Kinesthetic Teaching via Fast Marching Square

Javier V. Gómez, David Álvarez, Santiago Garrido and Luis Moreno

***Abstract*— This paper presents a novel robotic learning technique based on Fast Marching Square (FM^2). This method, which we have called FM Learning, is based on incorporating previous experience to the path planning system of the robot by taking into account paths taught to the robot via kinesthetic teaching, this is, guiding manually the robot through the desired path. The method proposed ensures that the path planning is always a globally asymptotically stable system at the target point, considering the motion as a nonlinear autonomous dynamical system. The few parameters the algorithm has can be tuned to get different behaviours of the learning system. The method has been evaluated through a set of simulations and also tested in the mobile manipulator Manfred V2.**

I. INTRODUCTION

During the last years, robot configurations are becoming more and more complex, with a larger number of degrees of freedom (DOF) involved in order to get better and more natural movements. Thus, the control of the robot becomes challenging and the commonly used techniques are unuseful. This problem has been faced by means of learning techniques. The movements the robot should execute are *shown* to the robot in many different ways and the robot learns how it has to behave.

It is important to remark that in learning by imitation (also referred to as programming by demonstration) [2], the demonstrations can be provided either by observing a demonstrator doing a task or by physical guiding of the robot during the task (kinesthetic teaching). While the first method requires the system to handle the re-targeting problem, the kinesthetic teaching method simplifies the problem using the same embodiment for both demonstration and reproduction. Data acquisition during demonstration can be also dealt with markers, as proposed in [3], where a 3D point measurement system is employed to transfer the motion patterns to the robot. In [4] programming by demonstration is used by teaching a robot to carry out specific tasks such as loading the dishwasher or taking out bottles from the refrigerator.

Many different approaches have been proposed to implement the robot learning. One approach has been to use the concept of motion primitives [5]. The dynamic movement primitives (DMP) are a set of nonlinear differential equations which creates smooth control policies. These primitives are learned by means of imitation learning and reinforcement learning. A more recent approach based on the motion primitives idea is proposed in [6], where the primitives learning is carried out using incremental kinesthetic teaching by means of Hidden Markov Models (HMM).

Robotics Lab, Department of Systems and Automatics, University Carlos III of Madrid, Leganes, 28911, Spain `jvgomez@pa.uc3m.es`, {`dasanche,sgarrido,moreno`}`@ing.uc3m.es`

Other approach is to show the robot how to perform a discrete motion (i.e. point-to-point trajectories) [7], where the robot performs a movement keeping the motion as similar as possible to the demonstrations. Calinon goes a step further and it is proposed a control strategy for a robotic manipulator operating in unstructured environments while interacting with human operators [8]. Such situations are starting to be common in manufacturing.

Lastly, a different, very interesting approach is given in [9]. In this case apprenticeship learning is carried out by supposing a Markov decision process where the reward function is not explicitly given. By observing an expert developing a task, this algorithm guesses that the expert is maximizing an unknown reward function, expressible as a linear combination of known features.

All the mentioned methods are a few examples of the wide literature about robot learning. All of them have proved a good performance within their objectives, but their underlying mathematical model is usually based on probabilistic terms, causing the learning to be stochastic and even unstable under certain conditions. Besides, these approaches are not able to take into account environment conditions, since their are based on modifying motion control parameters.

In this paper we propose a novel kinesthetic teaching method based on the Fast Marching (FM) algorithm [11]. The method assumes that the task taught to the robot can be *codified* into a path planning problem, either in joint coordinates or Cartesian coordinates. One of the main advantages of the proposed method is that it is very easy to implement and very intuitive, leaving aside complex theoretical formulation. The proposed method takes into account the environment, since it modifies the path planning algorithm of the system instead of modifying the motion control.

The rest of the paper is organized as follows. In section II the Fast Marching FM and Fast Marching square (FM^2) algorithms are summarized. Following, section III details the learning algorithm proposed and contains the simulation results. Section IV shows the experiments carried out and th results obtained. Lastly, section V summarizes the paper and outlines the main conclusions.

