2019 19th International Conference on Advanced Robotics (ICAR)
Belo Horizonte, Brasil, December 02-06, 2019

Dynamic Movement Primitives: Volumetric Obstacle Avoidance

Michele Ginesi, Daniele Meli, Andrea Calanca, Diego Dall'Alba, Nicola Sansonetto, and Paolo Fiorini

Abstract— **Dynamic Movement Primitives (DMPs) are a framework for learning a trajectory from a demonstration. The trajectory can be learned efficiently after only one demonstration, and it is immediate to adapt it to new goal positions and time duration. Moreover, the trajectory is also robust against perturbations. However, obstacle avoidance for DMPs is still an open problem. In this work, we propose an extension of DMPs to support volumetric obstacle avoidance based on the use of superquadric potentials. We show the advantages of this approach when obstacles have known shape, and we extend it to unknown objects using minimal enclosing ellipsoids. A simulation and experiments with a real robot validate the framework, and we make freely available our implementation.**

I. INTRODUCTION

Robots are now used in complex scenarios, ranging from industrial and manufacturing processes to aerospace and health care. As their involvement in common human tasks increases, adaptability and reliability at the motion planning level is often required, and imitation of human behavior often helps in this direction.
Standard motion planning techniques, such as splines, potentials and others [1], [2], [3], [4], work well when an objective function has to be optimized (e.g. minimize the time of execution of the trajectory, or the energy consumption). A *Learning from Demonstration* (LfD) approach, is usually preferable if one needs to learn human gestures. In LfD, a human operator shows an example trajectory or task execution, and parameters are learned for replication in different situations and environment. In last fifteen years, various LfD approaches (as *Gaussian Mixture Models* [5], *Extreme Learning Machines* [6], [7], and others [8]) have been developed in order to replicate human gestures. These LfD techniques may require a huge amount of demonstrations to be properly trained, which can represent a bottleneck when many different motion primitives have to be learned (e.g., for productive and cost reasons in industry).
In this paper we focus on Dynamic Movement Primitives (DMPs) [9], [10], [11], [12], which permit to learn a trajectory from just one demonstration. DMPs encode the trajectory in a system of second-order linear Ordinary Differential Equation (ODE) where a forcing term is learned as a linear combination of predefined time-dependent functions. They are successfully used in many robotic scenarios, such as cloth manufacturing [13], reproduction of human walk for exoskeletons [14], and collaborative bimanual tasks [15].
One open issue in the DMP framework is obstacle avoidance. This aspect has been successfully treated for point-like obstacles (e.g. [11] and [12]), while volumetric obstacle avoidance has been studied only using approaches that require multiple demonstrations with different type and sizes of the obstacles (e.g. [16], [17], [18], [19] that we will revise later).
In this work, we introduce an approach to volumetric obstacle avoidance, which does not require any additional learning process. We first learn the trajectory without obstacles. Then, we model the obstacles using superquadric potential functions [20], that allow to describe objects with generic shapes. Finally, we include the information of the obstacle in the DMP system (see Section III) which allows to successfully avoid the obstacle during the execution of the DMP. We remark that our approach can be applied both off-line (when the position and the shape of the obstacles are a-priori known) and on-line (when the obstacles are retrieved by visual information). From a computational point of view, if the geometry is not known, computing a superquadric from the point cloud or modeling the obstacle as a point cloud are computationally "equivalent". Nevertheless, if the geometry is known, our method is computationally faster (see Section IV-A and Figure 2). Moreover, our method generates more regular trajectories (see Figure 3b).
In Section II we recall the theory of DMPs and current advances in obstacle avoidance. Then, in Section III we recall the original formulation of superquadric potential functions, together with some modifications necessary to our study. In Section IV we show our results: in Section IV-A we compare the computational cost of using volumetric obstacles and point clouds; in Section IV-B we test the different behaviors of the trajectories between the two approaches; and in Section IV-C we test the effectiveness of our method using an industrial manipulator. Our code, which is freely available at `https://github.com/mginesi/dmp_vol_obst` includes a Python 3.5 implementation of DMPs and our proposed approach to volumetric obstacle avoidance.

