Coordinate Change Dynamic Movement Primitives – A Leader-Follower Approach

You Zhou, Martin Do, and Tamim Asfour

***Abstract*— Dynamic movement primitives prove to be a useful and effective way to represent a movement of a given agent. However, the original DMP formulation does not take the interaction among multiple agents into the consideration. Thus, many researchers focus on the development of a coupling term for the underlying dynamical system and its associated learning strategies. The result is highly dependent on the quality of the learning methods. In this paper, we present a new way to formulate and realize interactive movement primitive in a leader-follower configuration, where the relationship between the follower and the leader is explicitly represented via the new formulation. This new formulation does not only simplify the learning process, but it also meets the requirements of several applications. We separately tested our new formulation in the context of the handover task and the wiping task. The results prove the flexibility and simplicity of the new formulation.**

I. Introduction

Learning of tasks from human observation and sensorimotor experience is an essential skill in order to enable a humanoid robot towards assistance of a human within its environment. Especially when it comes to coordinated tasks which involve the cooperation and interaction of multiple agents, learning from demonstration and experience can boost the skill acquisition process.

The learning of skills for single agent, has been thoroughly addressed in previous approaches in the field of imitation learning or programming by demonstration [1]. In this context, generic models and representations have been proposed which are capable of encoding a demonstrated trajectory and which can be parameterized in order to adapt the encoded task to different situations such as Gaussian Mixture Models (see [2]) and Hidden Markov Models (see [3], [4]). In recent years, a popular approach using dynamical systems has been introduced by [5], [6] and [7] in the form of the Dynamic Movement Primitives (DMPs). A DMP uses a spring damping system to describe the goal attractor for a motion primitive, whose variety is captured by a coupling force term. With a regression learning algorithm such as locally weighted regression, the force term can be learned from a single demonstration.

A DMP has several beneficial properties. Due to the spring damping system, attraction to a specified goal is guaranteed. In addition, the force term profile encodes task-specific characteristics which allows the reproduction of topologically similar trajectories for different goal and start positions. Furthermore, by integrating a canonical system a DMP is time-independent which enables the generation of trajectories adapted to different task requirements.

However, the learned force term is only dependent on the canonical system, which decays or increases if time goes on. Hence, in interactive tasks where a moving goal represented by another agent might eventually alter the trajectory's shape which is determined by the force term profile.

The learning of such tasks for multiple agents requires a task representation which is capable of encoding the movements as well as the relations between the considered agents. In this work, we propose a novel method to combine multiple DMPs based on the leader-follower configuration. The task-space relation between a leading and a following agent is described by a translational and a rotational component which depend on the phase of the canonical system.

The paper is organized as follows. Section II provides an overview of existing approaches which have addressed the representation and encoding of coordinated multi-agent tasks. In Section III, our approach in the form of the Coordinate Changed Dynamic Movement Primitive (CCDMP) is introduced. We will describe the formulation, the corresponding learning strategy and possible applications. Subsequently, evaluation and experiments are presented in Section IV. In conclusions, the work is summarized and notes to future works are given.

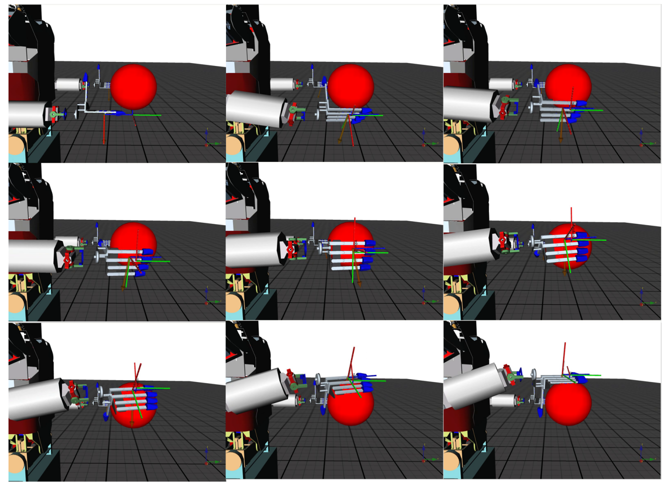

Fig. 1: The robot's hand follows the trajectory generated by CCDMP, which enables the learned wiping pattern to adapt to the ball surface.

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement no 643666 (I-Support).

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. `you.zhou@kit.edu`

II. Related Work

Recent research efforts, in particular, related to the field of learning from human observation have been dedicated to the modelling and the representation of complex manipulation and cooperative tasks which involve mostly two agents. In this context, the original DMP formulation has been extended in several works to address the requirements imposed by interactive tasks.

