IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

A Framework for Recognition and Prediction of Human Motions in Human-Robot Collaboration Using Probabilistic Motion Models

Thomas Callens[1,2], Tuur van der Have[3], Sam Van Rossom[3], Joris De Schutter[1,2] and Erwin Aertbeliën[1,2]

***Abstract*— This paper presents a framework for recognition and prediction of ongoing human motions. The predictions generated by this framework could be used in a controller for a robotic device, enabling the emergence of intuitive and predictable interactions between humans and a robotic collaborator. The framework includes motion onset detection, phase speed estimation, intent estimation and conditioning. For recognition and prediction of a motion, the framework makes use of a motion model database. This database contains several motion models learned using the probabilistic Principal Component Analysis (PPCA) method. The proposed framework is evaluated with joint angle trajectories of eight subjects performing squatting, stooping and lifting tasks. The motion onset and phase speed estimation modules are first evaluated separately. Next, an evaluation of the full framework provides more insight in the current challenges regarding motion prediction. A brief comparison between PPCA and the Probabilistic Movement Primitives (ProMP) method for learning motion models is made based on the influence of both methodologies on the performance of the framework. Both PPCA and ProMP motion models are able to predict motions over a short time horizon but struggle to predict motions over a longer horizon.**

I. INTRODUCTION

Human-robot collaboration (HRC) focuses on intuitive and predictable interactions between a human and a robotic device (e.g. a robotic arm as in [1] or an exoskeleton as in [2]). Using predictions of human motions in the controller of a robotic collaborator could be a crucial step in achieving such interactions. Thus, the need for automatic recognition and prediction of ongoing human motions arises. This paper presents a framework for recognition and prediction of ongoing human motions.

Crucial to this framework are mathematical representations of human motions. Such representations can be found in the Programming by Demonstration (PbD) paradigm. In PbD, a developer provides a number of demonstrations to teach a robot how to execute certain movements. Consequently, PbD requires models of human motion. One example of a technique to learn a human motion model makes use of probabilistic Principal Component Analysis (PPCA) [3]. Learning motion models using techniques such as PPCA requires formatting the raw human demonstrations provided by the developer. The recorded demonstrations need to be segmented and temporal differences need to be removed (i.e. "alignment" of the demonstrations). Once learned, the motion models are suitable tools for recognition and prediction of human motions. Using these motion models, a motion model database can be constructed containing separate motion models for every motion type (e.g. squatting, lifting etc.) that should be recognized and predicted.

Apart from the recognition and prediction of human motions, the framework takes two more steps into account: motion onset detection (i.e. detection of the start of a motion) and phase speed estimation (i.e. estimation of the execution speed of a motion). Consequently, the framework consists of four steps: *(i)* motion onset detection *(ii)* phase speed estimation *(iii)* intent estimation (i.e. recognition of an ongoing motion) and *(iv)* conditioning (i.e. fitting the most probable motion model to observations). The conditioned motion model is then used to generate predictions.

The contributions of this paper are the following: 1) A framework is proposed for human intent recognition and motion prediction. The framework combines several relevant methodologies and adds a novel motion onset detection module. To recognize and predict a motion, the framework makes use of a motion model database. 2) Methodologies are presented to construct a motion model database starting from raw (unprocessed) human motion data. PPCA is discussed as a method to learn motion models. However, a brief comparison between PPCA and the rather similar Probabilistic Movement Primitive (ProMP) method for learning motion models is made based on the influence of both methodologies on the performance of the framework. 3) The framework is evaluated on joint angle data of three different human motions: squatting, stooping and lifting tasks. Moreover, it is evaluated on its ability to predict the complete motion instead of only the end point of a motion.

The next section of this paper will discuss related work and motivate the choice for probabilistic motion models. A following section will present the framework. Subsequently, experiments and results are discussed. The paper ends with a discussion and some conclusions.

II. RELATED WORK

Some frameworks for human motion prediction have already been proposed. Luo and Mai as well as Luo, Hayne and Berenson propose frameworks to predict human reaching motions [4], [5]. However, these works do not address motion onset detection and phase speed estimation. Landi et al. propose a framework to predict if a human is reaching to grasp an object or not, but consider only one motion type (reaching motions) [6].

This work was supported by the Research Foundation-Flanders (FWO) through a research grant (Exo4Work, SBO-E-S000118N)

[1]Department of Mechanical Engineering, KU Leuven, Belgium thomas.callens@kuleuven.be

[2]Core Lab ROB, Flanders Make@KU Leuven

[3]Faculty of Kinesiology and Rehabilitation Sciences, KU Leuven, Belgium

A key aspect in the proposed framework is how to model human motions. A first group of motion models simulates dynamical systems to generate stable motions. Dynamic Movement Primitives (DMP) is a commonly used methodology [7]. Khoramshahi and Billard also propose a dynamical system but explicitly model interaction forces between human and robot [8].

A second group of motion models uses a probabilistic representation. An advantage of such models is their ability to capture the variance displayed by humans when providing demonstrations and to generate uncertainties on a prediction. These uncertainties can subsequently be used to adapt the control policy of a robotic device as discussed by Aertbeliën and De Schutter in [3]. Probabilistic Movement Primitives (ProMP) are well known probabilistic motion models [9] [10]. A lesser known but similar approach is based on probabilistic Principal Component Analysis and has been proposed by Aertebeliën and De Schutter [3]. It is used by Tanghe et al. to model gait [11] and by Vergara et al. in the control of a robotic arm [1]. Other probabilistic approaches use Gaussian Mixture Models (GMM) to recognize reaching motions [12]. Finally, some works already introduce unsupervised learning of new motions or sequencing of series of motions [5] [13].

Frameworks for recognition of human motions have also been proposed outside of the HRC domain in work by Lee and Nakamura on the classification of 3D whole-body motions of subjects wearing markers [14] and in work by Yang, Park and Lee on recognition of reaching or pointing motions [15]. In these vision based approaches, Hidden Markov Models (HMM) are used to model motions. However, processing images leads to higher computational load. In addition, those works do not discuss predicting ongoing motions.

