Learning from Demonstration: Trajectory Modelling Using DMPs and SEDS

Ahmed Elbary
Department of Computer Science
University of Lincoln
Lincoln, United Kingdom
29385647@students.lincoln.ac.uk

***Abstract*— This report investigates robot learning from demonstration by applying two main dynamical systems. The goal was to model demonstrated trajectories of varying complexity. For DMPs, the effect of varying the number of basis functions was explored the learned trajectories, evaluating performance using root mean squared error (RMSE). For SEDS, the number of Gaussians was tuned based on the Bayesian Information Criterion (BIC) to find the optimal parameter for each method. The results demonstrate that both DMPs and SEDS can successfully model and generalize motion trajectories, with DMPs offering more flexible trajectory shaping and SEDS ensuring stability and convergence.**

***Keywords*— LFD, DMP, SEDS, BIC, RMSE**

I. Introduction

Robots are noticeably expected to work in dynamic environments and beside non-expert users. To meet this required demands, Robot Learning from Demonstration (LfD) has appeared as a promising approach that enables the robots to have new skills by observing the human behaviour. In contrast with the traditional programming methods that ask for manually controlling, the LfD offers a more flexible framework for learning more complex behaviours from the data given.

In the context of a trajectory learning, the LfD is often used to model time movement patterns that a robot can learn and repeat the tasks across varying scenarios [1]. Two main studied techniques in this area are: Dynamic Movement Primitives (DMPs) and Stable Estimator of Dynamical Systems (SEDS), where Both of them aim to produce demonstrated trajectories but also achieving stability and generalization, however they differ in their mathematical formulations and guarantees. In the DMP it represents the trajectories through dynamical systems that are modulated by forcing terms, which allow for a time scalability and. In contrast, the SEDS models a general stable dynamical system using Gaussian Mixture Models (GMMs) and applies convergence through the Lyapunov constraints.

This report aims to explore and compare both approaches using a set of 2D demonstrated trajectories of varying complexity between them. The goal is to discuss their theoretical foundations based on two relevant papers and their practical implementation, while the rest of the report applies both methods to a real dataset, highlighting key differences in stability, generalization, and performance.

II. Literature Review

Robot Learning from Demonstration (LfD) enables the robots to learn more skills by understanding human provided instructions, avoiding the need for task specific programming for each scenario. This method has been widely used in robot manipulation, motion planning, and also in human-robot applications. Among the LfD methods two adopted frameworks have been widely used: Stable Estimator of Dynamical Systems (SEDS) and Dynamic Movement Primitives (DMPs) by demonstrating a strong ability for trajectory modelling where each of these methods has its own distinct advantages.

Khansari-Zadeh and Billard have introduced the SEDS framework in which described the motion learning as the estimation of a nonlinear time invariant dynamical system [1]. The system models the flow of velocity vectors across the state space using the Gaussian Mixture Model (GMM) and trained to produce the observed behaviour while also ensuring that all the trajectories is reaching to a desired goal. This formulation enables more robustness and allows the system to respond dynamically without relying much on time indexed trajectories. The key concept of the SEDS is its use of the Lyapunov stability theory to include global stability limitations directly into the GMM parameter optimization, which is described as a convex optimization problem [1]. This method ensures the theoretical convergence while keeping the computationally efficient.

The demonstration data for SEDS typically consists of position and velocity pairs, obtained using the kinaesthetic teaching or teleoperation technique [1]. The output of the learning is a usually stable and continuous velocity field to guide the system to reach the goal. However, this model is sensitive to the number of GMM parameters, which have to be tuned carefully to avoid any overfitting. Moreover, because the SEDS is a time invariant, it is usually lacks the mechanisms for encoding the time dependent behaviours which can be one of the limitations when temporal modulation is required.

In order to locate the SEDS within a wider landscape, Ravichandar et al. provided a comprehensive survey of LfD approaches, by offering some methods based on their learning outputs, generalization capabilities, and demonstration modalities [2]. They distinguish between the low-level primitives like those that are used in SEDS and DMPs, and the high-level task policies that abstract away from specific trajectories. The survey also emphasizes the power of SEDS in its ability to be stability, and real time application, especially in collaborative robotics.

