Dynamic Movement Primitives-Learning Movement from Demonstration and Obstacle Avoidance with Potential Fields

Yang Wang, Min Wu

***Abstract*— Dynamic Movement Primitives (DMPs) provide a general framework for movement generation and reproduction. This framework presents a flexible mathematical formulation capable to produce complex rhythmic (periodic) and discrete movements by learning from demonstrated motion behaviors. In this paper, we study the DMPs framework for movement reproduction and generalization and propose an online obstacle avoidance strategy integrated into the DMPs formulation. The obstacle avoidance is achieved by combining a dynamic potential field with the conventional DMPs framework. The reproduction and the generalization ability of the DMPs are demonstrated via simulations using a recorded arm movement presented by a human demonstrator. The performance of the online obstacle avoidance integrated into the DMPs framework is presented in a simulation.**

I. INTRODUCTION

rhythmic (e.g., locomotion) or discrete (e.g., hand-writing) Generation of complex movements has long been a key topic in the study of robotics. Traditional movement generation approach works by deriving a mathematical model of the desired movement and applying it to robotic systems. Though this method is efficient and straightforward for simple movement generation, it can lead to hard effort to deliver a dynamic model for complex actions applied in various applications such as bionic robotics, trajectory planning. Learning motor skills from human behavior is an essential component in the human-robot cooperation study, in which robots can imitate human motion activities to achieve a smooth and nature cooperation experience. However, directly deriving exact motion equations of complex human motion actions is tedious and error-prone and lacks of generalizability. To overcome the complex motion generation problem, a framework of Dynamic Movement Primitives (DMPs) was proposed by [1]. This method provides a simple way to describe rhythmic (e.g., locomotion) and discrete (e.g., hand-writing) motor motion by learning from a demonstrated movement and enjoys a theoretically guaranteed convergence. Through extracting characteristics of an observed movement, the DMPs framwork is capable to reproduce a motion trajectory resembling the demonstrated movement and can generalize it into a new scenario with the similar movement patterns as the observed action. The DMPs presents a recorded movement by using a set of differential equations including a linear second-order dynamic system and an additional nonlinear force term characterising the demonstrated movement. This framework has several significant advantages. Firstly, a generated trajectory affected by perturbations occurred during the pre-planned movement can be automatically corrected by the intrinsic system dynamics, i.e. robustness against perturbations. Secondly, the DMPs yields a high convenience for generalization of the learned movement into a new context with different time and spatial characteristics, thanks to the convenient formulation of motion description in which adaptations in time and spatial scaling can be easily achieved by changing the parameters of equations. Thirdly, as long as the linear dynamic system is stable, the convergence to a pre-specified goal position is guaranteed.

In real-world applications, a problem often occurred is how to reproduce a learned movement in the presence of an obstacle in a pre-planned trajectory. A common online obstacle avoidance strategy is the potential field method [2], [3], [4]. In this method, a potential field is created around the obstacle and the gradient of the potential field is utilized as a repellent force to drive the robot away from the obstacle. This approach is frequently prescribed in the applications of the motion planning of mobile robot and the robotic manipulators.

The DMPs framework has a nature that the system behaviors are driven by a learned nonlinear acceleration term in the differential equations, which shares the same idea of the potential field method in applications of trajectory planning. Motivated by the similarity between two approaches, we propose to combine the representative power of the DMPs and the applicability of the potential field method in obstacle avoidance by adding an additional acceleration term to the DMPs framework as a repellent force.

The original potential field method for obstacle avoidance is static, i.e., the repellent force is generated purely using the distance between the robot and the obstacle. This may introduce sudden changes in the driving force of dynamic systems when the robot moves across the boundary of the potential field and in reality it may even exceed the limitation of the motor capability since it does not consider the robot's current movement speed and direction. To produce a smoother obstacle avoidance trajectory, a dynamic potential field method was introduced [5], which takes the relative velocity between the robot and the obstacle into account and generates a smooth adjustment term added into the driving force of the system.

In this paper, we illustrate the framework of DMPs for movement learning and reproduction, and conduct several simulation experiments using a 3D moving trajectory recorded from a human demonstration for reproduction on a robotic arm. We also demonstrate the obstacle avoidance performance of the DMPs framework in combination of the dynamic potential field.

