Adaptive Collision-Free Reaching Skill Learning from Demonstration

Uchenna Emeoha Ogenyi[1], Dalin Zhou[2], Zhaojie Ju[3] and Honghai Liu[4]

[1] University of Portsmouth, United Kingdom
uchenna.ogenyi@port.ac.uk

[2] University of Portsmouth, United Kingdom
dalin.zhou@port.ac.uk

[3] University of Portsmouth, United Kingdom
zhaojie.ju@port.ac.uk

[4] University of Portsmouth, United Kingdom
honghai.liu@port.ac.uk

Abstract. In this paper, we considered the task of the robot learning low-level trajectory task in a novel clustered constraint environment.We propose a novel adaptive trajectory algorithm used to generate the necessary trajectory which satisfies the constraint of avoiding collision with an obstacle. Our approach is based on Gaussian mixture model which decomposes the trajectory into several ellipses since the isoline of a single Gaussian model is also an ellipse. Moreover, we employed the principle of the artificial potential field to modify the direction of the motion in the presence of obstacles. Since our approach is based on the underlying reactive skill dynamics, it does not share the same disadvantages as approaches which assume both the model of the task trajectory and the response from the obstacle should be learned from the demonstrations.

1 Introduction

The most essential skill in humans is the reaching skill. Human beings apply reaching skill in most of our daily life activities to bring a human hand to the object of interest without collision. This strong environmental adaptive skill is required for collaborative robot arms that must coexist with a human operator at work places such as in part assembly tasks. For industrial manipulators, it is usually assumed that the motion of the end-effector is mostly disturbed by any obstacle especially when the task learning is a low-level trajectory task. In such a situation, the task execution must be interrupted and a control algorithm has to be used to move the end-effector around the obstacle. Addressing this challenging problem requires a robot to possess a set of skills which may be difficult to pre-program since the position of the obstacle is usually not known in advance.

Ignorance of such circumstance led to most prior approaches proposed in the past [1], [2] assuming that demonstrations are performed in uncluttered constrained environments. But the presence of cluster in the environment can introduce additional constraints which if not accounted for can undermine the underlying human intent of reaching the desired goal. So, their approach is

impractical in a real-world scenario since human environment keeps changing with time.

Further approaches employed the principles of motion planning algorithms which can compute collision-free path required for a robot to successfully complete a task [3]. Researchers have employed various motion planning algorithm, such as Dynamic roadmaps (DRM) [4, 5], Rapidly Exploring Random Trees (RRT) [6, 7], and elastic [8, 9]. Although, significant improvements have been made using this approach, in general, motion planning algorithms is computational complex due to the high dimensionality of the configuration space that must be considered.

Another school of thought believe that Learning from Demonstration (LfD) could better be employed in learning skills from human demonstration in a clustered environment. Amongst other benefits of LfD approach is its flexibility in learning complex motions and simplicity/easy to implement [10]. Also, LfD has the advantage of allowing people without programming skills to transfer tasks skills to the robot just by demonstrating and allowing the robot to learn from those demonstrations [11]. This attribute of LfD makes the robot more adaptive to a new environment with only minor adjustment to the learning parameters.

Several works have been proposed to learn the dynamic movement primitive as well as coupling terms for obstacle avoidance from demonstrations. For example, [12] learn a DMP which encapsulates correlation information of the coupling motor control variables and employed reinforcement learning to modulate the optimal parameters of the dynamical system in a new an environment. Furthermore, Chi et al., [13] method integrated dynamic potential filed with the acceleration equation of the DMP to realise reactive action depending on the distance and velocity between the robot's end-effector and the obstacle. [14] also employed this method and further introduced cost function to estimate the deviation from the mean of demonstrations to the distance of the obstacle from the environment. A major drawback of this approach is the assumption that only the mean of the demonstration sufficiently expresses demonstrated skills. In contrast, we propose a control strategy which aims to in addition to the mean of the distributed demonstrations identify and incorporate how spread the distribution is in order to efficiently express the demonstrated skills. Moreover, DMPs allows only a single demonstration which limited its ability to potentially learn different ways of executing tasks skills, hence limiting its robustness in new scenarios.

