# Gaussian Mixture Model with Weighted Data for Learning by Demonstration

**A. Legeleux[a], C. Buche[b], D. Duhaut[a]**
[a]Lab-STICC, University of South Brittany, France
[b]IRL CROSSING, ENIB, Australia
legeleux@enib.fr, buche@enib.fr, dominique.duhaut@univ-ubs.fr

## Abstract

Cobots are robots specialized in collaborating with a human to do a task. These cobots needs to be easily re-programmed in order to adapt to a new task. Learning by Demonstration enables a non-expert user to program a cobot by demonstrating how to realize the task. Once the learning is done, the user can only improve the learning by adding new demonstrations or deleting existing ones. In this article, the proposed model gives the possibility to the user to impact the learning by choosing which parts of the demonstration has more importance. This model uses an extended version of Gaussian Mixture Model (GMM) with weighted data coupled with Gaussian Mixture Regression (GMR). This architecture was tested with two different tasks and with two robots. Results indicate better generated trajectory with the proposed approach.

## Introduction

Collaborative robots, or cobots, are developed to help humans in different situations such as factories, homes, or hospitals. Contrary to robots, cobots work in the same environment as a human to achieve a task together (El Zaatari et al. 2019). These cobots need to learn how to move before doing the collaborative work. Learning by Demonstration (Zhu and Hu 2018; Calinon 2009; Argall et al. 2009; Ravichandar et al. 2020; Chernova and Thomaz 2014) enables a non-expert user to easily teach a task to a cobot by demonstrating the correct movement. One possible way to demonstrate a movement is by physically moving the arm of the cobot. This type of demonstration is called kinesthetic (Zhu and Hu 2018; Calinon 2009; Argall et al. 2009; Ravichandar et al. 2020; Chernova and Thomaz 2014). The joints of the robot or the position of the end effector are directly recorded. So, there is no correspondence issue between the demonstrations and the movement's possibilities of the cobot (Chernova and Thomaz 2014). The user implicitly feels the arm's limits so the demonstration will be within the robot's workspace. Because of the added noise due to human manipulation, the kinesthetic demonstrations are not accurate (Chernova and Thomaz 2014). Therefore, doing multiple demonstrations for a single task improves the learning (Calinon 2009).



Learning by Demonstration, as its name indicates, uses demonstrations to learn. Without changes to the demonstration or with new entries, the model cannot create another learned trajectory. If the model includes weights on data, the user can choose weights to impact the learning. We propose to add the human-in-the-loop by coupling an extended version of Gaussian Mixture Model (GMM) with weighted data with Gaussian Mixture Regression (GMR) in the context of Learning by Demonstration.

The article is organized as follows. Related work is described before the presentation of the algorithms used in our approach. The proposed model and its experience with the results are described in order to arrive at the conclusion and perspectives.

## Related work

Learning by Demonstration involves a teacher who teaches, with demonstrations, how to perform a task to a robot. This learning may have two level (Calinon 2009; Chernova and Thomaz 2014): the high-level learning (symbolic learning) and the low-level learning (trajectory learning). The high-level learning extracts primitive or atomic actions of a task and created a symbolic representation of the logic sequence of atomic actions. The logic is defined by rules. This level of learning needs pre-existing atomic actions and additional data such as the position and orientation of the object or the state of the environment. In contrary, the low-level learning is based on the trajectory. This learning cannot adapt the trajectory to a new constraints such as a new position of the objects nor learning complex tasks. The learning needs only joint values of the robot. In this article, we focus on the low-level learning.

One common method to encode a movement is Dynamic Movement Primitives (DMP) (Ijspeert et al. 2013; Ijspeert, Nakanishi, and Schaal 2002; Schaal 2006; Schaal, Mohajerian, and Ijspeert 2007). This technique converts a demonstration into non-linear differential equations robust to external perturbations. DMP runs with only a single demonstration. Mixture of Motor Primitives (MoMP) (Mülling et al. 2013) joins multiple DMP to generalize a movement. Each demonstration is encoded with a DMP.

Hidden Markov Model (HMM) is a finite probabilistic state machine which splits a trajectory in states. Each state is modeled with a gaussian distribution. With HMM, the trajectory can be generated with interpolation or spline between key-points (Aleotti and Caselli 2005; Asfour et al. 2008; Billard, Calinon, and Guenter 2006; Brand and Hertzmann 2000; Calinon, Guenter, and Billard 2005; Calinon and Billard 2004) or with Gaussian Mixture Regression (GMR) (Calinon et al. 2010). Task-Parameterized Hidden Semi-Markov Model (TP-HSMM) (Pignat and Calinon 2017) uses Semi-Markov process combined with a model of parameterized tasks to adapt the movement with different environmental conditions.

