Arm Motion Gesture Recognition using Dynamic Movement Primitives and Gaussian Mixture Models

Steven Jens Jorgensen*, and Luis Sentis†
Department of Mechanical Engineering
The University of Texas at Austin, Austin, Texas 78712
Email: *stevenjj@utexas.edu , †lsentis@austin.texas.edu

***Abstract*—In collaborative interaction scenarios between a human and a robot, The robot's ability to recognize the movement gestures of a human is crucial to understanding the underlying intent. Gestures are particularly useful if there is some mapping (constant, time-varying, or task-dependent) between the gesture and the desired intention. As an effort towards recognizing movement gestures better, this work focuses on modeling human static, discrete, and rhythmic gestures as the forcing function of a discrete Dynamic Movement Primitive (DMP). In particular, the gestures are the gaussian basis weights that approximate the forcing function. It was found that a supervised Gaussian Mixture Model (GMM) classifer can recognize static and discrete gestures with high accuracy. Additionally, accuracy classification is still possible even when two discrete gestures are linear, a condition often avoided by other movement primitive recognition studies. Results also show that the GMM can also classify rhythmic gestures with only the discrete DMP formulation, while still performing much better than randomized guessing. For all types of classification recognition, it was also found that the classifier's bias-variance trade-off performance is sensitive to the number of basis weights used. The sensitivity finding is important as other movement primitive gesture recognition studies ignore tuning the number of basis weights, which can significantly improve or reduce performance.**

I. Introduction

In certain Human-Robot-Interaction(HRI) scenarios, recognizing human gestures is essential for efficient and safe human robot collaboration. Note that recognizing gestures is a key step to understanding the intent of a collaborative human, especially if there is a mapping between the provided movement gesture and the intent. This mapping may not necessarily be a constant one-to-one mapping but can also vary with time and task dependency.

This work models static, discrete, and rhythmic types of arm gestures as the forcing function of a Dynamic Movement Primitive (DMP) representing the gesture, where the basis weights of the forcing function were used as the gesture's features.

Using Gaussian Mixture Models (GMMs) as the primary classification tool, different experiments were made to show the practicality of using DMPs for gesture recognition.

The following hypotheses are addressed in this work:

1) An unsupervised learning algorithm such as an Expectation-Maximization (E-M) algorithm on GMMs can be used to automatically segment different static and discrete DMP demonstrations.
2) A supervised Gaussian Mixture Model (GMM) classifier can be used to classify different discrete DMP-based gestures.
3) A GMM classifier can distinguish between spatially different discrete DMP-based gestures
4) The classifier will fail to distinguish between two linear motions.
5) The GMM classifier will fail on classifying rhythmic gestures.
6) Using the discrete DMP formulation to represent all the gestures, the GMM classifier can classify static, discrete gestures but will fail to classify rhythmic gestures.
7) For a given set of data, there is an optimal number of weights that best represents the gestures.

In general, these experimental hypotheses were motivated to identify the limits, practicality and intricacies of using DMPs with GMMs for movement recognition. While not exhaustive, exploring the short list presented gives sufficient insight as the results and discussions presented in the paper show.

To test the hypotheses, we perform eight types of arm motion gestures. We have one static gesture, five discrete gestures, two of which are linear, and two rhythmic gestures. Figure 1 gives a visualization of the gestures. The static gesture is simply constant in space. Two of the discrete gestures are letters U and S, and another two are linear motions with different starting and ending positions. The last discrete gesture is a triangle shape with very similar starting and ending goal positions to test the stability of similar start and end-goal states (see Section III-A). The rhythmic gestures are a continuous circle motion and continuous waving motion.

From these gestures, it was found that hypothesis 1 is possible, but unreliable, hypotheses 2, 3, 7 are true with high confidence, hypothesis 4 is false with high confidence, and hypotheses 5 and 6 are true with low confidence.

In essence, this paper presents the following new findings: (a) As far as the authors know, the community who use movement primitives for recognition do not discuss how their systems are tuned, but here a performance sensitivity analysis is discussed with respect to the number of basis weights used for recognition. (b) Previous recognition studies using DMPs do not try to recognize spatially linear/straight motions as the forcing function may appear similar, but the experiments presented here give evidence that it is possible to discriminate between two straight motions. (c) By accident, it was found that the two rhythmic gestures used in this study can be recognized using the discrete formulation of DMPs with unexpectedly high recognition rates. Finally, (d) DMPs can also represent static-type gestures by setting the goal position constant.