In contrast, the DMPs represent motor skills using the second-order differential equations with spring damper dynamics, which allows for temporal modulation and a smooth trajectory generation [2]. Furthermore, the DMPs can be adapted to different goal positions or durations of motion execution which make them suitable for the tasks when changing in the temporal or spatial conditions are required. However, DMPs lack the built-in stability guarantees in contrast with SEDS, and extra effort is needed to ensure that the trajectories do not diverge or having unstable behaviour under any noisy [2].

Ruan et al. [3] has proposed PRIMP which refers to Probabilistically Informed Motion Primitives, a recent framework that combine the probabilistic modelling into the motion primitive learning. PRIMP captures the distribution of 6D trajectories and enables flexible adaptation to new task usage such as varying viewing perspectives. It is addressing a key limitation of the other LfD models. By combining the probabilistic trajectory modelling with the optimized planning, PRIMP supports the obstacle avoidance and the task generalization in random environments. Experiments demonstrated that PRIMP outperformed the traditional methods in both trajectory accuracy and the learning efficiency [3].

The survey also discusses the other ways through which demonstrations are introduced: the passive observation, teleoperation, and the kinaesthetic teaching and how each one of these affect the model performances [2]. For instance, kinaesthetic teaching produces a highly aligned input data, which suits the SEDS stability framework, but can be physically limited in scalability improvement. The DMPs, in contrast offer more variance to noisy or less consistent demonstrations because of their dynamic modulation capabilities [2].

From a machine learning perspective, both the SEDS and the DMPs are classified as non-deep, prioritizing transparency over raw model complexity [2]. This makes them especially suitable for the applications with a limited training data or real time control requirements, as in human robot collaboration. In contrast, the deep learning methods may offer better performance in more complex systems and datasets, but they for sure require greater computational resources.

In summary, SEDS provides a basic approach to learning converging motions from demonstrations, offering more formal guarantees of stability, generalization across initial conditions, and efficient learning [1]. While the DMPs are more flexible in encoding and adapting motion systems, they lack convergence assurances and require tuning to match the task goals [2]. PRIMP, as a recent advancement, using the probabilistic modelling with task the adaptation [3]. These trade-offs build the basis for the comparative evaluation of this report, where both methods the SEDS and the DMPs are applied to real-world 2D trajectory datasets and assessed in terms of accuracy and stability.

III. METHODOLOGY

In this section of the report, it outlines the two trajectory learning techniques explored in this study: Dynamic Movement Primitives (DMPs) and the Stable Estimator of Dynamical Systems (SEDS). Indeed, these methods are selected for their ability to model the continuous behaviours based on the demonstration data provided. Each subsection presents the mathematical formulation of the method used, followed by its key features and role in motion production.

A. *Dynamic Movement Primitives (DMPs)*

In the case of the Dynamic Movement Primitives (DMPs) are a trajectory learning framework that encodes the robot motions using a system of a second order differential equations where the goal is to reproduce the demonstrated movements but also to allow for adapting to any new goals. The DMPs are widely used because of their generalization capability, making them more suitable for encoding multiple degrees of freedom and compound behaviours.

The DMP describes the motion through the combination of a goal attracting term and a nonlinear forcing term. The basic formulation for a single dimension is:

$$\ddot{q} = k_{gain}(g - q) - d_{gain}\dot{q} + f(s)$$

Where the q represent the system state (e.g., position), g is the goal, while the $\dot{q}$ and $\ddot{q}$ represent the velocity and the acceleration, and the $f(s)$ is the learned nonlinear function that modulates the trajectory shape, and the s is the phase variable. The constants k_{gain} and d_{gain} are representing the proportional and derivative gains in a PD controller.

The nonlinear forcing term is expressed as a weighted combination of Gaussian basis functions as below, where the weights w_i are learned from the demonstration data to shape the trajectory, and the parameters μ_h, σ_h, and the N which is the number of basis functions should be predefined.

$$f(s) = \left(\frac{\sum_{i=1}^{N} w_i \cdot\ h_i(s; \mu_h, \sigma_h^2)}{\sum_{i=1}^{N} w_i}\right) \cdot s \cdot (g - q_0)$$

In summary, the DMPs enable a scalable encoding of more complex movements with the flexibility to change the goals or the timing. However, they suffer to achieve a global stability furthermore, their performance can be affected easily by the number of basis functions and also the quality of the demonstration data.