II. FAST MARCHING AND FAST MARCHING SQUARE

Fast Marching FM method is an algorithm proposed by J. Sethian in 1996 to approximate the viscosity solution of the Eikonal equation. We will use Sethian's notation [12].

Let us assume that a wave starts propagating in $T = 0$. This wave has a generic limit called frontwave (a curve in two dimensions, a surface in three and so on) that separates the region in which the limit has arrived from the region the front has not visited yet (see figure 1). A wavefront with velocity F, non-negative, can be characterized by the arrival time T for each position $\mathbf{x}$. For one dimension:

$$x = F \cdot T \tag{1}$$

The spatial derivative becomes the gradient:

$$1 = F\frac{dT}{dx} \tag{2}$$

and the magnitude of the gradient is:

$$\frac{1}{F} = |\nabla T| \tag{3}$$

Since the gradient is orthogonal to the level set of the arrival function $T(x)$, this concept can be applied for multiple dimensions. The final expression is known as Eikonal equation:

$$|\nabla T(\mathbf{x})| F(\mathbf{x}) = 1 \tag{4}$$

As an example for two dimensions, a frontwave expanding with unitary velocity in all directions can be expressed by the level set γ_t as follows:

$$\gamma_t = \{(x, y)/T(x, y) = t\} \tag{5}$$

Figure 2 depicts this example, adding the time as a third axis for the frontwave at $T = 0, 1, 2 \ldots$.

A. The Fast Marching Square Algorithm

Let us suppose that a robot wants to move from its current position $p = (x, y)$ to a goal position $p_g = (x_g, y_g)$. If a wave with constant velocity F is expanded from p_g until it reaches p, a scalar function $T(x, y)$ is obtained. Applying gradient descent to $T(x, y)$ from p a path to p_g will be found, which is optimal according to a minimal distance criterion.

Nevertheless, the obtained trajectory is not guaranteed to be smooth and safe since it runs too close from corners, obstacles and walls. The typical solution for this problem is to enlarge the obstacles by the size of the robot, but this still does not accomplish the smoothness requirements for a good trajectory. Figure 3 plots this problem.

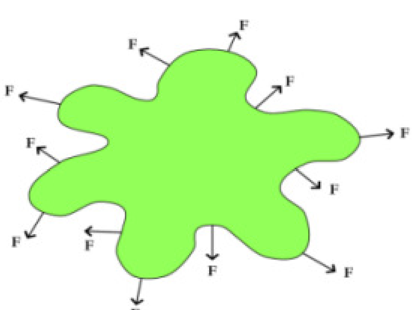


Fig. 1: Frontwave expanding with velocity F.

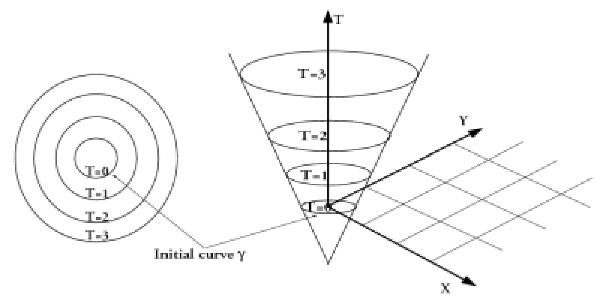


Fig. 2: Expansion over the time of a circular wavefront.

The Fast Marching square (FM^2) algorithm [13] solves this problem using a two-step wave propagation. It is conceptually close to the navigation functions of Rimon-Koditschek [14] since it uses a potential field with only one local minimum located at the goal point. However, FM^2 does not display the typical problems of the potential-based methods such as local minima, no passage between closely spaced obstacles and oscillations due to narrow passages or in the presence of obstacles [15].

