Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Trajectory Optimization and Force Control with Modified Dynamic Movement Primitives under Curved Surface Constraints *

Liang Han, Peng Kang, Yongting Chen, Wenfu Xu*, and Bing Li

School of Mechanical Engineering and Automation
Harbin Institute of Technology, Shenzhen
Shenzhen, Guangdong Province, China

lianghanhit_edu@163.com; kangpeng@stu.hit.edu.cn; qyz146006@163.com; wfxu@hit.edu.cn; libing.sgs@hit.edu.cn

***Abstract*—Dynamic Movement Primitives (DMPs) have been used extensively for trajectory planning due to robust against perturbations and excellent generalization performance. However, canonical discrete DMP could not generate right trajectories in three cases: the goal point is same as the start point, the goal point is close to the start point, and the goal point is changing across the start point. Moreover, when the original trajectory is on a curved surface, it is difficult to ensure that the generalized trajectory remains on this surface. In this paper, we propose a modified DMP method with the scaling factor and acceleration coupling term to solve the policy of the above learning problems using redundant robots. First, we introduce the adjusted cosine similarity as the performance index of the generalized curve. The cosine similarity is optimized to get the scaling factor. Then, by adding the acceleration coupling term to the original DMP, trajectory planning and force control are achieved on the curved surface. Finally, some simulations and experiments demonstrate the performance of our proposed method.**

Index Terms—DMP; Trajectory optimization; Force control; Curved surface; Redundant robot

I. Introduction

Robot learning from demonstration (LFD) has widely used in robotics that learn a policy from humans performing a task [1-3]. It can effectively solve the trajectory planning problems of robotic arms without kinematic models. DMP has wide applications in trajectory planning, obstacle avoidance, assembly, etc. due to its good performance. A DMP can represent any recorded movement with a set of differential equations. However, this basic formalism could not produce proper trajectories in some cases: the goal point is same as the start point, the goal point is close to the start point, and the goal point is changing across the start point. To solve these problems, some modified DMP formulas have been proposed. In [4], the modified DMP is proposed to overcome drawbacks motivated from human behavioral data. In [5], the original DMP is improved to generalize trajectories to new targets without singularities and large accelerations by adding an additional term to the differential equations. Although these methods can ensure that the position converges to the target point, the similarity of the trajectory cannot be guaranteed. In this paper, we use PSO method to optimize the parameter of the adjusted cosine similarity to maximize the similarity of trajectories.

Dynamic motion primitive is widely used in trajectory planning of robotic arms to imitate the behavior of the teacher. In [6], DMP method is proposed for on-line trajectory modification, and imitation learning by representing movements by a set of nonlinear differential equations. In [7], a three-level motion planning framework for an anthropomorphic arm is raised. In [8], a novel joining method for multiple DMPs is presented based on the modification of the original DMP. However, the original DMP does not consider the force information in the operation of the robot, for example, assembly. Therefore, the compliant movement primitive (CMP) method is proposed to record force information during teaching [9]. CMP method has been applied to robot assembly too. In [10, 11], low trajectory errors and compliance control without explicit models is effectively solved using CMP on the anthropomorphic dual-arm robot. However, the CMP method relies on the teacher's perception of force. It has poor adaptability to different tasks because of different force characteristics. In addition, the initial DMP algorithm is unconstrained. We cannot limit the trajectory to an unknown curved surface. If the teaching trajectory is on a plane, then the generalized trajectory is also in this plane. If the teaching trajectory is on a curved surface, the generalized trajectory is not guaranteed on the surface. One idea is to use impedance control to ensure contact between the end effector and the surface. In [12], reinforcement Learning is used for the 7 degree-of-freedom (DOF) robot to learn parameters of the impedance control in stochastic force fields. The trajectories are represented by DMPs. In [13], the human-in-the-loop approach using impedance control interface to solve assembly tasks is proposed. The demonstrated trajectories are encoded using DMP and learnt using locally weight regression. In [14], the constraint-aware learning is proposed in the null space of a constraint for the redundant robot. Some papers try to add coupling terms to initial DMP to perform specific tasks, such as

* This work is supported by t National Key R&D Program of China (2018YFB1304600), the Key Research and Development Program of Guangdong Province [grant numbers 2019B090915001] and the Basic Research Program of Shenzhen under Grant JCYJ20180507183610564). Wenfu Xu is the corresponding author.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

obstacle avoidance, temporal evolution, trajectory classification, and etc. [15].

