



Particle Filters vs Hidden Markov Models for Prosthetic Robot Hand Grasp Selection

Mohammadreza Sharif, Deniz Erdogmus, and Taskin Padir

Electrical and Computer Engineering Department, Northeastern University, 360 Huntington Ave
Boston, Massachusetts 02115, United States of America
{mrsharif,erdogmus}@ece.neu.edu
tpadir@northeastern.edu

Received (11/04/2018)

Revised (06/14/2019)

Accepted (07/20/2019)

Abstract. Robotic prosthetic hands are commonly controlled using electromyography (EMG) signals as a means of inferring user intention. However, relying on EMG signals alone, although provides very good results in lab settings, is not sufficiently robust to real-life conditions. For this reason, taking advantage of other contextual clues are proposed in previous works. In this work, we propose a method for intention inference based on particle filtering (PF) based on user hand's trajectory information. Our methodology, also provides an estimate of time-to-arrive, i.e. time left until reaching to the object, which is an essential variable in successful grasping of objects. The proposed probabilistic framework can incorporate available sources of information to improve the inference process. We also provide a data-driven method based on hidden Markov model (HMM) as a baseline for intention inference. HMM is widely used for human gesture classification. The algorithms were tested (and trained) with regards to 160 reaching trajectories collected from 10 subjects reaching to one of four objects at a time. The results show a classification accuracy of 91.3% and 82.2% for the entire reaching period for the PF and HMM methods, respectively.

Keywords: Prosthetic Robot Hand; Particle Filter Method; Human-in-the-loop System; Hidden Markov Model; Gesture Classification; Shared Control.

1 Introduction

Robotic prosthetic hands can compensate part of the lost ability for upper extremity amputees. Successful integration of the robot highly depends on how

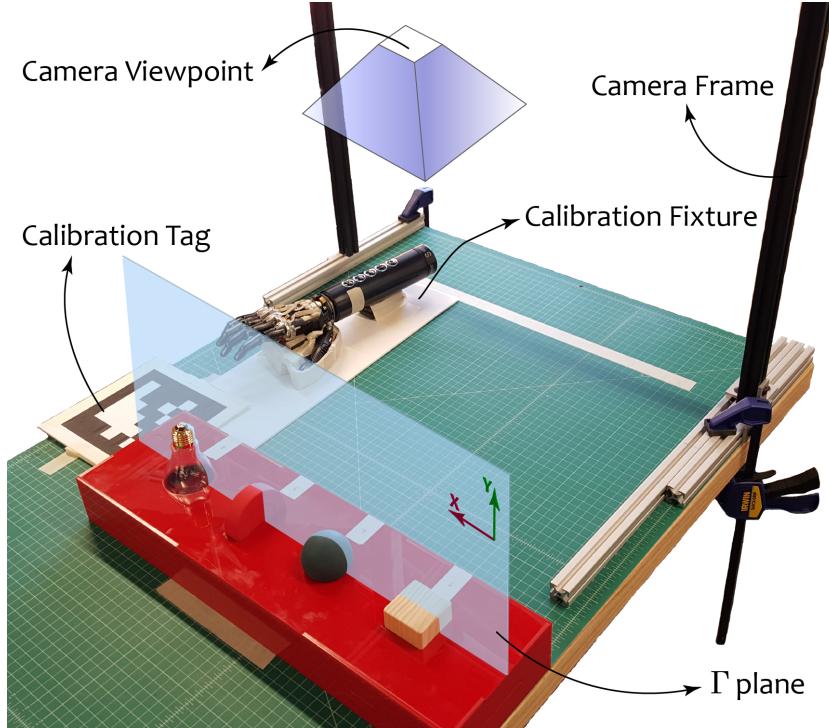


Fig. 1: The experimental setup for testing the hypothesis that our method can predict the projection of user intent on an arbitrary plane.

naturally it reacts to the environment and the user inputs. Thus, controlling the robot is very important besides its mechanical characteristics. Controlling robotic prosthetic hands is a good example of human-robot collaboration. It can be as simple as pressing a button for doing the grasp or changing the grasp type, or as complex as inferring user intent from the environment and bio-signals. In the latter case, the robot has to infer the grasp type it has to perform, the trajectory of finger tips and/or wrist joint(s), and the right time to execute the final grasp, on-the-fly. Intention inference can be achieved by utilizing cues from either the user or the environment.

Processing electromyography (EMG) signals from the user is one of the commonly used methods to infer grasp type or to trigger the grasp action. Using EMG alone, however, has been insufficient for real-life scenarios [14, 22, 50], due to issues such as electrode shift or change in number of electrodes [39], variations in electrode size and orientation [57], muscle fatigue [23, 36], variations in limb positioning [26], change of EMG signal patterns over time [2], and effect of prosthesis weight on signal patterns [8]. This can be partially explained by lack of enough diversity and variation in the training datasets [14, 24], however, one may argue that including such great diversity in datasets is at least very inconvenient for the end-users and clinicians, if not totally impossible [6, 7]. The situation becomes even more severe considering interpersonal differences and the

amount of data collection burden it may put on a single user. Although there are efforts on learning combined motions from single motions and hence relaxing this required variation partly [17], including all the different situations still needs a great amount of time and effort.

This can explain why most commercial prosthetic hands still use the fifty-year-old binary EMG signal inference technique [12], the two-channel open/close control, performed by monitoring levels of activation of EMG signals [22, 28]. High execution and reaction time as well as difficult training are two other issues with using EMG signals [44]. Re-calibration has been suggested as a simple workaround to adapt to new conditions and changing characteristics of the EMG signals from time to time. However, this usually puts extra effort and time on the user side, specially if they need to perform it on a daily basis [37].

To overcome the deficiencies of intent inference performed on merely EMG signals, it is proposed to use more inputs (or cues) [6, 28]. Markovic et al. [38], for instance, utilize vision (depth camera) and proprioception (by using IMU sensor) to help with the EMG classifier. Dosen et al. [10] use a camera-in-hand to infer grasp type and size, and a bipolar EMG sensor to trigger the grasp action. Patel et al. [41] adopt readings of IMU, force, and grip aperture sensors to help regulate the EMG signal classifier output. Gigli et al. [19] detect the object gazed at and uses vision to retrieve object information. A multimodal multi-kernel classifier is then used to find the most relevant grasp type given the extracted information. In a smart environment approach, Fukuda et al. [16] describe a platform in which each object is connected to the internet and has its own information stored, i.e. dimensions, shape, contact surface material, etc. The EMG signal is then used to decide the time of closing/opening. Knowledge of task *a priori* is another cue which can be used for intention inference. Patel et al. [42] ask the users the set of grasps they need in each experimental task before it starts and limit the grasp search space based on that. They show a 30% reduction in overall task completion time.

Inferring user intent in its most general case can be a very difficult task, since it is a hidden state of the system which can only be accessed through its signs (cues). Using more cues leads to more accurate intention inference. That is why in more recent works more cues are used to infer intention more accurately. However, to the best of our knowledge, no one has proposed a generalized framework that can systematically process the cues and give the distribution of user intent or at least its maximum likelihood estimate (MLE). Inferring distribution of the intention might be a bit complex though, as the Gaussian distribution assumption may not fit the system likelihood and posterior distributions or the system noises. So the conventional methods of Gaussian filtering (i.e. Kalman filter) may not apply to this problem. Because of this, non-Gaussian Bayes filters are considered in this paper as alternatives. Particle filter (PF) is a Bayes' filter in this class that can be used [54].

In this paper, we are proposing a framework based on the particle filter method which can systematically combine different cues and estimate distribution of the projection of the user intent on an arbitrary plane (we call it Γ

plane, see Fig. 1). Here, we use only kinematics of the motion obtained using a camera-based tracker. However, other cues can be fit in the framework readily.

As an alternative method and also a baseline to compare the PF algorithm with, we propose hidden Markov models (HMM) for intent inference. HMM has been widely used for modeling signals with temporal variations, e.g. speech recognition [18], hand-writing recognition [4,30], and human gesture detection [45,52]. An interesting property of HMM is its capability of incremental (lifelong) learning [33,34], where the model updates itself incrementally after failure cases or low-performance outcomes without retraining from scratch. Interestingly, incremental learning is also suggested to overcome the difficulties with the current approaches of EMG pattern recognition [5,7,20]. However, in this paper we consider using pure HMM with its ability to extend to incremental learning scenarios as an interesting future direction.