The main steps of the FM^2 method are:

1) *Modeling.* An *a priori* grid-based map is created by updating the corresponding occupied cells with black and white information avoiding complex modeling.
2) *Object enlarging.* The objects detected in the previous step are enlarged by the radius of the mobile robot to ensure that the robot does not collide or accept passages narrower than its size.
3) *FM 1st step.* A wave is propagated using the FM method from all the occupied cells. The result is a potential map W represented in gray scale in which black represents walls and obstacles and the farther the cells are from them, the lighter they become. W can be interpreted as a velocity (or slowness) map $F(x)$ in eq. 4. This step could be the bottleneck of the algorithm. Its complexity is $O(n)$, where n is the number of cells of the map, which increases exponentially with the dimensions taken into account. However, this step have to be carried out only once for every environment.
4) *FM 2nd step.* The FM method is applied again to the

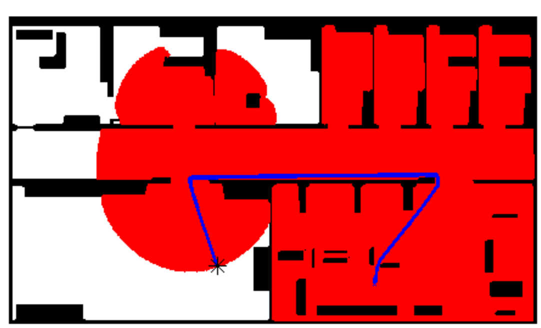


Fig. 3: Expansion over the time of a circular wavefront (constant velocity in all directions).

slowness map. The origin of the wave is the goal point and it propagates until it reaches the current position of the robot. A new potential map D is obtained. This can be interpreted as $T(x)$ in eq. 4 and it is used to calculate the trajectories using gradient descent from the robot current location.

Thanks to the laws of the electromagnetic waves propagation, we can ensure that the path obtained will be the shortest in terms of time due to the Fermat's *least time principle*. Figure 4 shows these steps where the smoothness and safeness of the path obtained is proved.

B. Key Characteristics of the Fast Marching Square Method

The key characteristics of the FM^2 method are:

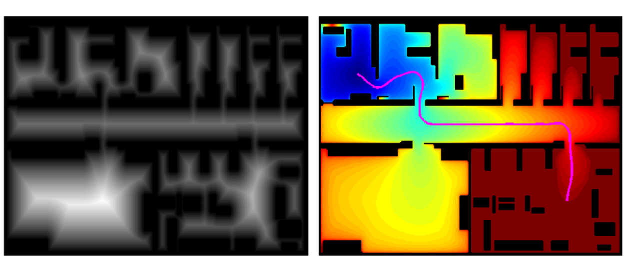
- *Absence of local minima.* $D(x)$ represents the expansion over the time of a wave in media with different velocities. As the time cannot go backwards, $D(x)$ will not have local minima since it is not possible to reach a farther place without visiting the closer places.
- *Fast response.* The planner has to be fast enough to be used reactively in case of unexpected obstacles. A simple treatment of the sensor information and a low complexity order algorithm is necessary.
- *Smooth and safe trajectories.* The trajectories do not need to be refined and they keep a safe distance to obstacles and walls.
- *Completeness.* As the method consists of the propagation of a wave, it will find a path from the initial position to the goal position, if it exists.

C. FM2 Saturated Variation

In large enough environments, it could be preferable to move the robot with constant velocity only reducing it when getting too close to obstacles or walls. Even more, it can be considered that maintaining a minimum distance to the obstacles is enough to consider the path as safe.

By saturating the slowness potential $F(x)$ at the maximum velocity allowable for the robot or at the minimum distance of the obstacles, the path will be more human-like and, in most cases, shorter (in terms of distance) than with the standard FM^2 method.

The performance of this variation is shown in figure 5, which is worthy to compare against figure 4.



(a) (b)

Fig. 4: a) Slowness potential $F(x)$. b) Second wave expansion $D(x)$ and the path obtained.

III. KINESTHETIC TEACHING WITH FM^2

The FM^2 has proved to work efficiently in path planning tasks. The results of this method only depend on the environment conditions, such as obstacles, walls or other robots. This means that the method will always give the same results when working under the same environment, without taking into account previous experience.

The objective of the learning is to introduce new data to the FM^2 algorithm to improve the motion planning modifying the slowness potential W depending on what an expert shows to the robot. This will cause that the paths given by a modified W potential will present different characteristics from those given by the standard FM^2 method. Since only W is being modified, the good characteristics of the FM^2 method remain, such as smoothness and local-minima-free.