B. *Stable Estimator of Dynamical Systems (SEDS)*

For the Stable Estimator of Dynamical Systems (SEDS) is a probabilistic framework that learns from the robot trajectories as a globally stable time invariant dynamical systems from the demonstrations. In contrast with the time dependent models like in the DMPs, the SEDS represents the movements using the velocity fields that are directly guiding the system to achieve the goal position, also to ensure that the system can convergence from any initial state. The key idea is to learn the mapping from the state $\xi \in \mathbb{R}^d$ to the velocity $\dot{\xi} \in \mathbb{R}^d$ such that:

$$\dot{\xi} = f(\xi)$$

This mapping is estimated using the Gaussian Mixture Regression (GMR) which forms the joint distribution $p(\xi, \dot{\xi})$ using the Gaussian Mixture Model (GMM) as below in this equation where N is the number of Gaussian components in the model:

$$p(\xi, \dot{\xi}) = \sum_{n=1}^{N} w_n \cdot N\left(\xi, \dot{\xi}; \mu_n, \sum n\right)$$

To ensure the convergence, SEDS is using the Lyapunov stability terms on the velocity field where this will guarantee that the learned system is globally stable toward a unique attractor ξ^* which is usually the task goal. Furthermore, the time invariant of the velocity field allows the robot to react to any external disturbances without any kind of delays. This ability makes the SEDS particularly more efficient in the dynamic environments, where the convergence is a critical matter. Its probabilistic basic also enables the extensions such as the confidence estimation which make it more flexible choice for motion learning in robotics.

In summary, the SEDS learns a smooth and stable vector field that helps to direct the robot motion model, making it much suitable for the tasks requiring safety and convergence from different initial conditions. However, it still requires a careful selection of the number of Gaussian number and may struggle with tasks that involve strongly nonlinear dynamics.

C. *Datasets and Preprocessing*

This study uses a collection of 2D trajectory datasets provided, where each file shows a series of motion demonstrations recorded, with each row representing a discrete time step and each column representing the dimensions. The datasets used to include some motion shapes such as CShape.csv, Sshape.csv, WShape.csv, and Line.csv.

Each demonstration was assumed to begin processing from a consistent initial point and progress to a clear defined goal which is necessary to convergence the modelling in both methods. Since the SEDS learns mapping from the position to the velocity, it required an accurate estimation of the velocity for each time step.

For the DMPs, the entire trajectory including the position, the velocity, and the acceleration are required for learning the shape of the motion inside the forcing term. The same demonstration data were used as input where the required derivatives are computed by the DMP library (e.g., via pydmps). The initial state q and the goal g are extracted directly from the first and last points respectively. Then the trajectory is encoded using a time independent variable to allow a temporal scaling during the reproduction.

D. *Parameter Tuning*

Both SEDS and DMPs require the selection of key hyperparameters that directly influence learning performance and trajectory reproduction quality. In this study, the parameters were chosen using different strategies appropriate to each model's underlying formulation.

1) *Tuning for SEDS*

The SEDS framework depends more on fitting a Gaussian Mixture Model (GMM) to a joint distribution of the position and the velocity values, where the number of Gaussian N is the main parameter; that is why tunning its value should be done gently where the low number of Gaussian may result in loss of trajectory detail, while the high value can lead to overfitting and instability in the learned velocity field.

To professionally do the process of finding the optimal number of components, the Bayesian Information Criterion (BIC) was employed. The main aim of the BIC is to evaluate the models by balancing the gap between the model's likelihood and its complexity, and is defined as:

$$BIC = k \cdot \ln(n) - 2 \cdot \ln(\mathcal{L})$$

Where the k is the number of the model parameters generally (the mean, covariance) for each GMM, the n is the number of data samples, and finally the $\mathcal{L}$ which is the likelihood of the model given. The GMMs with changing the numbers of Gaussians were trained on the demonstration data, and the BIC score was computed for each where the model with the lowest BIC was selected for the final SEDS fitting.