The contribution of this paper is thus the introduction of a PF framework for multi-cue intention inference as well as an HMM approach serving as a baseline and a possible future direction. This study extends the work in [51].

1.1 Motivation and Relevance

In order to motivate the research study presented in this paper, we consider a human prosthetic hand user performing a task which involves packaging of goods. This provides the research team with a controlled experimentation environment where additional information sources can be incorporated. Yet, the uncertainties in human decisions are maintained. The task involves the packing of a subset of known objects that can be picked up from pre-determined locations. This operational environment and task scenario justifies the assumptions made for the research study. The framework is aimed at enhancing the human-robot interaction for an amputee in the workplace by minimizing the time durations spent on information exchange between the user and the prosthetic hand.

1.2 Outline

The paper is organized as follows. Section 2 describes the problem using a probabilistic modeling framework as well as the baseline, followed by the methodology to perform user experiments. Experimental results are presented in Section 3. A discussion of results and conclusion and future work are included in Sections 4 and 5, respectively.

2 Methods

2.1 Problem Definition

If we define a plane Γ , which the user's hand is going to intercept while reaching to some objects, our objective is to find the projection of user intent on that plane. If we assume a discrete time system, we particularly define the system state to be:

$$x_t = [p_x, p_y, t_a] \quad (1)$$

where p_x is the x element of the projection of user intent on the plane, p_y is the y element of the projection of user intent on the plane, and t_a is the time remaining from instant t until the hand intercepts the plane.

We assume the system measurement z_t to be comprised of hand position elements:

$$\mathbf{z}_t = [H_{x,t}, H_{y,t}, H_{z,t}] \quad (2)$$

where $H_{x,t}$, $H_{y,t}$, and $H_{z,t}$ are the positions of the hand palm with respect to x axis, y axis, and z axis, respectively. The right-handed coordinate system origin is defined on the Γ plane where x and y axes are located on the plane, parallel and normal to the table top, respectively, and z axis is normal to the Γ plane (see Fig. 1).

If we assume the input to the system to be u_t , then our goal is to estimate the system posterior distribution P_t :

$$P_t = p(x_t | u_{1:t}, z_{1:t}) \quad (3)$$

where $1:t$ subscript indicates the series of that variable from instant 1 to instant t .

The main problem would then condense to finding the posterior distribution of system P_t given the sequence of measurements $z_{1:t}$ and inputs $u_{1:t}$. Note that in this problem we assume the input u_t to be null; however, we keep u_t in the formulation to keep the derivation general.

2.2 Particle Filter Method

Particle filter (PF) forms an estimate of the system state by a number of particles. Each particle represents a possible state of the system. These estimates are updated at every time step to improve the overall estimate and to match changes that have happened in the system state. Particles are first sampled randomly from the state transition distribution (i.e. prediction step), $p(x_t | u_t, x_{t-1})$. Since the distribution of particles may not match the posterior, they are then corrected using the likelihood (measurement distribution) of the system (i.e. correction step), $p(z_t | x_t)$. This is done through resampling in particle filter method. Resampling works by first assigning weights (based on the likelihood function) to each particle in the particle set, and then by randomly selecting particles (with replacement) based on their weights, so that particles with greater weights have more chance to survive. In other words, particles will gather around states in the state space that are believed to have higher weights. Sequential Importance Sampling (SIS) method, which particle filters is a special case of, elaborate more on the underlying mathematical foundations for resampling [3]. Algorithm 1 indicates a more formal description of the particle filter method, which recursively obtains the posterior distribution P_t . We assume the initial posterior to be uniform:

$$P_0 \propto \mathcal{U}(x_{min}, x_{max}) \quad (4)$$

Algorithm 1: Particle filter algorithm for intention inference. $x_t^{[m]}$ indicates the state of particle m at time t and η shows a normalization factor for the particles' weights to sum up to one.

Input: P_{t-1} , u_t , z_t
Output: P_t

- 1 N = Number of PF iterations;
- 2 M = Number of particles;
- 3 Sample $x_t^{[m]} \sim \mathcal{U}(x_{min}, x_{max})$ for $m = 1, \dots, M$;
- 4 **for** $k = 1$ to N **do**
- 5 Sample $x_t^{[m]} \sim p(x_t | u_t, x_{t-1})$ for $m = 1, \dots, M$;
- 6 Compute $\omega_t^{[m]} = \eta p(z_t | x_t^{[m]})$ for $m = 1, \dots, M$;
- 7 Resample particles (with replacement) based on computed weights $\omega_i^{[m]}$, for $m = 1, \dots, M$;
- 8 **end**

where x_{min} and x_{max} are minimum and maximum of system states, respectively. For a detailed derivation of the algorithm see [54].

State Transition Distribution State transition distribution is used to update the estimation of the states based on previous estimated states and current system inputs. We model the state transition distribution with a (multi-variate) Gaussian distribution:

$$p(\mathbf{x}_t | u_t, \mathbf{x}_{t-1}) \sim \mathcal{N}(\mu, \Sigma) \quad (5)$$

where μ is the mean and Σ is the covariance matrix. If we define the mean value of the particles' states,

$$\bar{\mathbf{x}}_t = \mathbb{E}[X_t]$$

we would then define the consecutive change in the mean particle state as the motion,

$$m_t = \bar{\mathbf{x}}_t - \bar{\mathbf{x}}_{t-1} \quad (6)$$

For the value of the μ in the above state transition equation we then assume

$$\mu = \mathbf{x}_{t-1} + \mathbf{m}_t \quad (7)$$

The covariance matrix Σ is chosen to be a diagonal matrix and its diagonal elements are determined by trial and error. Larger values of Σ diagonal elements yield better exploration of the state space in the corresponding eigenvector directions and easier avoidance of local minima, but smaller values lead to smoother results.

Measurement Distribution The measurement distributions are used to update the particle weights followed by the resampling step. The likelihood function might be assumed to be composed of several independent measurements, thus a product of several weighting functions:

$$p(\mathbf{z}_t | \mathbf{x}_t) \propto \prod_{i=1}^K w_i(\mathbf{x}_t) \quad (8)$$

where K is the number of available cues in the problem. Cues are problem-specific and depend on the information available from sensors. But we put no constraint on the number of weight functions used. For example, we may have weight functions corresponding to EMG/EEG signals, eye gaze trackers, hand pose and acceleration, etc. In this paper, our weight functions are based only on the hand trajectory for the purpose of demonstration. We define three different weight functions regarding different states in the state vector. In the next section we will go over the weight functions in detail (see Sec. 2.3).

As one of the improvements suggested for particle filter method [11], it is recommended to choose the weight functions to have “heavy” tails. It is possible to create the same effect by adding a small constant to our weight functions:

$$p(z_t | x_t) \propto \prod_{i=1}^K (\epsilon_{0,i} + w_i(x_t)) \quad (9)$$

This produces a “heavy-tailed” distribution even if we use an exponentially degrading tail distribution like Gaussian distributions. The value of $\epsilon_{0,i}$ may be determined by trial and error. For a certain weight function, the greater the value of its respective $\epsilon_{0,i}$ the more evenly it is distributed for different values of x_t . So we can adjust the sensitivity of model to the weight functions by tuning $\epsilon_{0,i}$.

Classification for item selection We use p_x and p_y from the system states to classify the item which the user is intending. We assume the ideal distribution of particles around item k on the Γ plane to be Gaussian:

$$I_k(x) \sim \mathcal{N}(\mu_k, \Sigma_k) \quad (10)$$

where μ_k is the projected location of item k on the Γ plane and Σ_k is the covariance matrix. As stated before, particle filter gives an approximation of the system posterior distribution function. So by comparing the distribution of particles in the Γ plane with I_k 's at each time instant t , we can estimate the item which is most likely to be intended at that time t . We compare the distribution using Kullback-Leibler (KL) divergence [35], which measures how close two distributions are:

$$D_{\text{KL}}(I_k \| \omega_t) = \sum_{m=1}^M I_k \left(\mathbf{p}_t^{[m]} \right) \log \frac{I_k \left(\mathbf{p}_t^{[m]} \right)}{\omega_t^{[m]}} \quad (11)$$

where $\mathbf{p}_t^{[m]} = [p_{x,t}^{[m]}, p_{y,t}^{[m]}]$ indicates the position for particle m on the Γ plane at time t . Then, we classify the user intent as the item with the lowest KL divergence.