For this project, the Matlab code used to recognize gestures is available at http://github.com/stevenjj/Gesture_Recognition, and the forcing function DMP code is available at https://github.com/stevenjj/myROS/tree/64-bit/gestures

II. Related Work

Military gesture recognition was previously implemented using nearest neighbors and an SVM classifier [2]. In their work, they focused significantly on recognizing only static type and rhythmic type gestures, as they are targeting military applications. Additionally their implementation throws many data points away while also being sensitive to temporal and spatial type of gestures.

In another work, the authors use a Hidden Markov Model (HMM) to automatically segment sequences of natural activities to automatically segment gestures and cluster them. After the primitive gestures are extracted, the gestures are represented as symbols and, the gestures' lexicon is extracted using their proposed algorithm [12]. Compared to their work, this paper focuses on the gestures that are already automatically segmented and only classification of the gestures is needed.

In [1], different unsupervised algorithms were tried to automatically detect gestures and test the performance of various unsupervised clustering methods. However, the features used in their algorithm were not specified, and their features only looked at static and rhythmic motions.

As for human robot collaboration scenarios, [6] uses Probabilistic Movement Primitives (ProMPs) [9] to detect human intentions for assembly hand-over tasks and spatial mimicking of pointing tasks. Probabilistic movements use spatial information as part of learning the movement primitive, and therefore may not recognize similar looking gestures that are spatially different. Thus, ProMPs do not have the spatial invariant property of DMPs.

The closest work to this paper is the extensive work done in [5] that details the mechanics of using DMPs. In their work, they performed motion recognition of discrete movements. In particular, they focused on showing that different alphabetical letters will have a consistent similarity matrix, and so classification is possible. The difference between their work and this paper is that GMMs were used to classify static, discrete, and rhythmic gestures using only the discere formulation of DMPs. Additionally, this paper shows that highly linear discrete motions can also be distinguished provided that the DMP parameters are specified properly. This paper also shows that it is possible to recognize rhythmic motions despite being modeled with the discrete formulation of DMPs.

III. Background Information

A. Dynamic Movement Primitives for Gesture Recognition

The Dynamic Movement Primitive (DMP) framework [5] is a powerful tool that enables dynamic representation of discrete and rhythmic movements. Here, a biologically-inspired discrete formulation of DMPs given in [10] and [4] is used. As noted in [4], the primary difference is that the differential equations are based on a sequence of convergent acceleration fields instead of force. Practically, this is similar to the original formulation, but with additional benefits such as better stability when the goal and initial positions are similar, invariance under transformations, and better generalization to new movement targets. From this discussion, any one-dimensional movement can be represented as a converging spring-damper system perturbed by a nonlinear forcing function $f(s)$:

$$\tau \dot{v}(t) = K(g - x(t)) - Dv(t) - K(g - x_o)s + Kf(s), \quad (1)$$

$$\tau \dot{x}(t) = v(t), \quad (2)$$

$$\tau \dot{s}(t) = -\alpha s(t), \quad (3)$$

where $x(t)$ and $v(t)$ are the position and velocity of the movement; K and D are the spring and damper terms; g and x_o are the goal and start positions of the movement; τ is the temporal scaling factor; and s is the phase variable that exponentially decreases from 1 to 0 with α to control the convergence time.

While representing motion as a DMP has many favourable properties [5], this work takes advantage of its temporal and spatial invariant property. In particular similar-looking motions can be demonstrated at varying durations with varying start and end goal positions. For example motion demonstrations can be spatially scaled and performed slowly but still have the same underlying DMP dynamics.

TABLE I
DMP Learning Parameters

$\tau(s)$	α	K	D
τ_{demo}	ln(0.01)	$400N/cm$	$2\sqrt{K}$

In this work, the parameters of the DMP are summarized in Table I . The spring term is set to be high, whose importance is described in Section VI , and here it was set to $K = 400N/cm$. The damping term is critically damped with $D = 2\sqrt{K}$. The temporal scaling term is set to $\tau = \tau_{demo}$, where τ_{demo} is the length of the movement demonstration. Finally, $\alpha = ln(0.01)$ to ensure that at $t = \tau_{demo}$, $s(t)$ is 99% converged.