Apart from motion recognition and prediction, the framework introduces a novel motion onset detection module based on Dynamic Time Warping (DTW) [16]. DTW is able to deal with temporal variations in human motion. It is often used to align two complete demonstrations. However, for motion onset detection, one of the demonstrations can be incomplete, limiting computational load.

Several aspects of the framework have been discussed in related work such as Tanghe et al [17] and Maeda et al. [10] and will be combined in this work. Motion models in the motion model database are learned using PPCA since it offers the advantages of a probabilistic approach. Given the similarities between PPCA and ProMP, it is interesting to quantify the impact of each method for learning motion models on the performance of the framework.

III. METHODS

Figure 1 gives an overview of the proposed framework. A motion model database (containing A motions models $a_1 \dots a_A$) is learned offline. The input to the framework (denoted as "sensor input") are human motion trajectories. For each of the tasks to be recognized and predicted, a motion model is learned and stored in the motion model database. The framework then exploits knowledge contained

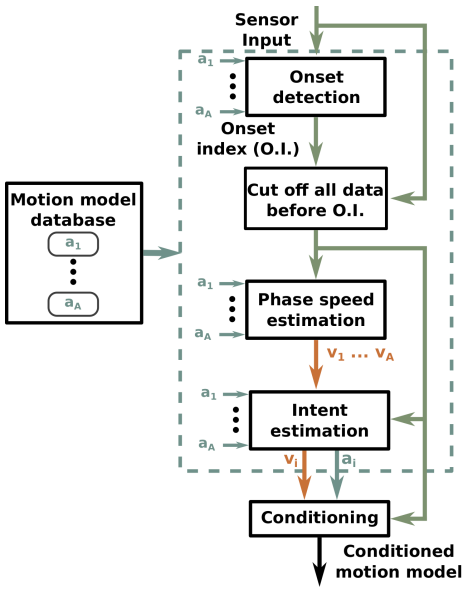

Fig. 1. Overview of the proposed framework.

in those motion models and outputs a "Conditioned motion model" used for prediction of the remainder of the human motion trajectory provided as input. In this paper, the sensor input is assumed to consist of joint angle trajectories. The first subsection discusses how to learn motion models for the database starting from raw joint angle trajectories. Next, The four steps mentioned in the introduction and visualized in the figure will be discussed. The acquisition of the joint angle trajectories for learning the motion models is discussed in section IV.

A. Learning motion models for the database

1) Formatting joint angle trajectories: This preprocessing step aims to obtain a set of demonstrations of equal length in which every demonstration is aligned with respect to a common time axis. Such a set of demonstrations optimally captures variability displayed by humans and will be used to learn a motion model. In a first formatting step, Dynamic Time Warping (DTW) [16] is used to extract individual demonstrations out of a series of demonstrations separated by brief pauses. Second, Local Time Warping is used to align segmented demonstrations with respect to a common time axis [18]. It is assumed that the demonstrations do not contain (partially) occluded datapoints.

Figure 2a) visualizes the segmentation process (for one dimensional trajectories). One manually segmented demonstration is used as a reference. The query sequence is the sequence to be segmented. The DTW-matrix is build using the symmetric stepping pattern with slope constraint zero and a squared euclidean distance as distance measure. After building the DTW matrix, warping paths are extracted wherever values of the final column of the DTW matrix go through a minimum. The first and last element of each warping path are written as $w_1 = (i_1, 1)$ and $w_K = (i_2, J)$ in which the first and second indices refer to the query and reference sequence, respectively. K is the length of the warping path and J the length of the reference series. Indices i_1 and

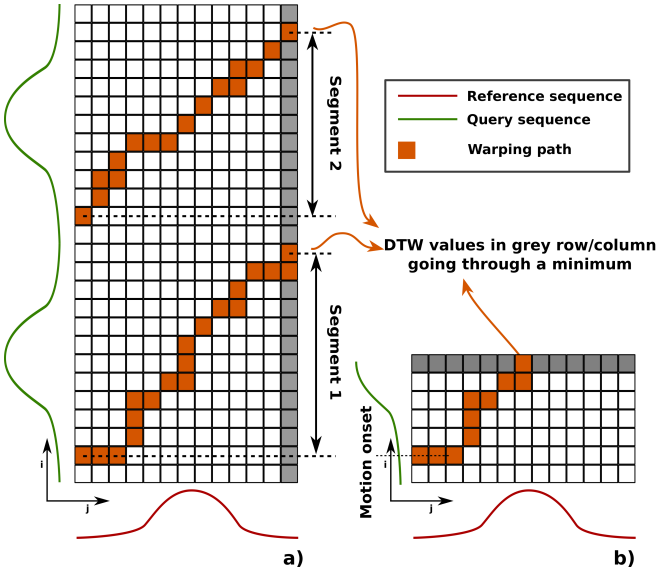

Fig. 2. The DTW processes for segmentation (part a) and motion onset detection (part b). The grid represents the DTW matrix. A match between the two time series is found whenever the values in the grey column (during segmentation) or row (during motion onset detection) go through a minimum.

i_2 provide the start and end points of the demonstrations in the query sequence. Note that this corresponds to a relaxation of the boundary conditions of Sakoe and Chiba [16].

After segmentation, the demonstrations have to be aligned to remove temporal differences between demonstrations. To avoid discontinuities when aligning the segmented demonstrations, Local Time Warping (LTW) is used [18] instead of the traditional DTW. Aligning a set of trajectories is not straightforward since a suitable reference sequence has to be picked. Gupta et al. propose the Nonlinear Alignment and Averaging Filter (NLAAF) to align sets of trajectories using DTW [19]. Although the procedure is aimed at symmetric DTW variants, it is also possible to use it with the asymmetric LTW variant used here. Using the LTW procedure in combination with the NLAAF method results in a set of demonstrations suitable for learning a motion model for every demonstrated motion type.