II. DYNAMIC MOVEMENT PRIMITIVES

Dynamic Movement Primitives were firstly introduced in [9], and then modified in [11] to solve some instability issues. We now present a brief review on the formulation of DMPs. For an exhaustive description of this framework refer to [11] and [12].
DMPs are based on a system of second order ordinary

*This research has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, ARS (Autonomous Robotic Surgery) project, grant agreement No. 742671.

Department of Computer Science, University of Verona, Verona, Italy {michele.ginesi, daniele.meli, andrea.calanca, diego.dallalba, nicola.sansonetto, paolo.fiorini}@univr.it

differential equations of "spring-mass-damping" type:

$$\begin{cases} \tau\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g}-\mathbf{x}_0)s + \mathbf{K}\mathbf{f}(s) & \text{(1a)} \\ \tau\dot{\mathbf{x}} = \mathbf{v} & \text{(1b)} \end{cases}$$

where $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{v} \in \mathbb{R}^d$ are, respectively, the current position and velocity of the system. Diagonal $d \times d$ matrices $\mathbf{K} = \mathrm{diag}[K_1, K_2, \ldots, K_d]$ and $\mathbf{D} = \mathrm{diag}[D_1, D_2, \ldots, D_d]$ are, respectively, the elastic and damping terms. The parameter $\tau > 0$ is a scalar quantity, which can be used to speed up or slow down the evolution of the solution. $\mathbf{g} \in \mathbb{R}^d$ is the "goal" position, and $\mathbf{x}_0$ is the initial position.
The vector valued function $\mathbf{f} : \mathbb{R} \to \mathbb{R}^d$ is a "forcing term", used to model a desired trajectory. $\mathbf{f}$ depends on the parameter s, which is a re-parametrization of time t governed by the so-called *canonical system*:

$$\tau\dot{s} = -\alpha s, \qquad \alpha > 0. \tag{2}$$

We assume that the elastic and damping constants in (1a) satisfy $D_i = 2\sqrt{K_i}$, so that the homogeneous version of (1) (i.e. setting $\mathbf{f} \equiv \mathbf{0}$) is critically damped and has a unique attractive equilibrium $(\mathbf{x}, \mathbf{v}) = (\mathbf{g}, \mathbf{0})$. We remark that the matrices $\mathbf{K}$ and $\mathbf{D}$ are chosen to be diagonal so that each direction is independent by the others, while being maintained aligned in time by sharing the same canonical system (2).
DMP approach works as follows: a curve in $\mathbb{R}^d$ is recorded, and each component $f_j(s)$, $j = 1, 2, \ldots, d$ of $\mathbf{f}(s)$ can be computed using (1a). Then, $f_j(s)$ is written in terms of basis functions:

$$f_j(s) = \frac{\sum_{i=1}^N w_i \psi_i(s)}{\sum_{i=1}^N \psi_i(s)} s. \tag{3}$$

In (3), functions $\psi_i(\cdot)$, $i = 1, 2, \ldots, N$ are *radial basis functions*:

$$\psi_i(s) = \exp\left(-h_i(s-c_i)^2\right),$$

with:

$$c_i = \exp\left(-\alpha \tfrac{i-1}{N-1}\right),$$
$$h_i = (c_{i+1} - c_i)^{-2}, \quad h_N = h_{N-1}.$$

The weights w_i in (3) are learned, using, for example, weighted linear regression [11].
We can then reproduce the desired trajectory by numerically integrating (1). It is important to notice that the forcing term (3) is written in such a way that $\mathbf{f}(s) \to \mathbf{0}$ when $s \to 0$ (i.e. $t \to +\infty$), thus guaranteeing convergence to the desired goal position $\mathbf{g}$, while the integrated trajectory maintains a similar shape to the learned one.
In robot trajectory learning, DMPs can be used both in joint and Cartesian space. In the following, we will model the end-effector of the robot as an ideal point, [1] thus we will define the obstacle only in Cartesian space: $\mathbb{R}^d$, for $d = 2$ (planar case) and $d = 3$ (spatial case). We remark that our approach holds for any arbitrary $d \in \mathbb{N}^+$ (thus allowing to deal with obstacles directly in joint space).

[1]We will take into account the volume of the end-effector by enlarging the volume occupied by the obstacle. See Section IV-C.