To obtain the forcing function that represents the gesture, a demonstration trajectory, $x_{demo}(t)$, is recorded and differentiated twice to get $v_{demo}(t)$ and $\dot{v}_{demo}(t)$, which is then substituted to the following equation:

$$f_{target}(s) = \frac{\tau \dot{v}(t) + Dv(t)}{K} - (g - x(t)) + (g - x_o)s, \quad (4)$$

with $s(t) = exp(\frac{\alpha}{\tau_{demo}} t)$ from solving Eq. 3 . Note that Eq. 4 is obtained by solving for $f(s)$ from Eq. 1. The target function can then be approximated by minimizing the squared error between Eq. 4 and the w_i weights of the following non-linear function:

$$f(s) = \frac{\sum_{i=1}^{n} w_i \psi_i(s) s}{\sum_{i=1}^{n} \psi_i(s) s} \tag{5}$$

where $\psi_i(s) = exp(-h_i(s - c_i)^2)$ is the i-th Gaussian basis function centered at c_i with width h_i. [3] empirically determined that the width and centers of the Gaussian basis functions can be set to $c_i = 1/n$ and $h_i = \frac{n}{c_i}$, where n is the number of basis weights used to approximate $f(s)$. Since all the parameters are fixed, local weighted regression of $f(s)$ will give consistent w_i weights.

B. Gaussian Mixture Models (GMMs)

Since DMPs are invariant to spatial and temporal motion demonstrations, it is reasonable to expect that the forcing function weights of the gestures will be clustered together in an n-dimensional plot [5]. This clustering may be oval in shape and so a multivariate gaussian is used to represent the cluster of the weights:

$$N(\boldsymbol{w}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{exp(-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\boldsymbol{w} - \boldsymbol{\mu}_k))}{(2\pi)^{n/2} |\boldsymbol{\Sigma}_k|^(1/2)} \tag{6}$$

where n is the dimension of the multivariate distribution, $\boldsymbol{w} \in \mathbb{R}^n$ is the input, $\boldsymbol{\mu}_k \in \mathbb{R}^n$ is the mean, and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{nxn}$ is the covariance. Now, a GMM [11] is defined to be

$$p(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{w}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{7}$$

where $p(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the probability of a particular feature, $\boldsymbol{w}$, given all the means, $\boldsymbol{\mu}$, and covariances $\boldsymbol{\Sigma}$ of the combined gaussians. The variable π_k is the mixture component representing the fraction of elements belonging to a mixture k such that

$$\sum_{k=1}^{K} \pi_k = 1. \tag{8}$$

As an intuition, if there are k clusters with equal number of elements in each cluster, the mixture component is $\pi_k = 1/k$, a uniform distribution.

IV. Methodology

A. Gesture Data Gathering

Eight gestures with 30 demonstrations each were recorded using the ROS package *ar track alvar* [7] to track a single AR marker with the Microsoft Kinect. There are five discrete gestures called "*U-shape, Letter-S, Triangle, LL-Swipe, and UL-Swipe*," one static gesture called "*Static*," and two rhytmic gestures called "*Wave and Circle*." The x-y plots of all the recorded gestures are shown in Figure 1.

Each gesture type served a purpose to maximize scientific findings. The *U-shape*, *Letter-S*, and *Triangle* discrete gestures have obvious descriptions. The *U-shape* and *Letter-S* gestures were provided as controls for hypothesis 2 since it has been previously show that they can be recognized with DMPs [5]. However, the *Triangle* gesture was selected since previous gesture recognition never dealt with motions that have almost identical start and goal positions. The *LL-Swipe* and *UL-Swipe* gestures are two discrete, linear-type gestures that starts from the lower-left corner and upper left corner respectively and ends in a corresponding opposite corner. The purpose of the discrete linear gestures is to test hypothesis 4 (that is, the linear DMP motions will be identical to each other and so any classifier will fail to distinguish the two gestures). Finally, two rhythmic gestures were added to test the hypothesis 5 (the discrete DMP formulation will fail recognizing rhythmic gesture). During the demonstration process, both rhythmic gestures *Wave* and *Circle* had no consistent starting and ending position. Sometimes it was difficult to manage the frequency of the rhythmic gesture, and these inconsistencies are kept as part of training data.