2) Learning motion models: A motion $\boldsymbol{f}$ (consisting of D, simultaneously recorded, joint angle trajectories) is represented (in discrete time) as:

$$\bar{\boldsymbol{f}}(t_n) = \begin{bmatrix} \bar{f}_1(t_n) & \dots & \bar{f}_D(t_n) \end{bmatrix} \tag{1}$$

$\bar{\boldsymbol{f}}$ indicates that $\boldsymbol{f}$ has been sampled at N discrete time steps (with $0 \leq n \leq N$).

A common modeling step is the normalization of time in (1) by replacing it with a phase variable s such that $s = 0$ at the start of the motion and $s = 1$ at the end of a motion. A linear progress model of the phase variable is a common choice:

$$s_n = v\frac{t_n}{T_{nom}} \tag{2}$$

with T_{nom} the duration of the demonstrations used to learn the motion model and t_n the time index of the ongoing motion. v is called the phase speed and can be interpreted as the execution speed of the motion. $v = 1$ means the ongoing motion is being executed at the same speed as the motion model while $v > 1$ or $v < 1$ indicate faster of slower execution, respectively.

PPCA now describes a model for $\bar{\boldsymbol{f}}$ using a vector $\boldsymbol{x}$ containing m latent variables [3] [20]:

$$\bar{\boldsymbol{f}}^T(s_n) = \bar{\boldsymbol{H}}(s_n)\boldsymbol{x} + \bar{\boldsymbol{b}}(s_n) + \bar{\boldsymbol{\epsilon}} \tag{3}$$

In which $\bar{\boldsymbol{f}}^T \in \mathbb{R}^{DN\times 1}$ and $\bar{\boldsymbol{b}} \in \mathbb{R}^{DN\times 1}$ represents the mean trajectory of the motion model. $\bar{\boldsymbol{H}} \in \mathbb{R}^{DN\times m}$ contains m basis functions that are weighted by the m latent variables in vector $\boldsymbol{x}$. Together, $\bar{\boldsymbol{H}}\boldsymbol{x}$ models the variation displayed by a subject during the demonstrations. Finally, $\bar{\boldsymbol{\epsilon}}$ is a $DN \times 1$ Gaussian distributed noise vector $\bar{\boldsymbol{\epsilon}} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$.

Aertbeliën and De Schutter [3] describe the procedure to learn values for $\bar{\boldsymbol{H}}$, $\bar{\boldsymbol{b}}$ and σ using the sample covariance matrix of $\bar{\boldsymbol{f}}^T$, which follows from the aligned set of demonstrations discussed in the previous subsection. The latent variables are normally distributed with $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$. Initially, this distribution satisfies $\boldsymbol{x}_{init} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. However, the conditioning (discussed later) modifies $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$.

The resulting distribution on $\bar{\boldsymbol{f}}$ at instance s_n now becomes:

$$\bar{\boldsymbol{f}}(s_n)^T \sim \mathcal{N}(\bar{\boldsymbol{H}}(s_n)\boldsymbol{\mu}_x + \bar{\boldsymbol{b}}(s_n), \bar{\boldsymbol{H}}(s_n)\boldsymbol{\Sigma}_x\bar{\boldsymbol{H}}(s_n)^T + \sigma^2\boldsymbol{I}) \tag{4}$$

Several different PPCA motion models are learned and stored in the motion model database.

B. Motion onset detection

The motion onset detection module outputs an "Onset Index" (O.I.). Using this index, only sensor input belonging to a motion is passed to subsequent modules. The approach followed is similar to the segmentation procedure (Figure 2b). However, the query sequence is now the "Sensor Input" from figure 1, while the reference sequence is chosen as the mean of a motion model.

Once more, for every match between query and reference series, a warping path can be constructed. The i value of w_1 (i.e. the first element of a warping path) is the O.I. In some warping paths, several consecutive points in the query series are matched with a single point in the reference series (or vice versa). Two additional constraints allow removal of a match with a warping path showing this behavior. Let $w_1 = (i_1, 1)$ and $w_K = (i_2, j_2)$ (with w_K the last element of a warping path), then any matches not satisfying $i_2 - i_1 > r_1$ and $j_2 - 1 > r_2$ are removed. This ensures that at least r_1 elements of the query sequence and r_2 elements of the reference sequence are used to estimate a motion onset index. Finally, if the value of w_K in the DTW matrix is too high i.e. it does not satisfy $DTW(i_2, j_2) < r_3$, the match is removed. If multiple matches satisfy all three conditions, then the match with lowest DTW distance is picked.

The reference series used in the onset detection process is given by the mean of a motion model. However, it is still unknown what motion model to pick (this is estimated later on in the intent estimation module). Therefore, this procedure

is repeated for every motion model in the database. To limit computation time, a window containing X observations is chosen. Only the X most recent observations of the "Sensor Input" are part of the query series.

C. Phase speed estimation

Once a motion onset has been detected, a logical next step is to select the correct motion model from the database (i.e. intent estimation). However, intent estimation is difficult since the phase speed variable (given by (2)) of the ongoing motion is unknown. On the other hand, estimating the phase speed variable is impossible without the correct motion model. Therefore, the phase speed variable is first estimated for every motion model in the database. Subsequently, the intent estimation module can take into account these estimated phase speed values.

Given the nonlinearity of the underlying problem, estimation of the phase speed variable is done with an Iterated Extended Kalman Filter (IEKF) similar to the approach followed by Tanghe et al. [17]. However, since Tanghe et al. only considered one motion model, no intent estimation was needed and the IEKF combined both phase speed estimation and conditioning. Since phase speed estimation and conditioning are now split up, the IEKF will estimate only one variable: v.