On the other hand, the statistical model (GMM/GMR) permits variant demonstrations for task learning, hence promising robust generalisation of tasks in new scenarios. The probabilistic approach 'GMM/GMR'was used for modelling and predicting motion trajectory [15]. In another work, [16] presented a method that uses GMM/GMR to encode and compute an average trajectory from a set of sub-optimal task demonstrations. Some approaches [17] added the skill constraints at the demonstration stage such that the influence of the constraints is learned to achieve the desirable trajectory at the reproduction stage. However, learning the influence of the entire constraints from the demonstration

stage is not feasible and therefore this approach is not suitable for an adaptive reaching tasks considered in our work. Our proposed approach estimates the parameter both from the demonstration state and the reproduction stage to more accurately acquire the desired trajectory.

In our work, we tackle the problem of learning skills from human demonstrations which can be influenced by the presence of obstacles. We use kinaesthetic teaching for data collection from human demonstrations. From these demonstrations, robot joint angles and its associated timesteps are recorded and used to learn the direction which will lead the end-effector through the desired motion. We adopt the GMM to compute the optimal path for the robot to reach the goal point and proposed a modified virtual repulsive potential function to avoid obstacles.

2 Motion Trajectory learning

The most popular method for learning human motion trajectory is by using Gaussian Mixture Model (GMM). The GMM is a mixture of a sequence of Gaussian distributions that can allow an arbitrarily large number of Gaussian components and a small number of variances. Rather than having hard assignments into clusters, like in K-means, GMM supports soft assignments which implies that each distribution has some level of responsibility for producing a given data point.

2.1 Gaussian Mixture Model

To extract the characteristics of a human arm motion trajectory for any collection of time $t = (t_1, t_2, ..., t_N)^T$, the probability density function of the Gaussian distribution of a vector $x = (x_1, x_2, ..., x_N)^T$ with K-components of a D-dimension is defined as:

$$p\left(x_j|\theta\right) = \sum_{m=1}^{M} \pi_m N\left(x_j|\mu_m,\ \Sigma_m\right) \tag{1}$$

Where $\mathrm{p}(x_j|k) = (x_j; \mu_m,\ \Sigma_m)$ is the set of the model parameters; μ_m is the mean and Σ_m is the covariance matrix of the Gaussian, and (π_m) is the prior of the j-th component, and exp is the exponential function. The priors are adjusted to satisfy the constraints in (9) and (3) where the sum of all the mixture weights must be equal to 1 and each of the weight must lie between 1 and 0.

$$\sum_{m=1}^{M} \pi_m = 1 form = 1, ..., M \tag{2}$$

$$0 < \pi_m < 1 \tag{3}$$

$$N\left(x_m; \mu_m,\ \Sigma_m\right) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma_m|}} \exp(-\frac{1}{2}(x_i - \mu_m)^{T} \Sigma^{-1}(x_i - \mu_m)) \tag{4}$$

In most cases, the optimal number of K-component is unknown. One way to estimate it is by using the information comparing criteria. The popular ones are the Baysian Information Criterion (BIC) and the Akaike Information Criterion (AIC). While both can estimate the required components, however, care must be taken on the choice of the information criteria, as the BIC tend to choose more complex models that might overfit whereas, the AIC tends to choose simple models that might underfit.

2.2 Gaussian Mixture Regression

In order to retrieve smooth trajectories of the estimated trajectories, the trajectories are considered as a regression problem and Gaussian Mixture Regression (GMR) is applied. Following the regression method, the conditional expectation of x_s given x_t is estimated as follows:

$$\mu_k = \begin{bmatrix} \mu_{t,m} \\ \mu_{s,m} \end{bmatrix}, \quad \Sigma_m = \begin{pmatrix} \Sigma_{tt,m} & \Sigma_{ts,m} \\ \Sigma_{st,m} & \Sigma_{ss,m} \end{pmatrix} \tag{5}$$

By applying the weighted mean and variance, the expected distribution of x_s given x_t can be computed as a block decomposition of the data-point x_j. For each component K, the conditional expectation of $x_{s,m}$ given the temporal value x_t and the estimated conditional covariance of x_s given x_t are given by the theorem of Gaussian conditioning based on combination property of Gaussian distribution as follows:

$$x_{s,m} = \mu_{s,m} + \Sigma_{st,m}(\Sigma_{t,m})^{-1}(x_t - \mu_{t,m}), \tag{6}$$

$$\Sigma_{s,m} = \Sigma_{s,m} - \Sigma_{st,m}(\Sigma_{t,m})^{-1}\Sigma_{ts,m} \tag{7}$$

$$\tag{8}$$

2.3 Learning GMM parameters

Learning requires estimating the model parameters (the mean, covariant and mixing coefficient) of the distribution to permit soft assignments of the K-components to the data points. It helps to find the best-fit parameters for the model. To find the maximum likelihood of a data point being fitted into the k-components, EM employs Bayes theorem to compute the probability that given observation belongs to each cluster. It works by choosing random values for the mixing data points and using those guesses to estimate the next set of the data [18]. The EM algorithm consists of two steps: E-step or Expectation step and the M-step or Maximisation step.

