Applying Dynamic Movement Primitives to Motion Planning and Inverse Kinematics Problems of an Upper Limb Exoskeleton Robot

Huy Nguyen, Trung Nguyen, Hung Dao, Dung Nguyen, and Ha Pham

Abstract Motion planning and inverse kinematics (IK) are two important parts of controlling the robot exoskeleton. To make the wearer feel comfortable, the joints variable values need to be similar to the values measured in the actual activities of the human arm. Dynamic movement primitives (DMPs) are a useful solution to this problem. In this study, we proposed to apply DMPs to generate the robot's end-effector position and orientation trajectory as well as the robot's joint variable values trajectory from the actual measured joints values of the robot exoskeleton during Activity of Daily Living exercise such as reaching a cup of water. Then, with the same ADL, when changing position of end-effector, the proposed algorithm could generate the end-effector trajectory and joints' values as well. The results showed that the end-effector position and orientation trajectory and the joints variable had the same form as the sample one. This proved that we can absolutely apply DMP to solve the Inverse Kinematics and Motion planning for robots without solving complicated inverse kinematics problems.

Keywords Motion planning · Forward kinematics · Inverse kinematics (IK) · Dynamic movement primitives (DMPs)

1 Introduction

The Motion planning and inverse kinematics (IK) problem is important problem in the field of robotics. IK problem is understood as finding the joints variables so that the robot has the desired position and direction of the end-effector [1]. The geometric method and the iterative method (Jacobian) are the classic method commonly used

H. Nguyen · T. Nguyen (✉) · H. Dao · D. Nguyen
Ha Noi University of Science and Technology, Hanoi, Vietnam
e-mail: trung.nguyenthanh@hust.edu.vn

H. Pham
Hanoi University of Industry, Hanoi, Vietnam

B. T. Long et al. (eds.), *Proceedings of the 3rd Annual International Conference on Material, Machines and Methods for Sustainable Development (MMMS2022)*, Lecture Notes in Mechanical Engineering, https://doi.org/10.1007/978-3-031-57460-3_15

to solve IK problems [2–4]. However, these methods have the limitation that they are only applicable to simple robots. For the exoskeleton rehabilitation robot, finding the joint values solution is extremely necessary because the bad joint variable can cause injury to the patient. The joints values trajectories of the robot need to be stable and continuous. In [5, 6], we have proposed the optimal approaches as improved PSO and DE algorithms to solve the IK problem, we have tested and used these proposed algorithms against the endpoint trajectory of Activities of Daily Living (ADL) as measured by an Exoskeleton type Human Motion Capture (E-HMCS) device [7].

However, in order for the user to be able to move in the most natural way, the trajectories need to be similar to nature human activities, Dynamic Movement Primitives (DMPs) is a commonly used solution to solve this problem. Saveriano et al. [8] provided a systematic and comprehensive review of the existing literature and classified the status of research on the DMPs. In [9] the article's authors reviewed the motion planning methods to control exoskeleton, especially for the upper extremity, and identified challenges in path planning for the upper extremities. Lauretti et al. [10] the article's authors presented a demonstration learning approach for planning the upper limb exoskeleton built based on the computation of DMPs and machine learning techniques to generate trajectories for the task and patient based on learned trajectories.

In this study, we experimented with applying DMPs to generate motion trajectories of robots from measured daily activities. At the same time, we also create joint variables by using DMPs instead of using traditional methods to solve IK problem. The results show that the joint values, the position and orientation trajectory of the robot in the 2 cases are similar. The following sections of the rest of the paper: the exoskeleton robot arm model describes in Sect. 2. Section 3 will provide the theory of the DMP algorithm and the application of DMPs in the IK problem. Section 4 shows the expected results after comparing the use of DMP to create angular trajectory matching and the use of DMPs combined with IK to solve motion planning and inverse kinematics. Finally, we will draw conclusions in Sect. 5.