A constant process model is proposed for the IEKF:

$$v_{n+1} = v_n + \rho_p \tag{5}$$

with n the n-th time step at which the IEKF is run. ρ_p is the normally distributed process noise: $\rho_p \sim \mathcal{N}(0, \boldsymbol{Q})$ with $\boldsymbol{Q}$ the covariance matrix of the process noise (which is a scalar in the case of only one state variable). The measurement model follows from a linearization around the current state estimate:

$$\boldsymbol{z} = \boldsymbol{h}(v_n, t_n) + \left.\frac{\partial \boldsymbol{h}}{\partial v}\right|_{v_n, t_n} (v - v_n) + \rho_{meas} \tag{6}$$

with ρ_{meas} the normally distributed measurement noise: $\rho_{meas} \sim \mathcal{N}(0, \boldsymbol{R})$. $\boldsymbol{h}$ is the measurement function. This measurement function is equal to (3) but is expressed as a function of v_n and t_n using (2) (therefore, $\boldsymbol{h}(v_n, t_n) = \boldsymbol{f}(s_n)^T$). The partial derivatives $\frac{\partial \boldsymbol{H}}{\partial v}$ and $\frac{\partial \boldsymbol{b}}{\partial v}$ are calculated using numerical differentiation of $\boldsymbol{H}$ and $\boldsymbol{b}$ during the learning phase of a motion model.

The Kalman filter gain is calculated in this problem similar as in [17]. This requires writing the posterior state covariance matrix $\hat{\boldsymbol{P}}_n$ in Joseph form. The Kalman gain is then obtained by minimizing the trace of this covariance matrix. Two additional constraints are specified while determining the Kalman gain matrix:

$$\boldsymbol{K}_n = \arg\min_{\boldsymbol{K}} \text{trace} \left[\left(\boldsymbol{I} - \boldsymbol{K} \left.\frac{\partial \boldsymbol{h}}{\partial v}\right|_{v_n} \right) \hat{\boldsymbol{P}}_n \left(\boldsymbol{I} - \boldsymbol{K} \left.\frac{\partial \boldsymbol{h}}{\partial v}\right|_{v_n} \right)^T + \boldsymbol{K} \boldsymbol{R}_n \boldsymbol{K}^T \right]$$

$$\text{subject to } \boldsymbol{A}_n (v_{n-1} + \boldsymbol{K}\boldsymbol{\nu}_n) \leq \boldsymbol{c}_n$$

with $\boldsymbol{\nu_n}$ the innovation of the IEKF such that $v_n = v_{n-1} + \boldsymbol{K}_n \boldsymbol{\nu}_n$. In this case, the constraints are set such that the phase variable s satisfies $0 \leq s_n \leq 1$, leading to $\boldsymbol{A}_n = \begin{bmatrix} -\frac{t_n}{T_{nom}} \\ \frac{t_n}{T_{nom}} \end{bmatrix}$ and $\boldsymbol{c}_n = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. The measurement noise covariance matrix $\boldsymbol{R}$ was set equal to $\sigma_{avg}^2 I$ with σ_{avg} the average standard deviation in the motion model, calculated using the covariance matrix given by (4). The process noise variance $\boldsymbol{Q}$ is used as a tuning parameter in the IEKF. The IEKF keeps running until the difference between two successive estimates of v falls below λ_{tol} (set to 10^{-6}) or if the maximum number of iterations (set to 200) has been reached.

D. Intent estimation

Classification is done by calculating the posterior probability of a motion model given its prior probability, estimated phase speed variable and the "Sensor Input" of figure 1. For a motion model a, the posterior probability then becomes:

$$p(a|\boldsymbol{y}_{obs}) = \frac{p(\boldsymbol{y}_{obs}|a)p(a)}{\sum_a p(\boldsymbol{y}_{obs}|a)p(a)} \tag{7}$$

with $p(a)$ the prior probability of motion model a. Assuming A motion models, a common choice is to set $p(a) = 1/A$. $\boldsymbol{y}_{obs} \in \mathbb{R}^{D \times l}$ specifies a sequence of l observations of the D dimensional "Sensor Input" at corresponding time instances t_n. $p(\boldsymbol{y}_{obs}|a)$ is calculated by evaluating the multivariate normal probability density function of model a given by (4) at the values s_n corresponding to t_n. Note that $p(\boldsymbol{y}_{obs}|a)$ will tend to zero as more observations are taken into account. After evaluating (7) for every motion model in the database, the model with highest probability is selected.

Finally, note that the onset detection module provides a first guess of the currently ongoing motion but neglects information contained in the covariance matrices. Therefore, this approach is preferred.

E. Conditioning

Once the appropriate motion model is selected, it is conditioned (i.e. "fitted") using the available data with the following equations (based on [9]):

$$\boldsymbol{\mu}_x^{[\text{new}]} = \boldsymbol{\mu}_x + \boldsymbol{L}\left(\boldsymbol{y}_{obs} - \bar{\boldsymbol{H}}_s \boldsymbol{\mu}_x - \bar{\boldsymbol{b}}_s\right) \tag{8}$$

$$\boldsymbol{\Sigma}_x^{[\text{new}]} = \boldsymbol{\Sigma}_x - \boldsymbol{L}\bar{\boldsymbol{H}}_s \boldsymbol{\Sigma}_x \tag{9}$$

with $\boldsymbol{L}$ given by:

$$\boldsymbol{L} = \boldsymbol{\Sigma}_x \bar{\boldsymbol{H}}_s^T \left(\boldsymbol{\Sigma}_{y_{obs}} + \bar{\boldsymbol{H}}_s \boldsymbol{\Sigma}_x \bar{\boldsymbol{H}}_s^T \right)^{-1} \tag{10}$$

with $\boldsymbol{\Sigma}_{y_{obs}}$ expressing the uncertainty on the observations $\boldsymbol{y}_{obs}$ and subscript s indicating that the preceding parameter should be evaluated at the indices s_n corresponding to the time indices t_n at which the observations $\boldsymbol{y}_{obs}$ were registered. Notice that these equations correspond to the update step of a Kalman filter. If a prediction step with corresponding process noise is specified, the NIS (Normalized Innovation Squared) values can be used to check the consistency of observations with the motion model. The process noise is modeled as $\boldsymbol{\epsilon}_{cond} \sim (\boldsymbol{0}, \rho_{cond}^2 \boldsymbol{I}_m)$ and a

constant process model is proposed in which ρ_{cond} is used as a tuning parameter. A choice is made to condition on a new observation every 5% progress. As a consequence, a model is conditioned on 20 observations spread evenly across the normalized time axis. Note, however, that phase speed estimation and intent estimation happens at every time step.