The rest of this paper is structured as follows. In Section

II, we explain the dynamic movement primitive framework and the modified form for online obstacle avoidance. Section III describes the potential field method for online obstacle avoidance. Section IV presents the simulation experiment demonstrating the movement reproduction and the effect of dynamic potential field in obstacle avoidance. We conclude this paper in Section V.

II. DYNAMIC MOVEMENT PRIMITIVES

This section briefly explains the framework of the dynamic movement primitives, discuss the movement generalization to new goal positions, and describes how a dynamic potential field is integrated to the DMPs framework for obstacle avoidance.

A. Original Dynamic Movement Primitives

Dynamic movement primitives can be used to generate discrete and rhythmic movements. In this study, we focus on the discrete movements. An one-dimensional movement is generated by combining the following set of differential equations[1], which can be interpreted as a linear spring-damper system perturbed by an external force:

$$\tau\dot{v} = K(g-x) - Dv + (g-x_0)f \tag{1}$$

$$\tau\dot{x} = v \tag{2}$$

where x and v are position and velocity of the one degree-of-freedom system; x_0 and g denote positions of the start point and the end point; τ is a temporal scaling factor which can extend or shorten the duration of the produced movement; K is similar to a spring constant and D acts as a damping term which is determined such that the system is critically damped; f is a nonlinear function. These parameters are set such that with the nonlinear function $f = 0$, the dynamic system is globally stable with a unique point attractor g, i.e., x converges to g after a transition from any initial state (x_0, v_0). The nonlinear function f is learned from an observed, arbitrarily complex movement and allows later to reproduce the learned movement. The set of equations is referred to as the "transformation system". The nonlinear function is defined as:

$$f(s) = \frac{\sum_{i=1}^{N} \omega_i \phi_i(s) s}{\sum_{i=1}^{N} \phi_i(s)} \tag{3}$$

$$\phi_i(s) = exp(-h_i(s-c_i)^2) \tag{4}$$

where $\phi_i(s),\ i = 1,\dots,N$ are Gaussian kernel functions with center c_i and width h_i and w_i are adjustable weighting factors. The function f does not depend directly on time but on a phase variable s of the following "canonical system":

$$\tau\dot{s} = -\alpha s \tag{5}$$

where τ is a temporal scaling factor as the same as the one in the transformation system (1) (2) and α is a predefined constant. The phase variable s varies from 1 to 0 during the movement. These equations characterize the advantages of the dynamic movement primitive framework presented as follows:

- The system guarantees the convergence to the goal position g when weighting factors w_i are bounded, since $f(s)$ tends to 0 when the object is moving towards the end of the trajectory.
- The weighting factors w_i can be learned to fit any non-linear force function $f(s)$ so that any desired movement can be generated using the learned nonlinear function.
- The equations are translation invariant[2].
- The duration of a movement can be simply adjusted by scaling τ.

In order to reproduce a demonstrated movement, the following steps are conducted:

1) For a movement in three-dimensional space, record the trajectory $x(t)$ (positions at each sampling instant $t = 0,...,T$) on each axes respectively. Compute the corresponding velocity $v(t)$ and the acceleration $\dot{v}(t)$.
2) Compute values of the nonlinear function f from the transformation system (1) (2) using data obtained above.
$$f_{target}(t) = -\frac{-K(g-x) + Dv + \tau\dot{v}}{g-x_0} \tag{6}$$
where x_0 and g are set to $x(0)$ and $x(T)$ respectively.
3) Build up a canonical system (5) such that a phase variable s of the system maps the time of the observed movement, e.g., $s(t)$ is computed, and goes from 1 to 0 as the object tending to the end of the movement. Derive the nonlinear function f with respect to phase variable s.
4) Fit the nonlinear function $f(s)$ by determining the weighting factors w_i of each Gaussian kernel function $\phi_i(s)$. In this paper, we fitted the nonlinear function using a radial basis function network.
5) To generate a movement, define firstly the starting point x_0 and the end point g, then construct transformation systems (1) (2) using the learned nonlinear function $f_{learned}(s)$ for each axis of the movement. After solving these dynamic systems, the demonstrated movement is reproduced.