In order to solve the problems that the original DMP could not produce the right trajectories under some special cases and the curved surface, we propose a modified dynamic movement primitive method for robot control in this paper. The proposed method has the following advantages. First, the optimized trajectories in any cases can be generated. Second, the trajectories can be generalized on an unknown surface. The end effector of the arm can achieve force control and keep tool aligned with the normal vector of the surface. Finally, the teachers can demonstrate the skills in the demonstration plane instead of the actual operating surface.

This paper is organized as follows. In Section Ⅱ, the initial DMP method and its limitations are introduced. In Section Ⅲ, the modified DMP with the scaling factor and acceleration coupling term is introduced. In Section Ⅳ, Simulations are carried out to verify the trajectory planning and force control of the modified DMP. In Section Ⅴ, the performance of the modified DMP is verified by experiments using a 7-DOF robotic arm. In Section Ⅵ, a conclusion is given.

II. Initial DMP Method and Its Limitations

We first describe initial DMP algorithm for state variables. The trajectory of a variable can be represented by a DMP. Multiple trajectories of variables can be represented by multiple independent DMPs. The equations of DMP are motivated from the second-order differential equation. The nonlinear term is added to the equation for changing the acceleration of the motion. Especially, the external force in the spring-mass-damper system is the sum of a set of Gaussian kernel functions.

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

where x_0 is the start point, K is the spring stiffness, D the damping damper, τ is the scaling factor of the movement duration, and f is the parametrized non-linear function. The function f is defined as:

$$f = \frac{\sum_{i=1}^{N} \omega_i \psi_i(s) s}{\sum_{i=1}^{N} \psi_i(s)} \tag{2}$$

where $\psi_i(s)$ is the Gaussian functions:

$$\psi_i(s) = \exp(-h_i(s-c_i)^2) \tag{3}$$

f is related to the phase variable s without time. c is the center, h is the width, and ω is the adjustable weight. Equations (1), (2), (3) are called transformation system. s is driven by the canonical system equation.

$$\tau\dot{s} = -\alpha_s s \tag{4}$$

where α_s is a predefined constant. The state variable s is set to 1 initially. When s approaches 0, the equation (1) becomes a linear second order differential equation that converges to g. In order to represent a DMP, we should learn ω_i from the demonstration data. In this paper, we use locally weighted regression to learn ω_i.

$$\omega_i = \frac{\boldsymbol{s}_x^T \boldsymbol{\Gamma}_i \boldsymbol{f}_{target}}{\boldsymbol{s}_x^T \boldsymbol{\Gamma}_i \boldsymbol{s}_x} \tag{5}$$

where $\boldsymbol{f}_{target}$ is calculated from the demonstration data.

$$(g-x_0)\boldsymbol{f}_{target} = \tau^2 \ddot{\boldsymbol{x}}_{demo} - K(g-\boldsymbol{x}_{demo}) + D\tau\dot{\boldsymbol{x}}_{demo}$$

$$\boldsymbol{s}_x = \begin{pmatrix} s_1(g-x_0) \\ s_2(g-x_0) \\ \cdots\cdots \\ s_p(g-x_0) \end{pmatrix} \quad \boldsymbol{\Gamma}_i = \begin{pmatrix} \psi_i(1) & & 0 \\ & \psi_i(2) & \\ & & \cdots & \\ 0 & & \psi_i(p) \end{pmatrix}$$

If the $g-x_0$ term is zero, ω_i will be infinite. If g is close to x_0, ω_i will be huge. A small change in g may lead to huge accelerations. If the goal point is changing across the initial point, the force term $(g-x_0)f$ inverts. The whole movement inverts too. The shortages of the initial DMP method are shown in [4].

When trajectory is generalized by the initial DMP algorithm, the nonlinear function f is proportionally changed. Therefore, when the original trajectory is on a plane, the generalized trajectory must also be on this plane. When the original trajectory is on a curved surface, the generalized trajectory is not guaranteed on this surface. Fig. 1 shows that the initial DMP does not have the ability to generalize trajectories to an unknown surface.