This paper is focused on kinesthetic teaching, more precisely, guiding the robot (usually robotics manipulators or humanoid robot's arms) through the desired trajectory. During the guidance, the robot records the data and later it adapts the parameters of the underlying mathematical model, commonly based on probabilistic formulas.

When being taught, the main objective of the robot is to be able to reproduce by itself the motion learned and adapt it to new motion requirements. Also, it is expected to improve the motion taught making it smoother, more efficient, faster, etc. Hence, the objective can be translated into learning a path and adapt it when necessary. Therefore the algorithm works over the path taught and over the path planning algorithm implemented. It does not mind if the teaching is being carried out in end-effector coordinates or joint coordinates.

A. Learning Algorithm

The proposed algorithm uses data gathered during a kinesthetic teaching process. During this learning, the end-effector's positions are stored with a time cycle T. The proposed algorithm can work as well with joint-coordinates, but to make it easier to understand the algorithm we use the end-effector's Cartesian coordinates.

The proposed algorithm starts with the slowness map of the environment $\mathbf{W}$ saturated at level sat, and the set of n points obtained during the kinesthetic guiding. The algorithm works as follows:

1) Connect all the points in the same order they were stored. This connection can be done using straight lines but we recommend to use $FM2$ to take advantage of this method. Since this is considered to be done offline the computational cost is not very important. These connections are stored in a binary map $\mathbf{W_p}$.
2) Dilate $\mathbf{W_p}$ using a structuring element SE, whose size aoi defines the *area of influence* of the learned data.
3) Fast Marching method is applied to the $\mathbf{W_p}$, in order to convert it to a gray scale map. This map has to be rescaled to a maximum value of $(1 - sat)$.
4) Add the rescaled map $\mathbf{W_p}$ to the initial map $\mathbf{W}$.
5) Restore the obstacles and walls to value 0 because the previous steps could delete this information.

6) (Optional) Apply a smoothing filter in order to do not have harsh changes in the final slowness map $\mathbf{W}$.

By following these simple steps, the slowness map is modified. The new paths provided by the path planning system will be very different depending on the experience of the robot. Figure 6 shows the different steps of the algorithm. In this case the path obtained is very similar to the one taught but much is smoother.

The size of the aoi becomes very important in the performance of the motion. If this size is too small, the robot will just repeat the movement taught. On the other side, if the size is too big, the motion will differ a lot from the demonstrations. Figure 7 depicts this fact. Here, the robot has been taught 5 similar trajectories. In the first case (figure 7 a) and b)) the aoi size is 10 pixels. The path obtained after learning is just the shortest path of those taught. However, in the second case (figure 7 c) and d)) the aoi size is 30 pixels. All the paths have turned into a unique learned, wide white zone in the slowness map. The path obtained in this case can be considered as a generalization of the taught paths. The aoi size also depends on whether we are carrying out one-shot learning or with multiple demonstrations.

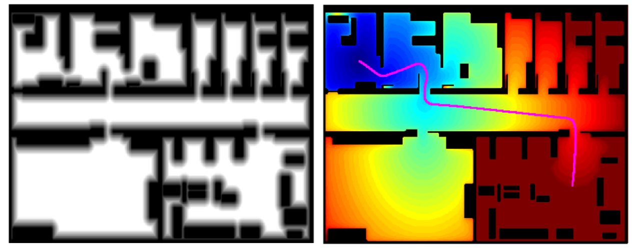
B. Stability Analysis

The motion of robots can be considered as a nonlinear autonomous dynamical system, where autonomous refers time-invariant. In this case, the proposed learning algorithm is asymptotically stable according to the Lyapunov Stability theorem [18]. This theorem expresses that a function $\dot{x} = f(x)$ is asymptotically stable at the point p_g if a continuous and continuously differentiable Lyapunov function $V(x)$ can be found such that it is always positive, its derivative is always negative and $V(x_g) = \dot{V}(x_g) = 0$.