Another model of Learning by Demonstration is the Gaussian Mixture Model (GMM). GMM encodes the trajectory of demonstrations with gaussian distributions of varying importance. The weight given at each gaussian creates a model of the movement with a mixture of a finite number of gaussians. The encoding of GMM extracts the constraints in time and space of the demonstrations. In combination with GMM, Gaussian Mixture Regression (GMR) generates the learned trajectory (Calinon, Guenter, and Billard 2007; Calinon and Billard 2008). Task-Parameterized GMM (TP-GMM) adds a model of parameterized tasks to deal with new environmental conditions (Calinon 2016; Rozo et al. 2016). GMM modification (m-GMM) adds an extra step to the process GMM/GMR by the modification of the means of learned GMM depending on new objects positions or obstacles. Contrary to TP-GMM, this model does not need demonstrations in various conditions.

HMM models only key-points contrary to GMM which encodes continuously the movement. GMM automatically represents the constraints of the trajectory (Calinon and Billard 2007). The motion's generation with GMR creates a smooth trajectory. In the presented approach, GMM is extended with weighted data to adapt the generated trajectory given by GMR to the user preference. The used model is based on (Gebru et al. 2016) which presented a weighted version of expectation-maximization (EM) algorithm for GMM with an application on clustering audio-visual data.

# Background

## Gaussian Mixture Model

Each demonstration has $N_j$ data points of $D$ dimension. A dataset $\xi$ with $J$ demonstrations is composed of $N$ data points with $N = \sum_{j=1}^{J} N_j$ and $\xi = \{\xi_j\}_{j=1}^{N}$. The gaussian mixture has $K$ gaussian distributions. The probability density function $P(\xi_j)$ of GMM is defined by Equation 1 (Calinon 2009):

$$P(\xi_j) = \sum_{k=1}^{K} P(k)\, P(\xi_j|k) \tag{1}$$

where the parameters $P(k)$ and $P(\xi_j|k)$ are defined by Equation 2 and Equation 3. GMM is made up of the parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ where $\pi_k$ is the prior probability, $\mu_k$ the mean vector and $\Sigma_k$ the covariance matrix of the gaussian $k$.

$$P(k) = \pi_k \tag{2}$$

$$\begin{aligned} P(\xi_j|k) &= \mathcal{N}(\xi_j; \mu_k, \Sigma_k) \\ &= \frac{1}{\sqrt{(2\pi)^D\, |\Sigma_k|}} e^{-\frac{1}{2}\left((\xi_j-\mu_k)^T\, \Sigma_k^{-1}\, (\xi_j-\mu_k)\right)} \end{aligned} \tag{3}$$

The GMM parameters are learned with the Expectation-Maximization algorithm. The parameters are initialized with K-means algorithm. At each u step, the E-step computes:

$$p_{k,j}^{u} = \frac{\pi_k^u\, \mathcal{N}(\xi_j; \mu_k^u, \Sigma_k^u)}{\sum_{i=1}^{K} \pi_i^u\, \mathcal{N}(\xi_j; \mu_i^u, \Sigma_i^u)} \tag{4}$$

$$E_k^u = \sum_{j=1}^{N} p_{k,j}^u \tag{5}$$

and the M-step:

$$\pi_k^{u+1} = \frac{E_k^u}{N} \tag{6}$$

$$\mu_k^{u+1} = \frac{\sum_{j=1}^{N} p_{k,j}^u\, \xi_j}{E_k^u} \tag{7}$$

$$\Sigma_k^{u+1} = \frac{\sum_{j=1}^{N} p_{k,j}^u\, (\xi_j - \mu_k^{u+1})\, (\xi_j - \mu_k^{u+1})^T}{E_k^u} \tag{8}$$

until the log-likelihood $\mathcal{L}(\xi_j)^u$ of Equation 9 increases of a threshold $C_1$. The EM algorithm is computed for each gaussian $k$. The BIC score is used to define the best number of gaussians.