The differential equations in DMPs represent an one-dimensional system, thus in applications of robotic system, for each degree of freedom we need to build up a transformation system and couple these system with one common canonical system.

B. Drawbacks of Original Formulation in Generalization to New Goals

The learned movement can be adapted to new scenarios by changing start point x_0 and goal position g. However, in the original DMPs formulation there exist three drawbacks:

[1]A notation different than in [1] and [6] is used to highlight the spring-like character of these equations.

[2]A function being translation invariant means that $f(x+h) = f(x)$, where h is a translation vector.

1) If the goal position g and the starting point x_0 are the same, the transformation system (1) (2) can not be driven by the nonlinear term f, since $(g-x_0)$ and v are zero from beginning to the end. Thus the system only remains at the initial position x_0.
2) The scaling of the nonlinear term f by $(g-x_0)$ introduces undesired problem if $(g-x_0)$ is closed to zero. In this case, a small adjustment in g may lead to a significant acceleration in the dynamic system, which may exceed the physical limitation of robots.
3) The changing in the sign of $(g_{new}-x_0)$ compared to $(g_{demo}-x_0)$ leads to a mirrored trajectory relative to the demonstrated movement.

C. Modified Dynamic Movement Primitives

Because of the problems mentioned above, researchers delivered a modified DMPs formulation, which solves the problems while keeping the favorable features of the original framework [2]. The new formulation of DMPs is motivated from human behaviors [7] and the convergent force fields, which were observed at the frog leg after spinal-cord stimulation [8]. There are three neurophysiological findings from the frog [9]:

- There is a force field occurred in frogs' leg after stimulating the spinal cord. The force field is often convergent.
- The magnitude of force fields is modulated in time by bell-shaped time pulses.
- Force fields can add up linearly after simultaneous stimulation.

These findings are utilized in formulation of the new framework of dynamic movement primitives. As result, the transformation system is replaced by the following equations:

$$\tau\dot{v} = K(g-x) - Dv - K(g-x_0)s + Kf(s) \tag{7}$$

$$\tau\dot{x} = v \tag{8}$$

where the nonlinear function $f(s)$ is the same as the one in (1). The differences between the original and the modified formulation are as follows:

- The nonlinear function is not scaled by $(g-x_0)$ anymore, which avoids problems mentioned above.
- The term $K(g-x_0)s$ counteracts the jumps caused by the term $K(g-x)$ at the beginning of a movement.
- The target nonlinear function $f_{target}(s)$ is now computed according to:

$$f_{target}(s) = \frac{\tau\dot{v}+Dv}{K} - (g-x) + (g-x_0)s \tag{9}$$

To achieve online obstacle avoidance, an adaptation on the modified DMPs framework is proposed to integrate the potential field method.

D. Combining Modified Dynamic Movement Primitives with Potential Fields

To integrate a potential field into the DMPs framework for obline obstacle avoidance, an additional repellent acceleration term $\varphi(x,v)$ is added to the transformation system (7) (8) as follows:

$$\tau\dot{v} = K(g-x) - Dv - K(g-x_0)s + Kf(s) + \varphi(\mathbf{x},\mathbf{v}) \tag{10}$$

$$\tau\dot{x} = v \tag{11}$$

The repellent acceleration term $\varphi(\mathbf{x},\mathbf{v})$ is the negative gradient of the potential field, which depends on the relative position and the velocity of the object to obstacles. This term can change the generated trajectory according to the properties of the potential field to avoid obstacles.

III. POTENTIAL FIELDS FOR OBSTACLE AVOIDANCE

The potential field method is commonly applied in trajectory planning, in which the same idea can be utilized here for obstacle avoidance. There are two types of potential field, static and dynamical. We start from the static one. In this paper, only a single point obstacle is considered.

A. Static Potential Field

The static potential field method was proposed by [2] and [4]. The principle is to create a potential field $U(\mathbf{x})$ at the position $\mathbf{x}$. In our method, the repellent term in (10) is computed by the gradient of the field, i.e., $\varphi(\mathbf{x}) = -\nabla U(\mathbf{x})$.