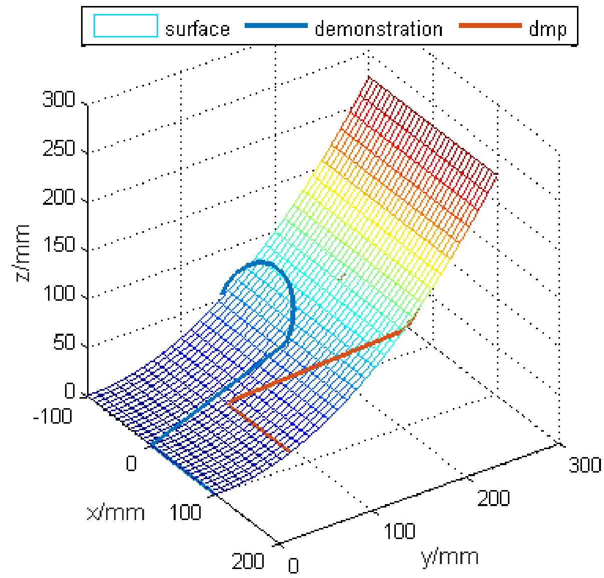


Fig. 1. The generalized trajectory using original DMP. When the original trajectory is in a plane, the result of generalization must still be in this plane.

III. Modified DMP Method for Trajectory Optimization and Force Control

A. Modified DMP Method

To solve the limitations using original DMP, we add the scaling factor α_f and the acceleration coupling term to the transformation system. The learning process and reproducing process of DMP are separately handled.

First, we discuss the learning process of the modified DMP. Refer to the papers [13, 14], we delete the $g - x0$ term in the learning process to avoid ω_i infinity. The transformation system in the learning process becomes the following equation.

$$\begin{aligned} \tau\dot{v} &= K(g-x) - Dv + f \\ \tau\dot{x} &= v \end{aligned} \tag{6}$$

The canonical system is also same as the initial DMP shown in equation (4). Then, we use locally weighted regression to learn ω_i when it is training.

$$\omega_i = \frac{\boldsymbol{s}^T \boldsymbol{\Gamma}_i \boldsymbol{f}_{target}}{\boldsymbol{s}^T \boldsymbol{\Gamma}_i \boldsymbol{s}} \tag{7}$$

where, $\boldsymbol{s}_x = \begin{pmatrix} s_1 & s_1 & \ldots\ldots & s_p \end{pmatrix}^T$.

More importantly, we add a scaling factor α_f and the force coupling term Δa to the transformation system in the reproducing process of modified DMP. The transformation system in the reproducing process becomes the following equation.

$$\begin{aligned} \tau\dot{v} &= K(g-x) - Dv + \alpha_f f + \Delta a \\ \tau\dot{x} &= v \end{aligned} \tag{8}$$

α_f is set to be greater than 0 to avoid the generalized trajectory inverting. If α_f equal to 1, the reproduced trajectory will be same as the initial one. Otherwise, the trajectory is similar to the original trajectory. Next, we get the optimized value of α_f through maximizing the adjusted cosine similarity of the trajectories. Δa is the acceleration coupling term related to the contact force error. Δa is used for trajectory planning and force control on the curved surface.

If the limit $\boldsymbol{s} \to 0$ and $\Delta a \to 0$, the function *f* vanishes and (6) and (8) becomes a linear spring equation that converges to *g*. The system is also stable and convergent.

B. Trajectory optimization

Algorithm 1 shows the program of the modified DMP method for trajectory optimization. To solve the problems in Fig. 1, we adopt the adjusted cosine similarity to evaluate the similarity of the two curves. The adjusted cosine similarity is defined as:

$$\cos(\theta) = \frac{(\boldsymbol{a}-k)\cdot(\boldsymbol{b}-k)}{\|\boldsymbol{a}-k\|\|\boldsymbol{b}-k\|} \tag{9}$$

where $\boldsymbol{a}$ is equal to the vector $\boldsymbol{x} - \min(\boldsymbol{x})$, $\boldsymbol{b}$ is equal to the vector $\boldsymbol{x}' - \min(\boldsymbol{x}')$., k is the average of augmented vectors $[\boldsymbol{a}\ \boldsymbol{b}]$. where $k = \sum(a_i + b_i)/(2N)\ \ i = 1,\ldots N$.