2.3 Selecting weight functions

We have selected three different weight functions which are functions of current system state. Weight functions are supposed to encourage “true” states and discourage “false” states. The more uniformly a weight function is distributed, the less sensitive it is to noises. However, convergence to the “true” state would also happen in more iterations due to lower information content.

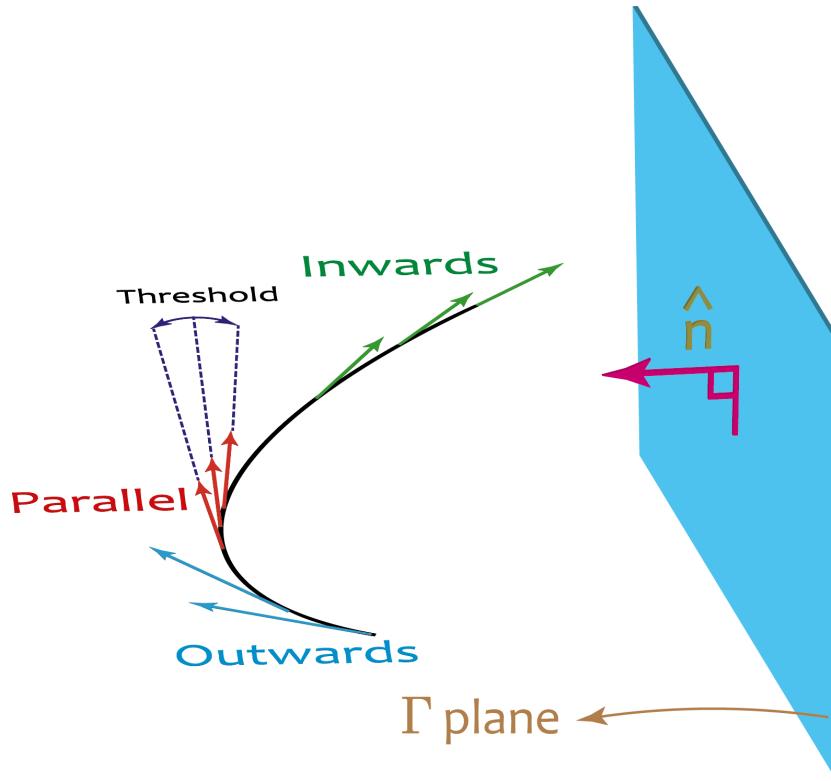


Fig. 2: We categorize hand trajectory into three regimes: “inwards”, “parallel”, and “outwards”. When the hand trajectory is pointing towards the Γ plane, we call it “inwards”, when it is parallel to the plane within a threshold, we call it “parallel”, and otherwise “outwards”.

Weight function for trajectory Depending on the direction of hand trajectory, we may have three cases, “outwards”, “parallel”, and “inwards” (see Fig. 2). The status of hand trajectory at instant i , s_i , is formally defined as:

$$s_i = \begin{cases} \text{outwards}, & \text{if } \mathbf{v}_i \cdot \hat{n} > \theta \\ \text{parallel}, & \text{if } |\mathbf{v}_i \cdot \hat{n}| \leq \theta \\ \text{inwards}, & \text{if } \mathbf{v}_i \cdot \hat{n} < -\theta \end{cases} \quad (12)$$

where \mathbf{v}_i is the hand's velocity vector at instant i and \hat{n} is the normal of the Γ plane.

We model the weight function for trajectory, W_r , by a Gaussian Mixture Model (GMM) for the last Q time instances, which is defined as following:

$$W_r = \sum_{i=t-Q}^t \omega_i \times \mathcal{N}(\mathbf{j}_i, \Sigma_r) \quad (13)$$

where ω_i specifies the weighing factor for the individual Gaussian models, \mathbf{j}_i identifies the mean used for the Gaussian models which will be discussed more, and Σ_r is a constant covariance matrix (see Fig. 3). In order to put more emphasis on the recent time instances, we assume the weighing factor ω_i of the Gaussian models to be exponentially decaying with time:

$$\omega_{t-i} = \lambda_0^i, \quad i = 0, 1, \dots, Q \quad (14)$$

where $\lambda_0 \leq 1$ is the decaying factor. By choosing $\lambda_0 = 1$ for example, all Gaussian models contribute equally to the GMM.

To define \mathbf{j}_i at time instance i , we fit both a first-order and a second-order curve to the last T points of the hand trajectory using least-squares method. Then, we find the intersection of these two curves with the Γ plane and call the intersection points (if they exist) $c_{1,i}$ and $c_{2,i}$, for the first-order and second-order curves, respectively. The second-order curve has always two real answers or none. $c_{2,i}$ is the one which satisfies the simple condition $\mathbf{v}_i \cdot (\mathbf{H}_i - c_{2,i}) > 0$, i.e. choosing the answer which is in the direction of hand velocity.

At time instance i , if the status of the hand trajectory, s_i , is "inwards", we define:

$$\mathbf{j}_i = \begin{cases} c_{2,i} & \text{if } c_{2,i} \text{ exists} \\ c_{1,i} & \text{otherwise} \end{cases} \quad (15)$$

If the status is "outwards" or "parallel" at instance i , we simply define \mathbf{j}_i to be projection of hand position on the Γ plane, i.e.:

$$\mathbf{j}_i = [H_{x,i}, H_{y,i}, 0] \quad (16)$$

This is a heuristic to maintain particle states in reasonable positions when the hand trajectory does not convey any useful information.

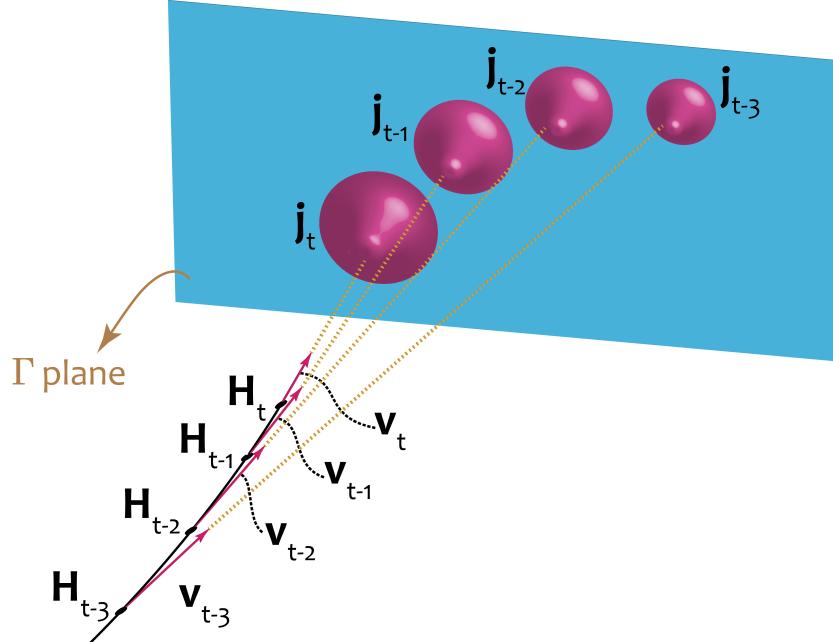


Fig. 3: We assume the extensions of user's hand trajectory to form a GMM on the Γ plane. The individual Gaussian models are depicted on the Γ plane. \mathbf{H}_i indicates hands position at instant i . \mathbf{v}_i indicates hand's velocity vector (tangent to trajectory) at instant i . We assume that previous time steps have a lower weight in the GMM.

Weight function for item proximity The trajectory weight function, W_r , helps us find an estimation of the user intent on the Γ plane. If we know the location of the objects *a priori*, and we assume that user's intention is to reach one of the objects, a helping weighting function would encourage particles remain close to one of the items. However, this multi-modal weight function may cause converging to local minima. Instead, one would limit the support for the weighting functions. In other words, we define:

$$W = \begin{cases} W_b^{[k]} & \text{if } \mathbf{H}_i \text{ in } \mathcal{R}_k \\ W_r & \text{otherwise} \end{cases} \quad (17)$$

where $W_b^{[k]}$ is the weight function of proximity to item k and \mathcal{R}_k is the region corresponding to item k . We simply define the regions \mathcal{R}_k to be spheres concentric with the objects center of geometries. We should note, however, for this method to work, regions must be mutually exclusive and therefore this weight function will not work for very cluttered environments.