2) *Tuning for DMPs*

In the case of the DMPs, the key parameter is the number of Radial Basis Functions (RBFs) which is as n_{bfs} , the idea of this is to determine the resolution of the forcing term that shapes the trajectory. The higher the number of RBFs enables the model to capture more complex trajectory shapes but also increases the risk of overfitting and higher computational cost.

The DMPs are not probabilistic models, and they do not have a formal model selection such as the BIC, that is why the Root Mean Squared Error (RMSE) was used instead. The DMPs were trained using multiple values for n_{bfs}. For each configuration, the model was trained, and the produced trajectory was compared to the original demonstration using the Root Mean Squared Error (RMSE) using the equation below where $y_{demo}(t)$ and $y_{pred}(t)$ are the demonstration and the DMP predicted values at time y respectively:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left\| y_{demo}(t) - y_{pred}(t)\right\|^2}$$

IV. Results and Discussion

In this section presents the results of applying the two learning from demonstration techniques to multiple 2D motion trajectories of varying complexity. Where each method was evaluated in terms of its ability to reproduce the demonstrated paths and the influence of key tuning parameters. RMSE was used as the primary performance metric for the DMP and the BIC was used in the SEDS, and qualitative plots were used to support these values.

A. *DMP Results*

The Dynamic Movement Primitive (DMP) framework was applied to the *WShape* trajectory dataset to evaluate its ability to reproduce the most complex motion pattern. The number of radial basis functions (RBFs) was varied between 10 and 300 to analyse its effect on the trajectory performance measured using the Root Mean Squared Error (RMSE).

As shown in Figure 1 the RMSE decreased rapidly up to approximately where the improvement slowed significantly. The reasonable RMSE was found around $n\_bfs = 150$, since the higher number of basis increases led to minimal gains.

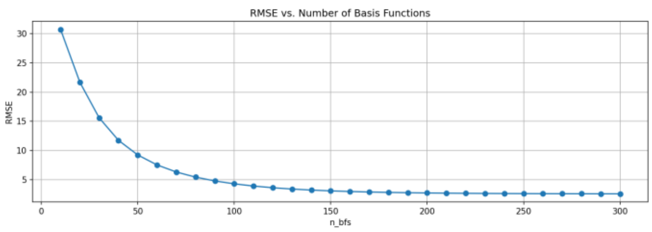


Figure 1 RMSE vs. number of basis functions on WShape dataset.

The optimal visual reproduction was achieved with 150 basis functions, producing a smooth path closely aligned with the demonstration but also not consuming high computational cost. The plot below in Figure 2 shows the best predicted DMP trajectory versus the demonstration input.

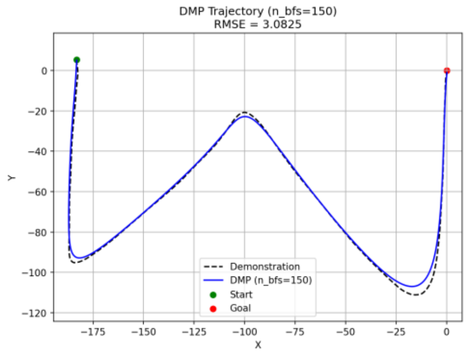


Figure 2 The Best DMP trajectory ($n\_bfs = 150$) on WShape

As shown in Table 1, the optimal number of basis functions varied based on the shape complexity, with the simpler trajectories requires fewer basis functions and more complex shapes needs higher values. The main results and plots for the WShape dataset are presented in the main body of this report due to its complexity and illustrative value. Plots of the DMP reproductions for the remaining shapes are provided in Appendix A.

Table 1 Optimal number of basis functions using DMP.

