International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Learning from Multiple Demonstrations with Different Modes of Operations

Utku Bozdoğan*[1], Emre Uğur[2]

Submitted: 05/09/2019 Accepted : 18/03/2020

Abstract: In this paper, teaching multiple types of complex trajectories at once to a robot in a robust, easy to train model using Learning from Demonstration is studied where the robot is expected to gain the capacity to differentiate between different types of demonstrated trajectories and be able to reproduce these trajectories correctly. Demonstrated trajectories are used to train a Hidden Markov Model (HMM) and a modified version of Gaussian Mixture Regression (GMR) -which utilizes state transition probabilities between states of the HMM, the most probable state the end effector of the robot belongs to in the current reproduction of the trajectory, and previous points in the current reproduction of the trajectory- is used to estimate the trajectory iteratively. A Proportional Derivative (PD) controller is employed for the reproduction. Starting points that are intended to correspond to different types of trajectories which the robot is expected to differentiate between are tested on numerical and simulation experiments. Multiple numerical experiments and simulation experiments showed that our modified algorithm produced comparable results to previous work, and in certain complex trajectories our algorithm was successful where previous work has failed to produce expected results.

Keywords: *Gaussian Mixture Regression, Hidden Markov Models, Learning from Demonstration, trajectory reproduction*

1. Introduction

Robots that have the capacity of generalization will perform skilfully both in situations which they possess experience, and by adjusting their behaviours accordingly in situations which they have no previous experience. Today, robots without the capacity of generalization are able to perform skilfully in structured environments in which the tasks they perform are based solely on experience. The attention is therefore shifted to building models which give the robots the ability to generalize to perform in more dynamic environments, where changes occur, and the desired reproduction of the task has potential variations in each reproduction. Recent advances in materials science, computer vision, machine learning and control fields, combined with increasing computing power of CPUs and GPUs, and their common availability enable researchers to utilize powerful models and create robots which can perform successfully in uncertain and dynamic environments. Robots that have some capacity to generalize will find applications that are not limited to structured environments like factories and assembly lines but also semi-structured ones that give the robot some required freedom to handle variations in the task while still limiting the robot's operation to a certain degree, where the desired robot functionality can be achieved through limited autonomy. Robots are able to perform successfully in environments such as warehouses, shops behind a counter where interaction is limited and where manual labor is essential and the task and objects are well-defined, such as pick and place, push and pull etc. with limited variations in the task definition. This will soon be followed by robots aiding humans in their daily lives in their homes or in their work as well, in collaboration and without any need for expertise in robotics [1]. In this paper, we build such a model, able to learn and reproduce different and complex trajectory following tasks with some generalization capability.

In the context of this paper, a complex trajectory means that in a single trajectory following task, among multiple types of demonstrated trajectories, there are position or velocity values which are the same, and at these points different types of demonstrations cross each other. These points are named as crossing points.

Learning from Demonstration (LfD) [1] has been suggested as an efficient and intuitive way to teach new skills to the robots, where the robot observes, learns and imitates the actions demonstrated by the human tutors. LfD is a more time efficient technique than exploration-based ones as the search space is vastly reduced since learning process happens around desired outcomes, provided by an expert demonstrator which bears similarity to how humans learn many skills [2]. This shortens the amount of time spent learning compared to trial-and-error methods. Time and engineering effort are no longer needed to be spent creating a complex model with reward functions, tuning parameters and other requirements that may follow.

Multiple types of trajectories that belong to the same task was taught to the robot, in order to give it the ability to choose the most appropriate way to perform a given variation of the task, and the crossing points provide a challenge to the technique utilized, by resulting in undesired trajectory estimations which will be explained in more detail in Section 4.

In this work, using Hidden Markov Models (HMM) with a modified version of Gaussian Mixture Regression (GMR) was proposed. A model capable of reproducing a trajectory despite the differences coming from different demonstrations was intended. The robot was taught multiple types of trajectories together, with a small number of demonstrations for each. With no common

[1] *Boğaziçi University, Department of Computer Engineering, İstanbul, Turkey; ORCID: 0000-0002-8912-9446*
[2] *Boğaziçi University, Department of Computer Engineering, İstanbul, Turkey; ORCID: 0000-0001-9597-2731*
* *Corresponding Author Email: utku.bozdogan@boun.edu.tr*

parts in demonstrations of different types except for the crossing points, an HMM is able to correctly encode each demonstrated type using state transitions. State transition probabilities and the probabilistic information from the already produced portion of the trajectory is utilized in the GMR estimation. The robot is then able to decide on the type of trajectory it will reproduce based on the point it starts its execution and can reproduce the trajectory successfully.

2. Related Work

In this section, we provide a brief overview of the techniques applied previously with success. It was seen in the literature that Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs) [3] and Conditional Random Fields [4] can correctly model the probability distributions of the data from demonstrations. In [5], Task Parametrized GMMs are used, where after demonstrations have been processed, changing the parameter changes the reproduction of the task to accommodate the variability of the task. [16] tied the covariance matrices of the members of the mixture, which enabled reusing the parts of the task with similar pattern. They also used this method with task-parametrized GMMs and task-parametrized Hidden semi-Markov models. Their approach was successful in tasks such as pick and place and object avoidance. [22] shows task-parametrized GMMs in a wide range of tasks including handling high dimensional data by clustering, performing control etc. and extends parametrization to different models, which take GMMs as a starting point. GMM is a powerful and an easily applied technique, however not for modelling tasks with multiple types of trajectories that have crossing points due to the potential of generating a state around these points. In such tasks GMM risks failure by starting the reproduction as one type of trajectory but after passing through the area with crossing points continuing reproduction as a different type of trajectory. This happens due to GMMs being only able to encode spatial information in the model. Our method can deal with this situation by exploiting the state transition probabilities encoded in the HMM, and only using the influence from the state with the maximum probability. An example of GMM failing as described will be presented in Section 4. [7] used a Task Parametrized GMM to compare with the reproduction from a Parametric HMM. When the direction parameter is not provided to the GMM, the reproduction is shown to fail since the algorithm is influenced by states belonging to multiple types of trajectories and the result is unlike any demonstrated trajectory.