The weight function for proximity to items is written as:

$$W_b^{[k]} = \frac{1}{1 + C_b \left((p_x - o_x^{[k]})^2 + (p_y - o_y^{[k]})^2 \right)} \quad (18)$$

where C_b is a constant which tunes importance of the weight function with respect to other weight functions, $o_x^{[k]}$ is the geometric center of item k in the x direction, and $o_y^{[k]}$ is the geometric center of item k in the y direction.

Weight function for time-to-arrive One of the states of the system is time-to-arrive t_a , i.e. the remaining time that we expect the hand to reach an object. This state is important because it can help to parameterize the grasp, i.e. adjusting the grasp aperture with respect to the time remaining to the object.

As denoted before, \mathbf{v}_i represents the hand velocity at instance i . We thus indicate the time-to-arrive implied by \mathbf{v}_i at instance i to be linearly defined as:

$$t_{a,i} = \frac{H_{z,i}}{\mathbf{v}_i \cdot \hat{n}} - t_{elapsed} \quad (19)$$

where $t_{elapsed}$ indicates the elapsed time since hand was at \mathbf{H}_i .

Assuming a GMM representation like Eq. (13) based on the history of the hand's trajectory, we have:

$$W_e^{[m]} = \sum_{i=t-Q}^t \omega_i \times \mathcal{N}(t_{a,i}, \Sigma_e) \quad (20)$$

where Σ_e is a constant covariance matrix. Gaussian model weighing factors, ω_i , are obtained as in Eq. (14).

2.4 Hidden Markov model

Hidden Markov model is a two-layer statistical model with the inner hidden layer being a discrete Markov model. The outer measurable layer simply emits observations from the inner Markov states [47].

HMM is parameterized by a transition matrix A which characterizes the transition probabilities between the Markov model states, emission probability matrix B which denotes the probability of the observation from each Markov state, and initial state distribution π which determines the probability of starting from each Markov model state. Hence, an HMM is fully characterized by $\lambda = (A, B, \pi)$. HMM is basically defined for discrete emissions probabilities. However, it can easily be generalized to continuous emissions by providing continuous emission probabilities.

HMM can both be trained in a supervised or unsupervised manner. In the unsupervised way the model learns a generative model of the provided data sequences. In the application of HMM for speech or motion recognition usually several HMMs are trained, one for each class, in an unsupervised manner. Although the training process of a single HMM is done in unsupervised mode, because one model is trained for each label, the overall process requires label, i.e.

a supervised training process. If we consider an observation sequence of length τ and associated with class κ defined as

$$\mathcal{O}^{[\kappa]} = \{\mathcal{O}_1^{[\kappa]}, \dots, \mathcal{O}_{\tau}^{[\kappa]}\} \quad (21)$$

then we are interested in the following maximum likelihood problem for training the HMM:

$$\lambda^{[\kappa]} = \underset{\lambda}{\operatorname{argmax}} P(\mathcal{O}^{[\kappa]} | \lambda), \quad \text{for } \kappa = 1, \dots, \Omega \quad (22)$$

where Ω is the number of classes.

Once model $\lambda^{[\kappa]}$ is trained, a new data sequence \mathcal{O} can be classified as the class with the maximum likelihood:

$$\operatorname{class}(\mathcal{O}) = \underset{\kappa \in \{1, \dots, \Omega\}}{\operatorname{argmax}} P(\mathcal{O} | \lambda^{[\kappa]}) \quad (23)$$

The likelihood of an HMM can be computed efficiently by the forward-backward algorithm [46] which utilizes dynamic programming in its core. To solve the maximum-likelihood problem of Eq. 22 one may use Baum-Welch algorithm [46] which uses forward-backward algorithm iteratively in an expectation-maximization (EM) framework. Baum-Welch algorithm first assumes the unknown parameters to be known with an arbitrary value, then computes the current likelihood. Then it tries to find modifications in the model parameters such that likelihood will increase an so on.

Since Baum-Welch algorithm—like all other EM class algorithms—can only find a *local* maximum, we run the algorithm for N_B times with different initialization and choose the one with the highest score (i.e. likelihood).

There are two popular architectures for the HMM's underlying Markov model, an ergodic (fully-connected) architecture and a left-to-right architecture. In the ergodic architecture, each state can have transitions to any other state, whereas in left-to-right architecture states can only have transitions to the states on their right, thus creating a one-directional architecture. Left-to-right architecture is particularly popular in the motion detection literature [32, 34, 53], however, [45] has shown that ergodic and left-to-right architectures have mostly the same performance for the purpose of motion detection. Therefore, we use an ergodic architecture in this paper for its better numerical stability.

For initialization of the HMM parameters, we use random initializatin for the transition matrix A and initial probability π . We use multivariate Gaussian distribution as the emission probability. Then, we would initialize the respective mean parameter by using a k-means clustering over training data sequences, with number of clusters k equal to the number of Markov model states N_M . We also assume the covariance matrix Σ_E of the multivariate Gaussian to be a diagonal matrix of the form:

$$\Sigma_E = \operatorname{diag}\left(\operatorname{cov}(\mathbf{X}^T) + \epsilon I\right) \quad (24)$$

where \mathbf{X} is the sequence of all training data concatenated into a large sequence, ϵ is a small constant to avoid non-singularity in the covariance matrix, and I is the identity matrix.

Since k-means clustering depends on the distance between data points, each feature within the input sequence must be scaled. We scale each feature with regards to the mean and variance of that feature in the training data for each class. In other words, test data is not used to find the scaling factors in order to avoid data leakage.

The features in the observation input can be any combination of the available sensor readings. The features for the motion detection are usually designed to be insensitive to displacement, i.e. translational invariance. While favorable in motion detection, this attribute would be detrimental to the intent detection in a setting where relative position of user's hand and objects matters. In this paper, we consider several possible combinations of features based on the hand position readings z_t and objects' locations prior. Each combination is trained within a range of different number of Markov states and then the best combination and number of Markov states are chosen as the final configuration. The candidate feature sets include:

$$\begin{aligned} (1) \mathcal{O}_t^{[\kappa]} &= \left[z_t^{[\kappa]} \right] \\ (2) \mathcal{O}_t^{[\kappa]} &= \left[\tilde{z}_t^{[\kappa]} \right] \\ (3) \mathcal{O}_t^{[\kappa]} &= \left[z_t^{[\kappa]}, \Delta z_t \right] \\ (4) \mathcal{O}_t^{[\kappa]} &= \left[z_0^{[\kappa]}, \Delta z_t \right] \end{aligned} \quad (25)$$

where $z_t^{[\kappa]} = z_t - o^{[\kappa]}$ is the hand position relative to item κ , $\tilde{z}_t^{[\kappa]} = \text{MovingAverage}(z_t^{[\kappa]})$ is the filtered relative hand position, $\Delta z_t = z_t - z_{t-1}$ is the first order difference of the hand position, and $z_0^{[\kappa]}$ is the initial relative hand position. Note that all candidate feature sets incorporate the knowledge of item locations by considering relative positions. This also obviates the need for knowing the origin of the world coordinate frame to compute the feature sets.

Here, we use $[\kappa]$ superscript with an abuse of notation to denote observation calculated *with respect to* class κ , which observation \mathcal{O} may or may not be associated with. In (21) this notation was used to denote observations that were knowingly associated with class κ . In order to calculate the likelihood of an observation associated with a particular class κ in Eq. (23), we use the observation calculated with respect to that class κ .

2.5 Experimental Setup

We have set up an experiment in which there are four objects and a robotic hand (see Fig. 1). The subjects were told to reach to the object identified by an announcement of the object. Each subject performed 16 reaching trajectories. The position of the objects are assumed to be known *a priori*, but the position

of the hand is obtained using depth sensor and a particle filter tracker from the PCL library [49]. After removing the points related to the table from the point cloud using RANSAC (random sample consensus) algorithm, the points related to the robotic hand are computed through clustering of the points and choosing the largest particle set. Later, particle filter algorithm is used to find the best match from the point cloud to the initially found cluster. The camera sees the table from top. The tracker was calibrated during the initialization of the system by putting it in a known position with respect to the *Calibration April Tag* [40] (shown in Fig. 1). The order of tasks was chosen based on a 4×4 balanced Latin square [55] to nullify order effects. The PF intention inference algorithm was then used offline using the real data to obtain results. Our PF algorithm produces inferred item and time-to-arrive at each time instance. The data were also used to train and test the HMM through k-fold cross-validation. The HMM algorithm only produces the inferred item and not the time-to-arrive. All the reported performance measures are averages over all folds.