2 Human Arm with E-HMCS Model Description

2.1 E-HMCS Model

The human arm is composed of three anatomical joints (shoulder, elbow, and wrist joints) while ignoring the clavicle and clavicle muscles [11]. We also propose a 3D model to simulate the human arm to ensure the basic anthropometric parameters of Asians in general and especially of Vietnamese people in particular [12]. The length of the joints of the model is selected which basing on the length of the person's arm who has worn the E-HMCS. To evaluate the efficiency of the proposed algorithm and also determine the endpoint trajectories in the ADL operation, we have also developed the Exoskeleton-type Human Motion Measurement System (E-HMCS)

[13]. It is an exoskeleton measuring device made by 3D printing technology and has a low-cost, simple calculation process. The arm model of the person wearing the device is depicted as shown in Fig. 1 with kinematic parameters, and Denavit–Hartenberg parameters (DH) as shown in Table 1.

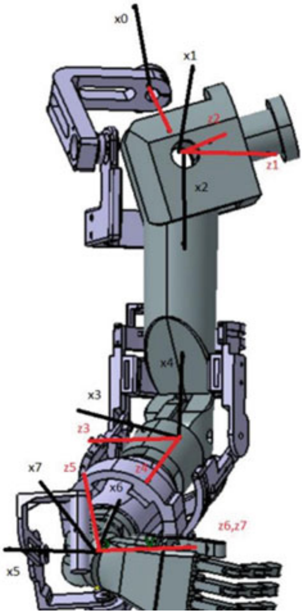

Fig. 1 Human arm model wearing the E-HMCS

Table 1 Denavit–Hartenberg parameters of E-HMCS

θ (°)	d (mm)	a (mm)	α (°)
$q1$	143	0	90
$90 - q2$	7	0	− 90
$q3$	286.9	0	− 90
$q4$	0	0	90
$90 + q5$	336	0	90
$90 + q6$	0	0	− 90
$q7$	0	0	0

3 Dynamic Movement Primitives to Motion Planning and Inverse Kinematics Problems

3.1 Dynamic Movement Primitives Algorithm

The theory behind DMP is well established in [14] where the heart of the model is a point-attractor system modulated with a nonlinear function to enable the generation of complex movements. One point-attractor system is the spring-damper system:

$$\ddot{y} = \alpha_y \big(\beta_y (g - y) - \dot{y}\big) + f \tag{1}$$

where α_y and β_y are positive constants, y and g are the initial and final points of the trajectory (system state and goal). The crux of the DMP framework is an additional nonlinear system used to define the forcing function f over time, giving the problem a well-defined structure that can be solved straightforwardly and easily generalized. The introduced system is called the canonical dynamical system

$$\dot{x} = \alpha_x x \tag{2}$$

The forcing function f is defined as a function of the canonical system:

$$f(x, g) = \frac{\sum_{i=1}^{N} \psi_i w_i}{\sum_{i=1}^{N} \psi_i} x(g - y_0) \tag{3}$$

where y_0 is the initial position of the system:

$$\psi_i = \exp\big(-h_i (x - c_i)^2\big) \tag{4}$$

And w_i is a weighting for a given basis function ψ_i. The ψ_i equation above defines a Gaussian centered at c_i, where h_i is the variance. w_i has solution:

$$w_i = \frac{S^T \psi_i \mathbf{f}_d}{S^T \psi_i \mathbf{S}} \tag{5}$$

where:

$$\mathrm{s} = \begin{pmatrix} x_{t_0}(g - y_0) \\ \vdots \\ x_{t_N}(g - y_0) \end{pmatrix}, \quad \psi_i = \begin{pmatrix} \psi_i(t_0) & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \psi_i(t_n) \end{pmatrix} \tag{6}$$

3.2 Dynamic Movement Primitives for Inverse Kinematic

In most of the previous studies, the DMP algorithm often used to create a motion trajectory for the robot and be able to control the robot to move along that trajectory, we have to solve the inverse kinematics problem. For non-skeletal robotic arms, joint values solutions need to be naturally obtained to provide comfortable to the user. So why don't we apply DMP to create a most natural trajectories?