HMMs use state transition probabilities between states, ensuring correct transitions. This has parallels with how a trajectory is followed by humans such as starting from a point and moving through the areas belonging to the trajectory in a specific order, occurring in a single motion. HMMs can encode both spatial information, and through state transition probabilities multiple trajectories in the model, and therefore are able to correctly model the mentioned complex trajectories, which is why HMMs are preferred in this work.

HMMs have been extensively used in the literature because of their ability to handle spatial and temporal variations in the data. Use of HMMs are combined with various techniques for trajectory reproduction.

It has been found in the literature that multiple previous work utilized key points. An early approach using key points takes a single demonstration, which is segmented from its inflection points in a preprocessing phase, then HMM training occurs on the segments and the hidden states are determined. Spline interpolation is used to reproduce the trajectories and the model can differentiate different types of trajectories, tested on drawing letters [10]. Another approach with key points utilizes the Dynamic Time Warping algorithm to align the key points from all demonstrations. In the preprocessing phase, Linde-Buzo-Gray algorithm is used for the segmentation of demonstration data and key points are chosen based on position and velocity. Then key points are temporally aligned and HMMs are used for analysis of the variability of the motion and weighting of the key points. Lastly, spline interpolation is used to reproduce the trajectory [11].

Another work uses Principal Component Analysis or Independent Component Analysis to remove the redundancies in the data such as variances and correlations caused by demonstrators, thus being able to work with the meaningful part of the data. Using dimensionality reduction techniques also helps with the training of HMM and the result is a model able to recognize and reproduce trajectories drawing letters, with the extra benefit of being robust to noise [12]. The same authors had another valuable work in which they used PCA to project the data and then apply a mixture of Gaussians and Bernoulli distributions to encode the information which they then used GMR to reproduce generalized trajectories [17].

Calinon et al. [6] used HMMs alongside Gaussian Mixture Regression (GMR) to build robust models to reproduce the dynamics of the observed movements. Aside from using state transition probabilities of the HMM, this work also benefits from using a recursive approach for points that are traversed during the reproduction of the trajectory, further taking advantage of the available information. Previous work in [7] employed a Parametric HMM with Dynamic Movement Primitives (DMP), able to learn complex action trajectories parametrically and used a mechanism to extract from the HMM states separate chains representing a type of trajectory to reproduce it. DMP framework enabled creating nonlinear trajectories that are robust to perturbations and variations in the motion [9]. The probabilistic information encoded in the HMM is utilized to identify the presence of high variance portions of the trajectory and enables a human to scaffold the movement during reproduction. Previous work in [8] used PHMMs with DMPs as well, integrating a coupling term in the DMP equation to increase robustness to perturbation using Associative Skill Memories. [20] utilized Locally Weighted Regression (LWR) [18] and Gaussian Process Regression (GPR) [19] to compute generalized DMP trajectories which are able to perform according to different goals and constraints. In their experiment, the robot was able to perform ball throwing and drumming tasks, which are similar to trajectory following. They also argue that using a latent space representation generated by a dimensionality reduction technique may result in loss of important details within the task. Another work succeeding in nonlinear settings but without DMP is LWPR [21] which uses many locally linear models to approximate nonlinear functions such as trajectories for a task and learns a dimensionality reducing transform.

Our work uses HMMs alongside GMR, much like [6] but instead of summing up the influences from all states, we use only the one with the maximum probability. This enables our algorithm to decide on a type of trajectory to reproduce immediately without fail and be robust to multiple state transitions possibly occurring in the HMM.

3. Method

HMMs can give information, based only on the sequence of observations, about the state sequence generating the observations. This makes HMM a very important model that finds use in prediction and recognition models. Here, a brief explanation will be given and for the interested reader, Rabiner provided a thorough tutorial on HMMs and their applications in speech processing [13].

3.1. Background

3.1.1. Hidden Markov Models

A Hidden Markov Model is a model of a system which is assumed to be a Markov process with hidden states. The state sequence and the distributions producing the observations are not visible, however the observations coming from these hidden states are visible. A local maximum solution for the parameters belonging to the states can be found by applying the Baum-Welch algorithm based on the provided observations [13].

An HMM is denoted as $\lambda = \{\pi_i, A, B\}$ where $A = \{a_{ij}\}$ is the state transition matrix, a_{ij} is the state transition probability from state i to j and $B = \{b_i(k)\}$ is the observation probability matrix, $b_i(k)$ is the observation probability of an observation k under state i. The states are denoted with $i, j = \{1, 2, ...N\}$ where N is the number of states in the model.

3.1.2. Gaussian Mixture Regression

Distributions which the observed data come from are assumed to be Gaussian, after applying the Expectation-Maximization algorithm, a local maximum solution for the parameters of Gaussians are found. This is called a GMM. GMR is a supervised learning technique which can be used to estimate unknown portions of incomplete data. [14] The estimation is performed as follows:

$$\hat{x}^{\mathrm{o}}\left(x_i^{\mathrm{i}}\right) = \frac{\sum_{j=1}^{M} h_{ij}\left[\mu_j^{\mathrm{o}} + \Sigma_j^{\mathrm{oi}}\left(\Sigma_j^{\mathrm{ii}}\right)^{-1}\left(x_i^{\mathrm{i}} - \mu_j^{\mathrm{i}}\right)\right]}{\sum_{j=1}^{M} h_{ij}} \tag{1}$$

where o denotes unknown features and i denotes known features of data x. $j = \{1, 2, ..., M\}$ denotes a Gaussian distribution where M is the number of Gaussians. μ_j^{i} denotes the mean of jth Gaussian over known features, $\sum_{\mathrm{j}}$ is the covariance matrix of jth Gaussian and h_{ij} denotes the fractional probability of data x_i to belong to the jth Gaussian and is calculated as follows:

$$h_i = \frac{\pi_i N(x; \mu_i^x, \Sigma_i^x)}{\sum_{k=1}^{K} \pi_k N(x; \mu_k^x, \Sigma_k^x)} \tag{2}$$

where $N(x; \mu_i^x, \Sigma_i^x)$ denotes the probability of data x belonging to ith Gaussian. π_i is the fraction of the ith Gaussian.

