{\mathrm{HLS}}$ be their corresponding mean trajectories. E_{HLS} is defined as:

$$E_{\mathrm{HLS}} = \mathbb{E}\left[\frac{1}{M} \sum_{i=1}^{M} \left\| \mathbf{X}^{i,K}_{\mathrm{HLS}} - \bar{\mathbf{X}}^{i,K}_{\mathrm{HLS}} \right\|^2_{\mathbb{S}^{-1}} \Big|_{\{K=L,N,H\}} \right] \tag{2}$$

where distances in above Eq. 2 are Mahalanobis distances expressed in the corresponding ellipsoids of HLS (Fig. 5 left). $\mathbb{E}[\cdot]$ denotes the mean value of the three quantities that result for K values marked as N, L, H.

4.1.2. RLS consistency and repeatability—E_{RLS}. In a similar manner as above, E_{RLS} is obtained as the relevant sum of Mahalanobis distances expressed in RLS:

$$E_{\mathrm{RLS}} = \mathbb{E}\left[\frac{1}{M} \sum_{i=1}^{M} \left\| \mathbf{X}^{i,K}_{\mathrm{RLS}} - \bar{\mathbf{X}}^{i,K}_{\mathrm{RLS}} \right\|^2_{\mathbb{S}^{-1}} \Big|_{\{K=L,N,H\}} \right] \tag{3}$$

where symbols in Eq. 3 above are interpreted as in Eq. 2.

4.1.3. Robotic end-effector trajectory—E_{EF}. A Euclidean distance metric is calculated as an index of imitation of the end-effector's movement. More specifically, E_{EF} is assumed to quantify the precise reproduction of a demonstrated act by the robotic end-effector, and it is defined as the 3D-error in the latter's trajectory. By this metric, only the spatial information is isolated, as the execution time is not considered in the relevant equation. In other words, this measure computes the Euclidean (3D) differences in trajectories between the robotic action reproduction and the demonstrated one, assuming the same execution times; it is obtained as:

$$E_{EF} = \mathbb{E}\left[\frac{1}{M} \sum_{i=1}^{M} \left\| \mathrm{X}^{i,K}_{H_{EF}} - \mathrm{X}^{i,K}_{R_{EF}} \right\|^2 \Big|_{\{K=L,N,H\}} \right] \tag{4}$$

where $\mathrm{X}^{i,K}_{H_{EF}}$, $\mathrm{X}^{i,K}_{R_{EF}}$ indicate the end-effector trajectories of the human actor and the robotic arm, respectively, at K speed, that is low, normal, and high.

4.1.4. Temporal consistency—E_t. A final metric is evaluated to describe the differences rendered in the execution times. More specifically, let ${t_{\mathrm{HAS}}}^{i,N}$ be the normal execution time of human action $\mathrm{X}^{i,N}_{\mathrm{HAS}}$ and similarly for the execution times at low and high speeds. Let also ${t_{\mathrm{RAS}}}^{i,N}$ be the execution time of the robotic reproduced action $\mathrm{X}^{i,N}_{\mathrm{RAS}}$, with similar definitions again for reproduced actions at low

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

Table I. Mean error values (μ) and standard deviation (σ) of the computed metrics for each group of speed, namely low (L), normal (N), and high (H).

	Low (L)		**Normal (N)**		**High (H)**	
Metrics	μ (%)	σ (%)	μ (%)	σ (%)	μ (%)	σ (%)
E_{HLS}	2.7	0.1	3.2	0.1	3.4	0.2
E_{RLS}	2.9	0.1	4.6	0.3	5.1	0.2
E_{EF}	2.7	0.2	3.7	0.3	4.4	0.3
E_t	2.4	0.1	2.8	0.3	3.2	0.3

and high speeds. The differences in the execution times averaged over all actions at a certain speed give rise to E_t as follows:

$$E_t = \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\left\| t_{\mathrm{HAS}}{}^{i,K} - t_{\mathrm{RAS}}{}^{i,K}\right\|^2 \Bigg|_{\{K=L,N,H\}}\right] \tag{5}$$

The complete set of 30 demonstrated actions was employed for assessment purposes. The values obtained in this case, for the above-described performance metrics, are summarized below in Table I as percentage figures. Following common practice in the area,[3,5] mean values of the obtained errors are presented, along with the relevant standard deviations.