A. Previous Works on Obstacle Avoidance

Obstacle avoidance for DMPs has been treated in different ways. In [16] *Stylistic* DMPs are introduced, in which instead of learning the weights w_i in (3), a probability distribution $p(w_i|\varsigma)$ is learned, where ς is a *style parameter*. This parameter can be, for example, the height of a cylindrical obstacle.
Other approaches add an additional term, called *coupling term*, $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{v})$ to (1a) obtaining

$$\tau\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g}-\mathbf{x}_0)s + \mathbf{K}\mathbf{f}(s) + \boldsymbol{\varphi}(\mathbf{x}, \mathbf{v}). \tag{4}$$

The works using this last approach differs one from the others in the way they model the coupling term $\boldsymbol{\varphi}$. In [18] and [19] a Neural Network is used to learn the coupling term from multiple human demonstrations. In [17] an analytical formulation is presented which, due to the number of free parameters, still need multiple observations (with and without the obstacle) to learn the parameters.
Approaches that do not need any learning, due to the small number of parameters that need to be tuned, are presented in [11] and [12]. This permit to have a robust method to obstacle avoidance without the need of executing more demonstration (both with and without obstacles).
However, while the "learning" approaches in [16], [17], [18], [19] can treat volumetric obstacles, approaches presented in [11] and [12] only work for point obstacles.
At the best of the authors' knowledge, our proposed approach is the first one which is able to handle volumetric obstacle avoidance without the need of any additional learning phase. Since our proposed approach falls in the "purely hand-designed" category, and due to the fact that [12] is presented as an upgrade of [11], in Section IV we will compare our approach to [12]. This approach, which works only in $\mathbb{R}^2$ and $\mathbb{R}^3$, is based on a ODE that models how humans may avoid obstacles. We start by giving the definition of *steering angle* (depicted in Figure 1):

$$\theta = \arccos\left(\frac{\langle \mathbf{o}-\mathbf{x}, \mathbf{v}-\dot{\mathbf{o}}\rangle}{\|\mathbf{o}-\mathbf{x}\|\ \|\mathbf{v}-\dot{\mathbf{o}}\|}\right), \tag{5}$$

where $\mathbf{o}$ is the position of the obstacle, $\dot{\mathbf{o}}$ is its velocity, and $\|\cdot\|$ denotes the Euclidean norm with respect to the standard scalar product $\langle\cdot,\cdot\rangle$ in $\mathbb{R}^d$. The evolution of the steering angle θ is modeled as follows:

$$\dot{\theta} = \gamma\theta\exp(-\beta|\theta|),$$

where γ and β are positive constants. Then, the coupling term $\boldsymbol{\varphi}(\mathbf{x}, \mathbf{v})$ is written as

$$\boldsymbol{\varphi}(\mathbf{x}, \mathbf{v}) = \gamma\mathbf{R}(\mathbf{v}-\dot{\mathbf{o}})\theta\exp(-\beta\theta), \tag{6}$$

where $\mathbf{R}$ is the rotation matrix of angle $\frac{\pi}{2}$ w.r.t. the axis generated by $(\mathbf{o}-\mathbf{x}) \times \mathbf{v}$, where $\times$ is the cross product in $\mathbb{R}^3$. Multiple obstacles are treated by simply summing the contributions of each one of them.

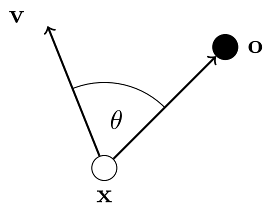


Fig. 1: Scheme of the steering angle θ as defined in (5), where $\mathbf{x}$ denotes the current position, the vector $\mathbf{v}$ the current velocity, and $\mathbf{o}$ the obstacle.

III. SUPERQUADRIC POTENTIAL FUNCTIONS

When modeling solid obstacles in the environment, the shape of the surrounding potential field is crucial. Two opposite objectives must be satisfied. First, the potential has to enclose the whole object in order to avoid collision during the motion; second, the potential shall not be oversized with respect to the obstacle, to avoid strong reduction of the available robot workspace. If the object to be modeled has a smooth standard shape (e.g., a sphere or an ellipsoid), the obstacle's potential function can be easily derived. However, when sharp edges are involved (e.g., in a cube), a function which properly surrounds the object while preserving smoothness must be designed. To this purpose, superquadric potential functions are used in this work.