A static potential field is defined as follows:

$$U_{static}(\mathbf{x}) = \begin{cases} \frac{\eta}{2}\left(\frac{1}{p(\mathbf{x})} - \frac{1}{p_0}\right)^2 & , p(\mathbf{x}) \le p_0 \\ 0 & , p(\mathbf{x}) > p_0 \end{cases} \tag{12}$$

where p_0 is the radius of influence of the obstacle, and η is a constant gain. This potential field is static since it considers only the distance $p(\mathbf{x})$ between the current position and the obstacle. A drawback of this potential field is that it does not produce a smooth obstacle avoidance as the speed and the direction of the movement are not considered.

B. Dynamic Potential Field

A dynamic potential field was proposed by [10] to alleviate the drawbacks of the static potential field method. The dynamic potential field considers both the robots' current position $\mathbf{x}$ and the velocity $\mathbf{v}$. Because of the velocity dependence, this potential field is "dynamic". The dynamic potential field has the following properties:

1) The magnitude of the potential field decreases with the distance from $\mathbf{x}$ to the obstacle.
2) The magnitude of the potential field increases with the speed of $\mathbf{x}$ and is zero when the speed of $\mathbf{x}$ is zero.
3) The magnitude of the potential field decreases with the angle between the direction of the current velocity $\mathbf{v}$ and the direction of the movement towards the obstacle. In addition, if the angle is over 90°, i.e., robots move away from the obstacle, the magnitude of the potential field is zero.

According to these properties, the dynamic potential field is defined as

$$U_{dynamic}(\mathbf{x},\mathbf{v}) = \begin{cases} \lambda(-\cos\theta)^{\beta}\frac{||\mathbf{v}||}{p(\mathbf{x})} & , \frac{\pi}{2} < \theta \le \pi \\ 0 & , 0 \le \theta \le \frac{\pi}{2} \end{cases} \tag{13}$$

where λ is a constant gain of the entire field and β is a constant. In the following simulation we choose $\beta = 2$. θ is the angle between the current velocity $\mathbf{v}$ and the robots' position $\mathbf{x}$ relative to the obstacle,

$$\cos\theta = \frac{\mathbf{v}^{\mathrm{T}}\mathbf{x}}{||\mathbf{v}||p(\mathbf{x})} \tag{14}$$

The angle θ is bounded in range from 0 to π. The repellent force is derived from a negative gradient of the potential function as:

$$\begin{aligned}\varphi(\mathbf{x},\mathbf{v}) &= -\nabla_x U_{dynamic}(\mathbf{x},\mathbf{v}) \\ &= \lambda(-\cos\theta)^{\beta-1}\frac{||\mathbf{v}||}{p}(\beta\nabla_x\cos\theta - \frac{\cos\theta}{p}\nabla_x p)\end{aligned} \tag{15}$$

where $\frac{\pi}{2} \leq \theta \leq \pi$. The dynamic movement primitive can thus be modified with this dynamic potential field.

IV. SIMULATION

For simulation analysis, a set of three-dimensional trajectories of arm movements from human demonstration were recorded. In this simulation, we first demonstrated computing the nonlinear function $f(s)$ from the recorded trajectory. Then, we learned the weighting parameters w_i for encoding the target nonlinear function $f_{target}(s)$ using a radial basis function network and reproduced the learned nonlinear function $f_{learned}(s)$. Finally, we reproduced the demonstrated trajectory using the learned nonlinear function $f_{learned}(s)$.

A. Computing Nonlinear Function

We can computed the nonlinear function $f_{target}(s)$ for a demonstrated trajectory using method described in Section II-C. Figure 1 shows the demonstrated trajectory in three-dimensional space. We need to build three transformation

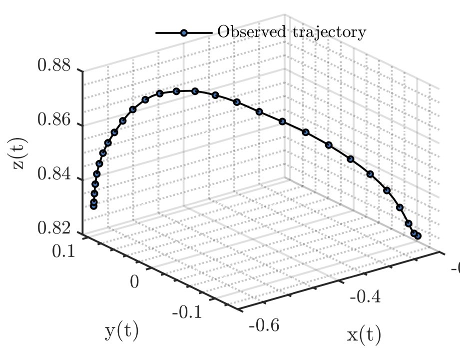

Fig. 1. Demonstrated trajectory

systems (7) (8) for each axes respectively. Before computing nonlinear function $f(s)$ we need to integrate a canonical system (5). Figure 2 shows the change of the phase variable s in this simulation. Ideally, it goes from one to zero in the duration of the demonstrated trajectory.