3.2. Proposed Model: Hidden Markov Mixture Regression (HMMR)

An HMM trained with position and velocity data assuming Gaussian emissions (observations are assumed to be coming from Gaussian distributions) is used for encoding the observed trajectories. In order to reproduce learned trajectories, on the other hand, GMR is used for velocity and position estimations and iteration is performed with a PD-controller. Here, we propose to modify GMR estimation at the level of calculating fractional probabilities, as follows:

$$h_i(x_t) = \frac{\left(\max_{j \in \{1,K)} h_j(x_{t-1})\right) a_{ji}\, N(x_t; \mu_i^x, \Sigma_i^x)}{\sum_{k=1}^{K}\left[\left(\max_{j \in \{1,K)} h_j(x_{t-1})\right) a_{jk} N(x_t; \mu_k^x, \Sigma_k^x)\right]} \tag{3}$$

where a_{ji} denotes the state transition probability from state j to state i, K is the number of HMM states and x_t is the position data at time t.

Our proposed trajectory generation approach uses only the most probable state as a guide for estimation, unlike using the sum of influences from all states as in [6]. With this, we aim to let the robot decide when the algorithm is initialized, and the starting position is almost equally likely to be one of multiple types of training, and at the start there is no past sequence to look back and decide for the algorithm. We expect that our approach can accurately reproduce the trajectory when such a starting point was provided.

The algorithm works iteratively, and at the beginning a starting position needs to be provided. At each step, the next velocity is estimated through GMR from the current position. Then the next position is estimated through GMR from the current position and the estimated velocity. Then the velocity estimation and position estimation are passed to the PD-controller which iterates the motion one step and the new position is returned. h_i is calculated recursively, and $h_i(x_1)$ is estimated with Equation 2.

GMR estimations use the h_i from Equation 3 and are as follows:

$$\hat{\dot{x}} = \sum_{i=1}^{K} h_i(x)\left[\mu_i^{\dot{x}} + \Sigma_i^{\dot{x}x}(\Sigma_i^{x})^{-1}(x - \mu_i^{x})\right] \tag{4}$$

$$\dot{x} = \sum_{i=1}^{K} h_i(x)\left[\mu_i^{x} + \Sigma_i^{x\dot{x}}\left(\Sigma_i^{\dot{x}}\right)^{-1}\left(\dot{x} - \mu_i^{\dot{x}}\right)\right] \tag{5}$$

where x denotes position and $\dot{x}$ denotes velocity.

Input and output components of the hidden states which are multivariate Gaussian distributions are expressed as:

$$\mu_i = \begin{bmatrix} \mu_i^{x} \\ \mu_i^{\dot{x}} \end{bmatrix} \quad \text{and} \quad \Sigma_i = \begin{bmatrix} \Sigma_i^{x} & \Sigma_i^{x\dot{x}} \\ \Sigma_i^{\dot{x}x} & \Sigma_i^{\dot{x}} \end{bmatrix} \tag{6}$$

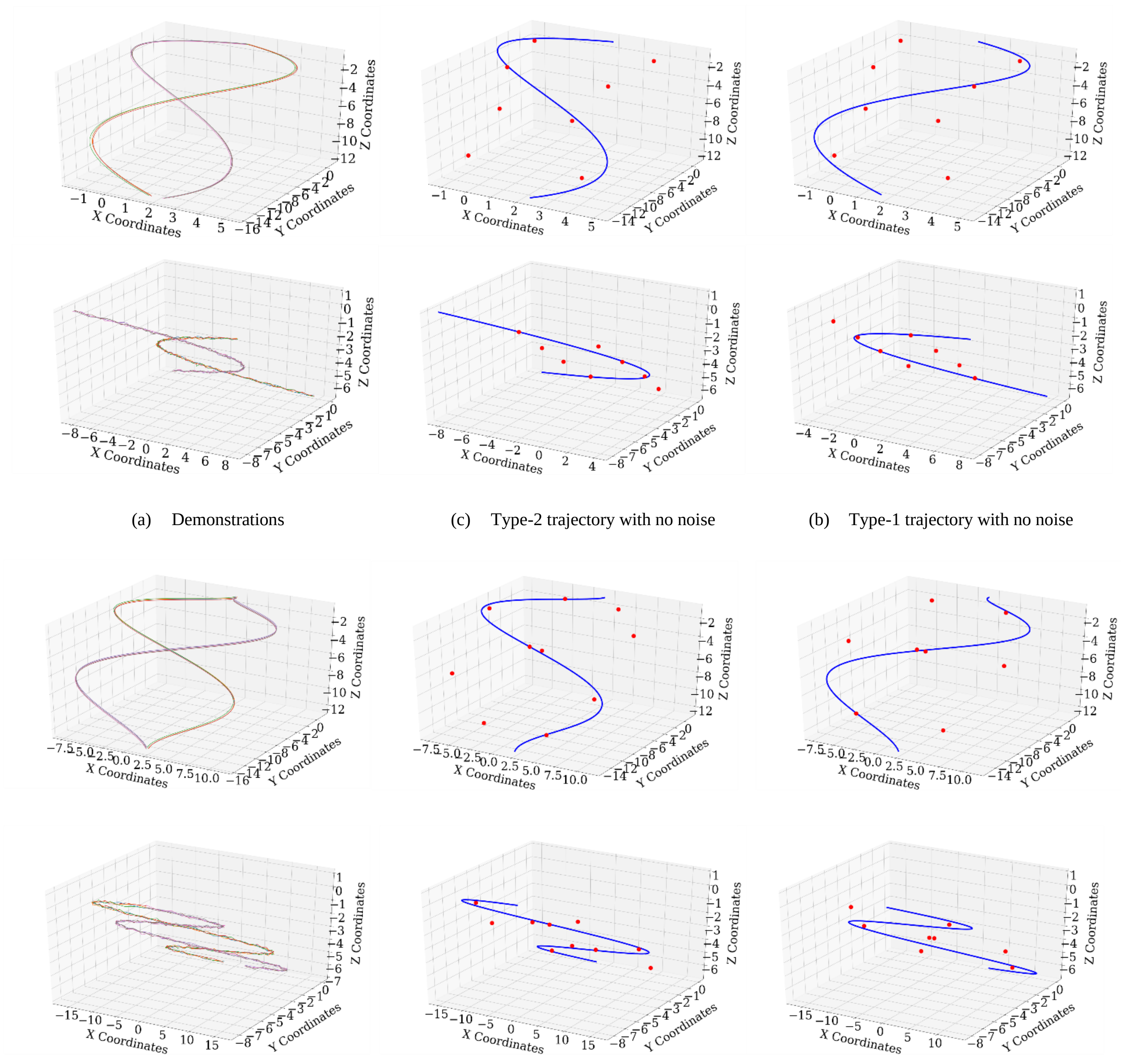
The PD-controller determines the acceleration as follows:

$$\ddot{x} = \left(\hat{\dot{x}} - \dot{x}\right)K^V + (\hat{x} - x)K^P \tag{7}$$

K^V and K^P are gain parameters and while $K^V = \frac{1}{\tau}$ where τ is the length of a timestep in the motion; K^P is adaptive and can be found from [15].

4. Experimental Results

Numerical and simulation experiments were performed on complex trajectories and the results are presented with a comparison between our algorithm and [6] in this section.



(a) Demonstrations

(c) Type-2 trajectory with no noise

(b) Type-1 trajectory with no noise

(d) Demonstrations

(e) Type-1 trajectory with no noise

(f) Type-2 trajectory with no noise

Fig 1. Experimental setup for S-shaped trajectory reproduction task. Upper plots denote the position profiles and lower plots denote the velocity profiles of the trajectory. (a) and (d) show the demonstrations together for two similar trajectories with some difference in velocity profiles. Red dots in (b), (c), (e), and (f) are the locations of HMM state centers and the blue lines show the expected trajectories without noise, starting from an unbiased starting point.

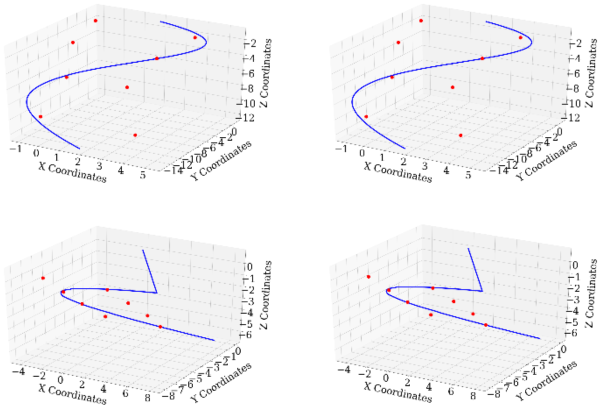
The experiments are aimed to test the ability of the algorithm to choose the correct type of trajectory to reproduce, and how accurate the reproduction is. Choosing the correct type of trajectory is related to the starting points we provide to the algorithm. We define two types of starting points: unbiased and biased starting points. Unbiased starting points correspond to the points that are equally likely to belong to multiple types of trajectories that are demonstrated, and biased starting points are starting points that are more likely to belong to one type of trajectory than the others. If multiple states exist around an unbiased starting point, each belonging to a different trajectory and with equal distance and therefore equal probability, summing up the influences from these states would guide the motion towards a trajectory unlike any demonstrated type.

In all experiments, Gaussian noise was added to the demonstrations in order to generate data similar to a human demonstrator.

4.1. S-shaped Trajectory Reproduction Task

In the first experiment, two types of S-shaped trajectories in three dimensional space were considered. 4 demonstrations were given for each type of trajectory, and Gaussian noise with 0 mean and 0.05 variance was added to the demonstration data. Fig. 1 shows the demonstrations and the types of trajectories. Providing biased starting points did not create notable differences between the reproductions as can be seen in Fig. 2.

A reproduction using GMM alongside GMR is presented in Fig. 3. Without utilizing the state transition probabilities provided by demonstrated trajectories when around a crossing point, the motion can be influenced by states belonging to different types of trajectories. This may result in a reproduction dissimilar to any demonstrated type or due to this undesired influence, moving



(a) Reproduction of Calinon et al. [6]

(b) Reproduction of HMMR

Fig 2. Experiment results from an S-shaped trajectory reproduction task using a biased starting point. Upper plots denote the position profiles and lower plots denote the velocity profiles of the trajectory. Red dots are the locations of HMM state centers and the blue lines show the reproduced trajectories, starting from a biased starting point.