E-step: Estimates the distribution of the hidden variable given the data and the current values of the parameters. To achieve that, a latent variable Z_m is introduced to each data point. The latent variable Z_m indicates the probability that the i^{th} data point is generated from the m^{th} Gaussian components. In order to start the E-step, we need to initialise the values of the parameters. A good practice is to estimate them using k-means.

$$\mathrm{Z}_m^i = \frac{\pi_m N\left(x_i|\mu_m,\ \Sigma_m\right)}{\sum_{m=1}^{M} \pi_m N\left(x_i|\mu_m,\ \Sigma_m\right)} \tag{9}$$

$$\mathrm{Z}_m = \sum_{i=1}^{N} \mathrm{Z}_m^i \tag{10}$$

M-step: After calculating the posterior, we then need to estimate the parameters of each Gaussian and evaluate the log-likelihood. To do this, we first compute the maximum likelihood of the parameters estimated in E-step. This set of iteration continues until the threshold defined in the log-likelihood is met.

$$\pi_m^{new} = \frac{1}{N} \sum_{j=1}^{N} \mathrm{Z}_m^i \tag{11}$$

$$\mu_m^{new} = \frac{1}{\mathrm{Z}_m} \sum_{i=1}^{N} \mathrm{Z}_m^i x_i \tag{12}$$

$$\Sigma_m^{new} = \frac{1}{\mathrm{Z}_m} \sum_{i=1}^{N} \mathrm{Z}_m^i (x_j - \mu_m^{new})^T (x_j - \mu_m^{new}) \tag{13}$$

The EM is well known for its convenient and easily extensibility to incremental learning, however, the major limitation of the EM algorithm is that it requires to keep all the historical data in memory for accurate order update to be achieved. Moreover, traditional GMM cannot adjust the parameters of the distribution as new data points arrive or are acquired, and can hone in on a local maxima that is not close to the optimal global maxima. To overcome this problem, [19] proposed an algorithm which merges density components to improve the log-likelihood and reduce the number of clusters. This proposed algorithm failed to account for the novel data points that might arrive one-by-one as it universally assumed that new data comes in blocks. Addressing this challenge, [20] proposed an approach to tackle novel data points which arrives one-by-one by keeping only two GMM components in the memory and without historical data.

3 Theoretical Background of Artificial Potential Field

The traditional potential field method as proposed by [21] assumes that the robot is moving by the influence of abstract artificial force fields. The artificial field

consists of two components: the repulsive potential field and attractive potential field. Considering a robot that needs to move from its current position "A" to a goal position "B" in the presence of an obstacle "O". Using the principle of APF, the goal point generates attractive potential fields which make the robot move towards it. Conversely, the obstacle generates a repulsive force which is inversely proportional to the distance between the robot and the obstacle with a defined distance threshold to push the robot away from the obstacle. In other words, attractive force (16) is applied for the robot to reach the goal point and repulsive force (18) to avoid obstacles. The attractive artificial filed is represented as:

$$U_{att}(q) = \frac{1}{2}\eta\rho^2 \tag{14}$$

Here, η is a positive scaling factor, ρ is the distance between the robot q and the goal. Assuming the goal is a point object in a 2-D plane. The position of the goal and that of the end-effector could be expressed as vectors of $[x_g, x_g]^T$ and $[x_r, x_r]^T$ respectively. So, the parameter ρ which is the Euclidean distance between the robot end-effector position and the goal position which is calculated as below.

$$\rho \;=\; \sqrt{(x_r - x_g)^2 \;+ (y_r - y_g)^2} \tag{15}$$

The corresponding attractive force is given by the negative gradient of the attractive potential (16). Consequently, there is a move from higher to lower potential field along the negative of the attractive field in other to reach the goal position.