One of the challenges when applying DMP to the inverse kinematics problem is that we don't know the value of the joint values at the end of the moving trajectories (the starting point of the trajectory is the current state of the robot). To solve this problem, the research applied the commonly used algorithms DMPs [15] and some IK methods to find the joint variables at the end point of the motion trajectory. After obtaining this value, we applied the DMP algorithm to achieve the desired match trajectory. We tested and compared the results in the following 2 cases:

- To use the DMPs algorithm to create the position trajectory and orientation trajectory of the robot (x, y, z) combined with the inverse kinematics problem-solving methods to find the joints values
- Use the algorithm for IK problem to find the joint values at the end point of the new position trajectory, then use DMPs to generate the joint values trajectory (q).

4 Experiments and Results

4.1 Experiments Setup

In this study, to evaluate the efficiency of the algorithms, we used the measuring results of the water cup movement task measured by E-HMCS (Fig. 2). With the matching joint values (q) measured by kinematic calculations, we obtain the positional and orientation trajectories of the respective end-effector (E). After having these parameters, we performed the following steps:

- Step 1: Run the algorithm to find DMP parameters corresponding to: joint values (DMP with q), both position and orientation (DMP with P, O)

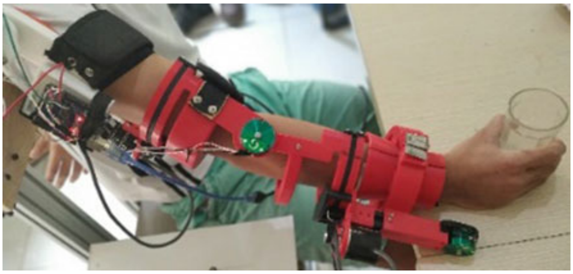

Fig. 2 Experiment for the water cup movement tasks

- Step 2: Generate new position and orientation trajectories of the end-effector (E_dmp) with the goal being the point around the measured endpoint of the end-effector (e_goal)
- Step 3: Solve inverse kinematic to find goal joint values (q_goal) from e_goal.
- Step 4: Generate new joint values trajectory (q_dmp) with q_goal and DMPs's parameters
- Step 5: Calculate inverse kinematic and forward kinematics with newly created trajectories: use forward kinematics to calculate the end-effector from q_dmp (q_dmp > FK > E_q_DMP) Calculate inverse kinematics to find to joint values from E_dmp (E_dmp > IK > q_ik_dmp,)
- Compare joint values and end-effectors with each other. (E_dmp and E_q_DMP, q_dmp and q_ik_dmp)

4.2 Results

We will compare and evaluate the obtained results based on the position trajectories of the end-effector and the resulting joints values trajectories. From Fig. 3, we can see that the trajectories (E_measure, E_dmp and E_q_dmp) all have similar form, although E_dmp and E_q_dmp share the same endpoint, but these two trajectories do not coincide. It can be explained causing by the error when generating data by DMP algorithm and the calculation process forward and reverse kinematics. When looking at the cosine similarity table of the end-effector orbital, we can see a specific result when E_q_dmp has a higher average score than E_dmp, two cosine similarity of trajectories are very high.

Figure 4 shows us the graph of the joint values trajectory in the cases. The q_dmp trajectories are smooth and stable, but the q_ik_dmp orbital has any unusually abrupt changes. The reason can be the inverse kinematics calculation. When we look at the scores cosine similarity again, we see that q_dmp is significantly higher than q_ik_dmp, especially at joints $q1$ and $q3$ exists a significant difference due to their instability in the q_ik_dmp orbital (Tables 2 and 3).