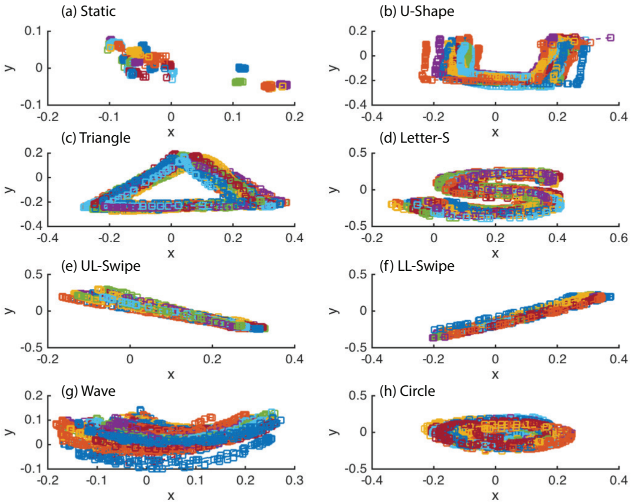

Fig. 1. The eight types of demonstrated gestures are shown. The sub-figures indicate (a) a static gesture, (b)-(f) five discrete gestures, and (g) and (h) are two rhythmic gestures. The gestures were made using a Kinect that recognized the x-y-z position of the AR marker held by the demonstrator. The static gesture, (a), is a demonstration where the marker never moves. (b) and (d) are discrete letter-type gestures which is used in existing DMP literature to show movement recognition [5]. (c) is a triangle shape gesture to test the ability of the DMP to recognize gestures with almost equal starting and ending positions. (e) and (f) are linear gestures with different starting and ending positions to test if DMPs can discriminate between two spatially different discrete motions. Finally, (g) and (h) represent a continuous circular and waving motion respectively. For each sub-figure, each colored trajectory represents the trajectory of a single demonstration.

Three additional types of discrete gestures were also gathered, but with only 5 demonstrations each. In particular, a spatially smaller versions of the discrete gestures *U-shape*, *Triangle*, and *Letter-S* were also provided as test data to test hypothesis 3.

B. Gesture Feature Representation

After all the demonstrations were made, using the DMP formulation with the constants listed in Table I, the data was pre-processed to calculate the 3-dimensional x,y,z forcing function of each gesture. We define n_b to be the number of basis weights on a particular dimension. Then, the n_b basis weights of each forcing function was extracted using local weighted regression, and the values of the weights were stored as vectors of w_x, w_y, w_z, where x, y, and z indicates the particular Cartesian axis the weight represents. Finally, each gesture is represented as

$$\boldsymbol{w}_g = [\boldsymbol{w}_x^T, \boldsymbol{w}_y^T, \boldsymbol{w}_z^T]^T, \tag{9}$$

where $\boldsymbol{w}_g$ is the concatenated vector of the forcing function's basis weights. Thus each gesture, $\boldsymbol{w}_g$, has $n = 3n_b$ dimension features.

In order to visualize the relationship of the basis weights between any two gestures, the similarity function

$$similarity = \frac{\boldsymbol{w}_{g_1}^T \boldsymbol{w}_{g_2}}{||\boldsymbol{w}_{g_1}|| \cdot ||\boldsymbol{w}_{g_2}||} \tag{10}$$

previously proposed by [5] is used. Note that Eq. 10 is 1 when two gestures are 100% similar and is 0 or below when there is minimum similarity. Figure 2 is a color map visualization of the similarity matrix between any two gestures, where each cell uses Eq. 10 with $n_b = 5$ basis weights per dimension.

C. GMM Supervised Classification

To perform supervised classification, a finite K number of gaussian mixtures are trained. Each mixture $k \in \{1, 2, ..., K\}$, representing one gesture, makes K total number of gestures to consider. For each $\boldsymbol{w}_g$ gesture, $D = 20$ random demonstrations were used as positive training examples. Then, for each k gaussian mixture, training is done by stacking the mean and covariance of the corresponding training examples:

$$\boldsymbol{\mu}_k = mean([\boldsymbol{w}_{g_1}, ..., \boldsymbol{w}_{g_2} ..., \boldsymbol{w}_{g_D}]^T) \tag{11}$$

$$\boldsymbol{\Sigma}_k = Cov([\boldsymbol{w}_{g_1}, ..., \boldsymbol{w}_{g_2} ..., \boldsymbol{w}_{g_D}]^T) \tag{12}$$

Now, given an unknown gesture, $\boldsymbol{w}_g$, the gesture's membership weight probability, r_k, is calculated for each k cluster using Baye's Rule. More specifically, r_k is the probability that the demonstration belongs to mixture k given a gesture demonstration, $\boldsymbol{w_g}$:

$$r_k = p(k|w_g) = \frac{\pi_k N(\boldsymbol{w_g}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum\limits_{\bar{k}=1}^{K} \pi_{\bar{k}} N(\boldsymbol{w_g}, \boldsymbol{\mu}_{\bar{k}}, \boldsymbol{\Sigma}_{\bar{k}})}, \tag{13}$$

where $p(k|\boldsymbol{w_g})$ represents the cluster membership probability given a $\boldsymbol{w_g}$ demonstration, and π_k is the k-th mixture weight representing that a randomly selected demonstration is part of the k-th mixture component. Note that $\sum_{k=1}^{K} \pi_k = 1$. Here, $\pi_k = \frac{D}{D \cdot K} = \frac{1}{K}$ since each mixture component was trained with D demonstrations and there are $D \cdot K$ total number of demonstrations. To identify the gesture, the cluster k that maximizes r_k is the gesture's classification. This is represented as:

$$p(k|\boldsymbol{w}_g) = \arg\max_k p(k|\boldsymbol{w}_g), \tag{14}$$

D. GMM Unsupervised Classification

For unsupervised classification, Gaussian Mixture Regression using an Expectation-Maximization [11] algorithm is performed on the static and discrete gestures data set and only the number of mixtures, K, is provided as input. If the mixture regression is 100% successful, it is expected that each mixture component π_k will reflect the the true mixture. Note that it is known there are $m_{per} = 30$ demonstrations for each gesture, and there are $m = m_{per} \cdot K$ total demonstrations. Thus, it is enough to see if each cluster has identified exactly 30 components. Suppose a cluster has specified $m_g(k)$ gestures to belong to cluster k. If $m_g(k) <= m_{per}$ then it is assumed that cluster k has found the correct m_g gestures. However, if $m_g(k) > m_{per}$ then cluster k has $m_{mistakes}(k) = m_g(k) - m_{per}$ mistakes since perfect clustering should contain m_{per} gestures for each cluster. Using this intuition, the following performance index is specified:

$$score = \frac{(m - m_{per} - \sum\limits_{k=1}^{K} m_{mistakes}(k))}{(m - m_{per})}, \tag{15}$$

Note that a perfect score of 1 means that each gaussian mixture has exactly 30 gestures and a 0 means that all gestures are classified as a single cluster. It is possible that a score of 1 can be obtained while the clustered gestures are a mix of other gestures. However, in general this is unlikely to happen as different gestures will have different target functions and therefore have different basis weights.

V. Experiment and Results

A. Unsupervised GMM Performance

TABLE II
UNSUPERVISED GMM

Weights per Dimension	Discrete Gestures	Weights per Dimension	Discrete Gestures
1	$(2.0 \pm 6.3)\%$	25	$(61.8 \pm 13.6)\%$
3	$(14.3 \pm 13.8)\%$	30	$(69.0 \pm 12.5)\%$
5	$(24.1 \pm 15.9)\%$	35	$(58.2 \pm 10.6)\%$
10	$(46.3 \pm 10.2)\%$	40	$(55.9 \pm 7.7)\%$
15	$(47.8 \pm 10.5)\%$	50	$(56.4 \pm 9.2)\%$

The first experiment was to see how well unsupervised classification works on the entire static and discrete gestures data set. The number of basis weights n_b per dimension was changed as experiments consistently show that performance is sensitive to the number of weights used to represent the gesture. Matlab has a built in gaussian mixture model fitting function, called *fitgmdist*, that utilizes the E-M algorithm. Using the criteria described in Eq. 15, the performance of the unsupervised clustering was recorded in Table II, where each cell in the table is the mean plus or minus the standard deviation of the score after 10 random trials.

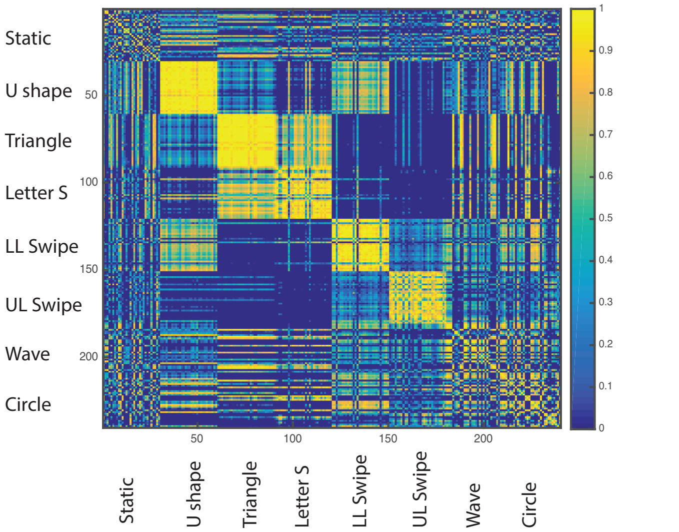

Fig. 2. The similarity matrix of all the gestures is visualized as a colormap. Each cell represents the similarity between any two gestures where colors closer to 1 indicates high similarity and those below 0 have minimum similarity

The results indicate that 30 basis weights per dimension has the best unsupervised GMM performance with $69\% \pm 12.5$ accuracy. However, as more basis weights are added to the dimension, the performance stagnates. Finally, the standard deviation for all the weights tested have high variance indicating unreliability due to its inconsistent performance. Thus, hypothesis 1 has potential but it is not reliable. The unsupervised GMM's performance sensitivity to n_b also confirms hypothesis 7.

