A general approach for generating artificial human-like motions from functional components of human upper limb movements

Control Engineering Practice 148 (2024) 105968

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A general approach for generating artificial human-like motions from functional components of human upper limb movements

Marco Baracca [a,*], Giuseppe Averta [b], Matteo Bianchi [a]

[a] Research Center E. Piaggio and Department of Information Engineering, University of Pisa, Pisa, 56122, Italy
[b] Department of Control and Computer Engineering, Politecnico di Torino, Turin, 10129, Italy

ARTICLE INFO

Keywords:
Human-like motion generation
Planning algorithms
Human–machine interaction

ABSTRACT

Anthropomorphism of artificial systems is a key enabling factor to ensure effective and compelling human–machine interactions in different domains, including immersive extended reality environments and cobotics applications. Among the different aspects that anthropomorphism refers to, the generation of human-like motions plays a crucial role. To this aim, optimization-based techniques, whose functional cost is devised from neuroscientific findings, or learning-based approaches have been proposed in literature. However, these methods come with limitations, e.g., limited motion variability or the need for high dimensional datasets. In previous works of our group, we proposed to exploit functional Principal Component Analysis (fPCA) of human upper limb movements, to extract principal motion modes in the joint domain and use them to directly embed the human-like behaviour in the planning algorithm. However, this approach faces with translational issues related to the computational burden and to the application to kinematic structures different from the one used to describe human movements. To overcome this problem, we propose a general framework to generate human-like motion directly in the Cartesian domain by exploiting fPCA. This solution permits to perform obstacle avoidance with low computational time and it can be applied to any kinematic chain. To prove the effectiveness of our approach, we tested it against a state-of-the-art human-like planning algorithm both in terms of the accuracy of target reaching and human-likeness features of the generated movement.

1. Introduction

Human-likeness (HL) is a key characteristic for artificial systems designed for a safe, effective and trustworthy human–machine interaction, e.g. with humanoid robots or virtual avatars (Bartneck, Kulić, Croft, & Zoghbi, 2009; Riek, Rabinowitch, Chakrabarti, & Robinson, 2009). HL is a broad term which encompasses several characteristics of the system, such as its design, appearance and motion (Rothstein, Kounios, Ayaz, & de Visser, 2021). While much has been done to increase the realism of body design, with a large spread of hyper-realistic humanoids (Cominelli, Hoegen, & De Rossi, 2021; Glas, Minato, Ishi, Kawahara, & Ishiguro, 2016; Hanson, 2023), the motion of these systems is far from the naturalness of human movements. The problem of generating Human-Like movements is still under-explored, albeit representing a key aspect for the realism, acceptability and predictability of human–machine interaction (Abubshait & Wiese, 2017; Duffy, 2003; Fong, Nourbakhsh, & Dautenhahn, 2003; Zanchettin, Bascetta, & Rocco, 2013).

Human motion has different key features which make it peculiar with respect to movements generated by classical planning algorithms. For example, in Grimme, Lipinski, and Schöner (2012) the authors analysed human hand motion in 3D space during reaching tasks, with and without obstacles, finding that the movement paths are largely planar. In Lacquaniti, Terzuolo, and Viviani (1983), the authors found that there is a relation between the curvature of a path and the velocity at which humans have to follow it. In Flash and Hogan (1985) the authors observed that humans tend to minimize jerk during movement execution.

Many researchers proposed strategies for generating Human-Like movements in different applications (Gulletta, Erlhagen, & Bicho, 2020; Nguiadem, Raison, & Achiche, 2020). One of the most popular solutions to achieve HL is to formalize an optimization problem whose functional cost is devised from neuroscientific observations. For example, in Piazzi and Visioli (2000) the authors developed an optimization-based framework to generate minimum-jerk trajectories building on Flash and Hogan (1985), while Klein Breteler, Gielen, and Meulenbroek (2001) exploited the minimization of the torque-change following the model proposed in Uno, Kawato, and Suzuki (1989). However, such optimization approaches usually build upon hypotheses on motion

* Corresponding author.
E-mail address: marco.baracca@phd.unipi.it (M. Baracca).

https://doi.org/10.1016/j.conengprac.2024.105968
Received 29 December 2023; Received in revised form 27 April 2024; Accepted 29 April 2024
Available online 7 May 2024

generation, which can reduce the variability of the planned movement (and, sometimes also lack experimental support (Miossec & Kheddar, 2009)). To cope with this problem, an interesting approach was the one proposed in Rosenbaum, Meulenbroek, Vaughan, and Jansen (2001) where the authors proposed a model of motion planning based on constraint hierarchy instead of optimizing some cost functions. In this way, the model focuses more on satisfying the constraints related to the desired task (e.g. final target arm posture, obstacle avoidance, etc.) instead of minimizing a predefined cost index.

Another possible approach to generate Human-Like movements exploits learning and data-driven methods. This is a solution used very often in the field of animations and computer graphics where, after an extensive campaign of data recording via motion capture systems, the recorded datasets are used to train neural networks to animate the avatars (Kwiatkowski et al., 2022; Mourot, Hoyet, Le Clerc, Schnitzler, & Hellier, 2022; Yin, Yin, Kragic, & Björkman, 2021). For example, in Mordatch, Lowrey, Andrew, Popovic, and Todorov (2015) the authors used a recurrent neural network to act as a near-optimal feedback controller generating stable and realistic behaviour. Another example is Rebol, Gütl, and Pietroszek (2021), where the authors used Generative Adversarial Neural Networks for synthesizing gestures directly from speech. Some of these approaches are also applied in robotic applications for the generation of human-like movements with humanoid robots (Schulz, Torresen, & Herstad, 2019). However, the common limitation of learning-based methods is related to the need for reliable datasets, whose dimensionality can be significant.

An interesting approach that lies in between model and learning-based solutions exploits Dynamic Movement Primitives (DMP) (Schaal, 2006). The idea is to use a dynamical system with convenient stability properties and modulate it with nonlinear terms such that it achieves a desired attractor behaviour. One of the strengths of this framework is the low number of demonstrations required to handle different situations (Lentini, Grioli, Catalano, & Bicchi, 2020). This approach has been extensively studied and used in literature (Saveriano, Abu-Dakka, Kramberger, & Peternel, 2023). One problem that this method has to deal with is related to obstacle avoidance. The solution presented in several papers is to add a second nonlinear term to guide the evolution of the dynamical system around obstacles. There are two ways to compute this term: (1) through potential fields to repulse the system from the obstacles (Ginesi, Meli, Roberti, Sansonetto, & Fiorini, 2021; Ijspeert, Nakanishi, Hoffmann, Pastor, & Schaal, 2013); (2) through neural networks learning the coupling term from a set of examples (Rai, Sutanto, Schaal, & Meier, 2017; Sutanto, Su, Schaal, & Meier, 2018). These approaches permit to handle also time-varying environments given the negligible computational time required to compute this term. However, the selection of the specific potential field could influence the behaviour of the trajectory, which could lose its desired characteristics, while implementing learning-based techniques could require a large number of examples to generalize for a wide range of obstacle setups.