The conditioned motion model is the output of the framework. Generating a prediction of a future time instance t_f is now possible by evaluating (4) at s_f corresponding to t_f and in which $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ are modified by (8) and (9).

IV. EXPERIMENTS AND RESULTS

The proposed methods are evaluated using joint angle data of eight subjects (5 men; age: 27.3 years ($\pm$8.8), body mass index: 21.6 kg/m^2 ($\pm$4.3)). All participants provided written informed consent prior to the start of the measurement and the local ethics committee (Universitair Ziekenhuis Leuven, S61611) approved all study procedures. Subjects were asked to perform squatting, stooping and lifting tasks with a box placed on the ground. During the tasks, 3D marker trajectories were captured using a 10 camera Vicon system (100 Hz, VICON, Oxford Metrics, Oxford, UK). The measuring protocol (including marker placement and the selection of a human model) is identical to the protocol mentioned in van der Have et al. [21]. Joint angles were calculated at each frame of the movement using a global optimization method for inverse kinematics implemented in OpenSim 3.3 [22], that minimized the weighted sum of squared differences between experimental and model marker positions. The shoulder elevation and hip flexion angles were selected to learn the motion models. The "Sensor Input" Is assumed to contain the same joint angles. Due to symmetry, only joint angle data from the right hand side was included in the motion models.

The squat and stoop procedures are the same as mentioned in [21]. For the lifting procedure, a rack was set in front of the participant at five different heights adapted to the individual participants' anthropometrics (ground level, knee height, hip height, shoulder height and 50 centimeters above shoulder height). Although participants performed the tasks with two different weights (a weight of 10 kg and a weight equal to 40 % of the arm lifting strength test [23]), no distinction was made between the corresponding joint angle trajectories in the evaluations discussed here. Only data from lifting the box from ground level to knee level and from ground level to above shoulder height was used. A total of 94 squatting motions, 94 stooping motions and 87 lifting motions were registered.

The lifting motions were segmented based on the velocity of the box (start/end point was selected whenever the velocity of the box rose above/fell below 0.0025m/s) and subsequently resampled to equal length. The squat and stoop demonstrations were segmented and aligned using the methods of section III. This ensured that the squat and stoop demonstrations included the part of the motion in which the subjects did not yet pick up the box.

Subsections IV-A until IV-D make use of the squat and stoop data while subsection IV-E makes use of the squat, stoop and lifting data. Sections IV-A and IV-B focus on assessing the performance of the motion onset detection and phase speed estimation modules and only consider a motion model database with PPCA motion models. Sections IV-C, IV-D and IV-E consider a motion model database with PPCA or ProMP motion models. Following the conclusion of [3], PPCA motion models were learned with a total of 5 latent variables. All ProMP motion models were learned with 20 basis functions per joint in the motion model (i.e. 40 basis functions in total). The evaluations were run on a laptop with a Intel Core i7-8650U 1.9GHz processor.

All evaluations followed an inter-subject cross validation scheme: formatted data (i.e. segmented and aligned data) from all but one subject was used as training data. Subsequently, raw data from the remaining subject was used as "Sensor Input" data in figure 1. This was repeated such that data from every subject was excluded once from the training data set. Note that each raw squat and stoop trial contains two demonstrations (one to pick up a box and one to put the box down). In between each demonstration, there is a limited period (of varying length) without motion.

A. Evaluation of onset detection

A first evaluation focuses only on the onset detection module using the squat and stoop data. The percentage of correctly detected motion onsets P_+, false positives P_- and false negatives N_- are reported (all with respect to the total number of motion onsets to be detected). Moreover, the raw data has been manually labeled and the average value of the absolute difference between manually labeled onset and detected onset $E_{abs,avg}$ is reported as well (in number of indices difference). Note that the sampling interval of the raw data is 100 Hz, meaning a difference of 10 indices corresponds to a timing difference of 0.1 s. Since the manual labeling is subject to interpretation, this metric only serves as a rough performance measure.

The values of thresholds r_1, r_2 and r_3 were set as follows:

$$r_1 = \frac{2}{3}X, \qquad r_2 = \frac{24}{30}X, \qquad r_3 = X\|\boldsymbol{q}\|_2^2 \tag{11}$$

with X (the number of observations taken into account for onset detection) set to 30, and $\boldsymbol{q} \in \mathbb{R}^{D\times 1}$ in which every element is set to 8. r_3 then models an average difference of 8° between query and reference series at every time step taken into account and for every joint in the query series.

As can be seen in table I, the algorithm is indeed able to detect most motion onsets. Table I also lists the average duration T_{avg} per evaluation of the motion onset detection module (it is evaluated whenever 10 new observations are available). Note that all the "missed" motion onsets belonged to demonstrations in which the subject was already holding a box.