Let m and n be natural numbers, and let $f_i(\mathbf{x}), i = 1, 2, 3$ be real-valued functions. We can define the *isopotential* function

$$C(\mathbf{x}) = \left(\left(\frac{x_1}{f_1(\mathbf{x})}\right)^{2n} + \left(\frac{x_2}{f_2(\mathbf{x})}\right)^{2n}\right)^{\frac{2m}{2n}} + \left(\frac{x_3}{f_3(\mathbf{x})}\right)^{2m} - 1 \tag{7}$$

that vanishes on the surface of a *generalized ellipsoid*. By tuning the parameters m, n and the functions f_i, $i = 1, 2, 3$, we can model obstacles of any shape (their boundary would be the zero-level set of (7)).

The superquadric potential function for the obstacle is:

$$U(\mathbf{x}) = \frac{Ae^{-\eta C(\mathbf{x})}}{C(\mathbf{x})}, \tag{8}$$

where A and η are positive parameters to be tuned. Note that potential function (8), see [21], is infinite on the surface of the generalized ellipsoid, thus preventing the robot from colliding with the enclosed obstacle. Moreover, the exponential factor determines a faster decrease of the potential function when away from the object, which is useful for saving more volume in the workspace.

Superquadric functions are currently widely used in obstacle avoidance thanks to their capability to adapt to completely generic shapes, see for example [22], [23], [24].

In our experiments, in Section IV, we will focus on the particular case in which $m = n$ and the f_i's in (7) are constant, obtaining the isopotential function

$$C(\mathbf{x}) = \left(\frac{x_1 - \hat{x}_1}{a}\right)^{2n} + \left(\frac{x_2 - \hat{x}_2}{b}\right)^{2n} + \left(\frac{x_3 - \hat{x}_3}{c}\right)^{2n} - 1. \tag{9}$$

The coupling term in (4) is given by

$$\boldsymbol{\varphi}(\mathbf{x}, \mathbf{v}) \equiv \boldsymbol{\varphi}_S(\mathbf{x}) \doteq -\nabla_{\mathbf{x}} U(\mathbf{x}), \tag{10}$$

where $U(\mathbf{x})$ is defined as in (8).

In Section IV-C, we will use a generalization of (9) by considering different exponents for each dimension, obtaining

$$C(\mathbf{x}) = \left(\frac{x_1 - \hat{x}_1}{a}\right)^{2m} + \left(\frac{x_2 - \hat{x}_2}{b}\right)^{2n} + \left(\frac{x_3 - \hat{x}_3}{c}\right)^{2p} - 1. \tag{11}$$

A. Minimal Enclosing Ellipsoid

When the obstacles presents locally flat boundary, as a parallelepiped, it can be more convenient to reduce the available workspace allowing for more rounded superquadric. Indeed, a locally flat isopotential increases the risk of being caught in a local minimum since the potential pushes along the normal of the surface. Therefore, it is sometimes more convenient to use classical ellipsoid ($n = 1$) when modeling obstacles in the workspace.

We now show how to create an ellipsoid that encloses a parallelepiped, while minimizing the difference of volume between the ellipsoid and the parallelepiped. Let us consider a cube of unitary edge length $\ell = 1$ with its center in the origin. The sphere containing it has radius $r = \frac{\sqrt{3}}{2}$:

$$x_1^2 + x_2^2 + x_3^2 = \frac{3}{4}. \tag{12}$$

It could be shown that a rescaling of three sphere axis is sufficient for obtaining the minimal enclosing ellipse. Suppose the edges of the parallelepiped measure, respectively ℓ_{x_1}, ℓ_{x_2} and ℓ_{x_3}. We obtain, by rescaling (12)

$$\left(\frac{x_1}{\ell_{x_1}}\right)^2 + \left(\frac{x_2}{\ell_{x_2}}\right)^2 + \left(\frac{x_3}{\ell_{x_3}}\right)^2 = \frac{3}{4}.$$

We can then define $C(\mathbf{x})$ as in (9):

$$C(\mathbf{x}) = \left(\frac{x_1 - \hat{x}_1}{\frac{\sqrt{3}}{2}\ell_{x_1}}\right)^2 + \left(\frac{x_2 - \hat{x}_2}{\frac{\sqrt{3}}{2}\ell_{x_2}}\right)^2 + \left(\frac{x_3 - \hat{x}_3}{\frac{\sqrt{3}}{2}\ell_{x_3}}\right)^2 - 1.$$

Remark 1: All the results presented in this Subsection can be easily restricted to the 2D case.