B. *Supervised GMM Performance*

TABLE III
SUPERVISED GMM ON ALL DATA SETS

Weights per Dimension	Discrete	Spatial	Rhythmic	Discrete and Rhythmic
1	$(78.7 \pm 0.7)\%$	$(54.7 \pm 4.2)\%$	$(31.5 \pm 8.5)\%$	$(62.8 \pm 1.2)\%$
3	$(98.3 \pm 0.6)\%$	$(73.7 \pm 7.1)\%$	$(93.2 \pm 4.0)\%$	$(96.3 \pm 1.7)\%$
5	$(98.6 \pm 1.2)\%$	$(88.0 \pm 5.3)\%$	$(97.0 \pm 2.2)\%$	$(95.1 \pm 7.1)\%$
10	$(89.3 \pm 1.5)\%$	$(43.3 \pm 5.7)\%$	$(82.7 \pm 2.9)\%$	$(86.1 \pm 1.3)\%$
15	$(71.6 \pm 3.1)\%$	$(11.3 \pm 3.2)\%$	$(58.8 \pm 11)\%$	$(62.7 \pm 2.8)\%$
25	$(78.7 \pm 2.5)\%$	$(33.3 \pm 6.3)\%$	$(76.3 \pm 4.0)\%$	$(77.0 \pm 2.4)\%$

TABLE IV
SUPERVISED GMM ON CROSS VALIDATION DATA SET

Weights per Dimension	Discrete	Spatial	Rhythmic	Discrete and Rhythmic
1	$(77.2 \pm 3.7)\%$	$(51.3 \pm 3.2)\%$	$(36.0 \pm 12.4)\%$	$(60.9 \pm 2.5)\%$
3	$(97.3 \pm 1.4)\%$	$(66.7 \pm 7.1)\%$	$(81.1 \pm 9.1)\%$	$(88.6 \pm 3.6)\%$
5	$(96.2 \pm 2.5)\%$	$(65.3 \pm 8.2)\%$	$(88.5 \pm 8.8)\%$	$(92.5 \pm 2.6)\%$
10	$(86.5 \pm 1.1)\%$	$(40.7 \pm 4.9)\%$	$(68.0 \pm 8.5)\%$	$(73.9 \pm 8.6)\%$
15	$(57.2 \pm 4.3)\%$	$(10.0 \pm 3.5)\%$	$(61.2 \pm 7.5)\%$	$(45.5 \pm 6.9)\%$
25	$(50.17 \pm 9.2)\%$	$(26.0 \pm 4.9)\%$	$(77.1 \pm 4.0)\%$	$(36.5 \pm 5.7)\%$

The next experiment was to test hypotheses (2-6) and further confirm hypothesis 7. Tables III and IV summarizes the results. For all scenarios, each GMM was trained using 20 random gestures from a corresponding gesture type. Except for the "Spatial" columns, Table III tests the performance against the entire $K_{test} \cdot 30$ gesture data set where $K_{test} \in \{K_{discrete}, K_{rhythmic},, K_{spatialdiscrete}, K_{all}\}$ is the number of gestures being considered.

In this work, there are $K_{rhythmic} = 2$ rhythmic gestures types, $K_{discrete} = 5$ discrete and static gesture types, $K_{spatialdiscrete} = 3$ spatially different discrete gestures and $K_{all} = K_{rhythmic} + K_{discrete}$ discrete and rhythmic gesture types.

Recall that for each gesture, $D = 20$ training data were used to train each mixture model. To ensure that the performance is not skewed by the trained data, Table IV tests the performance only on the remaining unseen $K_{test} \cdot 10$ gesture data set.

In the '*Discrete*" column, the supervised GMM was trained and tested only on the $K_{discrete}$ static and discrete gestures. The '*Spatial*" column was also trained using the $K_{discrete} \cdot 30$ static and discrete gestures but was tested using the $K_{spatialdiscrete}$ spatially different discrete gesture set. The '*Rhythmic*" column was trained and compared only on the $K_{rhythmic}$ rhyhtmic gestures. Finally, the '*Discrete and Rhythmic*" column was trained and tested on the K_{all} static, discrete, and rhythmic gestures without the spatially different gestures. For all types of tests, the number of basis weights per dimension were also changed to test hypothesis 7.

Tables III and IV show that in general, there is high accuracy in the recognition performance of static and discrete gestures, which confirms hypothesis 2 and disproves hypothesis 4. In general, recognizing spatially similar static and discrete gestures performs very well, and the accuracy drops below 80% only when more basis weights per dimension are used due to over fitting.