$$\mathcal{L}(\xi_j) = \sum_{j=1}^{N} \log(P(\xi_j)) \tag{9}$$

## Gaussian Mixture Model with Weighted Data

Gaussian Mixture Model with Weighted data (W-GMM) (Gebru et al. 2016) is identical to GMM with a weight $w_j$ on each data point. A weight $w_j > 0$ represents the importance of a given data point, such as this data point is observed $w_j$ times. If $w_j = 1$, the data point has the same importance than if it did not have weight. Equation 3 becomes Equation 10 because $\mathcal{N}(\xi_j; \mu_k, \Sigma_k)^{\boldsymbol{w_j}} \propto \mathcal{N}(\xi_j; \mu_k, \Sigma_k/\boldsymbol{w_j})$.

$$\begin{aligned} P(\xi_j|k) &= \mathcal{N}(\xi_j; \mu_k, \Sigma_k/\boldsymbol{w_j}) \\ &= \frac{1}{\sqrt{(2\pi)^D\, \left|\frac{\Sigma_k}{\boldsymbol{w_j}}\right|}} e^{-\frac{1}{2}\left((\xi_j-\mu_k)^T\, \left(\frac{\Sigma_k}{\boldsymbol{w_j}}\right)^{-1} (\xi_j-\mu_k)\right)} \end{aligned} \tag{10}$$

As consequence, inside the EM algorithm, Equation 4, Equation 7 and Equation 8 become respectively Equation 11, Equation 12 and Equation 13.

$$p_{k,j}^{u} = \frac{\pi_k^u\, \mathcal{N}(\xi_j; \mu_k^u, \Sigma_k^u/\boldsymbol{w_j})}{\sum_{i=1}^{K} \pi_i^u\, \mathcal{N}(\xi_j; \mu_i^u, \Sigma_i^u/\boldsymbol{w_j})} \tag{11}$$

$$\mu_k^{u+1} = \frac{\sum_{j=1}^{N} \boldsymbol{w_j}\, p_{k,j}^u\, \xi_j}{\sum_{j=1}^{N} \boldsymbol{w_j}\, p_{k,j}^u} \tag{12}$$

$$\Sigma_k^{u+1} = \frac{\sum_{j=1}^{N} \boldsymbol{w_j}\, p_{k,j}^u\, (\xi_j - \mu_k^{u+1})\, (\xi_j - \mu_k^{u+1})^T}{E_k^u} \tag{13}$$

W-GMM has no impact on the formulation of GMR equations. Only the parameters learned by the GMM are influenced by the weights.

## Gaussian Mixture Regression

The learned GMM parameters $\mu_k$ and $\Sigma_k$ are used by the GMR to generate a trajectory. The dataset $\xi = \{\xi^I, \xi^O\}$ contains the position vectors $\xi^O$ at the time step $\xi^I$. Equation 14 gives the components of the mean $\mu_k$ and the covariance $\Sigma_k$.

$$\mu_k = \begin{bmatrix} \mu_k^I \\ \\ \mu_k^O \end{bmatrix}, \Sigma_k = \begin{bmatrix} \Sigma_k^I & \Sigma_k^{IO} \\ \\ \Sigma_k^{OI} & \Sigma_k^O \end{bmatrix} \tag{14}$$

The trajectory $\hat{\xi}$ is generated by Equation 15:

$$\hat{\xi} = \sum_{k=1}^{K} h_k \, \hat{\xi}_k \tag{15}$$

$$\text{where:} \begin{cases} h_k = \frac{\pi_k \mathcal{N}(\xi^I; \mu_k^I, \Sigma_k^I)}{\sum_{i=1}^{K} \pi_i \mathcal{N}(\xi^I; \mu_i^I, \Sigma_i^I)} \\ \\ \hat{\xi}_k = \mu_k^O + \Sigma_k^{OI} \, (\Sigma_k^I)^{-1} \, (\xi^I - \mu_k^I) \end{cases}.$$

# Contribution

## Our approach

Our model proposes an extension to existing GMM/GMR with W-GMM (Gebru et al. 2016) in place of GMM. Weights for each data are chosen by the user to improve the learned trajectory depending on important points. Without weighted data, GMM/GMR accuracy depends only on demonstrations. In this case, the learning can only be improved by adding or deleting demonstrations. If all demonstrations are efficient and the generated trajectory not accurate enough, it is impossible to further improve the learning. Whereas weighted data can improve the learning and thus the generated motion.