In this paper, we use Particle Swarm Optimization (PSO) to obtain α_f because there is no explicit equation between $\cos(\theta)$ and α_f.

Algorithm 1 DMP_learning()

1. Input: $x\ \dot{x}\ \ddot{x}$
2. DMP_learning(): $\omega_i = \frac{\boldsymbol{s}^T \boldsymbol{\Gamma}_i \boldsymbol{f}_{target}}{\boldsymbol{s}^T \boldsymbol{\Gamma}_i \boldsymbol{s}}$
3. Randomly generate a set of values: α_f
4. **While** ($\min(\cos(\theta)-1)^2 > 1e^{-2}$) **do**
5. DMP_reproducing(): x'
6. $\boldsymbol{a} = \boldsymbol{x} - \min(\boldsymbol{x})$ $\boldsymbol{b} = \boldsymbol{x}' - \min(\boldsymbol{x}')$
   Normalized each vector: $k = \sum(a_i + b_i)/(2N)$
   Cosine similarity: $\cos(\theta) = \frac{(\boldsymbol{a}-k)\cdot(\boldsymbol{b}-k)}{\|\boldsymbol{a}-k\|\|\boldsymbol{b}-k\|}$
7. $\min(\cos(\theta)-1)^2 \Rightarrow$ *Produce new* α_f *by PSO*
8. **end**
9. DMP_ reproducing (): $[x_{dmp}, \dot{x}_{dmp}, \ddot{x}_{dmp}]$

C. Trajectory Planning and Force control on the Curved Surface

Considering an unknown surface, we use force control to make the end effector of the arm keep contact with the surface to ensure that the generalized trajectory lies on the surface. We directly add the acceleration term to initial DMP and consider six-dimension force control.

Fig. 2 shows robot performing a constrained task on the curved surface. In order to keep contact with the surface, the robot must maintain a certain contact force on the surface. In order to keep alignment with the normal vector of the surface, the robot must adjust the orientation of the end effector to be perpendicular to the surface. It means the contact force must be controlled to a given force. The force error is defined as [14]:

$$\begin{aligned} {}^T\tilde{\boldsymbol{F}} = &\begin{bmatrix} 0 & 0 & {}^TF_{zd} & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} - \\ &\begin{bmatrix} {}^TF_x & {}^TF_y & {}^TF_z & {}^TM_x & {}^TM_y & {}^TM_z \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{10}$$

where ${}^TF_{zd}$ is the desired contact force in the Z direction represented in coordinate system $\{T\}$, ${}^TF_x$, ${}^TF_y$, ${}^TF_z$, ${}^TM_x$, ${}^TM_x$, and ${}^TM_x$ are actual force and torque measured by six-axis force sensor. We directly add the force error to the transformation system to change the acceleration of the trajectory and the attitude of the arm. In order to fully represent the trajectory and the attitude of the end effector, we need 6 DMP equations. The six DMPs are shown as the following:

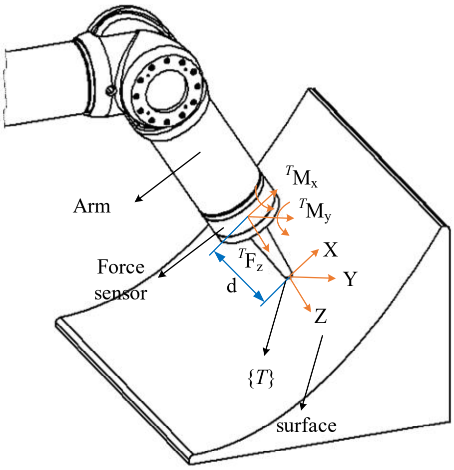


Fig. 2. Robot performs constrained task on curved surface

$$
\begin{aligned}
\tau\dot{v}_x &= K(g_x - x) - Dv_x + \alpha_{fx} f_x - K_{fx} F_x \\
\tau\dot{v}_y &= K(g_y - y) - Dv_y + \alpha_{fy} f_y - K_{fy} F_y \\
\tau\dot{v}_z &= K(g_z - z) - Dv_z + \alpha_{fz} f_z - K_{fz} (F_{dz} - F_z) \\
\tau\dot{\omega}_x &= K(g_{\varphi x} - \varphi_x) - D\omega_x + \alpha_{f\varphi x} f_{\varphi x} - K_{f\varphi x} M_{xt} \\
\tau\dot{\omega}_y &= K(g_{\varphi y} - \varphi_y) - D\omega_y + \alpha_{f\varphi y} f_{\varphi y} - K_{f\varphi y} M_{yt} \\
\tau\dot{\omega}_z &= K(g_{\varphi z} - \varphi_z) - D\omega_z + \alpha_{f\varphi z} f_{\varphi z} - K_{f\varphi y} M_{zt}
\end{aligned} \tag{11}
$$

Algorithm 2 shows the trajectory planning and force control with surface constraints. Fig. 3 shows its process.

Algorithm 2 Trajectory planning and force control with surface constraints

1. Using **Algorithm 1** to get α_f in the transition plane.
2. DMP_producing() for trajectory planning on the surface
3. **While** (norm(F_{err}) $> \sigma_F$) **do**
4. The modified DMP equation (24) for ${}^{\text{tool}}\dot{\boldsymbol{x}}$
5. $\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}_{\text{tool}} \cdot {}^{\text{tool}}\dot{\boldsymbol{x}}$, $\boldsymbol{q}(t+\Delta T) = \boldsymbol{q}(t) + \dot{\boldsymbol{q}}\Delta T$
6. PD position control law
7. **end**
8. Output: $x_{dmp}, \dot{x}_{dmp}, \ddot{x}_{dmp}$

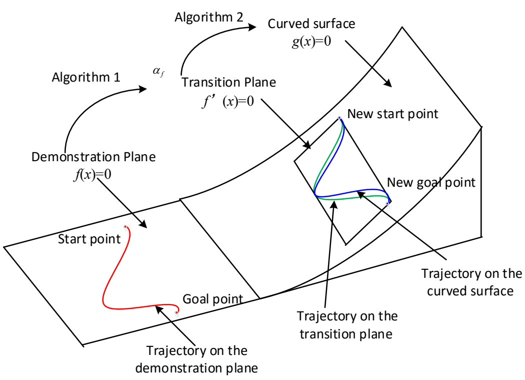


Fig. 3. The process of trajectory planning and force control with surface constraints

The trajectory planning on the surface consists of two steps. First, α_f is determined by trajectory optimization. The trajectory of the transition plane is obtained by Algorithm 1. Then, the trajectory on the curved surface can be obtained by Algorithm 2. The modified DMP is suitable for occasions where the surface is smooth.

IV. Simulations

A. Trajectory optimization

A proper trajectory can be produced using modified DMP in any case. We simulated a general case and three special cases to verify its performance.

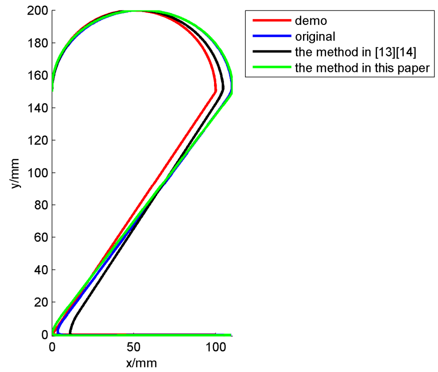
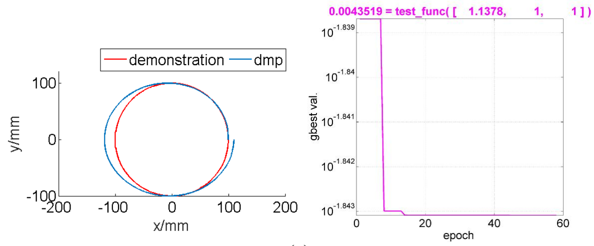
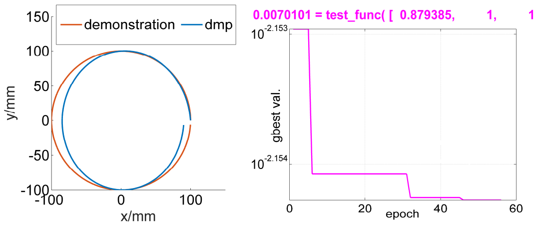