IV. RESULTS

To validate our approach we perform the following tests: in Section IV-A, we discuss the computational cost of our approach; in Section IV-B we make a synthetic experiment; and in Section IV-C we validate our approach on an industrial manipulator.

A. Computational Time

In this Section we analyze the computational cost of our approach in the worst case scenario, i.e. in the case in which an obstacle of a generic, not-known shape, is retrieved by a point cloud. To do so we generate a uniformly-distributed set of points on a generic shape, and evaluate the computational time for two different approaches. In the first approach,

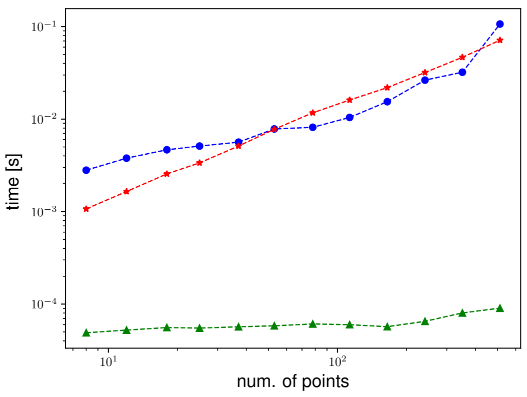


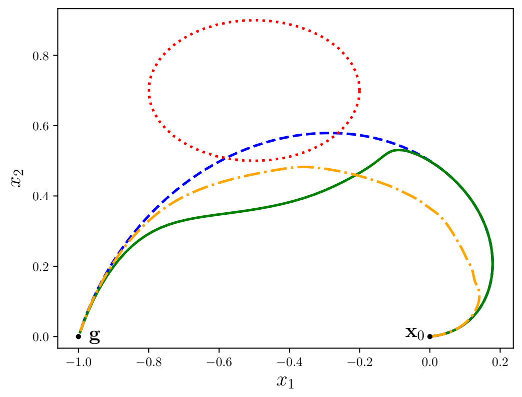
Fig. 2: Comparison of the computational time (in seconds) as function of the number of points in the point cloud. The blue circle-marked line represent the computational time needed to extract the minimal-volume ellipsoid from the point cloud and using it as an obstacle. The red star-marked line represent the cost of treating each element of the point cloud as a point obstacle. The green triangle-marked line shows the cost of computing the potential of the ellipsoid without extracting it from the point cloud.

we compute the minimal-volume enclosing ellipsoid using the algorithm presented in [25], and use it as an obstacle giving the perturbation term as defined in (10). In the second approach, we treat each point as an obstacle and compute the sum of each coupling term as in (6). Figure 2 shows that there is no meaningful difference in the computational time of the two approaches. However, we observe that our approach is more convenient if the obstacle is static or if it is moving with a known velocity. Indeed, in such cases the enclosing ellipsoid can be computed once for all, and the only computational time added by the presence of the obstacle at each time step is the evaluation of the expression (10). On the other hand, in the "steering angle" strategy (6), the coupling term must be computed for all the points of the cloud at each time step.

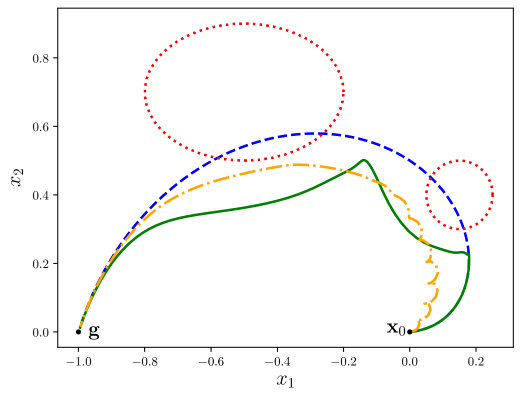
B. Synthetic Experiments

We tested the coupling term (10) in two dimensions. In Figure 3 two examples are shown where it is clearly visible the change of trajectory caused by the presence of the obstacles. For comparison we show also the trajectory obtained using (6) as coupling term. Since this last approach works only for point obstacles, we created a mesh of points on the boundary of the obstacle, and treated each point of the mesh as a point obstacle.