A possible solution to design an efficient Human-Like motion planning framework, overcoming the aforementioned issues, is to directly embed main human motion characteristics in the algorithmic structure. Many works in the literature addressed the analysis of human motion to extract movement patterns and obtain a reduced yet meaningful characterization of human kinematics (Santello et al., 2016). Regarding the upper limb motion, in Averta et al. (2017) we exploited functional Principal Component Analysis (fPCA) to identify a geometrical basis of mathematical functions whose elements can be combined to reconstruct the overall trajectory. These basis elements were also used to develop a planning algorithm in the joint space domain, which intrinsically embeds HL in the generated motion (Averta, Della Santina, Valenza, Bicchi and Bianchi, 2020). This planner, however, is strictly related to the kinematic description used to acquire human upper limb data, and a mapping strategy is needed to generalize the planning outcomes to manipulators with different kinematic structures. A solution to the latter problem was proposed in Averta, Caporale, Della Santina, Bicchi and Bianchi (2020), where Cartesian impedance control was used to implement fPCA-based planning with manipulators with redundant anthropomorphic kinematic architectures — although dissimilar with respect to the human model used for functional mode extraction. However, these approaches in the joint space are associated with non-negligible computational time: for example, while obstacle-free planning can be solved in a closed form, devising a trajectory in presence of obstacles requires solving an optimization problem, which can require up to several seconds.

To address both the problem of mapping and the reduction of the computational time, we propose a novel planning algorithm able to compute Human-Like trajectories of artificial upper limb/arm directly in the Cartesian domain. To this aim, we built upon the results we presented in Baracca et al. (2022), where we showed that a geometrical representation of the human end-effector trajectory in terms of functional elements still holds in the Cartesian space, confirming the outcomes reported in Averta et al. (2017) at the joint level. This approach permits to obtain a reference trajectory with an intrinsic Human-Like behaviour which can be applied to any kinematic chain used for describing an artificial manipulator in a lower time than the previous approach developed in the joint domain (less than 7 ms on average).

The paper is organized as follows: we first give a brief introduction to fPCA and we describe how the planning algorithm is structured; then we test the proposed method in different scenarios comparing the computed trajectories with the ones obtained via DMP; and in the end, we discuss our outcomes together with the possible application scenarios and future developments.

2. Methods

2.1. Functional Principal Component Analysis (fPCA)

Functional Principal Component Analysis (fPCA) is a statistical method to identify a geometrical basis of functions whose elements can be combined to reconstruct time series. In this section, we will provide a brief introduction to the underpinning theory and its application – without loss of generality – to the description of hand trajectories (i.e. the trajectories of the end-effector of the upper limb kinematic chain), while referring the interested reader to Ramsay, Hooker, and Graves (2009) for more details. Given a dataset of hand motions, the generic motion $x(t)$ can be represented as a weighted sum of a set of basis functions $S_i(t)$, or functional Principal Components (fPCs) extracted from the dataset, that is:

$$x(t) \simeq \bar{x} + S_0(t) + \sum_{i=1}^{s_{max}} \alpha_i \circ S_i(t) \tag{1}$$

where $\bar{x}$ is the average pose of the hand, $S_0(t)$ is the average trajectory across all the trajectories in the dataset, α_i is a vector of weights, s_{max} is the number of basis elements, $S_i(t)$ is the ith basis element, the symbol $\circ$ represents the Hadamard product (i.e. the element-wise product) and $t \in [0, 1]$ is the normalized time axis.

The first element of the functional basis or first fPC can be computed from the R motions of the dataset as:

$$\max_{S_1} \sum_{j=1}^{R} \left(\int S_1(t) x_j(t) dt \right)^2 \tag{2}$$

subject to

$$\|S_1(t)\|_2^2 = 1 \tag{3}$$

The other components $S_i(t)$ can be computed as:

$$\max_{S_i} \sum_{j=1}^{R} \left(\int S_i(t) x_j(t) dt \right)^2 \tag{4}$$

subject to

$$\|S_i(t)\|_2^2 = 1 \tag{5}$$

$$\int_0^{t_{end}} S_i(t)S_k(t)dt = 0, \forall k \in \{1, \dots, i-1\} \tag{6}$$

In this manner, we can identify a basis of functional elements, ordered in terms of the explained variance that each element accounts for.

For our purpose, to obtain a general representation of the human hand motion, we used the dataset proposed in Averta et al. (2021), containing the recording of 30 different activities of daily living performed by 30 different subjects belonging to three main classes of actions: Transitive, Intransitive and Tool-mediated, which were assumed to be representative of the human example (Averta et al., 2017). However, in Baracca et al. (2022), we recomputed fPCA independently for each participant (90 trajectories for each participant) showing that, with the same set of tasks, a stable functional representation in terms of the shape of the fPCs can be extracted from a reduced dataset (90 movements) even if it contains heterogeneous tasks. This is possible thanks to the structure of fPCA, which allows us to handle datasets containing time series that are very different from each other without requiring a large number of examples. Instead learning-based methods, to generalize on more variable patterns, require higher dimensionality of the datasets. For example, in He et al. (2021), the authors recorded, from 10 different subjects, a total of 400 trials taking into account only 8 different reaching movements (similar to the ones that, in our classification, belong to the Intransitive movements). Deep learning based solutions are notably even more data-hungry. To provide an example, recent generative based solutions require thousands of independent motion examples replicated on tenths of different characters (Raab et al., 2023).

2.2. Planning algorithm

The fPCs extracted from a dataset that can be considered representative of the most common upper limb movements can be used to plan trajectories that intrinsically embed HL. In the following section, we provide a formalization of the planning problem starting with the no-obstacle case, and then we extend the approach to deal with the presence of an arbitrary number of fixed obstacles. Of note, fPCA is performed for each Degree of Freedom (DoF) separately that, in our case, are the Cartesian position and orientation of the hand with respect to the centre of the chest. In the following, we report the equations for a single DoF of the end effector, while the extension to multiple DoFs (e.g. the six DoFs describing the pose of the end effector) is trivial.