3 Results

Our data consists of 10 subjects performing the tasks, 16 reaching trajectories each subject, 160 reaching trajectories overall. A sample hand trajectory for reaching to objects is depicted in Fig. 4. The item regions are also depicted. The data obtained from the users is segmented using a Matlab GUI so that each data sample only contains a single reaching trajectory, and user intent is known for each segment. We have used Matlab for implementation of the PF algorithm and Python with hmmlearn [1] and scikit-learn [43] modules for implementation of the HMM method.

Before feeding the input data to the PF inference engine, we passed the tracking information through a moving average filter with a window size equal to 10 to filter out high-frequency noises. The hyper-parameters of the PF method is selected through trial-and-error to be: number of particles $M = 100$, number of particle filter iterations $N = 10$, decaying factor $\lambda_0 = 0.99$, number of points in extension construction $T = 20$, and number of models in GMM $Q = 5$. A sample inference plot for the PF is depicted in Fig. 5. The different regimes of motion (“outwards”, “parallel”, and “inwards”) are also color-mapped in the figure.

The PF algorithm takes $T+Q = 25$ timesteps before making its first inference (shown as the white “none” region in Fig. 5). For fair comparison, this portion of data was not counted in the accuracy measurements of either PF and HMM methods. The classification error is computed on three different bases, the whole signal (100%), last 75% part of the signal, and last 50% part of the signal. Since the running time of each sample was different from others, we normalized each sample with respect to the number of data points in that sample in order to have a fair overall average.

The error of the time-to-arrive as an output of the PF algorithm was measured by comparing the Root Mean Square Error (RMSE) between estimated time-to-arrive and true time-to-arrive of the motion. The true time-to-arrive at

Sample hand trajectory for reaching to 4 objects

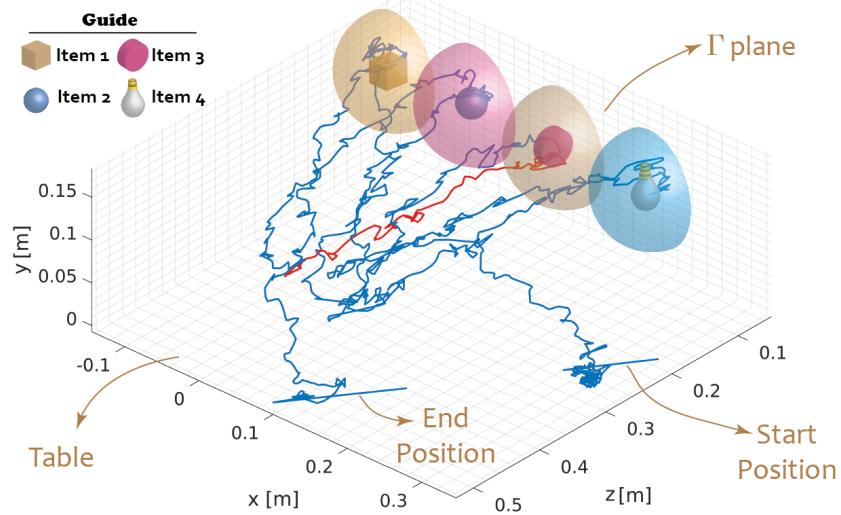


Fig. 4: A real sampled data. In this sample data, the subject reaches to four objects based on a specific order, and then retracts to the rest position. The reaching trajectory towards the third object is marked red as an example. The regions for item weight functions around objects are also shown.

time t was assumed to be linearly set by $t_f - t$, where t_f is the time that subject's hand reaches the object, i.e. the overall reaching duration. The normalized RMSE of the time-to-arrive based on the same three bases are $RMSE_{100\%} = 25.7\%$, $RMSE_{75\%} = 19.6\%$, and $RMSE_{50\%} = 14.8\%$.

For the HMM, 5-fold cross-validation was used for more accurate evaluation of performance. Also, the HMM was trained on different feature combinations (25) within a range of number of Markov states $N_M \in \{1, \dots, 42\}$. For the moving average in ${}^{(2)}\mathcal{O}_t^{[\kappa]}$, the window size is set equal to 10 similar to the moving average of the PF method. Each model was trained with $N_B = 15$ different initializations and the highest-score model was selected. The error plots for different feature sets for the HMM algorithm is depicted in Fig. 6. According to the results, the ${}^{(3)}\mathcal{O}_t^{[\kappa]}$ feature set is chosen with a number of Markov states $N_M = 4$ as the best result. Note that among models with similar performance the simpler one is preferred due to better generalization characteristics.

The confusion matrices of both PF and HMM (the chosen configuration) methods are depicted in one plot for easy comparison in Fig. 7.

4 Discussion

4.1 Particle filter method

Our results indicate that the proposed particle filter method can both classify items and infer time-to-arrive with acceptable accuracy and error, respectively. The accuracy improves over time as expected, and it particularly works very

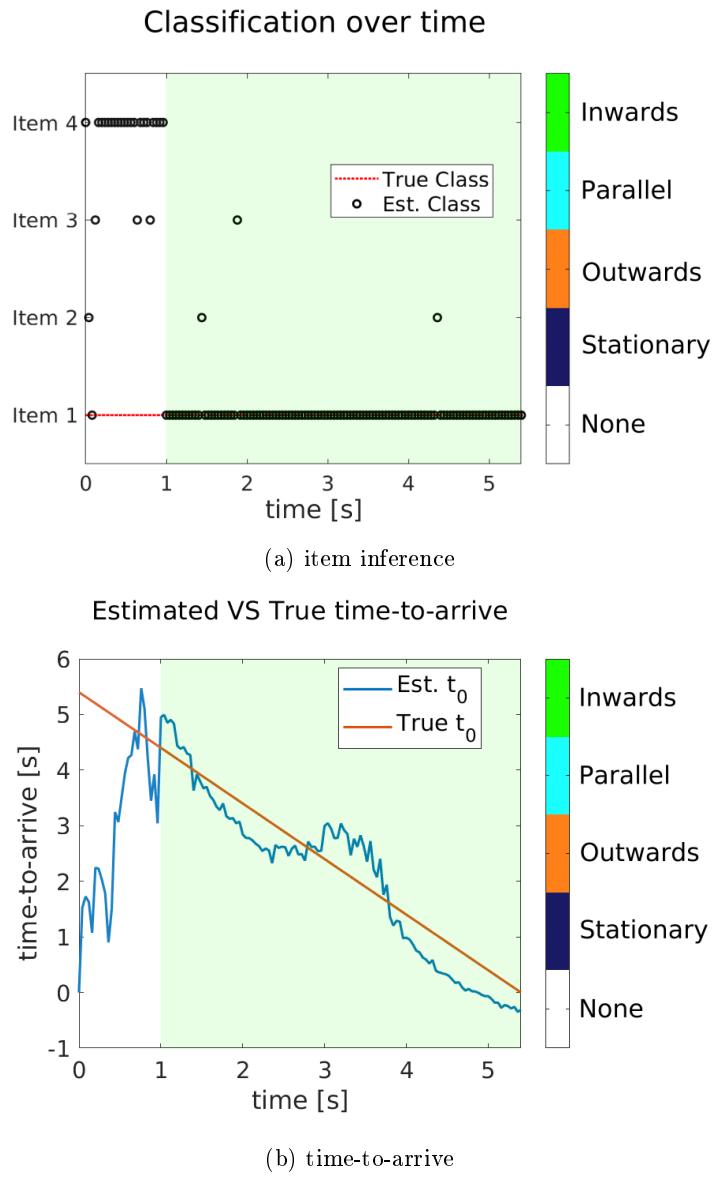


Fig. 5: Sample inference for one of the data samples.