5 Conclusion

In this study, the application of the DMPs algorithm generates the position and orientation trajectory of the end-effector like the joint variables trajectory. Therefore, the research team can conclude that the complete DMP can be applied to solve the invert kinematic problem and generate motion planning of the robot. The results showed that using DMP with joints helps to produce end-effector trajectories and joints, which are more similar to the measured values. It makes it possible for humans to manipulate the robot more naturally and comfortably. In the future, we can use DMP to generate a lot of end-effector position and orientation trajectories as well as

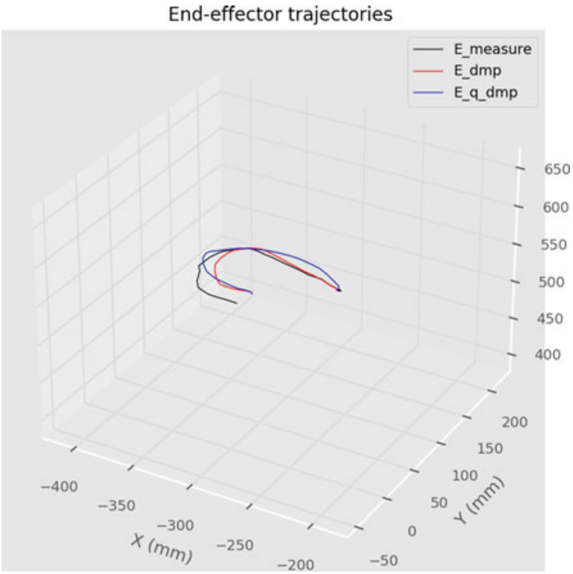

Fig. 3 Position trajectories of the end-effector

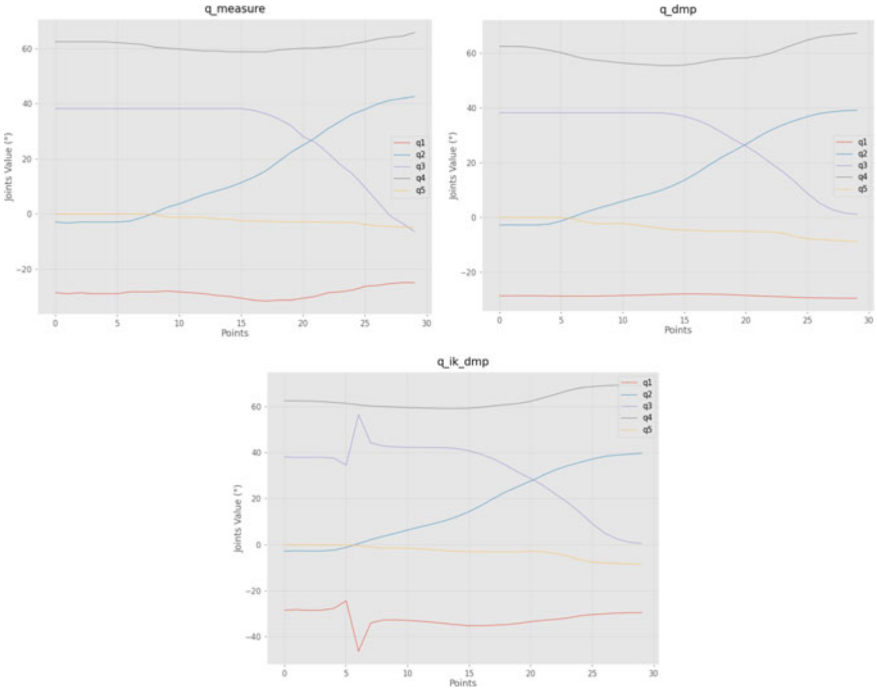

Fig. 4 Joints values of measure trajectories, DMP trajectories and IK DMP trajectories

Table 2 Cosine simility position trajectories end-effector

E_measure	x	y	Z	Avg
E_dmp	0.9991	0.9907	0.9998	0.9965
E_q_dmp	0.9992	0.9972	0.999	0.9987

Table 3 Cosine simility of joint's values trajectories

q_measure	$q1$	$q2$	$q3$	$q4$	$q5$	Avg
q_dmp	0.997	0.9956	0.9977	0.9994	0.9913	0.9962
q_ik_dmp	0.9938	0.9941	0.9934	0.9993	0.9732	0.9907

joint variable trajectories for control as well as data to train neural networks to help solve complex robot problems.

Acknowledgements Research reported in this paper was supported by Ministry of Science and Technology of Vietnam, under award number [ĐTĐLCN.28/20].