Data from demonstrations can be in joint mode or in cartesian mode. In joint mode, data is the position or speed of the joints of the robot. In cartesian mode, data corresponds to the position of the end effector of the robot (position and rotation). In this mode, it is common to give all data to the GMM algorithm. The number of gaussians is therefore the same for each dimension of data. We called this use of GMM, the multiple mode. In the case of joint mode, giving all joint in the algorithm may not be efficient. A joint may need a smaller number of gaussians than another one; if it is not often used for example. In this case, running new GMM for each joint may be more relevant. We called it the single mode. These two modes (single and multiple) are compared in the following experience.

## Experimentation

**Procedure** In the experience, the joints angles are recorded during demonstrations. Before the learning, the data needs to be temporally aligned and filtered. When someone teaches a movement to a robot, it is impossible to reproduce the same trajectory at the exact same speed so the temporal alignment is indispensable. A human may commit mistakes or add unwanted noise. These errors need to be removed. Without these two pre-processes, the result of the learning will be impacted because it depends only on the quality of the demonstrations. The temporal alignment is done with the proportionality in case all data is not periodic because of potential data losses. The filtering or selection of demonstrations is realized with Dynamic Time Warping (DTW) (Berndt and Clifford 1994). Three algorithms are tested:

- Baseline: GMM/GMR multiple mode (GG-mult),
- Proposition 1: GMM/GMR single mode (GG-sing),
- Proposition 2: W-GMM/GMR single mode (WGG-sing).

W-GMM/GMR was not tested with the multiple mode. In fact, this mode which is not suitable for joint values and thus poorly generated the trajectory.

The learning is difficult when data contains fast varying movement regarding the sampling frequency such as a sinusoid movement. The chosen tasks for the experience are *Wave* and *Goblet_into_Mug*, as shown in Figure 1. The task *Wave* consists of waving a greeting. The task *Goblet_into_Mug (GiM)* aims to place a goblet into a mug. The task *Wave* is done with the robots Nao and YuMi. The other task is realized only with the YuMi robot. Data from Nao is recorded at 15 Hz and 250 Hz for YuMi. For each task, ten good demonstrations are performed. Data from these demonstrations is refined using the two pre-processing explained earlier then the learning starts.

Figure 1: Experimental setup. From left to right: Nao task *Wave*, YuMi task *Wave*, YuMi task *Goblet_into_Mug (GiM)*

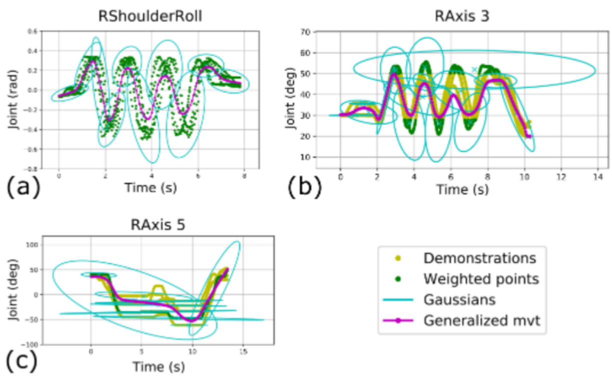


Figure 2: Weighted data on each task and the generated movement with WGG-sing. (a) Nao task *Wave*, (b) YuMi task *Wave*, (c) YuMi task *GiM*

For the WGG-sing, the weighted data are represented in Figure 2. The chosen weight, determined empirically, is different for each task. The weights chosen for each task illustrate some of the possibilities given by the weighted data. With Nao and the task *Wave*, all data are weighted with 1.15. Weighting all data points helps the learning when the sampling frequency is low. Although not intuitive, weighting all data with a value greater than one has an impact on the learning. For YuMi and the task *Wave*, the peaks of the sinusoid are weighted with 9.5. For YuMi *GiM*, the data points of a chosen demonstration get the weight of 5. Weighting a single demonstration, considered as the right one to follow, steers the learning by a specific trajectory. Except when weighting is applied on all data points or on a single demonstration, the points to weight are easily chosen by selecting points above or under a specific joint value or by choosing a time window.

**Measures** The success of a trajectory is hard to define because it is not possible to compare the generated movement with the perfect one which is unknown. The choice of the measures in the field of Learning by Demonstration is still an open challenge (Chernova and Thomaz 2014; Argall et al. 2009) The success of the learning is measured with a success rate, the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). The success rate is a score out of 100, see Table 1. The value 0 corresponds to an absolute failure and 100 to a perfect success of the task. The success rate is divided into two equal part. The first part evaluates the success of the task and the second part assesses the quality of the achievement of the task. MSE and MAE are computed between the generated movement and all demonstrations. For a single task, a single value for the MSE and MAE is computed by meaning MSE/MAE over all demonstrations.