$$\mathrm{F(att)}(q) = -\nabla U_{att}(q) = \eta\rho \tag{16}$$

To prevent collision between the robot and the obstacle, the traditional repulsive potential function is represented as follow:

$$U_{rep}(q) = \begin{cases} \frac{1}{2}k(\frac{1}{d(x)}) - \frac{1}{d_0})^2 & : d(x) \;\leq \mathrm{d_0} \\ 0, & : d(x) \;> \mathrm{d_0} \end{cases} \tag{17}$$

where k is the repulsive potential gain coefficient, $d(x)$ is the minimum distance between the robot's current position and the obstacle, and d_0 is the distance threshold which limits the range of the repulsive potential field. So, when the distance between the robot and the obstacle is greater than d_0, the robot will not be affected by the repulsive force. Assuming, the position of the obstacle is represented as $[x_o, yo]^T$. Then, the associated repulsive forces could be computed by finding the negative gradient of the repulsive potential function (17):

$$\mathrm{F}_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} k(\frac{1}{d(x)}) - \frac{1}{d_0})\frac{1}{d(x)^2}\frac{\partial d(x)}{\partial q} & : d(x) \;\leq \mathrm{d_0} \\ 0 & : d(x) > \mathrm{d_0} \end{cases} \tag{18}$$

$$d_x \;=\; \sqrt{(x_r - x_o)^2 \;+ (y_r - y_o)^2} \tag{19}$$

The total repulsive potential field can be obtained by summing up the potentials caused by all of the obstacles within the workspace.

$$U_{rept}(q) = \sum_{i=1}^{N} U_{repi}(q) \tag{20}$$

Using this function, moving the robot to the goal while avoiding obstacle is achieved by following the direction of the resultant forces obtained by summing all the negative gradients attractive/repulsive potentials of the target and obstacle obtained from a given point in the plane. Doing this, the total potential force field vectors will point away from the obstacle and towards the goal as influenced by the combined gradient of the attractive/repulsive potentials.

$$\mathrm{F}_{net} = F_{att} \;+\; F_{rep} \tag{21}$$

4 Desired Direction Estimation

In this section, we presented an approach to fit ellipses on the generalised trajectory to permit elliptical potential field force around the segmented ellipse that formed the trajectory. In addition, we modelled the optimal closest point from the ellipse to the obstacle to enhance the robot's ability to avoid obstacle with minimal computational power.

4.1 Trajectory Segmentation with Fitted Ellipse

Considering that a Gaussian mixture model consists of several single Gaussian models (SGM) which can be represented by an ellipse since the isoline of probability density is also an ellipse [22]. Based on that, we hypothesized to fit ellipses on the generalised trajectory from the GMR to permit elliptical potential field force influence around the ellipse rather than on the generalized trajectory. To formed the ellipse and computed the orientation of the ellipse; firstly we computed the eigenvalues and corresponding eigenvectors and used then to find the axes of the strain ellipses which is used to calculate the ellipse orientation.

Using the general equation of an ellipse which is centred at (0,0), aligned at the major and minor axes that are defined by σ_x and σ_y; where $\sigma_x > \sigma_y$ is given as:

$$\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2 = 1 \tag{22}$$

Assuming the same single GMM is sampled from the underlying Gaussian distribution having the centre determined by the mean μ_i. Then, the ellipsoidal

probabilistic contour of the eigenvectors $\vec{V}_i$ with a corresponding covariance matrix Σ_i for a given K-component is given as $[\vec{V}_i,\ D_i] = \text{Eig}(\Sigma_i)$. This is an indication that the shape and size of the ellipse are underpinned by its associated covariance matrix Σ_i.

The orientation of the component is constructed as the angle ϕ of largest eigenvector towards the x-axis:

$$\phi = arctan\frac{\vec{V}x_2}{\vec{V}x_1} \tag{23}$$

where $\vec{V}x_2$ and $\vec{V}x_1$ are the eigenvectors of the covariance matrix that corresponds to the largest eigenvalues. This is an indication that the magnitude of the component alignment to the desired direction can be tuned by modifying the eigenvalues of the confidence ellipse and computing the angle of the largest matrix eigenvalues.

4.2 Obstacle Closest Distance

Various distance threshold calculation approaches have been proposed in the literature [23]. The most common approach is to introduce a fixed threshold value which must be maintained at every situation. This approach is not adaptive since the threshold value is fixed and is independent of the inclined angle of the object to the position of the robot. The most adaptive approach to calculate the threshold distance by sampling different point locations on the surface of the obstacle in order to estimate the closest point from each of the K-component ellipses of the trajectory.