Then, we computed the nonlinear functions $f(s)$ of transformation systems in each axis according to (8). Figure 3 shows the computed nonlinear functions of the demonstrated trajectory.

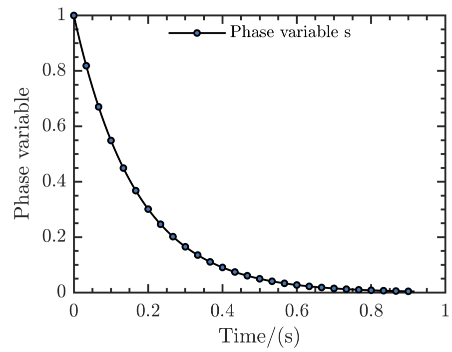

Fig. 2. Trajectory of phase variable s.

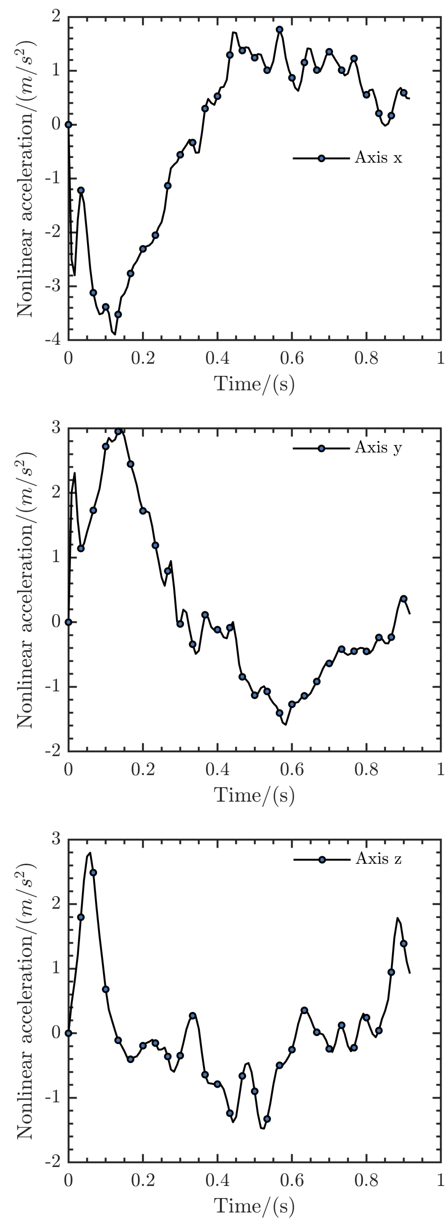

Fig. 3. Nonlinear force terms $f_{target}(s)$ of the transformation systems for demonstrated trajectory in each axis.

Next, we learned the weighting parameters w_i in (3) to represent the target nonlinear function. This can be implemented using a radial basis function network. An explanation of the radial basis function network is described in the APPENDIX. In this paper, we chose the number of the kernel functions $\phi_i(s)$ as 26 and the common width c_i as 200. Figure 4 presents an example of the kernel function set.

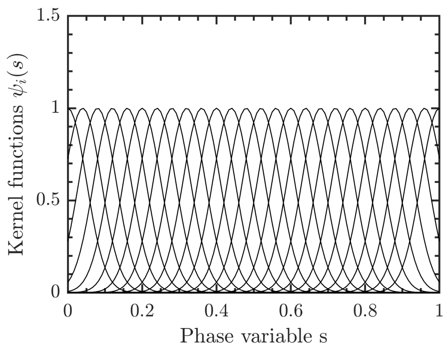

Fig. 4. Kernel function $\phi_i(s)$

According to formula (19) in the APPENDIX, the weighting parameters w_i are computed. Figure 5 shows the comparison of target nonlinear function and learned one.