The *Spatial* column confirms hypothesis 3. Concretely, spatially different discrete gestures can recognized with basis weights of 3 and 5 per dimension. As a reminder, the training set for the *Spatial* has never seen spatially smaller demonstrations, which makes this result more meaningful and significant.

What is surprising is that the *Rhythmic* column shows that even with using the discrete formulation of DMPs to represent rhythmic gestures, the supervised GMM can distinguish between the rhythmic "*Wave*" and "*Circle*" gestures. It was expected that the rhythmic gestures would appear as noise and the GMM will fail to recognize the rhythmic gestures completely. However, as the result shows, the accuracy is better than guessing between two rhythmic gestures at random.

To test if the GMM classifier can discriminate between static, discrete, and rhythmic gestures, the *Discrete and Rhythmic* columns shows that the presence of rhythmic gestures did not affect recognition performance as it reflects similar values to the *Discrete* column. From this study, it is surprising that hypotheses 5 and 6 are both false as rhythmic gestures were classified successfully.

For all of the gesture recognition tests, it is evident that the number of weights used to represent the gesture affected the performance of the classifier, which confirms hypothesis 7 convincingly. Using too many basis weights causes overfitting with high variance error, and not using enough basis

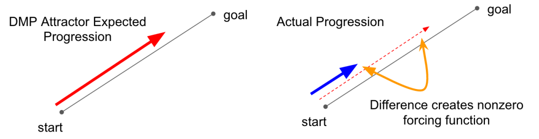
DMP Attractor Expected Progression
goal
start
Actual Progression
goal
start
Difference creates nonzero forcing function

Fig. 3. Linear Discrete Motion Gestures can be differentiated when K is high such that the DMP's attractor dynamics move faster than the actual demonstration making the forcing function non-zero.

weights (eg: when basis weights per dimension = 1) causes under-fitting with higher bias errors.

VI. Discussion

The results with regards to the ability of a supervised GMM to classify rhythmic gesture is strange and very unexpected. There are many possible explanations and some of are discussed here. It's possible that since there are only 2 rhythmic gestures, classifying between the two is easy as the GMM always return the best guess. The weights of each rhythmic gesture could also be sufficiently different in terms of forcing function noise, so fitting a GMM on two noise distributions was sufficient to discriminate between the two rhythmic gestures.

The hope was to show that rhythmic gestures will completely fail and using the rhythmic formulation of DMPs will be necessary. However, to even use the rhythmic DMP formulation for proper comparison, more rhythmic gesture types need to be recorded. Still, with the gestures used in this study, the static, discrete, and rhythmic gestures were classified successfully. Thus, until further study is conducted, hypotheses 5 and 6 are false but with low confidence.

The second surprising finding is that while the static and discrete gestures were classified successfully, confirming hypotheses 2 and 3, it did so while also classifying two different types of discrete linear gestures. The traditional thinking is that discrete linear gestures will have a 0 forcing function. This is why in [5] the motion gestures performed were all letters as trying to different linear motions could be problematic. However, here the results show that recognizing between two linear discrete gestures is possible. An intuitive explanation is provided in Figure 3. That is, if K of the DMP is set to be very high such that the attractor dynamics moves faster than the demonstration, the forcing function is non-zero and any type of linear motion in x-y-z can be classified.

In fact, this finding is predicted much earlier by looking at the similarity matrix between the two linear gestures in Figure 2. It is evident that they have no similarity at all.

This finding has an additional consequence. That is, it is also possible to detect richer types of static gestures. For example, suppose that recognizing between two types of static arm gestures is necessary. The coordinates can be set to the angle formed by the upper arm to the shoulder and the angle formed by the elbow to the upper arm as shown in [2]. Then, for all static gestures, the goal position can be set away from the user as indicated in Figure 4. However, the additional complication is that the goal position is now different. Thus, to make this work with the framework, a higher level classifier is needed to distinguish between static and discrete gestures.

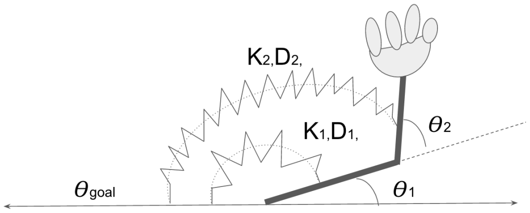

Fig. 4. Recognizing static gestures is possible by setting the goal position away from the user and using features such as arm angle relative to the body of the user.

VII. Conclusion

In this work the recognition of static, discrete, and rhythmic gestures were performed by using the discrete formulation of DMPs. In particular, the forcing function of the DMP was used to represent the gesture in which the weights obtained from local-weighted regression of equally-spaced gaussian functions were the features.