The reconstruction of the single DoF trajectory can be attained as:

$$x(t) \simeq \bar{x} + S_0(t) + \sum_{i=1}^{s_{max}} \alpha_i S_i(t) \tag{7}$$

To find the coefficients $\bar{x}$ and α_i given a set of constraints to be satisfied we can define an equation system to obtain the desired trajectory to be planned. For example, setting the initial and final position, velocity and acceleration, the following equation system is defined:

$$\begin{bmatrix} 1 & S_1(t_0) & \dots & S_5(t_0) \\ 1 & S_1(t_f) & \dots & S_5(t_f) \\ 0 & \dot{S}_1(t_0) & \dots & \dot{S}_5(t_0) \\ 0 & \dot{S}_1(t_f) & \dots & \dot{S}_5(t_f) \\ 0 & \ddot{S}_1(t_0) & \dots & \ddot{S}_5(t_0) \\ 0 & \ddot{S}_1(t_f) & \dots & \ddot{S}_5(t_f) \end{bmatrix} \begin{bmatrix} \bar{x} \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = \begin{bmatrix} x(t_0) - S_0(t_0) \\ x(t_f) - S_0(t_f) \\ \dot{x}(t_0) - \dot{S}_0(t_0) \\ \dot{x}(t_f) - \dot{S}_0(t_f) \\ \ddot{x}(t_0) - \ddot{S}_0(t_0) \\ \ddot{x}(t_f) - \ddot{S}_0(t_f) \end{bmatrix} \tag{8}$$

by solving the system we can obtain the desired planned trajectory

$$x(t) = \bar{x} + S_0(t) + \sum_{i=1}^{5} \alpha_i S_i(t) \tag{9}$$

In the presence of obstacles, instead of numerical optimization, it is possible to define a set of via points (e.g. points defined in the trajectory domain through which the trajectory itself has to go (Vaughan, Rosenbaum, & Meulenbroek, 2001)) and build a similar extended system of equations to solve the problem in a closed form. In a nutshell, the idea is to plan the trajectory in pieces between two successive points, ensuring continuity at the junction point. This strategy moves the problem of collision avoidance to the selection of the best via points to guide the trajectory around the obstacles. The optimal via point guarantees collision avoidance with a minimum path length. To find it, we can use a sampling-based algorithm that selects a set of possible points (that ensure obstacle avoidance) inside the workspace and then identify the one with the minimum path length. This approach exploits the low computational load in generating a single trajectory to perform a random sampling in space to find the best solution in a short time. For our purpose, the velocity and acceleration values have to be set in the via point as constraints to solve the equation system. Without loss of generality, we set the velocity as the mean velocity to go from the initial to the final target point, while we decide to pass the via point with an acceleration equal to zero. The implementation of the proposed algorithm in MATLAB can be found at the following repository[1]

3. Simulation framework

To evaluate the performance of our approach we tested it in simulation against a classical DMP approach. We have chosen this type of planner as a comparison for testing our method because it is placed in the middle between optimization-based techniques (usually computationally expensive) and learning-based techniques (that are faster to compute a solution but greedier in terms of data required for training and generalization). Briefly, a DMP for a single DoF trajectory y of a discrete movement (point-to-point) is defined by the following set of nonlinear differential equation

$$\begin{aligned} \tau \dot{z} &= K(g - y) - Dz + (g - y_0)f(x) \\ \tau \dot{y} &= z \\ \tau \dot{x} &= -\alpha x \end{aligned} \tag{10}$$

where x is the phase variable and z is an auxiliary variable. Parameters K and D are respectively the spring and the damping terms which define the behaviour of the second-order system described by in (10). With the choice $\tau > 0$, $D = 2\sqrt{K}$ and $\alpha > 0$ the convergence of the underlying dynamic system to a unique attractor point at $y = g$, $z = 0$ is ensured. The forcing term $f(x)$ is defined as a linear combination of N nonlinear basis functions, which enables the robot to follow any smooth trajectory from the initial position y_0 to the final configuration g

$$f(x) = \frac{\sum_{i=1}^{N} w_i \boldsymbol{\Phi}_i(x)}{\sum_{i=1}^{N} \boldsymbol{\Phi}_i(x)} x \tag{11}$$

Usually, the classical implementation of DMP uses for the forcing term a basis of Gaussian functions (Ijspeert et al., 2013). However, as discussed in Ginesi, Sansonetto and Fiorini (2021), there are also other types of basis functions that could be used, such as the mollifiers-like basis functions given by:

$$\boldsymbol{\Phi}_i(x) = \begin{cases} \exp\left(-\frac{1}{1-r(x)^2}\right), & \text{if } r < 1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

with $r(x) = |a_i(x - c_i)|$, where c_i is the centre and a_i is the width of the function. To select the suitable number of functions for implementing DMPs, we run a preliminary analysis with different cardinality of the basis of function elements. We started with 50 elements (which was stated as the usual maximum number of functions to be used in Ginesi, Sansonetto et al. (2021)) and we brought it down to 10 elements

[1] https://github.com/Marco-Baracca/HL-motion-generation.git.

evaluating the jerk and the tracking performances. We found that the performance levels of all the tested setups were comparable. For our purpose, we used 21 functions. This solution exhibited a slightly lower level of jerk compared to the other cardinalities we tested. To learn the weights for the forcing term, we considered human reaching motions extracted from the same dataset we used to perform fPCA (Averta et al., 2021) and we trained DMPs on one of these movements. For additional details on the building of this subset of motions, we refer the reader to Section 3.1.

To implement obstacle avoidance, we used the strategy presented in Park, Hoffmann, Pastor, and Schaal (2008), where a dynamic potential field was used in combination with DMP to deviate the trajectory around the obstacles. In this work, the repulsive field is defined as:

$$U_{dyn}(y,z) = \begin{cases} \lambda(-\cos\theta)^{\beta}\frac{\|z\|}{p(y)}, & \frac{\pi}{2} < \theta \leq \pi \\ 0, & 0 \leq \theta \leq \frac{\pi}{2} \end{cases} \tag{13}$$

where θ is the angle taken between the current velocity z and the relative position of the trajectory with respect to the obstacle (with respect to the latter the distance is defined as $p(y)$, while β is a scalar coefficient). Mathematically, given the equation in (13), the potential field generates a repulsion force only when the trajectory moves towards the obstacle, while in all the other cases it does not influence the evolution of the dynamic system.