Let us consider as Lyapunov function the one generated when expanding the second wave of FM^2, which we have called $D(x)$. This function starts at the goal point of the robot p_g, where the $D(x)$ value is 0. Given the fact that this wave expands always with non-negative velocities, the value of $D(x)$ will be higher (positive) as the wave gets farther from p_g. Finally, the derivative of the function is always negative since $D(x)$ is free of local minima.

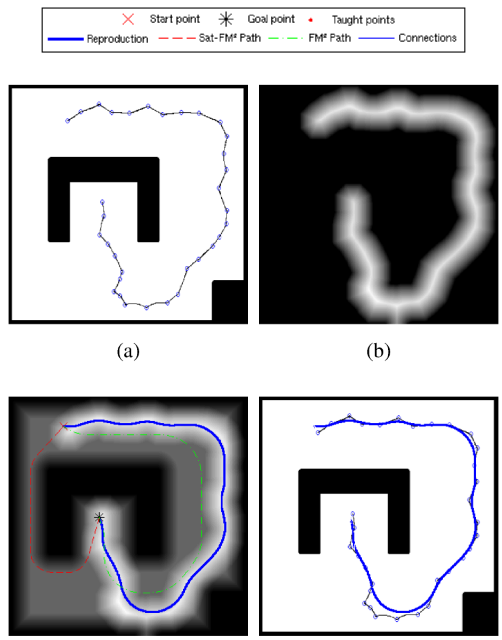
These conditions will be always satisfied, regardless the environment or even the number and shape of the given paths during the learning process. Figure 8 serves as an illustration of the aforementioned. In 8 a) it can be seen how all the possible points will converge to the destination but with the tendency of following the learned data. In 8 b) all the paths will also converge to the destination regardless the initial point. However, since the experience is very different, the robot will follow different behaviors depending on the starting point.

This algorithm has been compared against a method called *Stable Estimator of Dynamical Systems* (SEDS) [7]. In figure 9 a) the reproductions with SEDS are always similar to the taught data, this means that the streamlines are uniform. With the proposed algorithm (figure 9 b)) these streamlines converge to the goal point as with SEDS. However, if the



(a) (b)

Fig. 5: a) Saturated slowness potential $F(x)$. b) Second wave expansion $D(x)$ and the path obtained. Note that the level lines and the path are not equal than in figure 4.



(a) (b)

(c) (d)

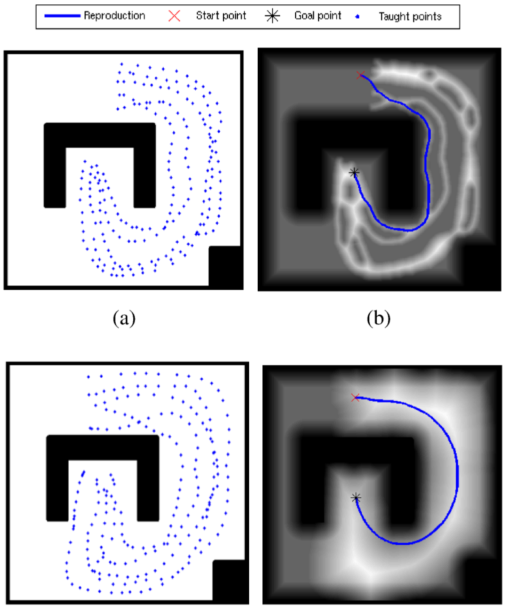
Fig. 6: Different steps of the learning algorithm. a) $\mathbf{W_p}$ first step. b) Dilated $\mathbf{W_p}$. c) Final $\mathbf{W}$ and different paths. d) Comparison between demonstration and reproduction.

starting point is out of the area of influence of the taught trajectories, the experience will not be taken into account. Therefore, it is possible to set this behaviour tuning the aoi and sat parameters. This could be considered as a drawback of the algorithm depeding on the application A further comparison among FM Learning and other methods and also a detailed study of the influence of the parameters is matter of future research.