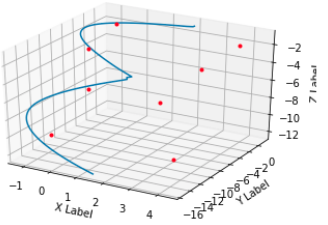
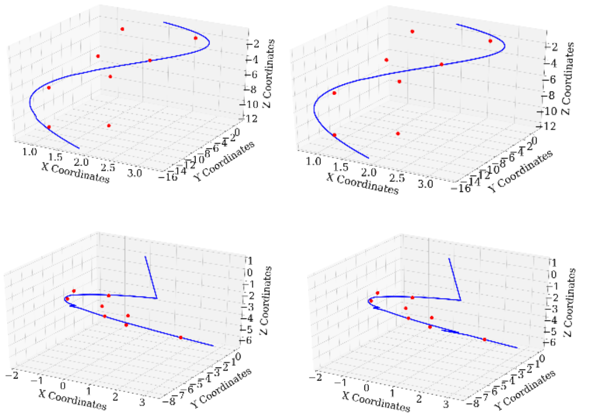


Fig 3. Position profile of an S-shaped trajectory reproduction task using GMM instead of HMM. Red dots are the locations of HMM state centers and the blue line shows the reproduced trajectory, starting from an unbiased starting point.



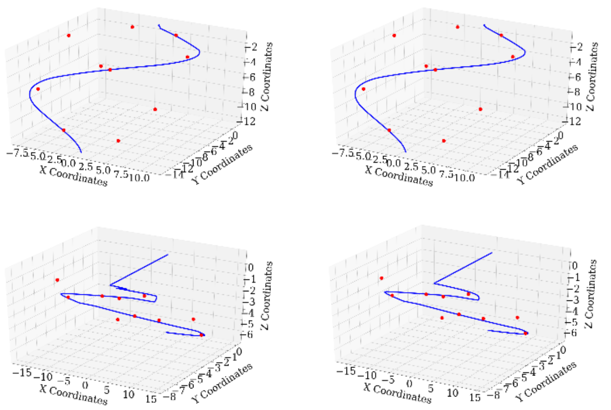
(a) Reproduction of Calinon et al. [6]

(b) Reproduction of HMMR

Fig 4. Experiment results from an S-shaped trajectory reproduction task using an unbiased starting point. Upper plots denote the position profiles and lower plots denote the velocity profiles of the trajectory. Red dots are the locations of HMM state centers and the blue lines show the reproduced trajectories. (a) is shown to be unable to decide on a type of trajectory briefly after the start for a few timesteps.

well within the area of a different state, thereby changing the type of trajectory produced after passing through the crossing point. The expected reproduction was a type-1 trajectory, similar to Fig. 1 (b).

Fig. 4 shows the results obtained from an unbiased starting point.



(a) Reproduction of Calinon et al. [6]

(b) Reproduction of HMMR

Fig 5. Experiment results from an S-shaped trajectory reproduction task where (b) shows the overcommitting towards the end of the trajectory. Red dots are the locations of HMM state centers and the blue dots show the reproduced trajectories, starting from an unbiased starting point.

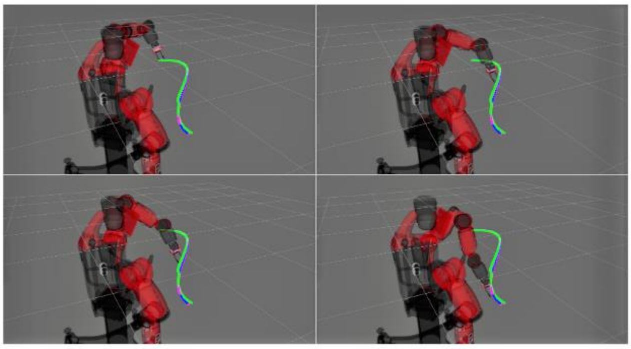


Fig 6. Snapshots from our simulation experiment from an S-shaped trajectory reproduction task using an unbiased starting point. Blue line denotes the expected reproduction, magenta line is reproduction of HMMR, and the green line is the reproduction of Calinon et al. [6].

The difference can be spotted easier from the velocity profiles of the trajectories. Our HMMR algorithm is immediately able to move towards the state with the most probability, which is the state closest in distance to the starting point. This creates a type-2 trajectory similar to Fig. 1 (f); whereas Calinon et al. [6] sums up the effects from probable states for both types of trajectories, leading to a response unlike any type of demonstrated trajectory. This lasts for a few timesteps and their algorithm also enabled the reproduced trajectory to converge towards a type-2 trajectory. The overall shape of the trajectory is preserved in this case, but due to time lost before reaching a decision there remains a region at the end of the trajectory that could not be reproduced.

Multiple experiments were performed for this type of trajectory, and a behavior of HMMR was observed which we call overcommitting. Due to using only the most probable state for estimation, the algorithm changes states only after the most probable state changes. This results in a sudden change in velocity, which results in a trajectory that is not seamless. Fig. 5. shows this phenomenon.

Table 1. Evaluation of simulation experiment results from an S-shaped trajectory reproduction task with RMS metric.