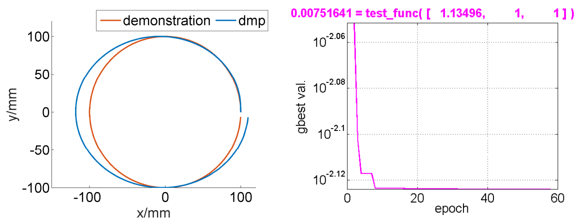
Fig. 4. The generalized trajectory using three DMP methods.



(a)



(b)



(c)

Fig. 5. The modified DMP produces the proper trajectories. a) the goal point is same as the start point; b) the goal point is close to the initial point; c) the goal point is changing across the initial point.

For the general case, three methods are used to generalize the trajectories under the same conditions. Fig. 4 shows the trajectory '2' which are generalized by the original DMP, the modified DMP in [4, 6], and the modified DMP based on PSO in this paper. According to fig. 4, the original DMP and the modified DMP in this paper can generalize trajectories with the highest similarity with the original trajectory, while the method in the paper [4, 6] has a worse similarity. A proper trajectory can be produced using modified DMP in any case if we give a suitable range. For the general case, modified DMP can produce the trajectory as the initial DMP.

For the case that initial DMP cannot produce the right trajectory, the modified DMP also could generalize a proper trajectory. Fig. 5 shows how the modified DMP produces the proper trajectories in special cases. In the simulation, we only change the coordinates of X. Through PSO algorithm, we get α_f that maximize the adjusted cosine similarity. Simulations show that DMP can reasonably generalize any trajectory and converge to the target point.

B. Trajectory Planning and Force Control with Surface Constraints

In order to verify the modified DMP method for trajectory planning and force control of the arm, we performed the MATLAB-ADAMS co-simulations. The simulation model was created in ADAMS, and the control algorithm was implemented in MATLAB and Simulink. The initial state of the simulation model is shown in Fig. 6. The simulation system contains a 7-DOF robotic arm, a 6-D force sensor, a tool, and an object. The arm would operate object on the curved surface of the object. In the simulation, we measure the force and torque of the fixed pair as the data of the six-axis force sensor.

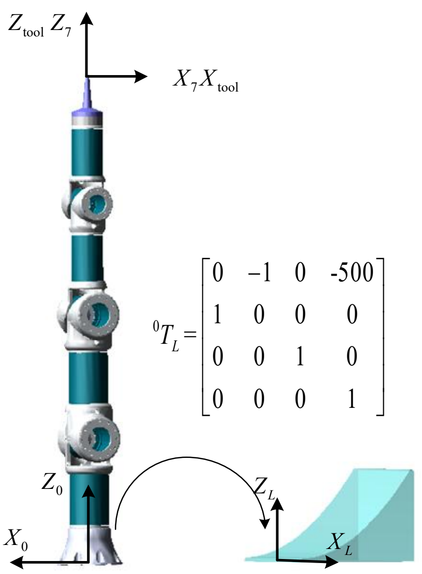


Fig. 6. The simulation model in ADAMS

The simulations include DMP learning from demonstration in the given plane and DMP trajectory reproducing in an unknown surface.

The demonstration data is given directly in MATLAB. The trajectory "2" is in the plane shown in Fig. 7. During the DMP learning, the parameters in the simulation are set as follows. The number of the Gaussian kernel functions N=100; $\boldsymbol{c}$ is the vector consisting of N distributed data between 0 and 1; h is the Gaussian distribution width: $h = 0.5N^{2.5}$; $\alpha_s = 2$; K=2000 ; $D = 2000/4$; The initial point is: $x_0 = \begin{bmatrix} 0 & 75 & 130 \end{bmatrix}$; The goal point is: $g = \begin{bmatrix} 100 & 0 & 0 \end{bmatrix}$; Through locally weighted regression, ω is learned.