Using only GMMs for classification, it was found that unsupervised clustering can potentially be used to automatically learn different gesture types. However the high variability of the unsupervised GMM in the results shows that it will be unreliable.

On the other hand, using supervised GMM clustering provided an easy way to train a classifier while performing reliable recognition at a high accuracy especially when the number of basis weights are tuned. In particular, the classifier was able to distinguish between discrete and static gestures. Additionally, the classifier was also able to recognize different types of discrete linear motion under the DMP framework. This is an unexpected result as the DMPs of the two linear motions were expected to be different.

Finally, another unexpected result shows that the GMM can also classify rhythmic gestures even though the gestures were represented as discrete motions. However, there are not enough rhythmic gestures in this data set to truly claim that the discrete DMP formulation can classify all types of rhythmic gestures.

Overall, this work demonstrates that using the new discrete formulation of DMPs is an effective method for recognizing spatially and temporally invariant movement gestures. Once the gestures are recognized, a mapping between the gesture to intention may be formulated.

VIII. Future Work

In this work, only one static gesture was tested. Still, experiments with the discrete linear gestures resulted into a finding that DMPs can also represent richer static gesture types, but experimental validation remains. As a potential approach, identifying static gestures can be recognized with the current framework. Since it is static, the forcing function will be close to 0 as the goal and start positions are very close. Then after recognizing that the gesture is a static type, another GMM that classifies different type of static gestures can be used with the goal position explicitly specified.

Another future work is on the topic of rhythmic gestures. It is still not convincing that the discrete formulation of DMPs is enough to classify rhythmic gestures. In the future, two better ways of recognizing rhythmic gestures exist. The first is to use the rhythmic formulation for DMPs and use the learned basis weights for classification. Second, performing alignment on the data and approximating one period of the demonstration using a fourier transform can give consistent basis function weights.

Another problem with the current classification scheme is that it cannot handle incorrect gestures as the current framework only assumes that all gesture demonstrations is represented by the GMM. Thus the classifier always returns the best maximum guess for any given gesture. This can be fixed by doing some threshold study after the best cluster membership is selected.

Finally, while using DMPs is invariant to different temporal demonstrations of similar gestures, the classifier will not be able to identify when the desired gesture has begun or ended. Thus, this will fail when a time series of data is given without some heuristics given to the system. An example heuristic for example could be detecting minimum velocity onset for both start and ending conditions [5]. However, this has the disadvantage that no gesture is ever given when the velocity is less than the specified threshold and gestures are assumed to be always given when the velocity is greater than the threshold. Perhaps a better approach to handle continuous time series is to use a change-point-detection algorithm [8].

Acknowledgment

This work was supported by a NASA Space Technology Research Fellowship (NSTRF) with grant number NNX15AQ42H.

References

[1] A. Ball, D. Rye, F. Ramos, and M. Velonaki. A comparison of unsupervised learning algorithms for gesture clustering. *Proceedings of the 6th international conference on Human-robot interaction - HRI '11*, page 111, 2011.

[2] G. Bernstein, N. Lotocky, and D. Gallagher. Robot Recognition of Military Gestures CS 4758 Term Project. 2012.

[3] T. Dewolf. Dynamic movement primitives: The basis part 1. https://studywolf.wordpress.com/2013/11/16/dynamic-movement-primitives-part-1-the-basics/. Last Accessed: 2015-12-04.

[4] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. *2009 IEEE International Conference on Robotics and Automation*, pages 2587–2592, 2009.

[5] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–73, 2013.

[6] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann. Learning Interaction for Collaborative Tasks with Probabilistic Movement Primitives. *International Conference on Humanoid Robots*, pages 527–534, 2014.

[7] S. Niekum. Ar tracker alvar ros package. http://wiki.ros.org/ar_track_alvar. Last Accessed: 2015-12-04.

[8] S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto. Online Bayesian Changepoint Detection for Articulated Motion Models. *2015 IEEE International Conference on Robotics and Automation*, 2015.

[9] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic Movement Primitives. *Neural Information Processing Systems*, pages 1–9, 2013.

[10] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. *2009 IEEE International Conference on Robotics and Automation*, pages 763–768, 2009.

[11] P. Smyth. The EM Algorithm for Gaussian Mixtures The EM Algorithm for Gaussian Mixture Models. http://www.ics.uci.edu/~smyth/courses/cs274/notes/EMnotes.pdf. Last Accessed: 2015-12-04.

[12] T. Wang, H. Shum, Y. Xu, and N. Zheng. Unsupervised analysis of human gestures. *Advances in Multimedia Information Processing. Lecture Notes in Computer Science*, 2195:174–181, 2001.