Technique used	RMS Error (m)
[6]	0.01371
HMMR	0.01368

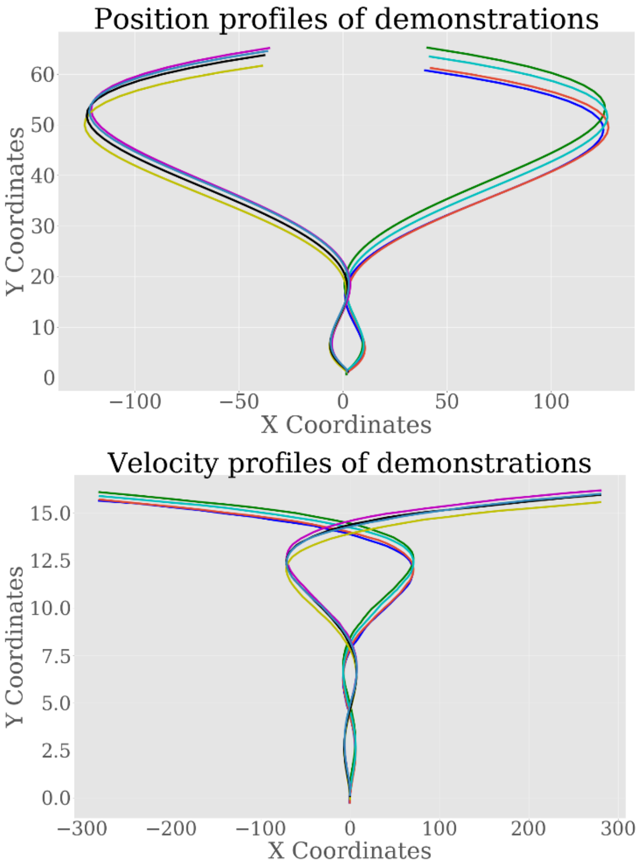


Fig. 7. Experimental setup for m-shaped trajectory reproduction task. Upper plot denotes the position profiles and lower plot denotes the velocity profiles of the demonstrated trajectories. Same color is used for the position and velocity of a demonstration, and there are 2 demonstrated types: left and right.

A simulation experiment using Baxter robot with 7 degrees of freedom arm was performed for the S-shaped trajectory. Inverse kinematics solver of Baxter is used to set angles of the joints in order to execute the generated trajectory with the end-effector. The obtained results were compatible with the previous results, as can be seen in Fig. 6. The results of the simulation experiments have been evaluated using Root Mean Square (RMS) error on position with respect to time. The dimensions of the workspace for the simulation experiment are 0.25 m × 0.60 m × 0.60 m. Reproductions of Calinon et al. [6] and HMMR are compared with the trajectory in Fig. 1 (c). Results presented in Table 1 show that the performance of the techniques are comparable in a case of correct reproduction.

From this set of experiments, we concluded that Calinon et al. has the potential to deviate more from the shape of demonstrated trajectories in non-starting points as HMM may create multiple state transitions between states. Based on this we designed experiments with trajectories that could create a state around the starting point and have multiple state transitions from the starting state, one for each type of trajectory demonstrated to be precise. In our previous experiments we used trajectories that diverge from the starting point, but there is no such requirement.

4.2. m-shaped Trajectory Reproduction Task

Trajectories which are more complex in two dimensions were encoded and generated and these trajectories did not immediately diverge from the starting point, enabling formation of a state close to the starting point with multiple state transitions.

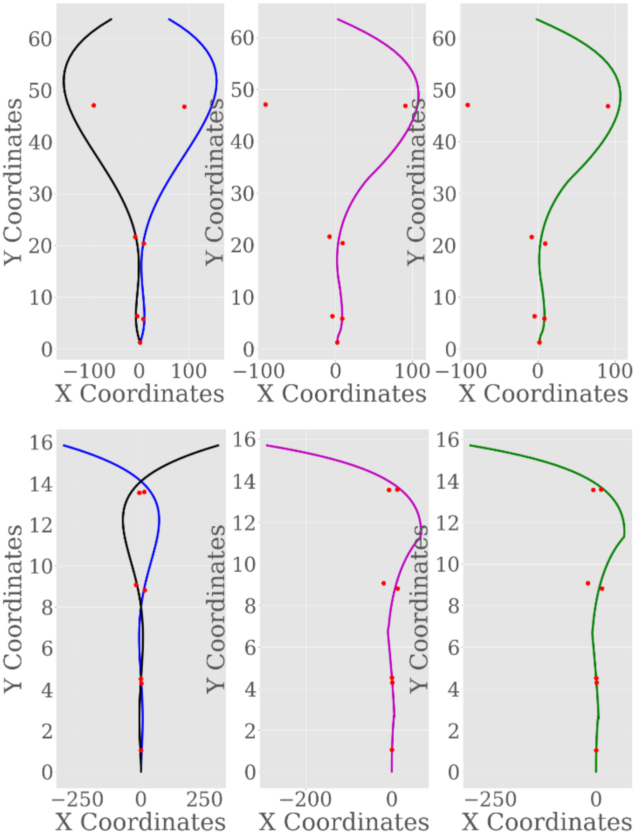


Fig. 8. Experiment for the ‘m’ shape which uses crossing points both for position and velocity simultaneously in two dimensions, from a biased starting point which is expected to generate the trajectory with the blue line. Upper plots denote the position profiles and lower plots denote the velocity profiles of the trajectory. Blue points belong to the other type of demonstrated trajectory, magenta points are the reproduction of HMMR, and the green points are the reproduction of Calinon et al. [6]. Red dots are the locations of HMM state centers.

Complexity of the trajectory comes from using crossing points both in position and velocity. Reproducing these trajectories are challenging because the encoded HMM would have multiple state transitions and both the position and velocity data are the same for all types of demonstrated trajectories around the crossing points. This means that around the crossing points, the algorithms would have no preference and choose to follow from any type of demonstrated trajectories, and not necessarily the same type they started the reproduction with, therefore failing the task, just like Fig. 3. However, both algorithms succeeded as can be seen in Fig. 8. We attribute this to HMM states not being exactly symmetrical and not always having equal state transition probabilities after training, enabling correct reproduction for both algorithms in practice. 4 demonstrations were given for each type of trajectory which can be seen in Fig. 7, and Gaussian noise with 0 mean and 0.2 variance was added to the demonstrated data for this experiment.