Trajectory	C Shape	Line	S Shape	W Shape
Optimal n_{bfs}	80	70	130	150

B. SEDS Results

For the SEDS learning method, the same four trajectory shapes were tested, where the goal was to model the dynamics of each demonstrated trajectory using the Gaussian Mixture Models (GMMs) and to learn the stable dynamical system that score to the goal position. For each shape, the number of Gaussian components K was varied from 2 to 10. The Bayesian Information Criterion (BIC) has been used to select the optimal number of Gaussian this time. Example of the BIC plot for the $WShape$ trajectory is shown in Figure 3, where each point presents a different number of Gaussians. The value of K that led the lowest BIC was selected as the optimal value for each shape.

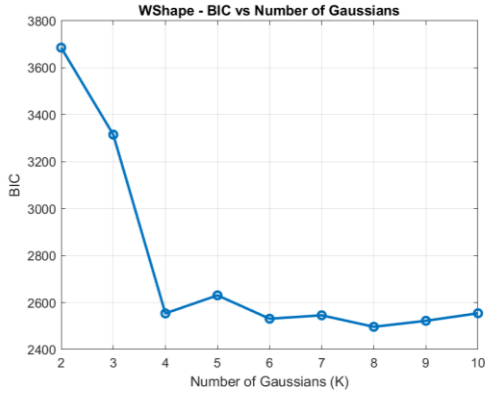


Figure 3 BIC plot for the $WShape$ trajectory

In the case of the $WShape$, the predicted trajectory was able to follow the demonstrated path closely while ensuring stability and convergence to the goal with 8 Gaussians. This is illustrated in Figure 4 below which shows the learned trajectory generated using the optimal value of $K = 8$ with the original shape input.

Figure 4 The Best SEDS trajectory (K=8) on $WShape$

Plots of the SEDS reproductions for the remaining shapes are provided in Appendix B. Table 2 below presents the optimal number of Gaussians selected for each trajectory based on the BIC value. This provides an overview of the model complexity required to learn the dynamics of each shape.

Table 2 The Optimal number of Gaussian using SEDS

Trajectory	C Shape	Line	S Shape	W Shape
Optimal n_{bfs}	7	5	5	8

Overall, more complex shapes like $WShape$ and $CShape$ required a higher number of Gaussian, but the simpler shapes like the $Line$ could be learned with fewer Gaussians. These results indicate that the SEDS framework is also capable of adapting to different trajectory complexities by adjusting it's the parameters.

V. Conclusion

This project demonstrated the application of the DMPs and the SEDS for learning from demonstration across multiple trajectory shapes of different complexity. In the case of DMPs, by increasing the number of basis functions lead to an improved in the trajectory accuracy up to a point, after which further gains become flattened. The SEDS models are evaluated using BIC to determine the optimal number of Gaussians all over the shapes. The findings suggest that DMPs are effective for capturing the fine details of trajectory shapes, but the SEDS provide robust and more stable properties. Both methods offer valuable capabilities for robot learning, with the choice of techniques guided by the nature of the task

Acknowledgement

I extend my deepest gratitude to Dr Athanasios Polydoros for his guidance and support in carrying out this project. Sincerely thank the authors of the referenced papers for their inspiring research.