Table 1: Detailed Success Rate

| Success rate out of 100 | |
|---|---|
| 50 | The goal of the task is successful. |
| 25 | None of the motion interferes with the task. |
| 10 | The movement is done without useless motions. |
| 10 | The robot moves during all the tasks without errors. |
| 5 | The movement is smooth. |

**Results** The results of the experience are presented in Table 2 for the success rate and in Table 3 for the MSE/MAE. The generated trajectory for each model is displayed in Figure 3 for Nao task *Wave*, Figure 4 for YuMi task *Wave* and Figure 5 for YuMi task *GiM*.

GMM/GMR with the multiple mode gives the worse results with all metrics. Indeed, this model attributes the same number of gaussians for each joint so the generated trajectory is not the optimal one. GMM/GMR with the single mode increases the results compared to the multiple mode. GMM/GMR with weighted data improves the single mode by helping the learning when needed. This model gets the best results regarding the success rate of each task. MSE and MAE are better with the weighted data model except for YuMi task *Wave*. Figure 4 shows that GMM/GMR with weighted data are the only model able to follow the sinusoid in the axis 3 of the right arm (RAxis 3). As a consequence, this model has the best success rate because it is the only one replicating the right waving sequence of the motion. With Nao robot, W-GMM/GMR single mode with weight on each data points reduces the impact of the low sampling frequency which complicated the learning specifically in case of sudden changes. The model can also weight a specific demonstration in order to steer the learning to a user chosen good demonstration, such as YuMi task *GiM*.

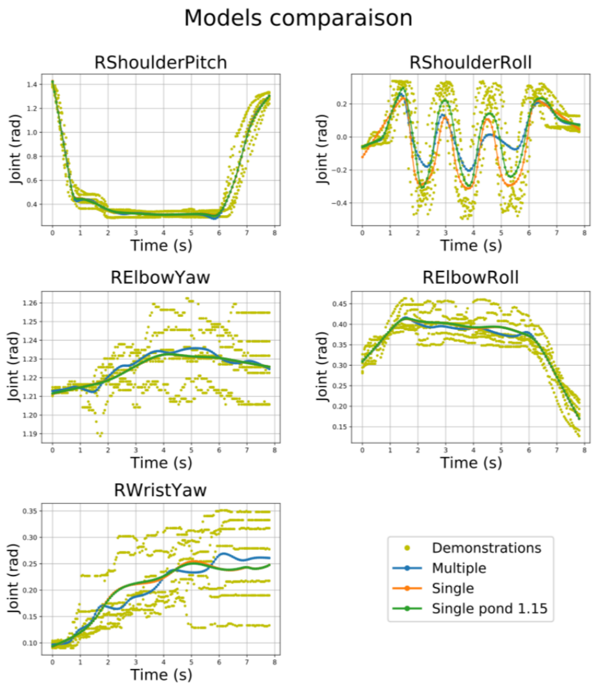


Figure 3: Result of the experience: Generated movement with Nao, task *Wave*

Table 2: Results of the experience: Success Rate

| Success Rate | Nao *Wave* | YuMi *Wave* | YuMi *GiM* |
|---|---|---|---|
| GG-mult | 75 | 38 | 35 |
| GG-sing | 92 | 45 | 40 |
| WGG-sing | **98** | **60** | **55** |

Table 3: Results of the experience: MSE and MAE

| MSE/MAE | Nao *Wave* | YuMi *Wave* | YuMi *GiM* |
|---|---|---|---|
| GG-mult | 0.0184 / 0.0556 | 0.0487 / 0.0693 | 0.0084 / 0.0390 |
| GG-sing | 0.0094 / 0.0453 | **0.0460 / 0.0668** | 0.0171 / 0.0465 |
| WGG-sing | **0.0077 / 0.0422** | 0.0684 / 0.0728 | **0.0077 / 0.0342** |

In case of data with joint values, GMM/GMR works better if it is used in single mode. The single mode can be improved by weighting some data points, chosen by the user.