B. Evaluation of phase speed estimation

This section focuses solely on the phase speed estimation module. By manually segmenting the raw demonstrations,

TABLE I

RESULTS OF THE ONSET DETECTION EXPERIMENT

	$P_+(\%)$	$P_-(\%)$	$N_-(\%)$	$E_{abs,avg}$ (#)	$T_{avg}(s)$
squat	93.55	1.06	7.45	15.34	0.095
stoop	94.68	1.06	5.32	18.19	0.096
total	**93.62**	**1.06**	**6.38**	**16.78**	**0.095**

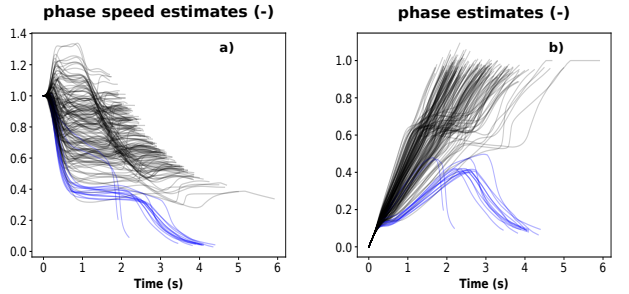

Fig. 3. Phase speed estimation throughout an ongoing motion as well as the corresponding phase values calculated through (2). The blue curves correspond to failures of the IEKF i.e. curves for which the estimated phase remains below 0.7 at the end of a motion.

no motion onset module was needed and demonstrations that would not be detected by the motion onset module could still be used to evaluate the phase speed estimation module.

Failure of the IEKF was decided based on the evolution of the motion phase variable s. As discussed earlier, s should progress linearly, should always satisfy $0 \leq s \leq 1$ and depends on v through (2). Figure 3 shows both the estimates of the phase speed variable and the corresponding estimates of the phase progress at every run of the IEKF. A run was classified as a failure whenever the final phase value was below 0.7 (shown in blue). Out of the 188 runs of the IEKF with both squat and stoop motions, failure occurred in 14 runs, leading to a success rate of 92.55%. Subject-specific tuning of the parameters of the IEKF might further improve the performance but was not done here. The process noise covariance matrix $\boldsymbol{Q}$ was set according to [24]. The initial estimate of v was set equal to 1 and its variance equal to 0.0038691, determined given the differences in execution time of the manually segmented raw demonstrations.

Finally, the performance of the IEKF was compared with the Moving Horizon Estimation (MHE) proposed in [11], with a horizon containing 10 observations. Comparison was done based on the average cumulative euclidean distance between estimated phase speed curves (such as shown in figure 3a) and benchmark curves. The benchmark was generated with a MHE with a horizon spanning a full trajectory. The average cumulative distance between IEKF and benchmark and between MHE and benchmark was 11.104 and 10.137, respectively. The average duration to estimate a phase speed at one time instance was 0.00341 s and 0.00499 s for the IEKF and MHE, respectively. While a MHE improves the performance slightly, it also increases computational load.

C. Evaluation of intent estimation accuracy

In this experiment, the full framework was used to evaluate the intent estimation performance. The next subsection evaluates the prediction performance. Thus, all modules of figure 1 are active. Whenever the phase variable s reaches

TABLE II

OVERVIEW OF THE INTENT ESTIMATION PERFORMANCE

	PPCA	ProMP
Total # demo's	188	188
Total # instances	564	564
$\rightarrow$ Onset detection failure		
$\rightarrow$ IEKF failure		
Registered instances	500	513
Correct estimated	435	456
Intent estimation accuracy (%)	**87**	**89**
Overall performance (%)	**77**	**81**

20, 50 or 80 % of a motion, the estimated motion type is compared with a ground truth. Since a total of 188 (squat and stoop) demonstrations are presented to the framework, the recognition of an ongoing motion is verified at 564 instances. The prior probabilities of squat and stoop motion ($p(a)$ in (7)) have been set to 0.5. A maximum of 30 points spread evenly across the observed motion interval are used for intent estimation. This evaluation was executed twice: Once with a database containing motion models generated with PPCA and once with a database containing motion models generated with the ProMP methodology. The measurement noise for the IEKF for ProMP motion models was set as discussed in the previous section but using equation (12) of Paraschos et al. [9]. All other parameters of the IEKF were identical as to the ones mentioned before.

Table II shows the results of this experiment. The intent estimation accuracy in table II is calculated as the number of instances at which the motion type (squat or stoop) was correctly estimated divided by the number of correctly registered instances. The overall performance is calculated as the number of correctly estimated instances divided by the total number of possible instances. As can be seen, using ProMP motion models leads to a higher overall performance. This follows from slightly higher success rates in the motion onset detection, phase speed estimation and intent estimation modules. Processing the intent estimation module took on average 0.0039 s for both ProMP and PPCA motion models

D. Evaluation of motion prediction accuracy

At each of the instances mentioned in the previous subsection, it is also possible to evaluate the prediction performance of the framework. This is visualized for one of the demonstrations in figure 4. At 20%, 50% and 80% motion progress, the selected and conditioned motion model (which is the output of the framework as shown in figure 1) is plotted. In all evaluations the conditioning process noise was set equal to $\rho^2_{cond} = 0.001$. A video visualizing the output of the framework is available as supplementary material as well.

Several remarks should be made. First, As can be seen, at 20 % progress, both motion models struggle to accurately predict both joint angles. However, at 50 % progress, some prediction errors have been compensated. Second, at 20 % progress, the phase speed is estimated to be rather high, leading to the assumption that the motion will end sooner than it does in reality. This high phase speed estimate is compensated as well as more observations become available. Finally, notice that at 50 % progress, the NIS value for the

PPCA motion model does not satisfy a 2 dof chi-squared consistency test with 2-sided 99% significance levels. Indeed, around the half-way point, squat and stoop motions are difficult to predict and show a lot of variability.

In general, predicting the shoulder elevation proves to be more difficult than the hip flexion (even over a short time horizon). This is visualized in the supplementary video indicating a horizon of $0.25s$ in the future following [17]. As a quantitative measure of prediction accuracy, the average absolute difference between each element of the mean of the conditioned motion model and the ground truth is calculated at the three different phase instances for all values after the time instance of last conditioning. This is done in such a way that all the elements of the shortest trajectory are used exactly once. With PPCA motion models this average difference amounted to 47.96°, 54.05° and 11.74° at 20%, 50% and 80% motion progress, respectively. Conversely, using ProMP motion models led to values of 38.21°, 42.96° and 14.65° at 20%, 50% and 80% motion progress. For PPCA and PromP motion models, the conditioning step takes $6 \cdot 10^{-5}s$ and $7 \cdot 10^{-5}s$, respectively.