Both approaches are able to avoid the obstacle. However, the "steering angle strategy" (6) has two drawbacks. First, in the example shown in Figure 3a, 50 points are needed in order to properly model the ellipse (with fewer the obstacle was not avoided). Second, formula (6) does not depends on the distance from the obstacle, thus giving the same "importance" to every obstacle in the scene. As shown in Figure 3b, this leads to unexpected behaviors (the oscillations in the first part of the trajectory).



(a) One obstacle.



(b) Two obstacles.

Fig. 3: Examples of obstacle avoidance in two dimensions, with one (Figure 3a) and two (Figure 3b) obstacles. The parameters for the DMP are $\mathbf{x}_0 = [0, 0]$, $\mathbf{g} = [-1, 0]$, $K = 1050$, $D = 2\sqrt{K}$, $\tau = 1$, and $\alpha = 3$; while the parameter for the potential are $A = 50$, and $\eta = 1$. The dashed blue line represents the learned trajectory, the dotted red ellipses represent the boundary of the superquadric. The solid green line represents the adaptation of the trajectory to the presence of the obstacles using the perturbation term in (10), while the dashdotted orange line represents the trajectory obtained using the perturbation term in (6).

We emphasize that, as proved in [12], the "steering angle approach" does not have undesired (i.e. different from the goal $\mathbf{g}$) local minima, while with superquadric potentials we do not have the same property. However, if a local minimum is reached, a perturbation term which drives the trajectory out of it can be easily added in the DMP formulation.

C. Experiments on Real Setup

In order to test the Cartesian DMP path planner with volumetric obstacles with a real robot, the setup shown in Figure 4 is used. It consists of a round base (approximately

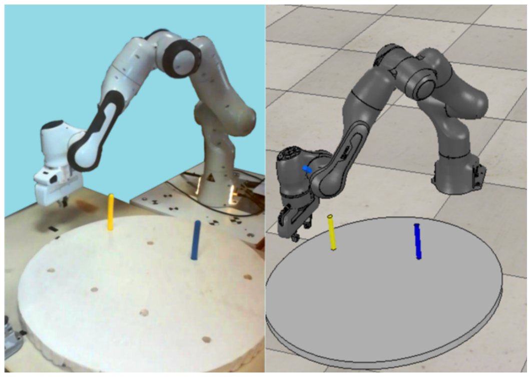


Fig. 4: Experimental setup with the Panda robot and the peg base (left), and its V-REP simulation (right).

DMPs		Obstacle	
Parameter	Value	Parameter	Value
K	3050	A	50
$D = 2\sqrt{K}$	≈ 110.45	η	1
τ	1	$n = m$	1
α	3	p	2
		a	$2\ \mathrm{cm}$ (test 1), $3\ \mathrm{cm}$ (test 2)
		b	$9\ \mathrm{cm}$ (test 1), $2\ \mathrm{cm}$ (test 2)
		c	$\approx 7\ \mathrm{cm}$

TABLE I: DMPs (left) and obstacle (right) parameters.

of radius of $25\,\mathrm{cm}$) with two pegs (approximately of radius of $1\,\mathrm{cm}$ and height $12\,\mathrm{cm}$) and a 7-DOF Panda industrial manipulator by Franka Emika. The robot, controlled using a ROS architecture and MoveIT interface, must move above the whole base at a fixed height, while avoiding the pegs on its way. Since we are mainly interested in the obstacle avoidance problem, no manual trajectory is learned; the forcing term for DMPs in (1a) is set to zero, so that the dynamic system naturally evolves towards the goal in a linear way, according to the elastic and damping terms. This choice does not limit the generality of the proposed approach, since the perturbation term guarantees the obstacle avoidance independent on the weights of the forcing term. Due to the big size of the hand of the manipulator (approx. $18\,\mathrm{cm}$ long and $5\,\mathrm{cm}$ wide) with respect to the pegs, avoiding obstacles while passing very close to them is a challenge which can be solved in two ways.