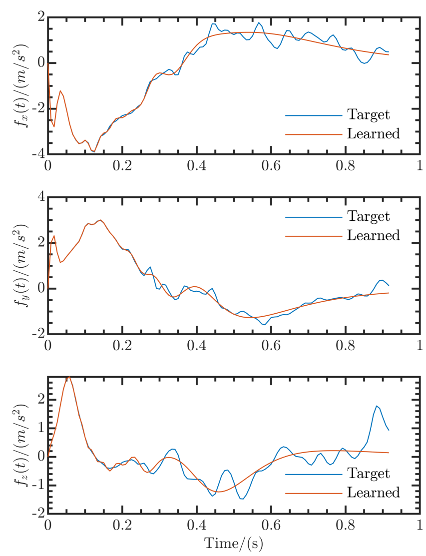

Fig. 5. Learned nonlinear functions compared with target functions

B. Reproduction of a Demonstrated Trajectory

Now we reproduce the demonstrated trajectory by computing the transformation system (7) (8) in each axis with the learned nonlinear function $f_{learned}(s)$. Figure 6 shows the comparison of demonstrated trajectory and reproduced one in three-dimensional space and Figure 7 to Figure 9 show the comparison in each axis respectively.

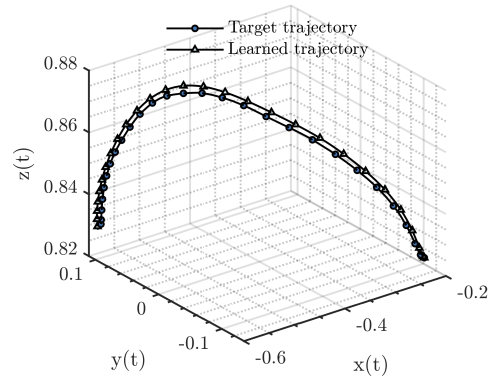

Fig. 6. Comparison of reproduced trajectory with observed trajectory

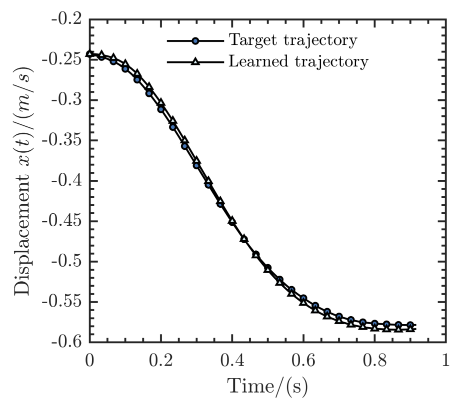

Fig. 7. Comparison of reproduced trajectory with observed trajectory on axis x

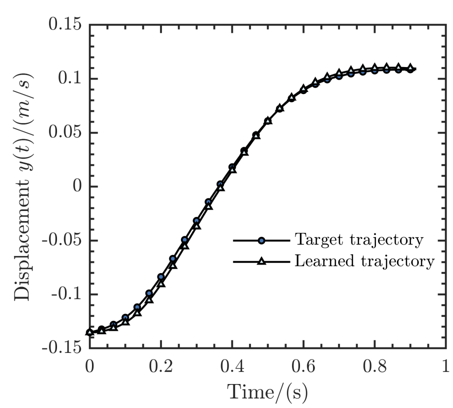

Fig. 8. Comparison of reproduced trajectory with observed trajectory on axis y

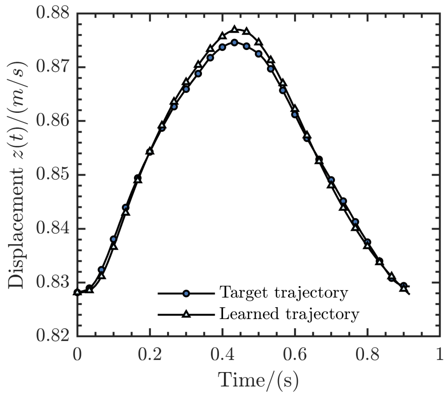

Fig. 9. Comparison of reproduced trajectory with observed trajectory on axis z

C. Generalization to a New Goal

Next we present generalization of the demonstrated trajectory to a new goal. For doing this, we need to change the goal position g in transformation system (7) (8) in each axis, and modify the canonical system (5) by changing the time scaling parameter τ and the constant α since by reaching to a new goal position the movement duration of system would change, which require the corresponding time scaling. Figure 10 shows the result of generalization of demonstrated trajectory to a new goal.