The validation of the planning algorithm is composed by different parts. First, given that our approach is based on sampling, we have studied how the output of the algorithm is influenced by the number of samples taken. After that, we move to the evaluation of the Human-Likeness of the generated trajectory which is divided into three parts: (1) comparison with a set of real reaching motions extracted by the original dataset; (2) large-scale simulation with different sets of obstacles; and (3) a reduced set of scenarios that reproduce some daily living activities, where the trajectories produced by the two planners served as references for the controllers of an anthropomorphic kinematic chain. More details regarding the simulation scenarios are provided in the following. All the validation is performed using MATLAB R2020b.

3.1. Comparison with real human motion

To evaluate the Human-Likeness of the proposed approach and DMPs, we compared it with a set of recorded human motions to check the similarity of the output of the planning algorithm with them. Given that the algorithm is designed to compute a trajectory to reach a desired position from an initial point, we extracted from the dataset proposed in Averta et al. (2021) 180 human upper limb reaching movements and compared the similarity of the generated path with respect to the human example. More specifically, given that the original dataset does not contain this type of movement, we extracted for 3 subjects the 30 intransitive motions performed (3 repetitions of 10 different tasks) and we manually segmented the forward and the backward motion of each trial. For each example movement contained in this subset, we extracted the initial and the final point, which were then used as start and goal positions for the planning algorithms.

3.2. Large scale obstacle avoidance simulation

Given that we do not have a dataset of real human motion performing reaching motion in the presence of obstacles, we performed this evaluation in simulation comparing the output of our planning algorithm with respect to DMP. To do this, we generated a large number of simulation settings, where each of them is defined in terms of a starting point, an ending point and a set of spherical obstacles. We have decided to use spherical obstacles due to the ease of verifying the collision on this shape. However, this choice does not limit the extension of this framework to any general obstacle shape (O'Rourke & Badler, 1979). We defined 4 different types of possible scenarios and, for each group, we generated 1000 different realizations of it. The 4 different classes are: (1) 5 spherical obstacles with $r = 0.1$ m; (2) 1 single spherical obstacle with $r = 0.2$ m; (3) 3 spherical obstacles with $r = 0.1$ m and (4) 3 spherical obstacles with $r = 0.2$ m. All these scenarios were generated randomly extracting starting points, target points and the centres of the obstacles.

3.3. Dynamic arm simulation

The second part of the validation regards the application of the planner to daily living sample tasks and the usage of these trajectories as a reference to control an anthropomorphic kinematic chain. The dynamic of the arm was simulated using the "Robotics Toolbox for MATLAB" developed by Peter Corke (Corke, 2007).

To follow the Cartesian reference, we implemented a Cartesian computed torque controller defined as:

$$\tau = J^T M_X(\ddot{\xi}_d + K_V\dot{\tilde{\xi}} + K_P\tilde{\xi}) + h(q,\dot{q}) - J^T M_X \dot{J}\dot{q} \tag{14}$$

where ξ_d, $\dot{\xi}_d$ and $\ddot{\xi}_d$ are Cartesian desired position, velocity and acceleration, $\tilde{\xi} = \xi - \xi_d$, $\dot{\tilde{\xi}} = \dot{\xi} - \dot{\xi}_d$, J is the Jacobian, M_X is the Cartesian generalized mass and $h(q,\dot{q})$ is the term related to gravity and coriolis effects.

However, the most common kinematic representation of the human arm has 7 DoFs (3 for the shoulder, 2 for the elbow and 2 for the wrist), while a Cartesian controller has only 6 independent inputs. To solve the redundancy of the kinematic chain, a common solution is to design additional controllers projected in the null space of the primary controller. In our case, we exploited this additional DoF to avoid collision with the whole arm (while the planner guarantees collision avoidance only for the end-point trajectory) and to control the elevation of the elbow. For collision avoidance, we can generate a set of repulsive forces that push away the arm from the obstacles and use them to control torque as:

$$\tau_{obs} = J_{elb}^T F_{elb} + J_{arm}^T F_{arm} + J_{forearm}^T F_{forearm} \tag{15}$$

The single repulsive force applied on the link is generated as:

$$F_i = \frac{\xi_i - \xi_{obs}}{\|\xi_i - \xi_{obs}\|} \cdot \frac{1}{d_i} \tag{16}$$

where d_i is the distance between the ith point in the kinematic chain and the centre of the obstacle. Regarding elbow posture, we can simply implement a joint impedance controller to keep a desired static posture. The control torque can be computed as:

$$\tau_{elb} = -K_v\dot{q} + K_p(\bar{q} - q) \tag{17}$$

where $\bar{q}$ is the desired arm posture. In our implementation, $\bar{q}$ is computed as mean posture when the participants were in the rest position during data recording in Averta et al. (2021). These two control actions can be added to the main controller (14) as τ_{null}:

$$\tau_{null} = P(\tau_{obs} + \tau_{elb}) \tag{18}$$

where $P = I - JJ^T$ is the Cartesian space null projector.

To perform these simulations, we selected 5 different tasks from the list of tasks contained in Averta et al. (2021) (Hitchhiking, Block out sun from own face, Stop gesture, Exultation and Self-feeding). These 5 tasks span most of the possible movement directions of the human hand. For each task, four different scenarios are simulated: (1) without obstacles, (2) a single obstacle in the middle of the hand path, (3) a single obstacle in the elbow proximity and (4) the previous 2 obstacles at the same time. Furthermore, to avoid collision with any part of the simulated hand, we set a clearance zone of 0.2 m from the surface of the obstacles to take into account the volumetric occupancy of the end-effector.

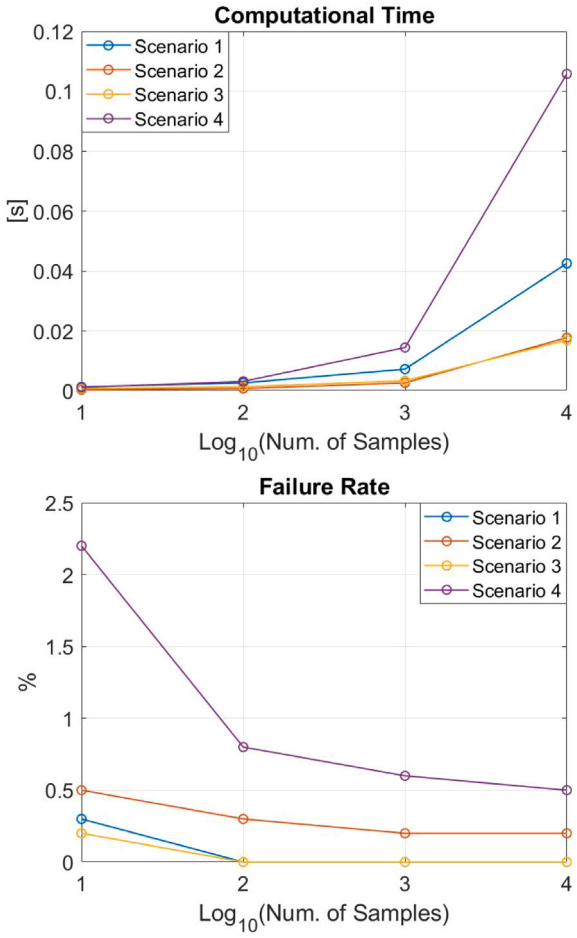

Fig. 1. Average computational time and failure rate for each scenario varying the number of samples used by the planning algorithm to find a suitable trajectory.