The trajectory is reproduced in an unknown surface by modified DMP. The new initial point is: $x_{\text{dmp0}} = \begin{bmatrix} 0 & 200 & 100 \end{bmatrix}$; The goal point is: $g_{dmp} = \begin{bmatrix} 100 & 80 & 16 \end{bmatrix}$; The new initial point and goal point are obtained by linear transformation of the original trajectory without changing the shape, so we choose α_f as 1. The generalized trajectory "2" is shown in Fig. 8. The dual-arm robot moves from the initial pose to the initial point first. Then, the modified DMP method is carried out. The position control will be implemented in the MATLAB and Simulink using PD control.

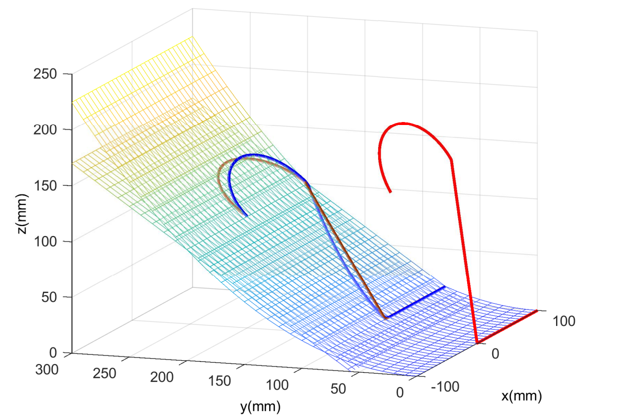


Fig. 7. The DMP learning and reproducing. The red line is the demonstrated trajectory. The brown line is the trajectory by the modified DMP based on the PSO to achieve α_f . The blue line is the trajectory on the surface by the modified DMP based on the form coupling term.

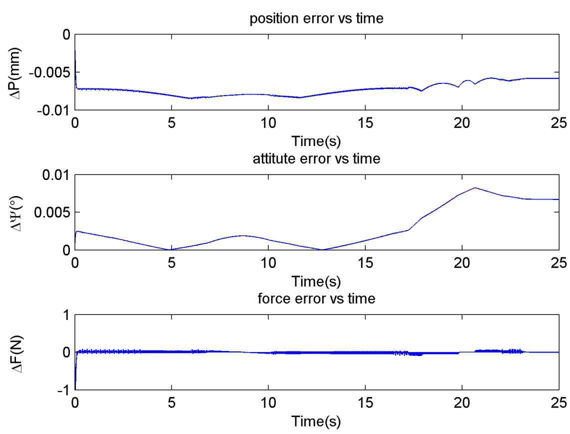


Fig. 8. Position error, attitude error, and force error by simulations

The trajectory is reproduced on the surface. The position error is defined as the shortest distance from the end point of the arm to the surface. The position error curve, the attitude error curve, and the force error curve are shown in Fig. 8.

From the curves, the position errors of the arm is very small, and the attitude errors do not exceed 0.01°. The force errors of the arm does not exceed 0.01N. These results show that the modified DMP method works well. The dual-arm robot successfully deal with the trajectory planning in an unknown surface and achieve the accurate force control.

V. Prototype and Experiments

A. Prototype

In order to verify the performance of the control system based on the modified DMP, we set up the prototype as shown in Fig. 9. The test system contains a 7-DOF robot and its control system, an object which contains a surface, power supply, a 6-D force sensor and its data acquisition system and so on.

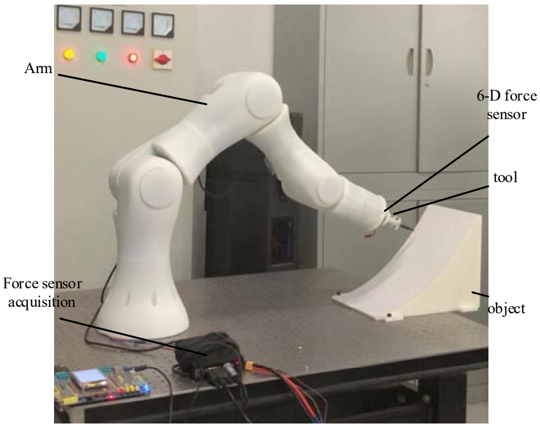


Fig. 9. The prototype of the system

B. Experiments

The trajectory planning and the force control on the surface of the object is carried out. In the experiment, the parameters of the modified DMP are set as same as the simulation. The control algorithms are implemented on the ARM processor.