At first, for simplicity, the orientation is kept constant along the whole trajectory, neglecting the problem of adjusting the end effector's orientation to avoid collisions. The cylindric shape of the peg is modeled using the superquadric formulation in (11), but it must be modified in such a way that the base is as large as the hand of the robot to avoid collisions because the DMP approach models the moving part as a point mass. To do so, we cover the obstacle with an ellipsoid superquadric which has as radii the semi-axes of the end-effector. As it can be seen in Figure 5a, this solution ensures the obstacle avoidance but it strongly reduces the available workspace for the robot, which is undesired when many objects are in the scene and cancels the advantage in using superquadric potential functions.

For this reason, in the second test the orientation adaptation of the end effector in order to avoid obstacles is introduced. This allows to restrict the size of the superquadric potentials, since the obstacle can be enlarged only by the smaller semi-axis of the end-effector (for safety reasons) and the orientation will then adapt to avoid the collision. Orientation adaptation is managed via Trac-IK [26], a modern efficient inverse kinematics plugin for generic industrial and humanoid manipulators. Given the target position computed by the DMP planner, multiple corresponding joint configurations are possible, due to the redundancy of the robot. Given the models of the obstacle and the robot as sets of voxels, Track-IK first performs random search to only consider joint values which avoid the intersection between them. The random increment of the joint values is limited by a step parameter which depends on the size of the obstacles. Here, it is chosen as half of the peg's radius, since the peg is the smallest object in the scene. Finally, Trac-IK selects the optimal configuration which maximizes manipulability, measured as $\sqrt{\det(\mathbf{J}\cdot\mathbf{J}^{\mathsf{T}})}$, being $\mathbf{J}$ the Jacobian matrix. This is an arbitrary optimization objective, others are possible (e.g., minimal joint displacement). In order to limit the search space of the algorithm, a bound of 45° on each joint is imposed, except for the hand joint, which is crucial for obstacle avoidance (90° bound).

Figure 5 shows the results of these experiments. Figure 5a-5b show the experiment maintaining a fixed orientation of the end-effector. In particular, Figure 5a shows the top view highlighting how the original trajectory would make the end-effector collide with the obstacles, while with the perturbed one the obstacles are avoided. Figure 5c shows the experiment with the orientation adaptation. It can be seen that the DMP path passes very close to the obstacles, but the end-effector is still able to avoid them successfully (see attached video for clarity).

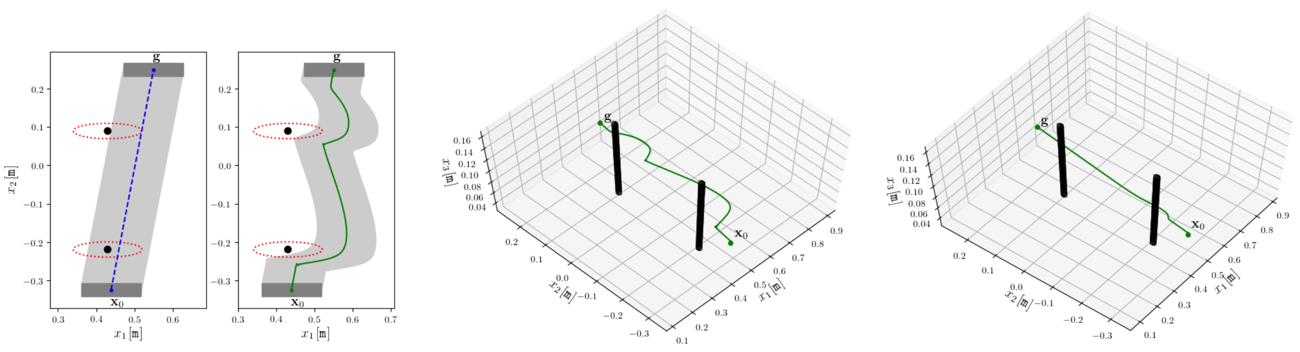
The parameters of the DMPs and the superquadric potentials for both tests are shown in Table I, with $c = \frac{h}{2}\,(2)^{\frac{1}{2p}}$, where h is the peg's height, from literature.

As a final test, in the attached video we also show how the presented approach can be implemented online when unknown obstacles with generic shapes are found in the scene, using least-squares fitting to compute the minimal enclosing ellipsoid from the camera point cloud.