References

1. R. Köker, T. Çakar, A neuro-genetic-simulated annealing approach to the inverse kinematics solution of robots: a simulation based study. Eng. Comput. **32**, 553–565 (2016)
2. S.-S. Liaw, H.-M. Huang, G.-D. Chen, Surveying instructor and learner attitudes toward e-learning. Comput. Educ. **49**, 1066–1080 (2007)
3. J.J. Rubio, A.G. Bravo, J. Pacheco, C. Aguilar, Passivity analysis and modeling of robotic arms. IEEE Lat. Am. Trans. **12**, 1381–1389 (2014)
4. Y.L. Kou, T.P. Lin, C.Y. Wu, Experimental and numerical study on the semi-closed loop control of a planar robot manipulator. Math. Probl. Eng. **114**, 1–9 (2014). https://doi.org/10.1155/2014/769038
5. T.T. Nguyen, N.L. Tao, V.T. Nguyen, N.T. Bui, V.H. Nguyen, D. Watanabe, Apply PSO algorithm with searching space improvements on a 5 degrees of freedom robot, in *Proceedings of the 2020 3rd International Conference on Intelligent Robotic and Control Engineering (IRCE)* (2020)
6. T.T. Nguyen, V.H. Nguyen, X.H. Nguyen, Comparing the results of applying DE, PSO and proposed Pro DE, Pro PSO algorithms for inverse kinematics problem of a 5-DOF scara robot, in *Proceedings of the 2020 International Conference on Advanced Mechatronic Systems (ICAMechS)* (2020)
7. T. Nguyen, H. Nguyen, K. Dang, P. Nguyen, H. Pham, A. Bui, Simulation and experiment in solving inverse kinematic for human upper limb by using optimization algorithm, in *ACIIDS 2021: Intelligent Information and Database Systems*, pp. 556–568 (2021)
8. M. Saveriano, F.J. Abu-Dakka, A. Kramberger, L. Peternel, Dynamic movement primitives in robotics: a tutorial survey. Comput. Sci. (2021)
9. C. Nguiadem, M. Raison, S. Achiche, Motion planning of upper-limb exoskeleton robots: a review. Appl. Sci. **10**, 7626 (2020). https://doi.org/10.3390/app10217626
10. C. Lauretti, F. Cordella, A.L. Ciancio, E. Trigili, J.M. Catalan, F.J. Badesa, S. Crea, S.M. Pagliara, S. Sterzi, N. Vitiello, A.N. Garcia, L. Zollo, Learning by demonstration for motion

planning of upper-limb exoskeletons. Front. Neurorobot. (2018). https://doi.org/10.3389/fnbot.2018.00005

11. N. Trung, D. Hiep, D. Thien, B. Tam, W. Dai, Design a human arm model supporting the design process of upper limb rehabilitation robot, in *Proceedings of the 14th South Asian Technical University Consortium Symposium 2020 (SEATUC 2020)*, pp. 2186–7631 (2020)
12. Tổng liên đoàn lao động Việt Nam, Viện nghiên cứu khoa học kỹ thuật bảo hộ lao động (Vietnam), Atlat nhân trắc học người Việt Nam trong lứa tuổi lao động: dấu hiệu tầm hoạt động khớp và trường thị giác, Nhà xuất bản Khoa học và kỹ thuật (2002)
13. N. Thanh-Trung, B. Ngoc-Tam, D. Watanabe, Design and manufacture a cheap equipment to measure human arm motion in developing countries, in *The 4th International Conference on Mechatronics Systems and Control Engineering (ICMSCE 2021)*, pp. 48–53 (2021)
14. A.J. Ijspeert, J. Nakanishi, H. Hoffmann, S.P. Schaal, Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput. **25**(2), 328–373 (2013)
15. C. Lopez-Franco, J. Hernandez-Barragan, A.Y. Alanis, N. Arana-Daniel, A soft computing approach for inverse kinematics of robot manipulators. Eng. Appl. Artif. Intell. **74**, 104–120 (2018)