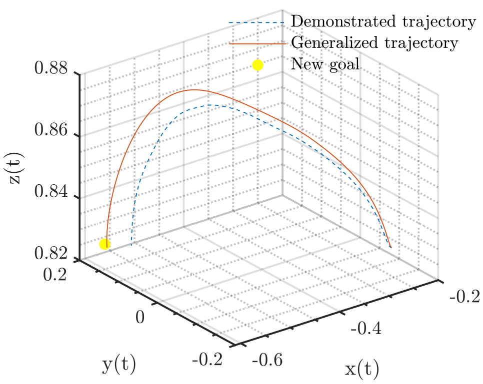

Fig. 10. Result of generalization of learned trajectory to a new goal

D. Obstacle Avoidance with Potential Fields

The DMP framework can be modified by adding a repellent acceleration term to the transformation system (7) for obstacle avoidance. This added acceleration term is the negative gradient of a potential field, which can be computed according to (15). Here we choose $\beta = 2$. The repellent acceleration term is online adapted during the movement, which means for each time instant the repellent acceleration is computed and added to the system at the next time instant. A computed repellent acceleration on axis x is shown on Figure 11. The accelerations on other axes are analogous.

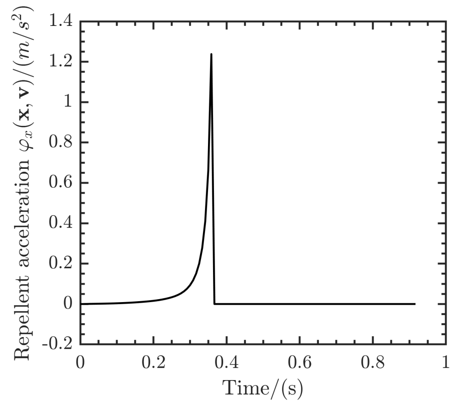

Fig. 11. Repellent acceleration on axis x

Solved the dynamic system (10) combined with repellent acceleration terms, we obtained a trajectory with obstacle avoidance. The result is shown on Figure 12.

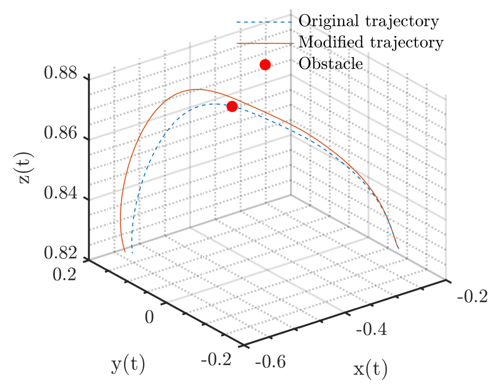

Fig. 12. Trajectory with obstacle avoidance using dynamic potential field

V. CONCLUSION

In this paper we studied the framework of dynamic movement primitives for movement generation. This framework allows to reproduce any complex rhythmic and discrete behaviors presented in a demonstrated movement by learning a nonlinear function characterizing the observed trajectory. We presented an online obstacle avoidance approach integrated into the dynamic movement primitives framework using a dynamic potential field method. We analyzed that a dynamic potential field can lead to a smoother obstacle avoidance behavior compared to a static potential field method as the result of considering both the distance between the current position and the obstacle and the movement velocity relative to the obstacle. We conducted several simulations presenting the procedure of learning an observed movement, reproducing the movement and generalizing it to a new goal position. Finally, we demonstrated the performance of the online obstacle avoidance of the dynamic movement primitives framework integrated with a dynamic potential field in simulation.