IV. EXPERIMENTS

To prove the feasibility of the proposed learning method, it has been implemented in the mobile manipulator Manfred V2. To gather the data, the arm is placed in different positions and the Cartesian coordinates of the end-effector are stored. After, the algorithm is run and the robot performs the learned trajectory.

For better understanding, the arm is moved in a 2-dimensional plane and only the XZ coordinates of the end-



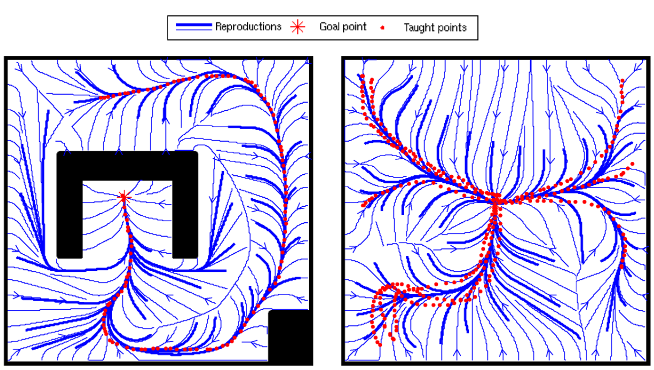
(a) (b)

(c) (d)

Fig. 7: Different paths using learned data depending on the aoi parameter in a 500x500 pixels map. a),b) $aoi = 10$ pixels. c),d) $aoi = 30$ pixels.

effector are stored. One-shoot learning is carried about. This is, the robot is taught only once since we assume that this is a desirable point by robot's end users.

With the learned data, the $D(x)$ map (figure 10 a)) converges always to the goal point, independently of the starting point of the trajectory and resembling as much as possible to the learned trajectory. Figure 10 b) compares the two trajectories carried out by the robot (with the initial taught data and with the learned data), using $sat = 0.5$ and $aoi = 35$ pixels in a 500x500 pixels region (each pixel corresponds to 1 millimeter). It is possible to see how the second trajectory adapts to the learned one and improves it, developing an smoother trajectory. Finally, figure 11 shows the robot Manfred V2 developing both motion sequences. A



(a) (b)

Fig. 8: Two examples of $D(x)$. a) $D(x)$ depends on the environment and also on the path taught. b) $D(x)$ depends only on the experience.

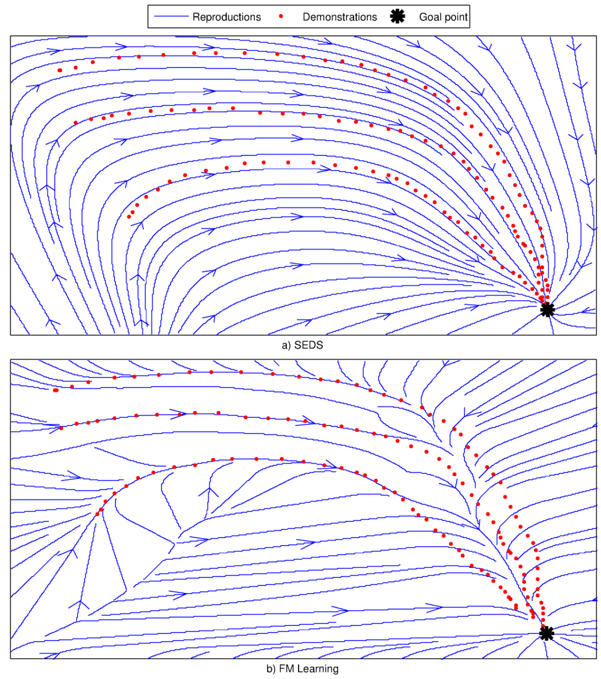


Fig. 9: Comparison between SEDS (a) and FM Learning (b).

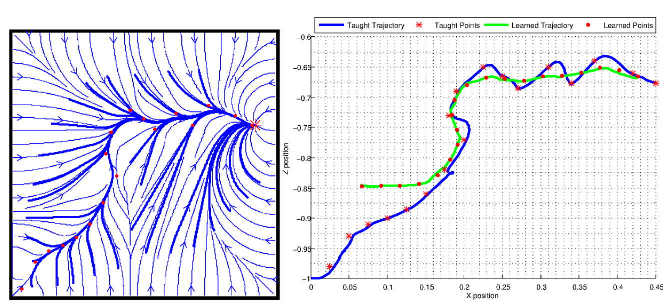
video is attached to the electronic document of this paper.