E. Evaluation with lifting motions

As mentioned, the lifting data set contains motions in which subjects lift a box and put it on a rack at knee height or 50 cm above shoulder height. Lifting a box to a lower or higher level can be considered the same "task", hence a natural choice would be to model it with only one motion model. However, joint angle trajectories could vary a lot within this same task. This experiment evaluates the performance of PPCA and ProMP motion models to recognize and predict lifting motions. For this evaluation, the start detection and phase speed estimation module were turned "off". The starting point for each demonstration was simply the first observation and the phase speed value was manually set equal to one. Apart from a lifting motion model, the motion model database contained also a squat and stoop motion model to see if motions were recognized correctly. The same inter-subject evaluation scheme as discussed earlier was used.

Classification accuracy was 100 % when using a database build using PPCA or ProMP's. The same metric to evaluate predictions was used. The difference at 20%, 50% and 80% amounted to 29.71°, 18.69° and 8.62° when using PPCA motion models and 23.53°, 22.59° and 21.10° when using ProMP motion models, respectively. At 20 %, the average difference is considerably lower than the value mentioned for squat and stoop motions in the previous section (which can be due to the phase speed being manually set to 1). Note that the ProMP motion model is not able to accurately predict the motion or improve its estimates. Moreover, visual inspection of the results through figures similar to figure 4 indicates that the ProMP motion model is overfitting to the observed part of a motion.

V. DISCUSSION

The most time-critical step of this framework is the motion onset detection module which takes 0.09 s to analyze a window of 30 elements. However, the module does not need to run with every new observation. The module can run only whenever 10 new observations are available as was done here. Moreover, once a motion onset has been detected, this module can remain inactive until the detected motion has ended. Nevertheless, more efforts could be made to further reduce the time complexity of each of the modules to ensure a timely reaction of a robotic device using this framework.

Second, a range of design choices has been made. The motion models encoded a complete squatting or stooping motion although this could be split up in a "going down" and "coming up" motion. Furthermore, a decision was made to only include shoulder elevation and hip flexion angles in the motion models, to assign each motion model an equal prior and to condition a motion model only every 5 % progress. All of these decisions influence the performance of the framework but were manually tuned. Better approaches to set these variables could lead to increased performance. Note also that motions were assumed to end first before a second motion starts. This clear distinction might not always be present in industrial settings.

Third, choosing PPCA or ProMP to learn motion models has an impact on the performance of the framework. ProMP motion models achieve higher prediction performance for squat and stoop motions but lead to lower prediction performance with lifting motions. However, it should be noted that PPCA motion models only use 5 degrees of freedom in the model as opposed to 40 degrees of freedom in the ProMP models.

Finally, quantitatively evaluating predictions is a tedious task given the different steps before a prediction can be generated. Moreover, no metrics exist yet to determine when a prediction is good enough. Current literature in the field of Programming by Demonstration often focuses on prediction the end point of a motion. However, in applications such as exoskeletons, predicting the entire motion trajectory is of importance. Additionally, the framework is only able to accurately predict the near future but fails to predict a complete motion. It is unclear to what extent these predictions would lead to discomfort if e.g. used to control actuation of the joints in an exoskeleton.

VI. CONCLUSION

A framework for recognition and prediction of motions was proposed. Relevant approaches from the literature were combined and a motion onset detection module was added. The framework was evaluated on relevant human motion data. Two different approaches for learning motion models were evaluated. Good performance was achieved for the motion onset detection module, phase speed estimation module and intent estimation module. Performance of the prediction module is difficult to evaluate quantitatively. Several design choices were discussed such as setting of priors for motion models, selecting informative motion segments and joint angles during modeling phase and tuning of parameters for groups of subjects. Future work should focus on a quantitative comparison between this framework and relevant

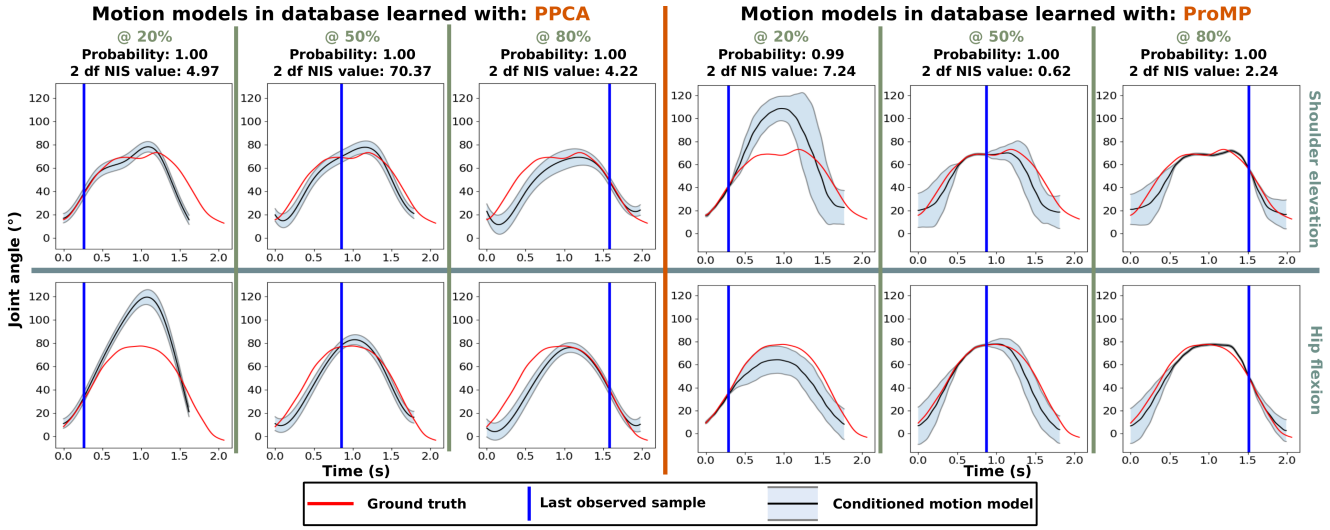

Fig. 4. Prediction performance of a PPCA or a ProMP motion model conditioned on data of an ongoing squat motion. The top row shows the shoulder elevation angle during a motion, the bottom row shows the hip flexion angle. The columns represent the different instances at which the conditioned motion models are plotted. The blue line indicates the last observation available to the motion model. Thus, everything to the right of it has to be predicted. The figure also shows the estimated probability of the squat motion as well as the 2 degree-of-freedom (df) NIS values.

methods from the state of the art, further improvements of prediction performance as well as an integration of this framework in the controller of a robotic device.