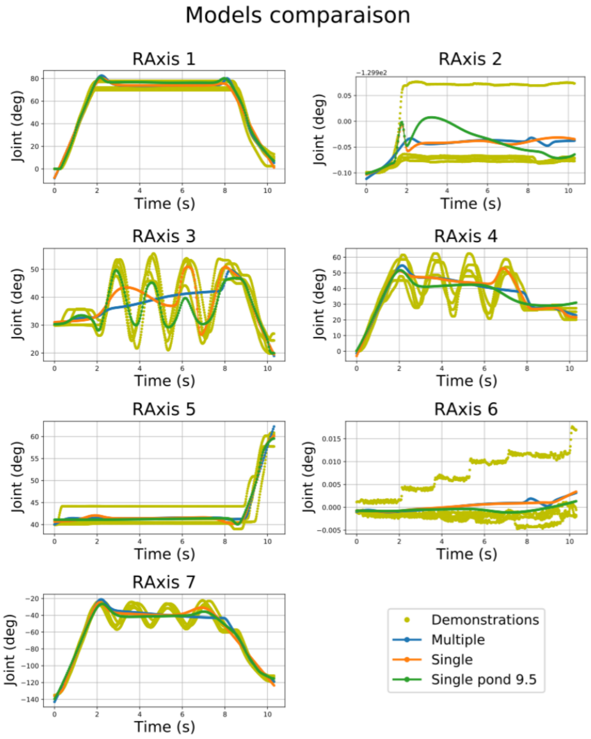


Figure 4: Result of the experience: Generated movement with YuMi, task *Wave*

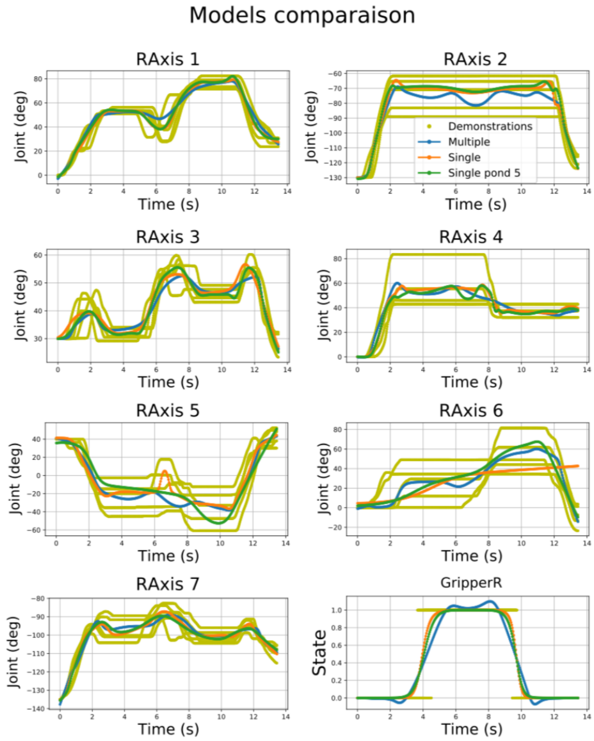


Figure 5: Result of the experience: Generated movement with YuMi, task *GiM*

## Conclusion

Learning by Demonstration becomes a good way to program a cobot by a non-expert user. The model GMM/GMR extracts the main constraints of the task from demonstration to generate a trajectory. The proposed approach extends this model with weighted data. The user chooses which data has more importance to improve the generated movement. This model has been tested on different tasks and robots. GMM/GMR has issues in modeling data containing quick changes. Using weighted data allows these curves to be learned better. The proposed approach can be further improved by adding an automatic algorithm which defines the best value of weight depending on chosen data points. This improvement will simplify the programming process of a cobot with the proposed learning approach for a non-expert user.

## Acknowledgment

This article benefited from the support of the project Prog4Yu ANR-18-CE10-0008 of the French National Research Agency (ANR).

## References

Aleotti, J., and Caselli, S. 2005. Trajectory clustering and stochastic approximation for robot programming by demonstration. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1029–1034. ISSN: 2153-0866.

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Asfour, T.; Azad, P.; Gyarfas, F.; and Dillmann, R. 2008. Imitation learning of dual-arm manipulation tasks in humanoid robots. *International Journal of Humanoid Robotics* 05(02):183–202. Publisher: World Scientific Publishing Co.

Berndt, D. J., and Clifford, J. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *AAAI-94 Workshop on Knowledge Discovery in Databases*, AAAIWS'94, 359–370. Seattle, WA: AAAI Press.