As can be observed, all performance metrics assumed very low values, indicating accurate and effective spatio-temporal representation and reproduction of actions. Interestingly, for the low (L) case the errors are slightly smaller, becoming gradually larger as speed increases. This is rather expected, since lower speed values give rise to smoother trajectories, which in turn facilitate more accurate reproduction of relevant actions.

4.2. Use case application—Service scenario

The second category of experiments regards the validation of robotic performance in a realistic service scenario. In particular, the available reaching, grasp, move actions are exploited to address time-constrained tray filling. The examined scenario is inspired by restaurant standing queues with customers served one at a time. Robot aims at filling the tray within the requested timeframe, in order to serve humans on time. Humans are supposed to wait in front of the serving queue. The simplified serving considered here assumes two cups and one bowl to be placed on each tray. We consider varying times of requested tray filling, centered at 2.0 min (the average period of customer arrival). In short, when serving a customer is delayed, the system tries to compensate this latency by asking for faster filling of future trays. Following this formulation/scenario, the repetitive tray filling task must be implemented at varying time limits and hence robot action speeds.

We have developed a simple setup that enables tray filling in naturalistic conditions (see Fig. 6). It is noted here that the proposed approach and the Daisy Planner in particular manage the temporal constraints on tasks by enabling both the increment and the decrement of speed. In general, both options are required and the proposed work provides a systematic approach to encompass both speed adaptations, and also demonstrates the latter in real application scenarios. To this end, a service scenario application case is used to demonstrate the validity of the approach, showing at the same time its ability of generalization and natural flow.

Additionally, for quantitative assessment of the action execution times, we conducted 20 repetitions of the tray filling task; cases where grasp failures were encountered were dropped out and the experiment was repeated. Accordingly, we ended up with 20 successful task completions for which we contrasted the actual execution times against the commanded ones by the planner. Time differences above 10% were regarded as failures. Interestingly, only three executions did not meet the latter criterion, and were marked unsuccessful. Given the complexity and variability of the studied scenario, the accomplished result is considered highly promising and indicative of

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

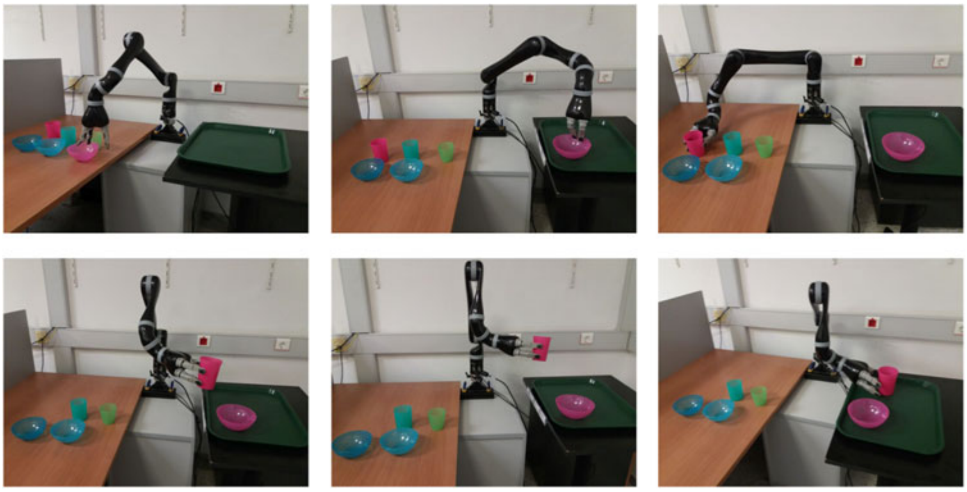

Fig. 6. Snapshots from the serving scenario.

the method's potential. A better appreciation of the described experiments can be acquired by the supplementary video in high resolution at https://youtu.be/WGG0vI6NiMU.

5. Discussion

In the current paper, we introduced a novel spatio-temporal formulation that compactly represents spatial and temporal aspects of studied actions. The latter is accomplished by assuming latent space representations and further combined with a time-informed task planner to effectively schedule actions in the course of a complete task. Our examination has revealed very useful and interesting properties of the formulated latent representations, namely (i) well-separated latent representations for actions executed at different speeds, (ii) neighboring points in the configuration space are mapped to neighboring points in the latent space, thereby preserving action continuity in the latent depiction, and (iii) errors in the spatial and temporal domains due to the latent transformation are very small and in practice do not affect the method's performance.