References

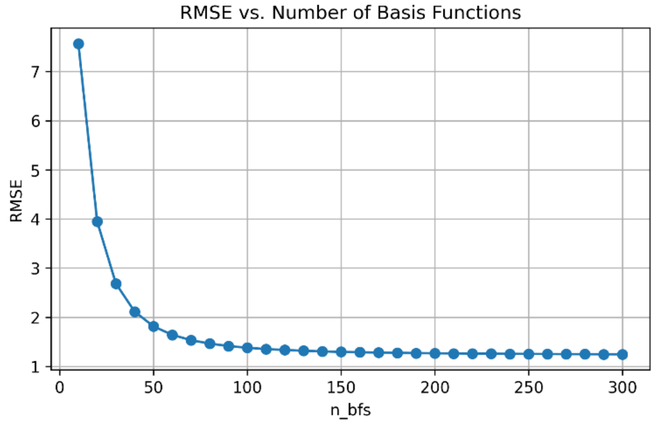
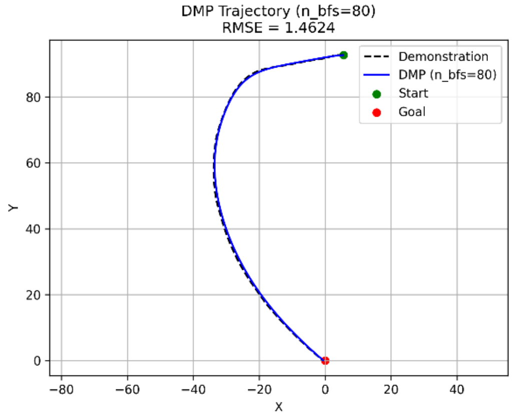
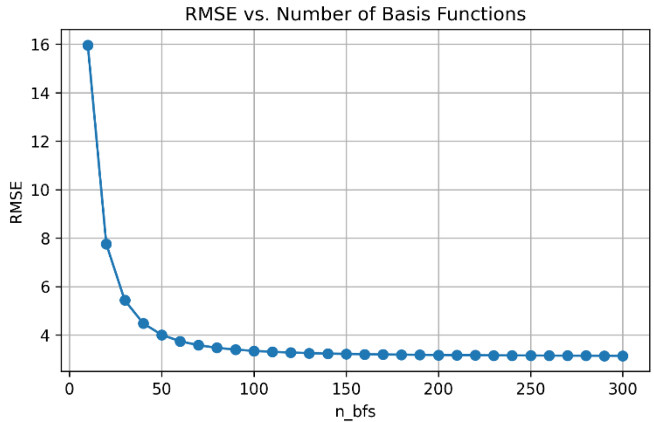
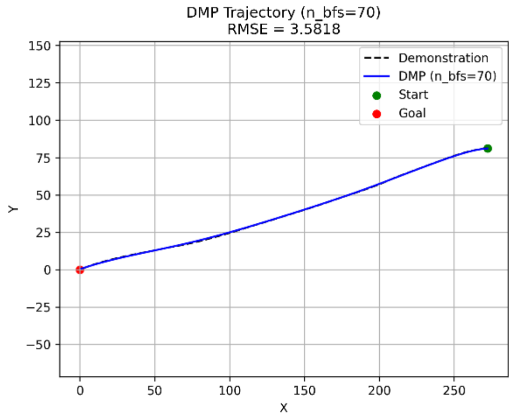
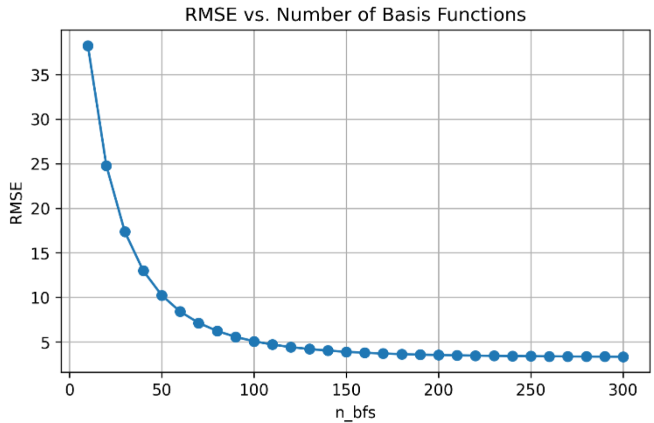
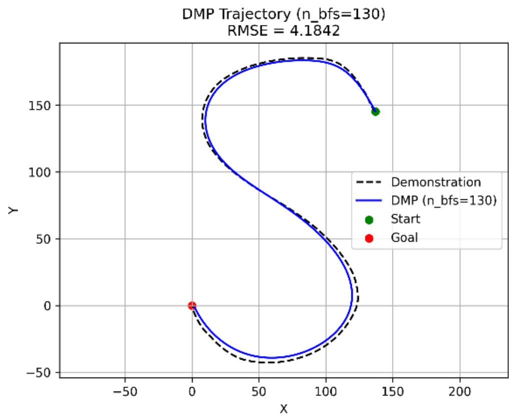
[1] A. Noohian, M. Raisi and S. Khodaygan, "A Framework for Learning Dynamic Movement Primitives with Deep Reinforcement Learning," *2022 10th RSI International Conference on Robotics and Mechatronics (ICRoM)*, Tehran, Iran, Islamic Republic of, 2022, pp. 329-334, doi: 10.1109/ICRoM57054.2022.10025346.