4. Results

4.1. Sampling density evaluation

Since our planning strategy is based on sampling, the first step is to evaluate how the number of samples used to find a possible trajectory affects the behaviour of the algorithm both in terms of computational time and characteristics of the trajectory returned. For this reason, we considered the same scenario described in Section 3.2 and we tested our algorithm using different numbers of samples, ranging from 10 to 10^4.

In terms of failure rate, the proposed algorithm achieved an overall failure rate which ranges from 0.8% using 10 samples only, to 0.18% using 10^4 samples. Regarding the computational time, it ranges from 0.848 ms using 10 samples to 46 ms using 10^4 samples. Merging these two pieces of information we found that 10^3 represents the most effective trade-off to fulfil performance and computational requirements, with an average computational time of 6.88 ms and a failure rate of 0.2%. In Fig. 1 the average computational time and failure rate are computed for each scenario separately for different numbers of samples. Of note, even if the computational time shown is low and suitable for one-shot planning of relatively long trajectories, it is generally larger compared to DMPs. In fact, while our approach has to compute the entire trajectory in advance, DMP allows computing it in real time, while the system is moving, making the latter more suitable for dealing with unstructured environments, e.g. where obstacles move and the trajectory must be recomputed at run time.

Table 1
Average RMS distance [m] between real human movement and the ones generated by the proposed algorithm (fPC) and DMP.

	fPC	DMP
RMS distance	0.0306 ± 0.0194	0.0411 ± 0.0256

Table 2
Median of average jerk of the real movements, our approach (fPC) and DMP [m/s^3].

Real	fPC	DMP
0.1976	0.0884	1.1503

Then, we evaluated if the different sampling densities affect the average jerk of the computed trajectory. To do this, we applied the Kruskal–Wallis test (Kruskal & Wallis, 1952) to check if the average jerk values obtained for each sampling density come from the same distribution. We applied this test for each scenario separately and we found that the number of samples used by the algorithm does not affect the jerk of the computed trajectory.

4.2. Comparison with real human motion

The first index we considered is the similarity between the Cartesian path of the human example movements and the ones produced by our proposed approach and DMPs. To this aim, we computed the Root Mean Square (RMS) distance between the real path and the one produced by the two algorithms. In Table 1 the values of this metric are reported for both our algorithm and DMP-based one, showing that they have similar performances in generating a path similar to the ones done by humans.

Another metric is related to the jerk of the computed trajectories. We computed the jerk for each trajectory contained in this subset and compared it with the ones obtained by the two planning algorithms. In Table 2 the median of the average jerk for each group is reported. We can see that DMP returns trajectory with a higher level of jerk with respect to our approach. We can also observe that real movements tend to have a jerk slightly higher with respect to the one returned by our approach. This can be explained by the fact that, given that the data of the real movements comes from measurements which, also after filtering, can contain residual noise that could amplify the level of computed jerk.

4.3. Large scale obstacle avoidance simulation

To compare the two planning algorithms we used different metrics to assess the effectiveness in reaching the target and the capability to maintain the human-likeness of the computed trajectory.

The first step is to check the capability of the proposed approaches to compute a trajectory which reaches the target in a given time. In our comparison, the two algorithms are based on completely different planning techniques that can have different problems in guaranteeing the reaching of the goal. For example, our approach is based on via-point sampling and, as shown in Section 4.1, it could be possible to not find a suitable trajectory. On the other hand, DMP generates trajectories exploiting a dynamic system and the convergence behaviour to the target point could strongly depend on the obstacle configuration in the scenario. To evaluate this aspect we have computed the distance from the desired point at the final time. In Table 3 the mean values and their relative standard deviation computed with respect to the five scenarios are reported. We can observe from these results that the fPCA-based planner can reach the target exactly at the desired time while DMP has a variable error depending on the configuration of the obstacles.

After this first assessment, we can move to evaluate the HL of the two algorithms. To do this we used the jerk of the trajectory. We can observe from Table 4 that the fPC-based solution outperforms consistently the DMP.

Table 3
Final target error at desired final time [m].

	fPC	DMP
Scenario 1	$2.12 \pm 2.02 \cdot 10^{-14}$	$3.24 \pm 5.42 \cdot 10^{-2}$
Scenario 2	$1.87 \pm 1.75 \cdot 10^{-14}$	$0.80 \pm 1.25 \cdot 10^{-2}$
Scenario 3	$1.87 \pm 1.71 \cdot 10^{-14}$	$1.84 \pm 3.24 \cdot 10^{-2}$
Scenario 4	$2.06 \pm 1.98 \cdot 10^{-14}$	$2.30 \pm 4.56 \cdot 10^{-2}$

Table 4
Median of the mean jerk [m/s^3].

	fPC	DMP
Scenario 1	0.22	1.19
Scenario 2	0.21	1.59
Scenario 3	0.22	1.31
Scenario 4	0.26	1.28

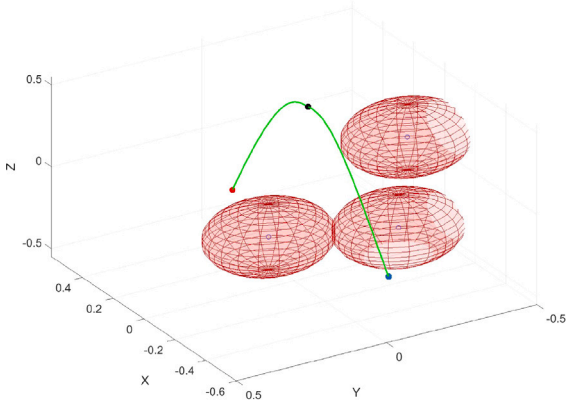

Fig. 2. An example of trajectory computed by our algorithm in the presence of obstacles. The starting point, the goal point and the viapoint are respectively drawn in blue, red and black. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