The case of finding the closet point from the obstacle to a single GMM component (the trajectory) is established by analyzing the steps captured by [24] and summarized here. Considering an ellipse that is centred along the x and y axes, and parameterised by (x_e, y_e) with radius a and b representing the semi-major and semi-minor axis of the ellipse at an angle ϕ. With the values of point $P_p(x_p, y_p)$ and point $P_e(x_e, y_e)$ given, the distance $D(P_p, \phi)$ between the two points can be calculated as in Eq. 24.

$$(D(P_p, \phi)) \ = \ \sqrt{(x_p - x_e)^2 \ + (y_p - y_e)^2} \tag{24}$$

From Eq. 24, we can deduce that the distance $D(P_p, \phi)$, is a function of ϕ, and any value of ϕ which drive the differential of $D(P_p, \phi)$ to zero will form the minimum distance which is D_{min}.

$$\frac{D(P_p, \phi)}{d\phi} \ = \ 0 \tag{25}$$

By substituting the values of $x_e = a.cos\phi$ and $y_e = a.sin\phi$, into the equation 24, we can compute the minimum distance as shown in Eq. 26.

$$D_{min} \ = \ \sqrt{(x_p - a.cos\phi)^2 \ + (y_p - b.sin\phi)^2} \tag{26}$$

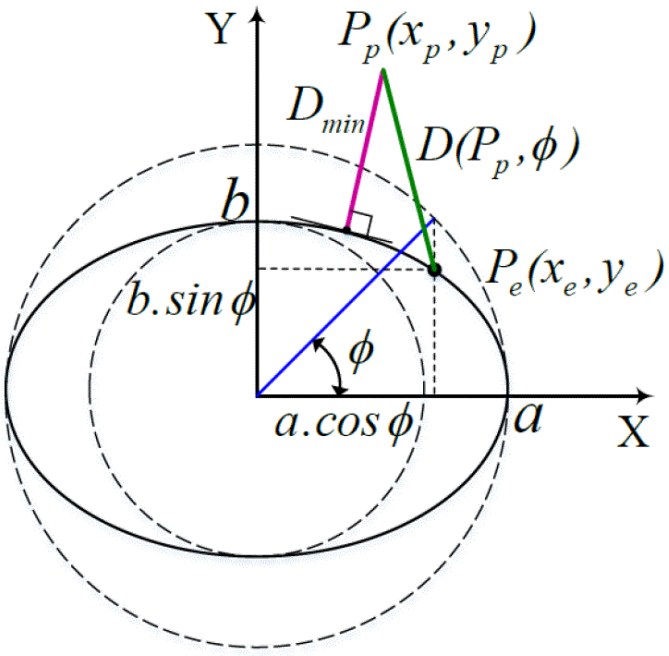


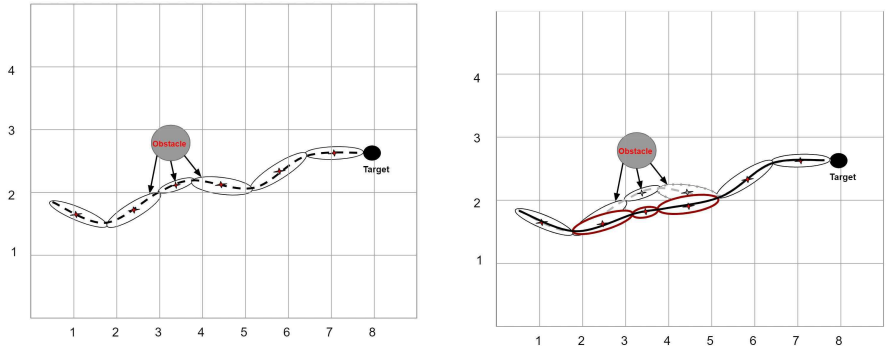
Fig. 1: Geometry for calculating the shortest distance between the robot and obstacle

4.3 Trajectory Component Adjustment

Haven computed the closest point from the obstacle point to the desired trajectory, and also the desired orientation of the fitted ellipse, this section presents the schematics for the component adjustment. The component adjustment will ensure that the robot maintains the threshold and adjust any component(s) that falls below it in the presence of an obstacle. By ensuring that only affected components are adjusted, the computational complexity is reduced and the motion of the robot will become similar to the way human being cloud have performed the same task.