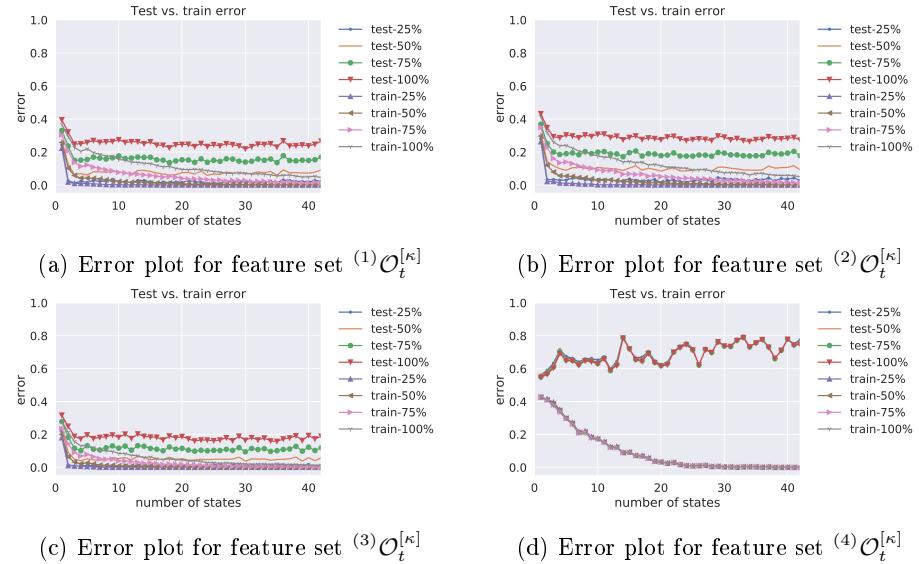


Fig. 6: Error plots for different feature sets (25) to be used with the HMM method. Both train and test errors are plotted on the same 100%, 75%, and 50% bases.

well in the last 75% of the trajectory. It is worth noting that at the very end of the hand trajectory the inference algorithm performs almost perfect due to the usage of items weight function W_b instead of the trajectory weight function W_r .

The accuracy for classifying items 2 and 3 is observed to be lower than items 1 and 4. This is because of the fact that items 1 and 4 are located at the boundaries, so are easier to infer.

It should also be noted that our model for time-to-arrive and trajectory weight functions do not include human trajectory models whatsoever and is rather based on heuristics for inferring the intention. In this paper we have emphasized more on a framework for multimodal intention inference rather than modeling individual cues per se. The algorithm also implies that the results might be improved by adding cues for more accurate inference.

For the PF algorithm to work in real-life, the implementation should be optimized for real-time inference. Due to the independence of particles, parallel-processing can readily be used for calculating particles' weights in parallel. Moreover, the moving average filter used for filtering the signal's noise may not be a good candidate for real-time applications, as it may incur very high phase lags in the signal, specifically with the current used window size. Although phase lags may not be completely removed from the filtering process unless the filter is memoryless, they can be minimized by optimizing the filter.

In the proposed PF algorithm, we constrain the intention inference to a plane. If the setting of objects is different, this method needs modifications to work.

Using time-to-arrive to modulate grasp types can be viewed very similar to the simultaneous and proportional (S/P) control paradigm versus on/off con-

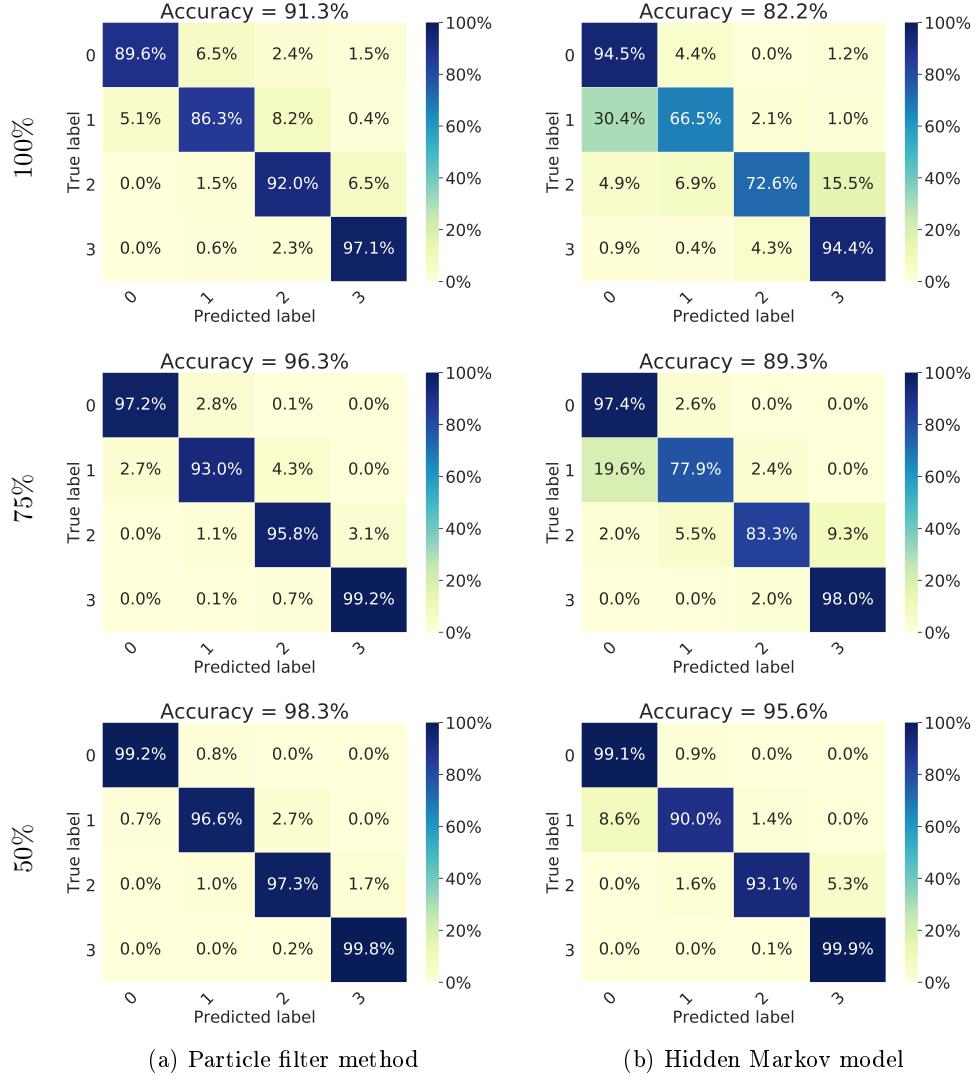


Fig. 7: Confusion matrices for the item inference by both PF and HMM methods. Confusion matrix of the HMM is computed over the test set while the confusion matrix of the PF method is computed over the whole dataset. The confusion matrices are depicted on the same 100%, 75%, and 50% bases.

trol paradigm which is normally resulted from classification-based EMG pattern recognition [6, 28]. In S/P control, the activation level of several degrees of freedom can be inferred simultaneously proportional to the degree of activation, which helps with a human-like motion [21, 22, 25, 29, 48, 56]. S/P control suggests smooth motions which do not suffer from mis-classifications of the pattern classifiers as severely as an EMG classifier totally mis-classifying a motion.

4.2 Hidden Markov model

In this paper, we have proposed HMM to infer intended object and not the time-to-arrive, as compared with the PF method. HMM is mostly suitable for classification tasks; however, there are variants like HMMR [15] that can be used for regression tasks and thus inferring time-to-arrive as well.

The feature sets used for testing the HMM show that $(^3)\mathcal{O}_t^{[\kappa]}$ (25), i.e. the observation featured with both the relative hand positions and the first order difference of it, performs the best. Interestingly, when hand position is filtered using moving average ($(^2)\mathcal{O}_t^{[\kappa]}$), the performance is degraded as compared to the unfiltered hand position ($(^1)\mathcal{O}_t^{[\kappa]}$). It could be because of the fact that moving average is removing some necessary information required by the HMM, so the performance is decreased. It should also be noted that $(^4)\mathcal{O}_t^{[\kappa]}$ has the worst performance. That could be due to the fact that this feature set is optimized for translational invariance which is not a desirable property for this problem.

The results for intent inference of our HMM method (in its best setting, i.e. using $(^3)\mathcal{O}_t^{[\kappa]}$ feature set) shows inferior performance to our PF method. That could be partially because of the fact that we have highly engineered features in the PF method (like the first and second order extension curves) that are missing from the HMM features. However, one could legitimately expect that the HMM performance becomes better by adding more engineered features like those of the PF method.

It should be noted, that just like the PF method, the HMM method improves in performance as greater portions of the signal is revealed to the algorithm (see Fig. 7). This behaviour is quite expected as greater parts of the input signal reveal more information about human intent.

In the proposed HMM method, as well as the PF framework, no ways are proposed to detect the onset of a motion towards an object. One may use EMG activation level for detection of the onset of a reaching trajectory or use threshold models [27] with the HMM method. Threshold models work by adjusting an adaptive threshold which shows the minimum amount of confidence we should have in a class in order to classify a motion within that class, thus safely discarding unrecognized patterns.