References

[1] C. Vergara, J. De Schutter, and E. Aertbeliën, "Combining Imitation Learning with Constraint-based Task Specification and Control," *IEEE Rob. and Autom. Letters*, vol. 4, no. 2, pp. 1–8, 2019.

[2] J. Vantilt, K. Tanghe, M. Afschrift, A. K. Bruijnes, K. Junius, J. Geeroms, E. Aertbeliën, F. De Groote, D. Lefeber, I. Jonkers, and J. De Schutter, "Model-based control for exoskeletons with series elastic actuators evaluated on sit-to-stand movements," *J. Neuroeng. Rehabil.*, vol. 16, no. 1, pp. 1–21, 2019.

[3] E. Aertbeliën and J. De Schutter, "Learning a Predictive Model of Human Gait for the Control of a Lower-limb Exoskeleton," in *Proc. 5th IEEE RAS/EMBS Int. Conf. Biomed. Robot. Biomechatron.*, São Paulo, 2014, pp. 520–525.

[4] R. C. Luo and L. Mai, "Human intention inference and on-line human hand motion prediction for human-robot collaboration," in *2019 IEEE/RSJ Int, Conf. on Int. Robots and Systems (IROS)*, 2019, pp. 5958–5964.

[5] R. Luo, R. Hayne, and D. Berenson, "Unsupervised early prediction of human reaching for humanrobot collaboration in shared workspaces," *Autonomous Robots*, vol. 42, no. 3, pp. 631–648, 2018.

[6] C. T. Landi, Y. Cheng, F. Ferraguti, M. Bonf, C. Secchi, and M. Tomizuka, "Prediction of human arm target for robot reaching movements," in *2019 IEEE/RSJ Int. Conf. on Int. Robots and Systems (IROS)*, 2019, pp. 5950–5957.

[7] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical Movement Primitives : Learning Attractor Models for Motor Behaviors," *Neural Computation*, vol. 25, pp. 328–373, 2013.

[8] M. Khoramshahi and A. Billard, "A dynamical system approach to task-adaptation in physical humanrobot interaction," *Autonomous Robots*, vol. 43, no. 4, pp. 927–946, 2018.

[9] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, no. 3, pp. 529–551, 2018.

[10] G. Maeda, M. Ewerton, G. Neumann, R. Lioutikov, and J. Peters, "Phase estimation for fast action recognition and trajectory generation in humanrobot collaboration," *Int. J. Robotics Research*, vol. 36, no. 13-14, pp. 1579–1594, 2017.

[11] K. Tanghe, F. De Groote, D. Lefeber, J. De Schutter, and E. Aertbelin, "Gait trajectory and event prediction from state estimation for exoskeletons during gait," *IEEE Trans. Neural Systems Rehabilitation Eng.*, vol. 28, no. 1, pp. 211–220, 2020.

[12] S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing and Generalizing a Task in a Humanoid Robot," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 2, pp. 286–298, 2007.

[13] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *Int. J. Robotics Research*, vol. 31, no. 3, pp. 330–345, 2011.

[14] D. Lee and Y. Nakamura, "Motion recognition and recovery from occluded monocular observations," *Rob. Auton. Systems*, vol. 62, no. 6, pp. 818–832, 2014.

[15] H. D. Yang, A.-Y. Park, and S.-W. Lee, "Gesture spotting and recognition for human-robot interaction," *IEEE Trans. Rob*, vol. 23, no. 2, pp. 256–270, 2007.

[16] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[17] K. Tanghe, A. Harutyunyan, E. Aertbeliën, F. De Groote, J. De Schutter, P. Vrancx, and A. Nowe, "Predicting Seat-Off and Detecting Start-of-Assistance Events for Assisting Sit-to-Stand with an Exoskeleton," *IEEE Rob. and Autom. Letters*, vol. 1, no. 2, pp. 792–799, 2016.

[18] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple humanrobot collaborative tasks," *Autonomous Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[19] L. Gupta, D. L. Molfese, R. Tammana, and P. G. Simos, "Nonlinear Alignment and Averaging for Estimating the Evoked Potential," *IEEE Trans. Biomed. Eng.*, vol. 43, no. 4, pp. 348–356, 1996.

[20] Bishop Christopher M., *Pattern Recognition and machine learning*, M. Jordan, J. Kleinberg, and B. Schölkopf, Eds. Singapore: Springer, 2006.

[21] A. van der Have, S. V. Rossom, and I. Jonkers, "Squat lifting imposes higher peak joint and muscle loading compared to stoop lifting," *Applied Sciences*, vol. 9, no. 18, pp. 1–20, 2019.

[22] S. Delp, F. Anderson, A. Arnold, P. Loan, A. Habib, C. John, E. Geundelman, and D. Thelen, "Opensim: open source to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[23] D. Chaffin, G. Herrin, and W. Keyserling, "Pre-employment strength testing: An updated position," *Journal of Occupational Medicine*, vol. 20, no. 6, pp. 403–408, 1978.

[24] F. De Groote, T. De Laet, I. Jonkers, and J. De Schutter, "Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis," *J. Biomechanics*, vol. 41, no. 16, pp. 3390–3398, 2008.