V. CONCLUSIONS

In this work, we investigated volumetric obstacle avoidance for the DMP framework. To handle volumetric obstacle avoidance, we have modified the standard formulation of superquadric potential functions for efficient modeling of generic 3D shapes, and we have introduced it into the framework of DMPs. We have shown the advantages of our method both in simulation and with a real robot when known obstacles are in the scene.

In the future, we aim to extend our approach of obstacle avoidance to quaternion space DMPs, so as to automatically account for the orientation adaptation when passing near objects in the scene at the path planning level, without invoking



(a) No orientation adaptation: top view. (b) No orientation adaptation. (c) Orientation adaptation.

Fig. 5: Experimental results on real robot. In all three pictures the pegs are colored in black and the green full line shows the adaptation of the trajectory to the presence of the obstacles. In the first plot we show also the enlarged potential in dashed red and the free (no obstacles) trajectory in dashed blue, together with the gray shade showing the area occupied by the end-effector during the movement.

time-consuming IK optimization algorithms. Moreover we will test it with more, and possibly moving, obstacles in order to reproduce a more realistic scenario for collaborative or medical tasks.

References

[1] C. Lin, P. Chang, and J. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Transactions on automatic control*, vol. 28, no. 12, pp. 1066–1074, 1983.

[2] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, Oct 1992.

[3] E. Magid, D. Keren, E. Rivlin, and I. Yavneh, "Spline-based robot navigation," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 2296–2301.

[4] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 489–494.

[5] 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, 2011.

[6] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[7] J. Duan, Y. Ou, J. Hu, Z. Wang, S. Jin, and C. Xu, "Fast and stable learning of dynamical systems based on extreme learning machine," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–11, 2017.

[8] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2011, pp. 602–607.

[9] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1398–1403.

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

[11] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*. IEEE, 2008, pp. 91–98.

[12] H. Hoffmann, P. Pastor, D. H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 2587–2592.

[13] R. P. Joshi, N. Koganti, and T. Shibata, "Robotic cloth manipulation for clothing assistance task using dynamic movement primitives," in *Proceedings of the Advances in Robotics*. ACM, 2017, p. 14.

[14] R. Huang, H. Cheng, H. Guo, Q. Chen, and X. Lin, "Hierarchical interactive learning for a human-powered augmentation lower exoskeleton," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 257–263.

[15] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 816–830, Aug 2014.

[16] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning stylistic dynamic movement primitives from multiple demonstrations," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. Citeseer, 2010, pp. 1277–1283.

[17] A. Rai, F. Meier, A. Ijspeert, and S. Schaal, "Learning coupling terms for obstacle avoidance," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 512–518.

[18] A. Rai, G. Sutanto, S. Schaal, and F. Meier, "Learning feedback terms for reactive planning and control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2184–2191.

[19] G. Sutanto, Z. Su, S. Schaal, and F. Meier, "Learning sensor feedback models from demonstrations via phase-modulated neural networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1142–1149.

[20] R. Volpe, "Real and artificial forces in the control of manipulators: theory and experiments," Ph.D. dissertation, PhD thesis, Carnegie Mellon University, Department of Physics, 1990.

[21] H. Yukawa, "On the interaction of elementary particles. i," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 17, pp. 48–57, 1935.

[22] V. Perdereau, C. Passi, and M. Drouin, "Real-time control of redundant robotic manipulators for mobile obstacle avoidance," *Robotics and Autonomous Systems*, vol. 41, no. 1, pp. 41–59, 2002.

[23] A. Badawy and C. McInnes, "Separation distance for robot motion control using superquadric obstacle potentials," in *International Control Conference, Glasgow, Scotland, paper*, no. 25, 2006.

[24] H. Wen, T. Chen, D. Jin, and H. Hu, "Passivity-based control with collision avoidance for a hub-beam spacecraft," *Advances in Space Research*, vol. 59, no. 1, pp. 425–433, 2017.

[25] N. Moshtagh *et al.*, "Minimum volume enclosing ellipsoid," *Convex optimization*, vol. 111, p. 112, 2005.

[26] P. Beeson and B. Ames, "Trac-ik: An open-source library for improved solving of generic inverse kinematics," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 928–935.