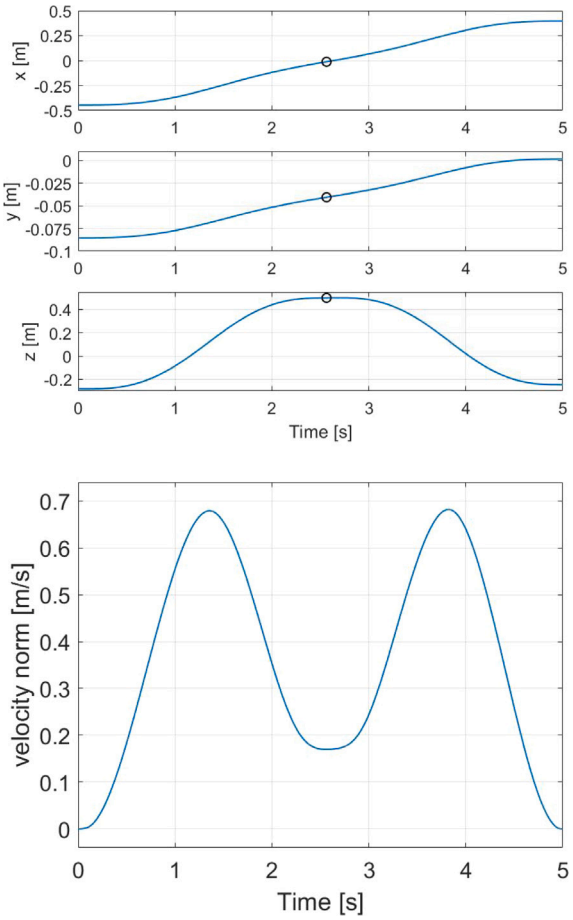

Fig. 3. Temporal evolution of the trajectory depicted in Fig. 2. In the first image, each Cartesian direction is shown separately (in black the viapoint used by the planning algorithm). In the second image, the plot of the velocity norm achieved by the computed motion is depicted.

Higher level of jerk ($p < 0.01$ Wilcoxon rank sum test) obtained by the DMP planner is related to the specific structure of the planner itself. In fact, the usage of a basis of function to learn the desired behaviour as done in DMP produces a ripple in the velocity profile, which brings to a higher level of jerk.

In Fig. 2 we have reported an example of a Cartesian trajectory computed with our approach while in Fig. 3 its position and velocity temporal evolutions are depicted. It is interesting to note that the behaviour shown by our planner, in terms of velocity profiles, is similar to the one reported in Flash and Hogan (1985).

4.4. Dynamics arm simulation

The second part of the validation involves also the dynamic of the kinematic chain. The goal is to check if the properties found in the previous analysis are also preserved by a manipulator which uses a Cartesian controller to follow the computed trajectory as reference. A snapshot of these simulations is reported in Figs. 4 and 5.

The first step, as done in the previous analysis, is to test the capability of the two planners to compute a trajectory to reach the target pose. Our planning algorithm is able to find a feasible trajectory for each of the 20 scenarios taken into account. Instead, the DMP has a final median error norm of 0.049 m. However, with this technique, there are two scenarios ("Hitchhiking" task with 2 obstacles and "Stop" task with 2 obstacles) which have noticeable errors in reaching the final pose (respectively 0.13 m and 0.48 m).

After that, we can evaluate the HL of the movement generated by the planners themselves and the combination between the planning algorithm and the controller. To quantify this property the main index used in this work is the jerk of the motion produced. Given that the end-effector trajectory has 6 DoFs, to keep the consistency of the measurement units, we split the jerk analysis into two values, one for the end-effector translation and one for the rotation. The first step is to evaluate jerk for the reference trajectories computed by the two algorithms. For each trajectory, the mean of the jerk norm of the movement is computed. The results are reported in Table 5. We can observe a significant difference between the jerk values produced by the two methods. The statistical difference between the two planning algorithms was tested with the Wilcoxon rank sum test which fails to reject the null hypothesis with a $p < 0.01$.

Table 5
Median of reference hand trajectory jerk.

	Translation [m/s^3]	Orientation [rad/s^3]
fPC	0.28	0.31
DMP	1.65	3.19

The same analysis was performed also on the actual simulated arm motion to check if the control policy affects the smoothness of the movement. In this case, we have evaluated both the Cartesian performance (as done for the reference movements) and the joint behaviour. For the joint domain, we have computed for each movement the average jerk across all the seven joints of the kinematic chain. From the results reported in Table 6, we can see that the proposed approach performs better than the DMP. Also in this case the statistical difference was tested using the Wilcoxon rank sum test and it fails to reject the null hypothesis with a $p < 0.01$.

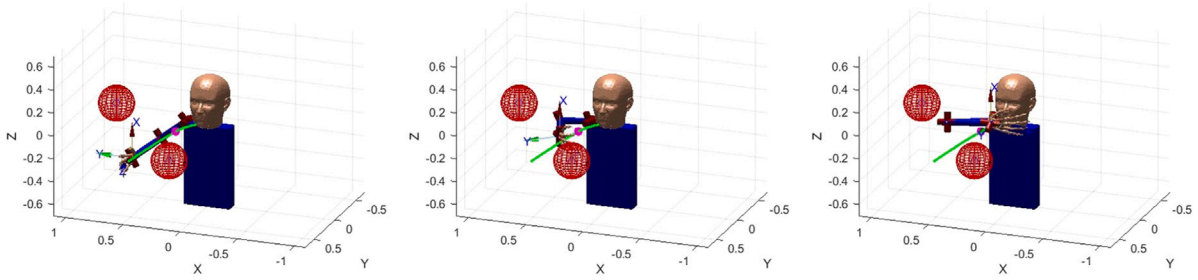

Fig. 4. Simulation of self-feeding task with two obstacles. The desired trajectory is depicted in green and the viapoint in purple. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

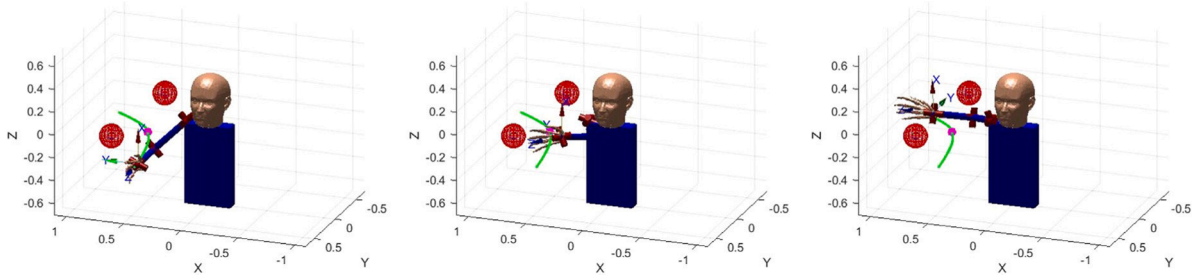

Fig. 5. Simulation of hitchhiking task with two obstacles. The desired trajectory is depicted in green and the viapoint in purple. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Median of real arm trajectory jerk.