Both the PF and HMM methods provide a task-based training versus a user-based training approach. Methods working by relying on user hand's trajectory have shown to provide a controlling framework without need to calibration [31]. In other words, the two proposed methods may be more reliably transferable from user to user as they do not rely on the highly variable EMG signals. However, our methods are limited to doing a certain task.

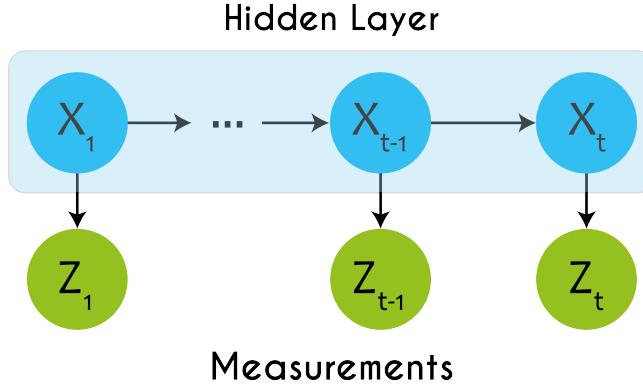


Fig. 8: Graphical model of the PF and HMM methods. Both methods have a latent variable X_t which is only observable through system observations Z_t .

Technically, both PF and HMM methods enjoy the same underlying graphical models as depicted in Fig. 8. The difference, however, is that in HMM all latent states are one-hot vectors with multinomial probability distributions. Each one-hot vector represents the active state at time t in the underlying Markov model. In the PF method, on the other hand, the latent variables are continuous random variables with a distribution determined by the particles. In the PF method, we are interested in the posterior distribution of latent states given the observations, while in the HMM we do not really care about the values of the latent states but the likelihood of an observation given the model parameters.

4.3 General points

Although the proposed algorithms assume no priors on the user intent, our experimental data does so. For instance, in the experiment we are assuming user intent cannot change during the reaching trajectory, and that, spatially, the user intent is placed somewhere on the object. For complex scenarios where the user intent might change while reaching to objects, the algorithms may still work, but as there are no ground-truth solutions available, it is hard to judge the performance of the algorithms in those cases.

In our experimental study, we assumed that we know the objects locations/affordances. In real-life this is usually not the case. To obtain object locations/affordances, assistive perception systems may be adopted to infer objects' locations/affordances.

Part of the error and noise is inflicted by the tracking system on the intention inference algorithms. The tracking system output has bias and noise. To improve tracking system output, one would use multiple depth cameras for tracking. Integration with other sensors like IMUs can be helpful as well. The algorithm used for doing tracking is another aspect worth improvement. Currently, the particle-filter based tracker used from the PCL library [49] cannot adapt itself to fast target movements above a certain velocity. That is why we have constrained ourselves to slow hand motions. Slow hand motions may not

be simply a scale of the normal fast hand motions, and this may lead to another source of error. However, the algorithm proposed here assumes very little about the kinematics of the hand trajectory and may still be used with faster motions without restrictions.

While we assumed a controlled smart environment for an amputee's workplace, we may still generalize this paper to more regular usage by using a head-mounted camera instead of an overhead camera. With smart glasses gaining more and more popularity [13], having a head-mounted camera is becoming a practical and valid assumption. Since the proposed algorithm relies on hand position information from the camera and not the camera position as such, it can be modified to include such cases.

Acceptance of new enabling technologies by patients has always been a challenge for researchers. Studies show that the rejection rate of electric upper limb prostheses differs from about 35% for children to about 23% for adults [9]. It is hard to discuss the rejection rate of the proposed method without actual tests by patients, in both controlled and real-life environments, specially over long periods of time. The proposed methods do not require extensive training by patients before use and are more task-oriented rather than user-oriented, which can be considered as an advantage by users doing activities in predefined environment.

5 Conclusion

In this paper, we proposed two methods for intention inference for interaction between human and a robotic prosthetic hand. The particle filter algorithm can predict which object the user is intending to reach and how much time is remaining before hand reaches the object. The time-to-arrive can then be used for finger trajectory planning so that they have a smooth reach-to-grasp motion. The hidden Markov model, on the other hand, only predicts which object the user has intended.

The PF algorithm also provides a general multimodal framework for incorporating several cues through weight functions. The weight functions come from different sources of information that may imply user intent. In this paper, by using heuristic weight functions based only on trajectory of the user hand we could reach an accuracy as high as 91.3%. The accuracy of the HMM was 82.2%, as a baseline model.

Generalization of both methods (PF and HMM) to intention inference in cluttered environments remains a task for the future. Currently both algorithms depend on prior knowledge about object positions. Removing known object position priors from the algorithm is another future task. Using the HMM model for incremental learning also seems a promising path to follow.

Acknowledgements

This work was supported by National Science Foundation (NSF) under the award number 1544895. Authors would like to thank Dr. Philip Long for his insightful comments on the paper draft and Peter G. Downward for his great help with setting up the experiments.

References

1. Hidden markov models in python, with scikit-learn like api, <https://github.com/hmlearn/hmlearn>
2. Amsüss, S., Paredes, L.P., Rudigkeit, N., Graimann, B., Herrmann, M.J., Farina, D.: Long term stability of surface EMG pattern classification for prosthetic control. Conf. Proc. IEEE Eng. Med. Biol. Soc. **2013**, 3622–3625 (2013)
3. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (Feb 2002)
4. Bunke, H., Roth, M., Schukat-Talamazzini, E.G.: Off-line cursive handwriting recognition using hidden markov models. Pattern Recognit. **28**(9), 1399–1413 (Sep 1995)
5. Castellini, C.: Incremental learning of muscle synergies: From calibration to interaction. In: Human and Robot Hands: Sensorimotor Synergies to Bridge the Gap Between Neuroscience and Robotics (Feb 2016)
6. Castellini, C., Artemiadis, P., Wninger, M., Ajoudani, A., Alimusaj, M., Bicchi, A., Caputo, B., Craelius, W., Dosen, S., Englehart, K., Farina, D., Gijsberts, A., Godfrey, S.B., Hargrove, L., Ison, M., Kuiken, T., Marković, M., Pilarski, P.M., Rupp, R., Scheme, E.: Proceedings of the first workshop on peripheral machine interfaces: going beyond traditional surface electromyography. Front. Neurorobot. **8**, 22 (Aug 2014)
7. Castellini, C., Bongers, R.M., Nowak, M., van der Sluis, C.K.: Upper-Limb prosthetic myocontrol: Two recommendations. Front. Neurosci. **9**, 496 (2015)
8. Cipriani, C., Sassu, R., Controzzi, M., Carrozza, M.C.: Influence of the weight actions of the hand prosthesis on the performance of pattern recognition based myoelectric control: preliminary study. Conf. Proc. IEEE Eng. Med. Biol. Soc. **2011**, 1620–1623 (2011)
9. Cordella, F., Ciancio, A.L., Sacchetti, R., Davalli, A., Cutti, A.G., Guglielmelli, E., Zollo, L.: Literature review on needs of upper limb prosthesis users. Front. Neurosci. **10**, 209 (May 2016)
10. Došen, S., Cipriani, C., Kostić, M., Controzzi, M., Carrozza, M.C., Popović, D.B.: Cognitive vision system for control of dexterous prosthetic hands: Experimental evaluation. J. Neuroeng. Rehabil. **7**(1), 42 (Aug 2010)
11. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later. In: OXFORD HANDBOOK OF NONLINEAR FILTERING (2011)
12. Farina, D., Jiang, N., Rehbaum, H., Holobar, A., Graimann, B., Dietl, H., Aszmann, O.C.: The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges. IEEE Trans. Neural Syst. Rehabil. Eng. **22**(4), 797–809 (Jul 2014)
13. Forrester Research: Number of smart glasses users in the U.S. workforce from 2016 to 2025 (in millions). www.statista.com/statistics/731753/us-number-of-workers-adopting-smart-glasses/, accessed: 2019-1-14
14. Fougner, A., Scheme, E., Chan, A.D.C., Englehart, K., Stavdahl, O.: Resolving the limb position effect in myoelectric pattern recognition. IEEE Trans. Neural Syst. Rehabil. Eng. **19**(6), 644–651 (Dec 2011)
15. Fridman, M.: Hidden Markov model regression. Ph.D. thesis, University of Pennsylvania (1993)
16. Fukuda, O., Takahashi, Y., Bu, N., Okumura, H., Arai, K.: Development of an IoT-Based prosthetic control system. Journal of Robotics and Mechatronics **29**(6), 1049–1056 (2017)