V. CONCLUSIONS AND FUTURE WORK

This paper presents a novel point of view for robot learning based on fast marching techniques. The proposed algorithm is not based on probabilistic approaches which can derive in local instabilities and non-convergences in case of a incomplete data during the learning process.

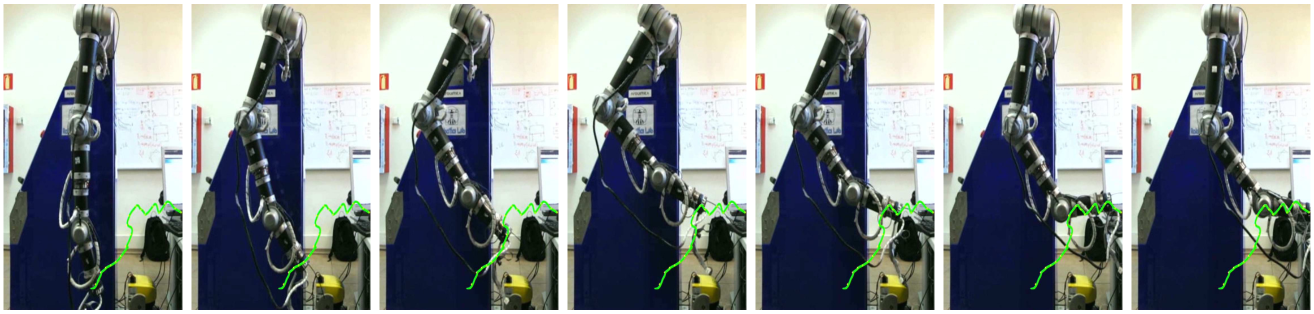
The presented method erases the stochasticity of most of the learning methods, whose results vary depending on the sequence followed when learning. The kinesthetic teaching using FM^2 ensures global stability.

In section III-A it has been discussed how the different parameters of the method change the behavior of the learning, being possible to set whether the robot will learn and follow very well the shown trajectories or if it is going to extract the main information of a set of paths, generalizing the information provided. The learning algorithm has been

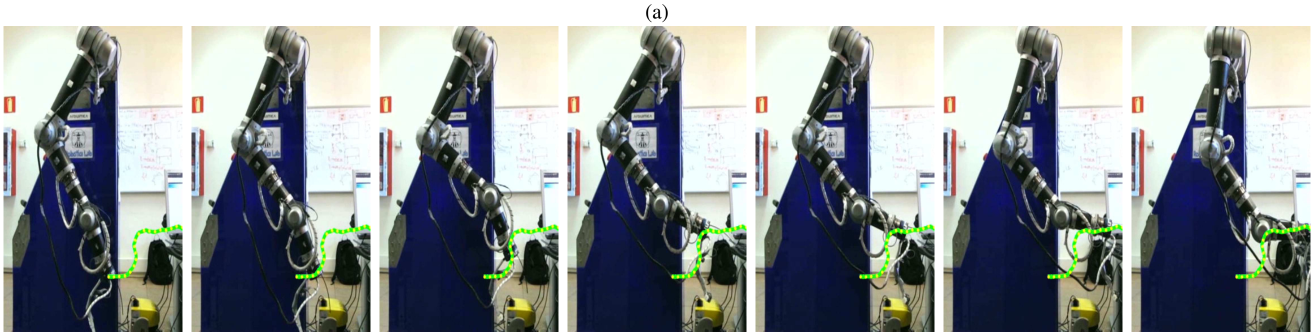


(a) (b)

Fig. 10: a) $D(x)$ map obtained from Manfred. b) Comparison between the taught trajectory and the one reproduced once the robot has learned.