The overview of component adjustment approach is illustrated in Fig. 2. Figure 2 (a) shows the presence of an obstacle within the workspace of the robot. The black dotted lines represent the optimal trajectory the robot would follow in the absence of an obstacle. Now that the environment has changed, the robot must avoid the obstacle while moving to the target point and it must maintain a close distance from the optimal trajectory. To apply our approach, we fitted the acquired trajectory with ellipses in the form of 6-GMM components and measures the distance of the obstacle to the components. It was computed that only 3-components are closer to the robot the obstacle. Thus, they are the only

components that must be adjusted in order to satisfy the task and environmental constraints. As illustrated in Fig. 2 (b), those identified components were adjusted by applying the modified AFP presented in subsection 4.4



(a) (b)

Fig. 2: Figure showing (a) the trajectory at the point of obstacle approach (b) the response of the robot on applying the component adjustment method.

4.4 Modified APF

We modified the traditional APF so we can apply it in our obstacle avoidance case. Here the focus is on the repulsive potential field as the attractive potential field has the form of the traditional attractive potential field. To find the repulsive points, the trajectory is segmented and interest is focused on the areas with potential of been influenced by the obstacle. This approach reduces the computational complexity as it only samples few points to estimate the closet point as against considering the entire demonstration. The minimum distance between the fitted ellipse and the obstacle is used to modify the repulsive force computational equation in Eq. 17. In the equation, instead of using $d(x)$, the distance between the robot and the obstacle, we used d_{min} which is the closet point from the robot to the segmented GMM component as shown in Fig. 1 and presented in Eq. 27.

$$\hat{U}_{rep}(q) = \begin{cases} \frac{1}{2}k(\frac{1}{(d_{min})} - \frac{1}{d_{th}})^2 - e^{\sigma_i 2} & : d_{min} \leq \mathrm{d}_{th} \\ 0, & : d_{min} > \mathrm{d}_{th} \end{cases} \tag{27}$$

Where σ_i represents the width of the distribution for each of the given K-component of the Gaussian.

$$\hat{F}_{net} = \hat{F}_a(q) \ + \ \hat{F}_r(q) \tag{28}$$

5 Experimental Setup

The operator is tasked to perform a realistic robot sweeping an object into a dustpan while avoiding multiple stationary obstacles within the scene. Using the robot, a user pushes a piece of rubbish denoted by T on a 14cm x 14cm table into a dustpan (G) while avoiding obstacles (o1, o2 and o3) as depicted in Fig. 3. The operator repeated this task for three time in order to collect a set of demonstrations required for the GMM/GMR.

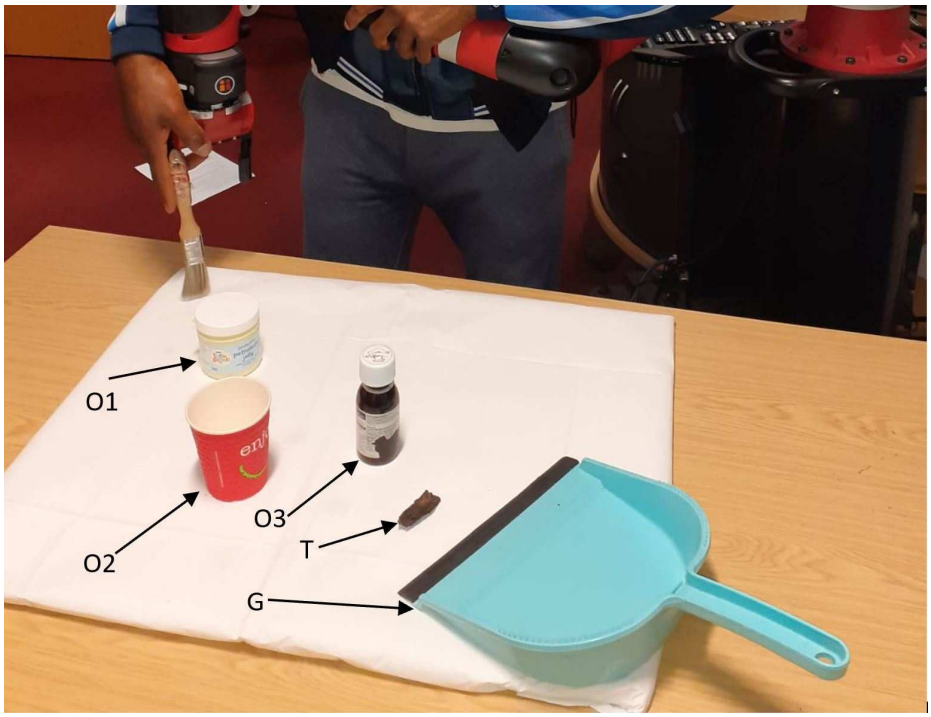


Fig. 3: In the experiment, a sample demonstration of a sweeping task is performed.