The described framework has been employed in the execution of a realistic service scenario. Our results demonstrate the successful involvement of the robotic system in the accomplishment of relevant tasks, whereby learned action behaviors are appropriately executed at varying speeds. The main contributions of the proposed method are summarized as follows:

- Enhance robots that learn from demonstration to execute actions at variable speeds;
- Provide insight to the key role of temporal information in obtaining compact, latent representations of human arm motions;
- Demonstrate the reversibility of the spatio-temporal aspects of actions from the low-dimensional space to the actual action space;
- Combine temporal planning with LfD.

The so far encouraging results attest for the validity and effectiveness of the proposed approach. Nevertheless, open issues for investigation still exist. To this end, in future work, we aim at extending this methodological framework in the case of multi-robot collaborative systems, where time-constrained task fulfillment is a critical issue. In addition, the method's scalability by incorporating further dynamic parameters, such as force and torque, will be examined. Finally, one of our immediate goals for the continuation of the proposed work is to examine the capability of obstacle avoidance and on-line motion adaptation during robotic reproduction.

Acknowledgements

The current work was partially supported by the EU FET Open grant (GA: 754423) EnTiment: Industrial Exploitation and Market Uptake of a Temporal Cognition Toolbox for Commercial

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

Robots, and the FORTH-HONDA Research Institute-JP joint research project (Ref.No.350/23-6/18-03-2017): Predicting the temporal properties of activities facilitates human-robot symbiosis & cooperative intelligence.

Supplementary Material

To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574719001449.

References

1. M. Li, Y. Bekiroglu, D. Kragic and A. Billard, "Learning of Grasp Adaptation Through Experience and Tactile Sensing," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, Chicago, IL (2014).
2. S. Schaal, J. Peters, J. Nakanishi and A. Ijspeert, "Control, Planning, Learning Andimitation with Dynamic Movement Primitives," *IROS 2003*, Las Vegas, NV (2003) pp. 1–21.
3. M. Koskinopoulou and P. E. Trahanias, "A Methodological Framework for Robotic Reproduction of Observed Human Actions: Formulation Using Latent Space Representation," *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Cancun, Mexico (IEEE Press, 2016) pp. 565–572.
4. A. Shon, D. B. Grimes, C. Baker and R. P. N. Rao, "A Probabilistic Framework for Model-Based Imitation Learning," *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society*, Berkeley, CA (2004) pp. 1237–1242.
5. S. Calinon, F. Guenterlorent and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* **37**(2), 286–298 (2007).
6. R. Vuga, B. Nemec and A. Ude, "Speed adaptation for self-improvement of skills learned from user demonstrations," *Robotica* **34**(12), 2806–2822 (2016).
7. F. Ficuciello, P. Falco and S. Calinon, "A brief survey on the role of dimensionality reduction in manipulation learning and control," *IEEE Robot. Auto. Lett.* **3**(3), 2608–2615 (2018).
8. T. Hofmann, "Probabilistic Latent Semantic Analysis," *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden (1999) pp. 289–296.
9. M. Maniadakis, E. Aksoy, T. Asfour and P. Trahanias, "Collaboration of Heterogeneous Agents in Time Constrained Tasks," *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Cancun, Mexico (2016).
10. M. Maniadakis and P. Trahanias, "Time-Informed, Adaptive Multi-robot Synchronization," *Proceedings of the Simulation of Adaptive Behavior (SAB)*, Aberystwyth, UK (2016).
11. A. de Rengervé, J. Hirel, P. Andry, M. Quoy and P. Gaussier, "On-Line Learning and Planning in a Pick-and-Place Task Demonstrated Through Body Manipulation," *IEEE International Conference on Development and Learning (ICDL) and on Epigenetic Robotics (Epirob)*, Frankfurt am Main, Germany (2011) pp. 1–6.
12. B. D. Argall and A. G. Billard, "A survey of tactile human-robot interactions," *Robot. Auton. Syst*. **58**(10), 1159–1176 (2010).
13. T. Brys, A. Harutyunyan, H. Suay, S. Chernova, M. Taylor and A. Now, "Learning from Demonstration and Reinforcement," *International Joint Conference on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina (2015) pp. 1–7.
14. R. Toris and S. Chernova, "Goal-Based Learning from Demonstration for Mobile Pick-and-Place," *AAAI Fall Symposium on Artificial Intelligence and Human-Robot Interaction*, Arlington, Virginia (2014).
15. A. Gupta, C. Eppner, S. Levine and P. Abbeel, "Learning Dexterous Manipulation for a Soft Robotic Hand from Human Demonstrations," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016*, Daejeon, Korea (2016) pp. 3786–3793.
16. L. P. Kaelbling, M. L. Littman and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.* **101**(1–2), 99–134 (1998).
17. P. Evrard, E. Gribovskaya, S. Calinon, A. Billard and A. Kheddar, "Teaching physical collaborative tasks: object-lifting case study with a humanoid," *Humanoids*, Paris, France (2009) pp. 399–404.
18. S. Hou, A. Galata, F. Caillette, N. Thacker and P. Bromiley, "Articulated Pose Estimation in a Learned Smooth Space of Feasible Solutions," *IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil (2007).
19. S. Quirion, C. Duchesne, D. Laurendeau and M. Marchand, "Comparing gplvm approaches for dimensionality reduction in character animation," *J. WSCG* **16**(2–3), 41–48 (2008).
20. S. Hou, A. Galata, F. Caillette, N. Thacker and P. Bromiley, "Real-Time Body Tracking Using a Gaussian Process Latent Variable Model," *ICCV*, Rio de Janeiro, Brazil (2007) pp. 1–8.
21. O. Kramer, *On Missing Data Hybridizations for Dimensionality Reduction* (Springer, Berlin, Heidelberg, 2013) pp. 189–197.
22. M. Koskinopoulou, S. Piperakis and P. E. Trahanias, "Learning from Demonstration Facilitates Human–Robot Collaborative Task Execution," *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, Christchurch (2016) pp. 59–66.