(a)



(b)

Fig. 11: The robot Manfred V2 developing the trajectories. a) Taught trajectory. b) Reproduction with FM Learning starting from a different point and with almost the same goal point.

implemented in Manfred V2 to prove the feasibility of the system. The results are shown in section IV.

The paper has focused on considering end-effector coordinates. To simplify, it has been supposed that such end-effector is moving in two dimensions. Nevertheless, the proposed algorithm, since it is based on Fast Marching, can be applied to 3 or more dimensions, being possible to apply the algorithm to joint coordinates if desired.

This paper proves that Fast Marching can be applied to approach learning solutions and hence it opens a new point of view in learning techniques. Fast Marching is a very easy to implement and to understand technique. Therefore, the future work will focus on maturating Fast Marching learning techniques at the level of the current ones, including among others: different velocities to the motions taught to the robot, a forgetting factor, deeper comparisons, and so on. Another very interesting case to study is the behaviour of the proposed method under perturbations.

Acknowledgment

This work is funded by the project number DPI2010-17772 by the Spanish Ministry of Science and Innovation.

References

[1] A. Billard, S. Calinon, R. Dillmann and S. Schaal, Handbook of Robotics, *MIT Press*, 2008, ch. Robot Programming by Demonstration.
[2] S. Calinon, Robot Programming by Demonstration: A Probabilistic Approach, *EPFL/CRC Press*, 2009.
[3] C. Ott, D. Lee, and Y. Nakamura, Motion Capture based Human Motion Recognition and Imitation by Direct Marker Control, *IEEE-RAS Intl. Conf. on Humanoid Robots*, 2008, pp. 399-405.
[4] R. Dillmann, Teaching and Learning of Robot Tasks via Observation of Human Performance, *Robotics and Autonomous Systems*, 2004, pp. 109-116.
[5] S. Schaal, J. Peters, J. Nakanishi and A. Ijspeert, Learning Movement Primitives, *Intl. Symposium on Robotics Research*, 2004.
[6] A. D. Lee and C. Ott, Incremental Motion Primitive Learning by Physical Coaching Using Impedance Control, *IEEE Intl. Conf. on Intelligent Robots and Systems*, 2010, pp. 4133-4140.
[7] S. M. Khansari-Zadeh and A. Billard, Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models, *IEEE Transactions on Robotics*, 2011, Vol 27, Issue 5.
[8] S. Calinon, I. Sardellitti and D. G. Caldwell, Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies.*Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
[9] P. Abbeel, A. Y. Ng, Apprenticeship Learning via Inverse Reinforcement Learning, *Proc. of the 21st Intl. Conf. of Machine Learning*, 2004.
[10] S. Calinon and A. Billard, Incremental Learning of Gestures by Imitation in a Humanoid Robot, *ACM/IEEE Intl. Conf. on Human-Robot Interaction*, 2007, pp. 255-262.
[11] J. A. Sethian, A Fast Marching Level Set Method for Monotonically Advancing Fronts, *Proc. of the National Academy of Science*, 1996, Vol 93, Issue 4, pp. 1591-1595.
[12] J. A. Sethian , Level Set Methods, *Cambridge University Press*.
[13] S. Garrido, L. Moreno, M. Abderrahim and D. Blanco, FM2: A Real-time Sensor-based Feedback Controller for Mobile Robots, *Intl. Journal of Robotics and Automation*, 2009, Vol 24, Issue 1, pp. 3169-3192.
[14] E. Rimon and D.E. Koditschek, Exact Robot Navigation Using Artificial Functions, *IEEE Trans. on Robotics and Automation*, 1992, Vol 8, Issue 5, pp. 501-5018.
[15] Y. Koren and J. Borenstein, Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation, *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, 1991, pp. 1398-1404.
[16] 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*, 1977, Vol 39, Issue 1, pp. 1-38.
[17] S. Garrido, L. Moreno and P.U. Lima, Robot Formations Motion Planning using Fast Marching, *Robotics and Autonomous Systems*, In Press, Corrected Proof, 2011, pp. 1-25.
[18] J. Slotine and W. Li, Applied Nonlinear Control, *Prentice-Hall*, 1991.