5.1 Data Acquisition and Preproccessing

The experimental data is acquired from a widely used Sawyer robot platform. The research version of Sawyer robot is compatible with ROS and integrated with a variety of sensors such as vision and force sensors. In the experiment, the Sawyer robot learns how to sweep an object into the dustpan while avoiding multiple objects (obstacles). A similar approach has been studied to allow a robot such as Roomba Vacuum Cleaning Robot to better adapt to an unseen environment [25]. The teaching process can be achieved in two ways: (i) using a remote joystick to teleoperation through the desired task (ii) using kinaesthetic

teaching. In this work, we adopt the second method which allows the human demonstrator to move the robot manipulator through the desired motion while the robot records the trajectory. The key benefits of this method are that it allows for accuracy and direct recording of the control commands, and ensures that the demonstrations are constrained to actions that are within the robotâs abilities, hence eliminating correspondence problem.

We assume that the important sensing information comes from (i) the state of the robot's end-effector which is obtained using the Sawyer robot built-in encoder, (ii) the distance between the robot and the target, and (iii) the shortest distance between the robot and the obstacles retrieved from the Kinect sensor since it can return distance information representing an absolute position of the obstacle in a specific range and we choose the shortest distance as our desired value as illustrated in Eq. 26.

5.2 Experimental Result

In order to validate the proposed learning by demonstration with obstacle avoidance approach proposed, we conducted several experiments for a robot to generate a trajectory needed to satisfy both the task constraints and the environmental constraints. The experiment is a sweeping task which involves using the robot to push a piece of rubbish denoted by T on a 14cm x 14cm table into a dustpan (G) while avoiding obstacles (o1, o2 and o3) as depicted in Fig.4 (a). From Fig. 4 (a), the robot successfully moved through the path position of the rubbish to the dustpan without colliding with any obstacles objects on the table as presented in Fig. 3. Further experiments with similar scenarios were performed and the outcome came out successful as shown in (b) - (d). Overall, the experiment proved that the proposed approach can permit successful object reaching tasks in a clustered environment.

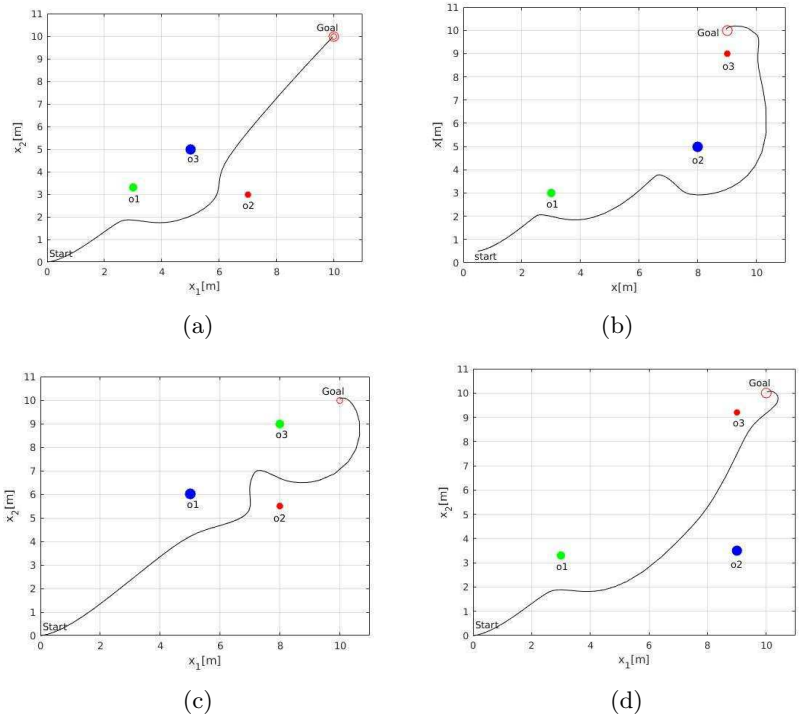


Fig. 4: Figure (a)-(d) shown samples of the reproduced trajectories. The circular points represent the position of obstacles in the environment while the start point is the origin and the goal point is defined ahead.