Billard, A. G.; Calinon, S.; and Guenter, F. 2006. Discriminative and Adaptive Imitation in Uni-Manual and Bi-Manual Tasks. *Robotics and Autonomous Systems* 54(5):370–384.

Brand, M., and Hertzmann, A. 2000. Style Machines. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, 183–192. New Orleans, Louisiana, USA: ACM Press/Addison-Wesley Publishing Co.

Calinon, S., and Billard, A. 2004. 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)*, volume 3, 2769–2774.

Calinon, S., and Billard, A. 2007. Incremental Learning of Gestures by Imitation in a Humanoid Robot. In *Proceed-*

*ings of the ACM/IEEE International Conference on Human-Robot Interaction*, HRI '07, 255–262. Arlington, VA, USA: Association for Computing Machinery.

Calinon, S., and Billard, A. 2008. A Probabilistic Programming by Demonstration Framework Handling Constraints in Joint Space and Task Space. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 367–372. ISSN: 2153-0866.

Calinon, S.; D'halluin, F.; Sauser, E. L.; Caldwell, D. G.; and Billard, A. G. 2010. Learning and Reproduction of Gestures by Imitation. *IEEE Robotics & Automation Magazine* 17(2):44–54. Conference Name: IEEE Robotics Automation Magazine.

Calinon, S.; Guenter, F.; and Billard, A. 2005. Goal-Directed Imitation in a Humanoid Robot. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 299–304. ISSN: 1050-4729.

Calinon, S.; Guenter, F.; and Billard, A. 2007. On Learning, Representing, and Generalizing a Task in a Humanoid Robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37(2):286–298. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics).

Calinon, S. 2009. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press.

Calinon, S. 2016. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics* 9(1):1–29.

Chernova, S., and Thomaz, A. L. 2014. Robot Learning from Human Teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8(3):1–121. Publisher: Morgan & Claypool Publishers.

El Zaatari, S.; Marei, M.; Li, W.; and Usman, Z. 2019. Cobot programming for collaborative industrial tasks: An overview. *Robotics and Autonomous Systems* 116:162–180.

Gebru, I. D.; Alameda-Pineda, X.; Forbes, F.; and Horaud, R. 2016. EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(12):2402–2415. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Ijspeert, A. J.; Nakanishi, J.; Hoffmann, H.; Pastor, P.; and Schaal, S. 2013. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation* 25(2):328–373.

Ijspeert, A. J.; Nakanishi, J.; and Schaal, S. 2002. Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation*, volume 2, 1398–1403.

Mülling, K.; Kober, J.; Kroemer, O.; and Peters, J. 2013. Learning to Select and Generalize Striking Movements in Robot Table Tennis. *The International Journal of Robotics Research* 32(3):263–279. Publisher: SAGE Publications Ltd STM.

Pignat, E., and Calinon, S. 2017. Learning adaptive dressing assistance from human demonstration. *Robotics and Autonomous Systems* 93:61–75.

Ravichandar, H.; Polydoros, A. S.; Chernova, S.; and Billard, A. 2020. Recent Advances in Robot Learning from Demonstration. *Annual Review of Control, Robotics, and Autonomous Systems* 3(1):297–330. _eprint: https://doi.org/10.1146/annurev-control-100819-063206.

Rozo, L.; Calinon, S.; Caldwell, D. G.; Jiménez, P.; and Torras, C. 2016. Learning Physical Collaborative Robot Behaviors From Human Demonstrations. *IEEE Transactions on Robotics* 32(3):513–527. Conference Name: IEEE Transactions on Robotics.

Schaal, S.; Mohajerian, P.; and Ijspeert, A. 2007. Dynamics systems vs. optimal control—a unifying view. In Cisek, P.; Drew, T.; and Kalaska, J. F., eds., *Progress in Brain Research*, volume 165 of *Computational Neuroscience: Theoretical Insights into Brain Function*. Elsevier. 425–445.

Schaal, S. 2006. Dynamic Movement Primitives—A Framework for Motor Control in Humans and Humanoid Robotics. In Kimura, H.; Tsuchiya, K.; Ishiguro, A.; and Witte, H., eds., *Adaptive Motion of Animals and Machines*. Tokyo: Springer. 261–280.

Zhu, Z., and Hu, H. 2018. Robot Learning from Demonstration in Robotic Assembly: A Survey. *Robotics* 7(2):17. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.