17. Furui, A., Eto, S., Nakagaki, K., Shimada, K., Nakamura, G., Masuda, A., Chin, T., Tsuji, T.: A myoelectric prosthetic hand with muscle synergy-based motion determination and impedance model-based biomimetic control. *Science Robotics* **4**(31), eaaw6339 (Jun 2019)
18. Gales, M., Young, S.: The application of hidden markov models in speech recognition. *Found. Signal. Process. Commun. Netw.* **1**(3), 195–304 (2008)
19. Gigli, A., Gijsberts, A., Gregori, V., Cognolato, M., Atzori, M., Caputo, B.: Visual cues to improve myoelectric control of upper limb prostheses (Aug 2017)
20. Gijsberts, A., Bohra, R., Sierra González, D., Werner, A., Nowak, M., Caputo, B., Roa, M.A., Castellini, C.: Stable myoelectric control of a hand prosthesis using non-linear incremental learning. *Front. Neurorobot.* **8**, 8 (Feb 2014)
21. Hahne, J.M., Rehbaum, H., Biessmann, F., Meinecke, F.C., Müller, K.R., Jiang, N., Farina, D., Parra, L.C.: Simultaneous and proportional control of 2D wrist movements with myoelectric signals. In: 2012 IEEE International Workshop on Machine Learning for Signal Processing. pp. 1–6 (2012)
22. Hahne, J.M., Schweisfurth, M.A., Koppe, M., Farina, D.: Simultaneous control of multiple functions of bionic hand prostheses: Performance and robustness in end users. *Science Robotics* **3**(19), eaat3630 (Jun 2018)
23. Hakonen, M., Piitulainen, H., Visala, A.: Current state of digital signal processing in myoelectric interfaces and related applications. *Biomed. Signal Process. Control* **18**, 334–359 (Apr 2015)
24. Hargrove, L., Englehart, K., Hudgins, B.: The effect of electrode displacements on pattern recognition based myoelectric control. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **1**, 2203–2206 (2006)
25. Hwang, H.J., Hahne, J.M., Müller, K.R.: Channel selection for simultaneous and proportional myoelectric prosthesis control of multiple degrees-of-freedom. *J. Neural Eng.* **11**(5), 056008 (Oct 2014)
26. Hwang, H.J., Hahne, J.M., Müller, K.R.: Real-time robustness evaluation of regression based myoelectric control against arm position change and donning/doffing. *PLoS One* **12**(11), e0186318 (Nov 2017)
27. Hyeon-Kyu Lee, Kim, J.H.: An HMM-based threshold model approach for gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(10), 961–973 (Oct 1999)
28. Jiang, N., Dosen, S., Muller, K.R., others: Myoelectric control of artificial limbs—is there a need to change focus?[in the spotlight]. *IEEE Signal Processing* (2012)
29. Jiang, N., Englehart, K.B., Parker, P.A.: Extracting simultaneous and proportional neural control information for multiple-DOF prostheses from the surface electromyographic signal. *IEEE Trans. Biomed. Eng.* **56**(4), 1070–1080 (Apr 2009)
30. Jianying Hu, Brown, M.K., Turin, W.: HMM based online handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(10), 1039–1045 (Oct 1996)
31. Kim, D., Kang, B.B., Kim, K.B., Choi, H., Ha, J., Cho, K.J., Jo, S.: Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view. *Science Robotics* **4**(26), eaav2949 (Jan 2019)
32. Kulic, D., Takano, W., Nakamura, Y.: Incremental on-line hierarchical clustering of whole body motion patterns. In: RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication. pp. 1016–1021 (Aug 2007)
33. Kulic, D., Ott, C., Lee, D., Ishikawa, J., Nakamura, Y.: Incremental learning of full body motion primitives and their sequencing through human motion observation. *Int. J. Rob. Res.* **31**(3), 330–345 (Mar 2012)

34. Kulić, D., Takano, W., Nakamura, Y.: Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *Int. J. Rob. Res.* **27**(7), 761–784 (Jul 2008)
35. Kullback, S., Leibler, R.A.: On information and sufficiency. *Ann. Math. Stat.* **22**(1), 79–86 (Mar 1951)
36. Kyranou, I., Vijayakumar, S., Erden, M.S.: Causes of performance degradation in non-invasive electromyographic pattern recognition in upper limb prostheses. *Front. Neurorobot.* **12**, 58 (Sep 2018)
37. Liu, J., Sheng, X., Zhang, D., He, J., Zhu, X.: Reduced daily recalibration of myoelectric prosthesis classifiers based on domain adaptation. *IEEE J Biomed Health Inform* **20**(1), 166–176 (Jan 2016)
38. Markovic, M., Dosen, S., Popovic, D., Graimann, B., Farina, D.: Sensor fusion and computer vision for context-aware control of a multi degree-of-freedom prosthesis. *J. Neural Eng.* **12**(6), 066022 (Dec 2015)
39. Muceli, S., Jiang, N., Farina, D.: Extracting signals robust to electrode number and shift for online simultaneous and proportional myoelectric control by factorization algorithms. *IEEE Trans. Neural Syst. Rehabil. Eng.* **22**(3), 623–633 (May 2014)
40. Olson, E.: AprilTag: A robust and flexible visual fiducial system. In: 2011 IEEE International Conference on Robotics and Automation. pp. 3400–3407 (May 2011)
41. Patel, G.K., Hahne, J.M., Castellini, C., Farina, D., Dosen, S.: Context-dependent adaptation improves robustness of myoelectric control for upper-limb prostheses. *J. Neural Eng.* **14**(5), 056016 (Oct 2017)
42. Patel, G.K., Nowak, M., Castellini, C.: Exploiting knowledge composition to improve Real-Life hand prosthetic control. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**(7), 967–975 (Jul 2017)
43. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python . *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
44. Peerdeman, B., Boere, D., Witteveen, H., in 't Veld, R.H., Hermens, H., Stramigioli, S., Rietman, H., Veltink, P., Misra, S.: Myoelectric forearm prostheses: state of the art from a user-centered perspective. *J. Rehabil. Res. Dev.* **48**(6), 719–737 (2011)
45. Plappert, M.: Classification of Human Whole-Body Motion using Hidden Markov Models. Ph.D. thesis, Institute for Anthropomatics and Robotics (IAR) (2015)
46. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE* **77**(2), 257–286 (Feb 1989)
47. Rabiner, L.R., Juang, B.H.: An introduction to hidden markov models. *ieee assp magazine* **3**(1), 4–16 (1986)
48. Radhakrishnan, S.M., Baker, S.N., Jackson, A.: Learning a novel myoelectric-controlled interface task. *J. Neurophysiol.* **100**(4), 2397–2408 (Oct 2008)
49. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: 2011 IEEE International Conference on Robotics and Automation. pp. 1–4 (May 2011)
50. Scheme, E., Englehart, K.: Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *J. Rehabil. Res. Dev.* **48**(6), 643–659 (2011)
51. Sharif, M., Erdoganmus, D., Padir, T.: Human-in-the-Loop prosthetic robot hand control using particle filters for grasp selection. In: 2019 Third IEEE International Conference on Robotic Computing (IRC). pp. 293–300 (Feb 2019)
52. Starner, T.: Visual recognition of American sign language using hidden Markov models. Ph.D. thesis, Massachusetts Institute of Technology (1995)

53. Takano, W., Yamane, K., Sugihara, T., Kou Yamamoto, Nakamura, Y.: Primitive communication based on motion recognition and generation with hierarchical mimesis model. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. pp. 3602–3609 (May 2006)
54. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (Aug 2005)
55. Williams, E.J.: Experimental designs balanced for the estimation of residual effects of treatments. Aust. J. Chem. (1949)
56. Xiloyannis, M., Gavriel, C., Thomik, A.A.C., Faisal, A.A.: Gaussian process autoregression for simultaneous proportional Multi-Modal prosthetic control with natural hand kinematics. IEEE Trans. Neural Syst. Rehabil. Eng. **25**(10), 1785–1801 (Oct 2017)
57. Young, A.J., Hargrove, L.J., Kuiken, T.A.: The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift. IEEE Trans. Biomed. Eng. **58**(9), 2537–2544 (Sep 2011)