For this task, we performed another experiment where 2 HMMs were utilized in the algorithms; one for position and velocity, which was used in previous experiments and the other for velocity and acceleration. This was thought as a countermeasure in case crossing points both in position and velocity caused the algorithms to fail but the results were similar, and we did not observe any significant effects from using 2 HMMs.

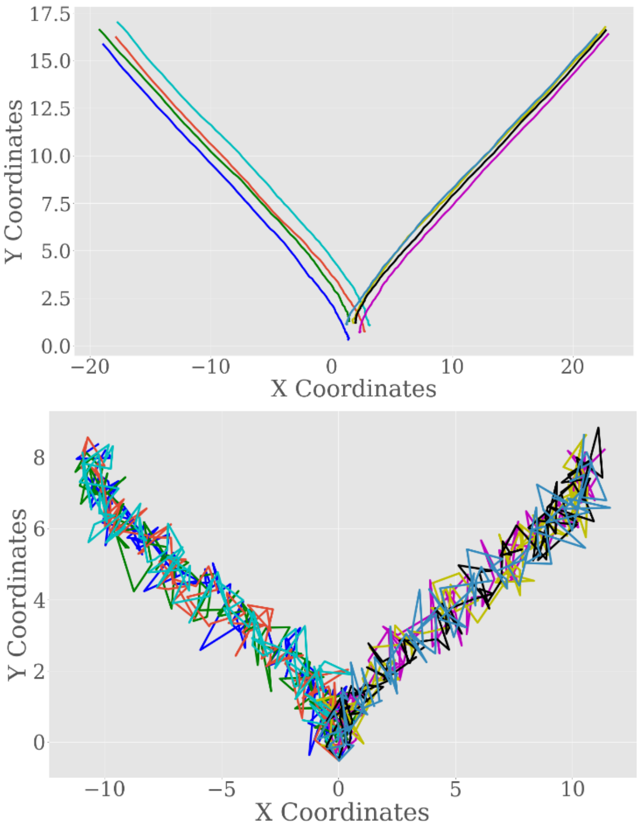


Fig. 9. Experimental setup for V-shaped trajectory reproduction task. Upper plot denotes the position profiles and lower plot denotes the velocity profiles of the demonstrated trajectories. Same color is used for the position and velocity of a demonstration, and there are 2 demonstrated types: left and right.

4.3. V-shaped Trajectory Reproduction Task

We performed an experiment in two dimensions with a trajectory where the trajectories did not immediately diverge from the starting point, enabling formation of a state close to the starting point with multiple state transitions. 2 types of trajectories were used, one moving towards the left after the start, and the other towards the right. 4 demonstrations for each type were used to train the HMM and Gaussian noise with 0 mean and 0.5 variance was added to the data. The trajectories together form a V-shape as can be seen in Fig. 9, and the motion starts from the bottom. Resulting HMM has the initial state at the bottom of the 'V', with a total of three state transitions: one transition to itself, one towards the left and one towards the right. Multiple starting points were tested, and an interesting result was found when a biased starting point towards the left was provided. Reproduction of a trajectory that moved towards the left was expected, and while HMMR performed successfully, Calinon et al. [6] failed and created a trajectory that moved towards the right, as shown in Fig. 10. In this experiment, the starting point at coordinates x = 2, y = 1 was set up as an unbiased starting point. Any point with an x-coordinate smaller than 2 is expected to reproduce a trajectory going towards the left. Due to HMM states not being exactly symmetrical however, slight variance is expected. In HMMR reproduction, at x = 1.94, the change from a right trajectory to a left one is observed whereas in the reproduction of Calinon et al. [6] the change came at x = 1.62, which is a highly biased point, and the algorithm was expected to respond to the change in coordinates at less biased points, such as at x = 1.8. The summing up behavior of Calinon et al. [6] resulted in a much-delayed response in this case, and HMMR was more robust in its reproduction.

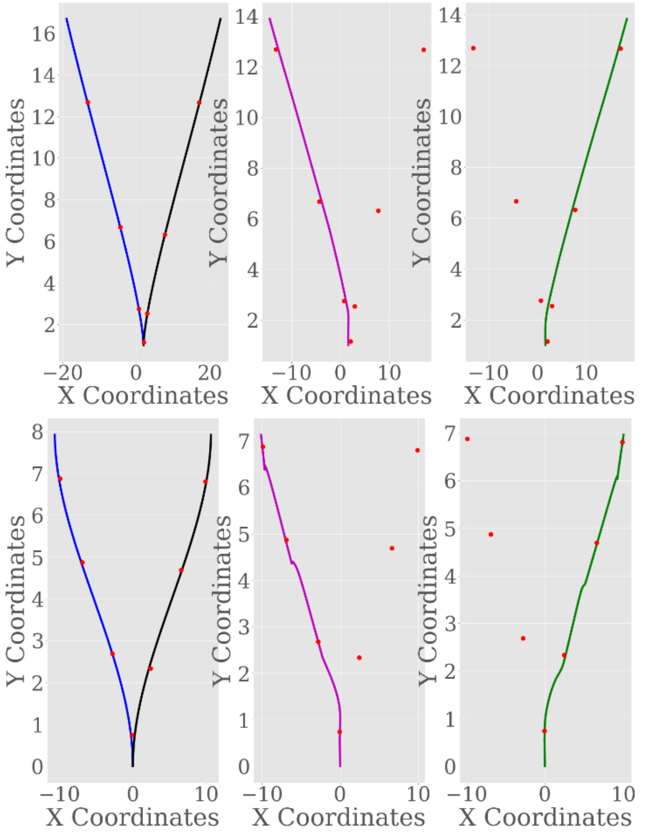


Fig. 10. Experiment for the 'V' shape in two dimensions, using a biased starting point towards the left, which is expected to generate a trajectory moving towards the left. Upper plots denote the position profiles and lower plots denote the velocity profiles of the trajectory. Blue lines are the expected reproduction, magenta lines are the reproduction of HMMR, and the green lines are the reproduction of Calinon et al. [6]. Red dots are the locations of HMM state centers.