5.3 Conclusion

A learning by demonstration framework to enable a robot to move its manipulator from a start position to a goal position while avoiding obstacles is presented in this work. The kinesthetic teaching technique was employed to acquire motion demonstrations from the human teacher while DTW was used to align the demonstrated trajectories. The optimal trajectory is modelled using the Gaussian mixture model and Artificial potential difference. Our approach allows for the part of the demonstration prone to collision to be identified such that only a section of the demonstration will be adjusted to satisfy both the task and environmental constraints. Hence, reducing the amount to data to be left in the memory to achieve a satisfactory reproduction.

Although, obstacle avoidance is studied, there are still many problems to be further discussed. While our approach can thrive in an environment with static obstacles it is not guaranteed to perform greatly in an environment with moving obstacles. This area will be further investigated in future

References

[1] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, Oct 2011.

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

[3] Kang Shin and N. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491–500, June 1986.

[4] T. Kunz, U. Reiser, M. Stilman, and A. Verl, "Real-time path planning for a robot arm in changing environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 5906–5911.

[5] P. Leven and S. Hutchinson, "A framework for real-time path planning in changing environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 999–1030, 2002.

[6] A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri, and R. Siegwart, "An incremental sampling-based approach to inspection planning: the rapidly exploring random tree of trees," *Robotica*, vol. 35, no. 6, pp. 1327–1340, 2017.

[7] A. Zamani, J. D. Galloway, and P. A. Bhounsule, "Feedback motion planning of legged robots by composing orbital lyapunov functions using rapidly-exploring random trees," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1410–1416.

[8] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[9] Y. Yang and O. Brock, "Elastic roadmapsâmotion generation for autonomous mobile manipulation," *Autonomous Robots*, vol. 28, no. 1, p. 113, 2010.

[10] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz, "Keyframe-based learning from demonstration," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, 2012.

[11] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[12] P. Kormushev, S. Calinon, and D. G. Caldwell, "Robot motor skill coordination with em-based reinforcement learning," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 3232–3237.

[13] M. Chi, Y. Yao, Y. Liu, and M. Zhong, "Learning, generalization, and obstacle avoidance with dynamic movement primitives and dynamic potential fields," *Applied Sciences*, vol. 9, no. 8, p. 1535, 2019.

[14] C. Paxton, G. D. Hager, L. Bascetta *et al.*, "An incremental approach to learning generalizable robot tasks from human demonstration," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 5616–5621.

[15] J. Wiest, M. HÃ¶ffken, U. KreÃel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *2012 IEEE Intelligent Vehicles Symposium*, June 2012, pp. 141–146.

[16] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," *Handbook of robotics*, vol. 59, no. BOOK_CHAP, 2008.

[17] M. A. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots, "Learning generalizable robot skills from demonstrations in cluttered environments," 2018.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[19] M. Song and H. Wang, "Highly efficient incremental estimation of gaussian mixture models for online data stream clustering," in *Intelligent Computing: Theory and Applications III*, vol. 5803. International Society for Optics and Photonics, 2005, pp. 174–183.

[20] O. Arandjelovic and R. Cipolla, "Incremental learning of temporally-coherent gaussian mixture models," *Society of Manufacturing Engineers (SME) Technical Papers*, pp. 1–1, 2006.

[21] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, March 1985, pp. 500–505.

[22] H. Lin, T. Zhang, Z. Chen, H. Song, and C. Yang, "Adaptive fuzzy gaussian mixture models for shape approximation in robot grasping," *International Journal of Fuzzy Systems*, vol. 21, no. 4, pp. 1026–1037, 2019.

[23] N. Certad, R. Acuna, A. Terrones, D. Ralev, J. Cappelletto, and J. C. Grieco, "Study and improvements in landmarks extraction in 2d range images based on an adaptive curvature estimation," in *2012 VI Andean Region International Conference*. IEEE, 2012, pp. 95–98.

[24] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "An artificial potential field based mobile robot navigation method to prevent from deadlock," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 5, no. 3, pp. 189–203, 2015.

[25] T. Denning, C. Matuszek, K. Koscher, J. R. Smith, and T. Kohno, "A spotlight on security and privacy risks with future household robots: attacks and lessons," in *Proceedings of the 11th international conference on Ubiquitous computing*, 2009, pp. 105–114.