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press

23. S. Calinon, A. Pistillo and D. G. Caldwell, "Encoding the Time and Space Constraints of a Task in Explicit Duration Hidden Markov Model," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA (2011) pp. 3413–3418.
24. A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Comput.* **25**(2), 328–373 (2013).
25. L. Rozo, P. Jimenez and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *Intell. Service Robot.* **6**(1), 33–51 (2013).
26. L. Rozo, J. Silverio, S. Calinon and D. G. Caldwell, "Learning controllers for reactive and proactive behaviors in human-robot collaboration," *Front. Robot. AI* **3**, 1–111 (2016).
27. M. Ewerton, G. Maeda, G. Neumann, V. Kisner, G. Kollegger, J. Wiemeyer and J. Peters, "Movement Primitives with Multiple Phase Parameters," *IEEE International Conference on Robotics and Automation (ICRA),* Stockholm, Sweden (2016) pp. 201–206.
28. R. Dechter, I. Meiri and J. Pearl, "Temporal constraint networks," *Artif. Intell.* **49**(1–3), 61–95 (1991).
29. P. Morris, "Dynamic Controllability and Dispatchability Relationships," **In**: *Integration of AI and OR Techniques in Constraint Programming*, Lecture Notes in Computer Science (Springer International Publishing, Cham, 2014) pp. 464–479.
30. M. P. Sigalas, M. Markos and P. Trahanias, "Full-body pose tracking the top view reprojection approach," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(8), 1569–1582 (2015).
31. N. Lawrence and A. Hyvarinen, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *J. Mach. Learn. Res.* **6**, 1783–1816 (2005).
32. E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," *NIPS* (2004).
33. S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, Quebec, Canada (2001).
34. N. D. Monnig, B. Fornberg and F. G. Meyer, "Inverting non-linear dimensionality reduction with scale-free radial basis interpolation," Technical Report (2013).
35. E. Amorim, E. V. Brazil, J. P. Mena-Chalco, L. Velho, L. G. Nonato, F. F. Samavati and M. C. Sousa, "Facing the high-dimensions: Inverse projection with radial basis functions," *Comput. Graph.* **48**, 35–47 (2015).
36. D. Dubois and H. Prade, *Possibility Theory. An approach to Computerized Processing of Uncertainty* (Plenum Press, New York, 1988).
37. B. Khadar, A. Rajesh, R. Madhusudhan, M. V. Ramanaiah and K. Karthikeyan, "Statistical optimization for generalised fuzzy number," *Int. J. Modern Eng. Res.* **3**(2), 647–651 (2013).

https://doi.org/10.1017/S0263574719001449 Published online by Cambridge University Press