5. Conclusion

In this paper we proposed and implemented a solution able to successfully reproduce the desired trajectory among multiple types taught in succession without any requirement for an additional mechanism or parameter. A solution utilizing HMM state transition probabilities during the GMR estimation was proposed, since this would satisfy the expected decision capabilities. We verified our approach via experiments, testing the ability of our algorithm to reproduce simple and complex trajectories accurately. Apart from multiple other trajectory reproduction tasks, our algorithm succeeded in reproducing successfully the trajectories mentioned in this work, namely shaped, m-shaped and V-shaped trajectories. We also compared our experimental results with Calinon et al. [6] who also utilized state transition probabilities of HMM during GMR estimation, but their work took into account all the hidden states instead of the most probable one as we did. The results of our simulation experiments with S-shaped trajectories are provided in Table 1, where the achieved RMS error is shown to be comparable to results reproduced from Calinon et al. [6]. We observed that using only the most probable hidden state created significant differences in certain reproductions. We qualitatively evaluate the reproductions starting from biased points towards one side when the reproductions tend to the other side, since the task is then a failure and quantitative metrics do not yield useful information since the expected reproduction and the actual reproduction are very far from each other. An example is shown in Fig. 10 where the reproduction of Calinon et al. [6] is a failure whereas HMMR reproduction is a success. Using starting points that are unbiased created difficulties for choosing a type of trajectory in [6], and HMM states having multiple state transitions reduced [6]'s ability to respond correctly when biased starting points were provided, due to it summing up the influences from all the states with a transition from the current state. We observed that our approach had no problem handling multiple state transitions to different states, which may occur as a mistake when the demonstrated trajectories have no such property and multiple state transitions from one state to different states is undesired, or characteristic to the type of trajectory in which case the transitions are desired. HMMR is shown to perform better in such cases and the performance of the two algorithms are shown to be comparable in other experimented cases with the exception of the overcommitting phenomenon, explained in Section 4 that can lead to sudden jerks during reproduction, which must be considered while using the algorithm.

Cases where both algorithms were expected to fail were also considered, however the experiment results were successful reproductions. Still, we believe that there is merit to consider such cases.

In the future, we plan to exploit our HMMR method in real robotic industrial settings where the task is taught to the robot through kinesthetic teaching or demonstrations; and the robot is able to successfully perform the expected version of a task, among multiple possible ways to perform it, based on the starting conditions provided.

Acknowledgment

This research has received funding from the European Union's Horizon2020 research and innovation programme under grant agreement no. 731761, IMAGINE.

References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in Handbook of Robotics, B. Siciliano and O. Khatib, Eds. Secaucus, NJ, USA: Springer-Verlag, 2008, pp. 1371–1394.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," Robotics and autonomous systems, Vol. 57, No. 5, pp. 469-483, 2009.

[3] M. Khansari and A. Billard, "BM: An iterative method to learn stable non-linear dynamical systems with Gaussian mixture models," in Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Anchorage, Alaska, USA, May 2010.

[4] A. Vakanski, F. Janabi-Sharifi, I. Mantegh, and A. Irish, "Trajectory learning based on conditional random fields for robot programming by demonstration," in Proceedings of the IASTED International Conference on Robotics and Applications (RA'2010).

[5] T. Alizadeh, S. Calinon, and D. G. Caldwell, "Learning from demonstrations with partially observable task parameters," 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 3309-3314, IEEE, 2014.

[6] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell and A. G. Billard, "Learning and reproduction of gestures by imitation," IEEE Robotics & Automation Magazine, Vol. 17, No. 2, pp. 44-54, 2010.

[7] E. Ugur and H. Girgin, "Compliant Parametric Dynamic Movement Primitives," Robotica, vol. 38, no. 3, pp. 457–474, 2020.

[8] H. Girgin, and E. Ugur, "Associative skill memory models," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6043-6048, IEEE, October 2018.

[9] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in Adaptive motion of animals and machines, pp. 261-280, Springer, Tokyo, 2006.

[10] S. Calinon, and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot," in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), Vol. 3, pp. 2769-2774, IEEE, September 2004.

[11] A. Vakanski, I. Mantegh, A. Irish and F. Janabi-Shari, "Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 42, No. 4, pp. 1039-1052, 2012.

[12] S. Calinon, and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," Proceedings of the 22nd international conference on Machine learning, pp. 105-112, ACM, 2005.

[13] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[14] Z. Ghahramani, and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," Advances in neural information processing systems, pp. 120-127, 1994.

[15] S. Calinon, E. L. Sauser, A. G. Billard and D. G. Caldwell, "Evaluation of a probabilistic approach to learn and reproduce gestures by imitation," 2010 IEEE International Conference on Robotics and Automation, pp. 2671-2676, IEEE, 2010.

[16] A. K. Tanwani and S. Calinon, "Learning robot manipulation tasks with task-parameterized semitied hidden semi-markov model," IEEE Robotics and Automation Letters, vol. 1, no. 1, pp. 235-242, Jan 2016.

[17] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 37, no. 2, pp. 286-298, 2007.

[18] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," in Lazy learning. Springer, 1997, pp. 75-113.

[19] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local Gaussian process regression," Adv. Robot., vol. 23, pp. 2015–2034, 2009.

[20] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," IEEE Transactions on Robotics, vol. 26, no. 5, pp. 800-815, 2010.

[21] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: Incremental real time learning in high dimensional space," in Proceedings of the Seventeenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., 2000, pp. 1079-1086.

[22] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," Intelligent Service Robotics, vol. 9, no. 1, pp. 1-29, 2016.