[2] S. M. Khansari-Zadeh and A. Billard, "Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models," in *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943-957, Oct. 2011, doi: 10.1109/TRO.2011.2159412.

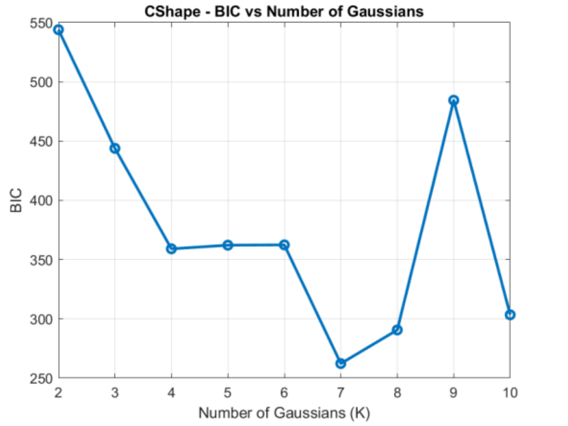
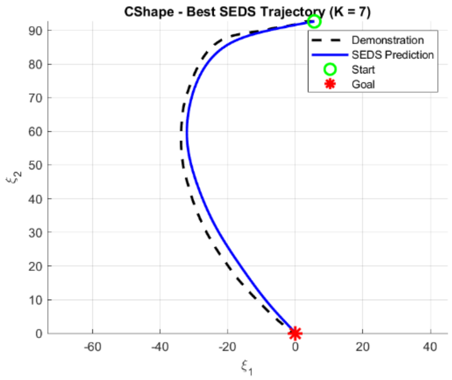
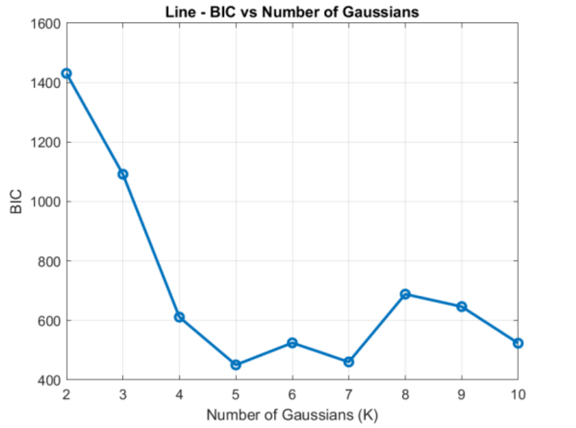
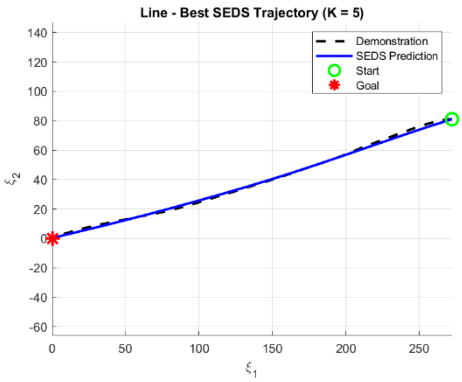
[3] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent Advances in Robot Learning from Demonstration," *Social Science Research Network*, vol. 3, no. 1, pp. 297–330, May 2020, doi: 10.1146/ANNUREV-CONTROL-100819-063206. Available: https://www.annualreviews.org/doi/pdf/10.1146/annurev-control-100819-063206

[4] S. Ruan, W. Liu, X. Wang, X. Meng and G. S. Chirikjian, "PRIMP: PRobabilistically-Informed Motion Primitives for Efficient Affordance Learning From Demonstration," in *IEEE Transactions on Robotics*, vol. 40, pp. 2868-2887, 2024, doi: 10.1109/TRO.2024.339005

APPENDIX A (DMP PLOTS)

C Shape		
Line		
S Shape		

APPENDIX B (SEDS PLOTS)

C Shape		
Line		
S Shape