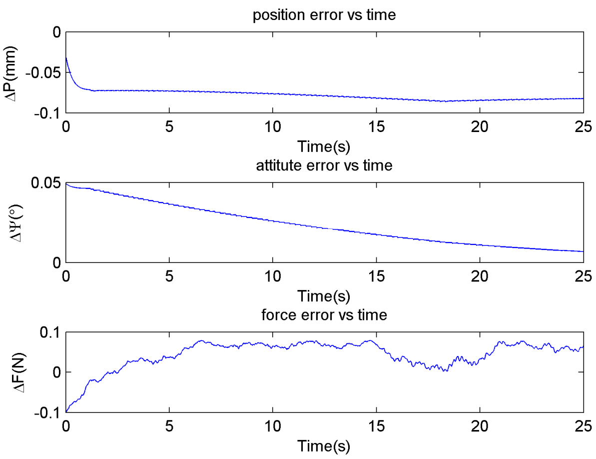


Fig. 10. Position error, attitude error, and force error by experiments

The position error curve, the attitude error curve, and the force error curve are shown as the Fig. 11. From the curves, the position errors of the arm is less than 0.1mm, and the attitude errors do not exceed 0.05°. The force errors of the arm does not exceed 0.1N. The effectiveness of this method is verified in the actual robot arm control.

VI. Conclusion

In this paper, the modified DMP method for trajectory optimization and force control with surface constraints is developed. We optimize the adjusted cosine similarity to get proper scaling factor. Moreover, we add the acceleration coupling term related to the force information to achieve trajectory planning and force control on an unknown surface. To verify the effectiveness of this method, we carried out the simulations and experiments using the 7-DOF robotic arm. From simulations and experiments, the modified DMP method is verified.

References

[1] B. D. Argall, S. Chernova, M. Veloso and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol.57, no.5, pp.469-483, 2009.

[2] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol.3, no.6, pp.233-242, 1999.

[3] A. Billard, S. Calinon, R. Dillmann and S. Schaal. Robot programming by demonstration: Springer, 2008. 1371-1394.

[4] D. Park, H. Hoffmann, P. Pastor and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Proc.*2008, pp.91-98.

[5] H. Hoffmann, P. Pastor, D. Park and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Proc.*2009, pp.2587-2592.

[6] A. J. Ijspeert, J. Nakanishi and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc.*2002, pp.1398-1403.

[7] X. Ding and C. Fang, "A novel method of motion planning for an anthropomorphic arm based on movement primitives," *IEEE/ASME Transactions on Mechatronics*, vol.18, no.2, pp.624-636, 2013.

[8] T. Kulvicius, K. Ning, M. Tamosiunaite and F. Worgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Transactions on Robotics*, vol.28, no.1, pp.145-157, 2012.

[9] A. Batinica, B. Nemec, A. Ude, M. Raković and A. Gams, "Compliant movement primitives in a bimanual setting," in *Proc.*2017, pp.365-371.

[10] T. Petrič, A. Gams, L. Colasanto, A. J. Ijspeert and A. Ude, "Accelerated Sensorimotor Learning of Compliant Movement Primitives," *IEEE Transactions on Robotics,* no.99, pp.1-7, 2018.

[11] M. Deniša, A. Gams, A. Ude and T. Petrič, "Learning compliant movement primitives through demonstration and statistical generalization," *IEEE/ASME transactions on mechatronics*, vol.21, no.5, pp.2581-2594, 2016.

[12] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. Theodorou and S. Schaal, "Reinforcement learning of impedance control in stochastic force fields," in *Proc.*2011, pp.1-6.

[13] L. Peternel, T. Petrič and J. Babič, "Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface," in *Proc.*2015, pp.1497-1502.

[14] L. Armesto, J. Moura, V. Ivan, M. S. Erden, A. Sala and S. Vijayakumar, "Constraint-aware learning of policies by demonstration," *The International Journal of Robotics Research, vol. 37, pp.1673-1689*, 2018.

[15] J. I. A., N. J., H. H., P. P. and S. S., "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors," *Neural Computation*, vol.25, no.2, pp.328-373, 2013.