	Translation [m/s^3]	Orientation [rad/s^3]	Joint [rad/s^3]
fPC	0.32	0.97	0.78
DMP	1.87	5.77	3.00

5. Conclusion

To conclude, in this paper, we present a new approach to compute Human-Like movement directly in the Cartesian domain. This algorithm exploits fPCA to extract human motion principal features and it directly embeds them in the trajectory without using optimization to guarantee Human-Likeness. We compared our method with Dynamic Motion Primitives and we tested for the effectiveness and the Human-Likeness of the planners. We demonstrated that using as reference real human motion, our approach shows a similar accuracy in terms of Cartesian path and a lower level of jerk with respect to DMP. We also showed that this lower level of jerk is maintained in the presence of obstacles and performing dynamic simulation with a simulated arm. We also tested the computational time required by our approach to provide a complete characterization. We found that it needs 6.88 ms on average to return a feasible trajectory.

In future developments, this planner will be implemented also in real robotic manipulators to further test the adaptability of our approach to real scenarios. This step will open a new test phase where this planning algorithm could be used in human–robot interaction scenarios to study if Human-Like behaviour produced by the framework improves the safety and the efficiency of this interaction. Other possible future developments could be the exploitation of the low computational time of the algorithm to perform dynamic tasks, such as the grasping of moving objects, or in the case of unstructured scenarios where fast replanning is required to avoid dangerous collisions. In the latter case, a possible solution could be also the integration of the proposed planner with a contact detection system (such as artificial skin (Luu, Nguyen, Nguyen, et al., 2023) or proprioceptive information (Zurlo, Heitmann, Morlock, & De Luca, 2023)). This type of framework would permit a rapid reaction to the collision and perform a prompt replanning of the trajectory exploiting the contact information.

CRediT authorship contribution statement

Marco Baracca: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Giuseppe Averta:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Matteo Bianchi:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Investigation, Funding acquisition, Conceptualization, Project administration, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 101017274 (DARKO), and Grant Agreement No. 871237 (SOPHIA); the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab and ForeLab projects (Departments of Excellence); the FAIR - Future Artificial Intelligence Research; and the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013; and project ECS00000017 'Ecosistema dell'Innovazione' Tuscany Health Ecosystem - THE, PNRR, Spoke 9: Robotics and Automation for Health).

References

Abubshait, A., & Wiese, E. (2017). You look human, but act like a machine: agent appearance and behavior modulate different aspects of human–robot interaction. *Frontiers in Psychology*, *8*, 1393.

Averta, G., Barontini, F., Catrambone, V., Haddadin, S., Handjaras, G., Held, J. P., et al. (2021). U-Limb: A multi-modal, multi-center database on arm motion control in healthy and post-stroke conditions. *GigaScience*, *10*(6), giab043.

Averta, G., Caporale, D., Della Santina, C., Bicchi, A., & Bianchi, M. (2020). A technical framework for human-like motion generation with autonomous anthropomorphic redundant manipulators. In *2020 IEEE international conference on robotics and automation* (pp. 3853–3859). IEEE.

Averta, G., Della Santina, C., Battaglia, E., Felici, F., Bianchi, M., & Bicchi, A. (2017). Unvealing the principal modes of human upper limb movements through functional analysis. *Frontiers in Robotics and AI*, *4*, 37.

Averta, G., Della Santina, C., Valenza, G., Bicchi, A., & Bianchi, M. (2020). Exploiting upper-limb functional principal components for human-like motion generation of anthropomorphic robots. *Journal of NeuroEngineering and Rehabilitation*, *17*(1), 1–15.

Baracca, M., Bonifati, P., Nisticò, Y., Catrambone, V., Valenza, G., Bicchi, A., et al. (2022). Functional analysis of upper-limb movements in the Cartesian domain. In *Converging clinical and engineering research on neurorehabilitation IV: proceedings of the 5th international conference on neurorehabilitation (ICNR2020), October 13–16, 2020* (pp. 339–343). Springer.

Bartneck, C., Kulić, D., Croft, E., & Zoghbi, S. (2009). Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International Journal of Social Robotics*, *1*, 71–81.

Cominelli, L., Hoegen, G., & De Rossi, D. (2021). Abel: integrating humanoid body, emotions, and time perception to investigate social interaction and human cognition. *Applied Sciences*, *11*(3), 1070.

Corke, P. (2007). MATLAB toolboxes: robotics and vision for students and teachers. *IEEE Robotics & Automation Magazine*, *14*(4), 16–17.

Duffy, B. R. (2003). Anthropomorphism and the social robot. *Robotics and Autonomous Systems*, *42*(3–4), 177–190.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, *5*(7), 1688–1703.

Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, *42*(3–4), 143–166.

Ginesi, M., Meli, D., Roberti, A., Sansonetto, N., & Fiorini, P. (2021). Dynamic movement primitives: Volumetric obstacle avoidance using dynamic potential functions. *Journal of Intelligent and Robotic Systems*, *101*, 1–20.

Ginesi, M., Sansonetto, N., & Fiorini, P. (2021). Overcoming some drawbacks of dynamic movement primitives. *Robotics and Autonomous Systems*, *144*, Article 103844.

Glas, D. F., Minato, T., Ishi, C. T., Kawahara, T., & Ishiguro, H. (2016). Erica: The erato intelligent conversational android. In *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)* (pp. 22–29). IEEE.

Grimme, B., Lipinski, J., & Schöner, G. (2012). Naturalistic arm movements during obstacle avoidance in 3D and the identification of movement primitives. *Experimental Brain Research*, *222*, 185–200.

Gulletta, G., Erlhagen, W., & Bicho, E. (2020). Human-like arm motion generation: A review. *Robotics*, *9*(4), 102.

Hanson, D. (2023). Hanson robotics website. URL https://www.hansonrobotics.com/sophia/.

He, C., Xu, X.-W., Zheng, X.-F., Xiong, C.-H., Li, Q.-L., Chen, W.-B., et al. (2021). Anthropomorphic reaching movement generating method for human-like upper limb robot. *IEEE Transactions on Cybernetics*, *52*(12), 13225–13236.

Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., & Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Computation*, *25*(2), 328–373.

Klein Breteler, M. D., Gielen, S. C., & Meulenbroek, R. G. (2001). End-point constraints in aiming movements: effects of approach angle and speed. *Biological Cybernetics*, *85*, 65–75.

Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, *47*(260), 583–621.

Kwiatkowski, A., Alvarado, E., Kalogeiton, V., Liu, C. K., Pettré, J., van de Panne, M., et al. (2022). A survey on reinforcement learning methods in character animation. In *Computer graphics forum: Vol. 41*, (pp. 613–639). Wiley Online Library.

Lacquaniti, F., Terzuolo, C., & Viviani, P. (1983). The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica*, *54*(1–3), 115–130.

Lentini, G., Grioli, G., Catalano, M. G., & Bicchi, A. (2020). Robot programming without coding. In *2020 IEEE international conference on robotics and automation* (pp. 7576–7582). IEEE.

Luu, Q. K., Nguyen, D. Q., Nguyen, N. H., et al. (2023). Soft robotic link with controllable transparency for vision-based tactile and proximity sensing. In *2023 IEEE international conference on soft robotics (roboSoft)* (pp. 1–6). IEEE.

Miossec, S., & Kheddar, A. (2009). Human motion in cooperative tasks: Moving object case study. In *2008 IEEE international conference on robotics and biomimetics* (pp. 1509–1514). http://dx.doi.org/10.1109/ROBIO.2009.4913224.

Mordatch, I., Lowrey, K., Andrew, G., Popovic, Z., & Todorov, E. V. (2015). Interactive control of diverse complex characters with neural networks. *Advances in Neural Information Processing Systems*, *28*.

Mourot, L., Hoyet, L., Le Clerc, F., Schnitzler, F., & Hellier, P. (2022). A survey on deep learning for skeleton-based human animation. In *Computer graphics forum: Vol. 41*, (pp. 122–157). Wiley Online Library.

Nguiadem, C., Raison, M., & Achiche, S. (2020). Motion planning of upper-limb exoskeleton robots: a review. *Applied Sciences*, *10*(21), 7626.

O'Rourke, J., & Badler, N. (1979). Decomposition of three-dimensional objects into spheres. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3), 295–305.

Park, D.-H., Hoffmann, H., Pastor, P., & Schaal, S. (2008). Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoids 2008-8th IEEE-RAS international conference on humanoid robots* (pp. 91–98). IEEE.

Piazzi, A., & Visioli, A. (2000). Global minimum-jerk trajectory planning of robot manipulators. *IEEE Transactions on Industrial Electronics*, *47*(1), 140–149.

Raab, S., Leibovitch, I., Li, P., Aberman, K., Sorkine-Hornung, O., & Cohen-Or, D. (2023). Modi: Unconditional motion synthesis from diverse data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13873–13883).

Rai, A., Sutanto, G., Schaal, S., & Meier, F. (2017). Learning feedback terms for reactive planning and control. In *2017 IEEE international conference on robotics and automation* (pp. 2184–2191). IEEE.

Ramsay, J. O., Hooker, G., & Graves, S. (2009). *Functional data analysis with R and MATLAB*. Springer Science & Business Media.

Rebol, M., Gütl, C., & Pietroszek, K. (2021). Real-time gesture animation generation from speech for virtual human interaction. In *Extended abstracts of the 2021 CHI conference on human factors in computing systems* (pp. 1–4).

Riek, L. D., Rabinowitch, T.-C., Chakrabarti, B., & Robinson, P. (2009). How anthropomorphism affects empathy toward robots. In *Proceedings of the 4th ACM/IEEE international conference on human robot interaction* (pp. 245–246).

Rosenbaum, D. A., Meulenbroek, R. G., Vaughan, J., & Jansen, C. (2001). Posture-based motion planning: Applications to grasping. *Psychological Review*, *4*(108), 709–734.

Rothstein, N., Kounios, J., Ayaz, H., & de Visser, E. J. (2021). Assessment of human-likeness and anthropomorphism of robots: A literature review. In *Advances in neuroergonomics and cognitive engineering: proceedings of the AHFE 2020 virtual conferences on neuroergonomics and cognitive engineering, and industrial cognitive ergonomics and engineering psychology, July 16-20, 2020, USA* (pp. 190–196). Springer.

Santello, M., Bianchi, M., Gabiccini, M., Ricciardi, E., Salvietti, G., Prattichizzo, D., et al. (2016). Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands. *Physics of Life Reviews*, [ISSN: 1571-0645] *17*, 1–23. http://dx.doi.org/10.1016/j.plrev.2016.02.001, URL https://www.sciencedirect.com/science/article/pii/S1571064516000269.

Saveriano, M., Abu-Dakka, F. J., Kramberger, A., & Peternel, L. (2023). Dynamic movement primitives in robotics: A tutorial survey. *International Journal of Robotics Research*, *42*(13), 1133–1184.

Schaal, S. (2006). Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. *Adaptive Motion of Animals and Machines*, 261–280.

Schulz, T., Torresen, J., & Herstad, J. (2019). Animation techniques in human-robot interaction user studies: A systematic literature review. *ACM Transactions on Human-Robot Interaction (THRI)*, *8*(2), 1–22.

Sutanto, G., Su, Z., Schaal, S., & Meier, F. (2018). Learning sensor feedback models from demonstrations via phase-modulated neural networks. In *2018 IEEE international conference on robotics and automation* (pp. 1142–1149). IEEE.

Uno, Y., Kawato, M., & Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, *61*(2), 89–101.

Vaughan, J., Rosenbaum, D. A., & Meulenbroek, R. G. (2001). Planning reaching and grasping movements: The problem of obstacle avoidance. *Motor Control*, *5*(2), 116–135.

Yin, W., Yin, H., Kragic, D., & Björkman, M. (2021). Graph-based normalizing flow for human motion generation and reconstruction. In *2021 30th IEEE international conference on robot & human interactive communication (RO-MAN)* (pp. 641–648). IEEE.

Zanchettin, A. M., Bascetta, L., & Rocco, P. (2013). Acceptability of robotic manipulators in shared working environments through human-like redundancy resolution. *Applied Ergonomics*, *44*(6), 982–989.

Zurlo, D., Heitmann, T., Morlock, M., & De Luca, A. (2023). Collision detection and contact point estimation using virtual joint torque sensing applied to a cobot. In *2023 IEEE international conference on robotics and automation* (pp. 7533–7539). IEEE.