APPENDIX

A nonlinear function can be fitted using Radial Basis Network (RBF). A RBF network is consisted of three layers: input layer; one hidden layer; output layer. The structure of RBF network is shown on Figure 13, where x_i are scalar input; y_i are sampled value of the target function; f is a Gaussian kernel function with center μ_k, width h_k and amplitude 1. Generally, each output value y_i is a linear combination of all kernel functions amplified by different weighting parameters ω_i. This can be shown by the following equation:

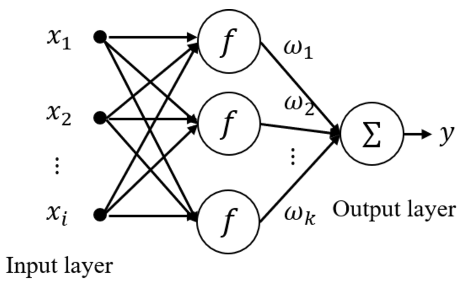

Fig. 13. Structure of radial basis network

$$y_i = \begin{bmatrix} f(||x_i-\mu_1||) & f(||x_i-\mu_2||) & \cdots & f(||x_i-\mu_k||) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_k \end{bmatrix} \tag{16}$$

where ω_k are the weighting parameters that are to be trained. For determining the weighting vector W we use Least-square approach. Combining all data pair (x_i, y_i), we obtain the following matrix equation,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix} = \begin{bmatrix} f(||x_1-\mu_1||) & \cdots & f(||x_1-\mu_k||) \\ \vdots & & \vdots \\ f(||x_i-\mu_1||) & \cdots & f(||x_i-\mu_k||) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_k \end{bmatrix} \tag{17}$$

According to Least-square approach, the weighting vector W can be computed by:

$$W = (F^T F)^{-1} F^T Y \tag{18}$$

where F and Y denote the matrix constructed with elements $f(||x_i - \mu_k||)$ and the output vector in (17) respectively. For our application, the weighting parameter w_i in (4) can be computed by:

$$W = (F'^T F')^{-1} F'^T Y' \tag{19}$$

where $F' = \begin{bmatrix} f(||s_1-c_1||) & \cdots & f(||s_1-c_i||) \\ \vdots & & \vdots \\ f(||s_k-c_1||) & \cdots & f(||s_k-c_i||) \end{bmatrix}$, and $f(||s_k - c_i||) = \frac{exp(-h_i(s_k-c_i)^2)}{\sum_{i=1}^{N} exp(-h_i(s_k-c_i)^2)}$, N is the number of kernel functions, $Y' = \begin{bmatrix} f(s_1) & f(s_2) & \cdots & f(s_k) \end{bmatrix}^T$.

REFERENCES

[1] Stefan 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.

[2] Bruce Krogh. "A generalized potential field approach to obstacle avoidance control". In: *Proc. SME Conf. on Robotics Research: The Next Five Years and Beyond, Bethlehem, PA, 1984*. 1984, pp. 11–22.

[3] Johann Borenstein and Yoram Koren. "Real-time obstacle avoidance for fast mobile robots". In: *IEEE Transactions on systems, Man, and Cybernetics* 19.5 (1989), pp. 1179–1187.

[4] Oussama Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". In: *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[5] Oliver Brock and Oussama Khatib. "Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 1. IEEE. 2000, pp. 550–555.

[6] Stefan Schaal et al. "Control, planning, learning, and imitation with dynamic movement primitives". In: *Workshop on Bilateral Paradigms on Humans and Humanoids: IEEE International Conference on Intelligent Robots and Systems (IROS 2003)*. 2003, pp. 1–21.

[7] Peter Pastor et al. "Learning and generalization of motor skills by learning from demonstration". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 763–768.

[8] Simon F Giszter, Ferdinando A Mussa-Ivaldi, and Emilio Bizzi. "Convergent force fields organized in the frog's spinal cord". In: *Journal of neuroscience* 13.2 (1993), pp. 467–491.

[9] Heiko Hoffmann and Stefan Schaal. "Human movement generation based on convergent flow fields: a computational model and a behavioral experiment". In: *Advances in Computational Motor Control VI, R. Shadmehr and E. Todorov, Eds., San Diego, CA* (2007).

[10] Dae-Hyung Park et al. "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields". In: *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*. IEEE. 2008, pp. 91–98.