[8] introduced interactive DMPs for the representation of cooperative tasks by incorporating two dynamical systems in order to encode the movements of the involved agents. To ensure that both agents reach a common goal configuration, the dynamical systems are extended by a force term which emerges from a virtual spring with a fixed, task-specific length or distance spanned between the two agents.

In [9], the interaction forces between agents are encoded in a continuous coupling term which is added to the DMP formulation. The coupling term is gradually learned through interactive learning control (ILC) based on sensory feedback. In this case, the impact of the leader's behaviour on the follower is considered as an external force changing the follower's acceleration. The coupling term is merely phase-dependent and does not consider the relation between the agents in task space which makes the adaptation of the proposed DMP formulation to different scenarios more difficult.

In [10], interactive movement primitives are presented which combine separately trained DMPs for each agent in cooperative tasks with a predictive distribution. The distribution allows the inference DMP parameters of one agent based on the observed trajectory of the other agent in order to obtain a temporally synchronized task execution. However, the presented approach is restricted to a single interaction pattern and not easily generalizable. Thus, in [11], interactive movement primitives and Gaussian mixture models are combined to obtain a mixture of interactive movement primitives. Using this representation, the variance in multiple demonstrations of an interactive task can be encoded.

In [12], handover tasks are addressed by reformulating DMP with weighted shape and goal attractors. The introduction of different weighing function allows better control on the evolution of the DMP and the transition between following the encoded trajectory and reaching the goal.

Instead of using an external force, a more general idea is to define a mapping from the leader to the follower with regard to position and orientation. CCDMP realizes this idea by using coordinate transformation matrix R_G^L, where G is for the global coordinate and L for the leader's coordinate, to transform the follower's state from the leader's coordinate to the global one. The entries of this matrix are changed with the leader's state.

III. The CCDMP Formulation

In this section, we will briefly describe the formulation of the Dynamic Movement Primitive (DMP). Subsequently, the Coordinate Changed Dynamic Movement Primitive (CCDMP) formulation is introduced.

A. Dynamic Movement Primitive

The original formulation of DMP mainly consists of the transformation systems, the corresponding force terms, and a canonical system. A transformation system can be described as follows:

$$\begin{array}{rcl}\tau \cdot \dot{v} & = & K \cdot (g - y) - D \cdot v + scale * f \\ \tau \cdot \dot{y} & = & v,\end{array} \tag{1}$$

where $(v, y)^T$ is the state describing the velocity and the position of an agent which is attracted to the current goal g by critically-damped spring damping system with spring constant K and a damper factor D. The temporal vector tau can be used to scale the evolution of the dynamical system in time, and the scaling term $scale$ is a matrix which avoids unexpected movement when adapting to new start and goal. The force term f can be described as follows:

$$f(u) = \frac{\sum_{i=1}^{N} \Psi_i(u) w_i}{\sum_{i=1}^{N} \Psi_i} u, \tag{2}$$

where Ψ_i is the i-th kernel function, N is the total number of the functions, and u is the canonical value determined by the canonical system. The canonical system is used to make the entire dynamical system time-invariant and drives the evolution of the transformation systems in a synchronized way. The formulation of the canonical system depends on the type of the movement, discrete or rhythmic, to be represented. For example, the canonical system for a discrete movement primitive has following form:

$$\tau \cdot \dot{u} = -\alpha_u u, \tag{3}$$

where α_u is the parameter of the canonical system.

B. General Idea

We assume that in an coordinated task between two agent, one agent is taking a leading role and the other one is following. The follower's state $(y_f) \in \mathbb{R}^m$ depends on the leader's state $(y_l) \in \mathbb{R}^n$. The follower needs to be adjusted based on the leader's state. The adjustment of the follower's state is defined as a function in $\mathbb{R}^m$. Hence, the mapping from the leader's state onto the follower's adjustment is defined as following:

$$\begin{array}{rccl}\phi : & \mathbb{R}^n & \rightarrow & \mathbb{H} : \{h : \mathbb{R}^m \rightarrow \mathbb{R}^m\} \\ & (x_1^l, x_2^l, ..., x_n^l)^T & \rightarrow & h(x_1^l, x_2^l, ..., x_n^l)(\cdot),\end{array} \tag{4}$$

where the function h on $\mathbb{R}^m$, is determined, once the leader's state is fixed. Thus, a trajectory in $\mathbb{R}^n$ is corresponding to a trajectory in the continuous function space $\mathbb{H}$, if ϕ is a continuous mapping.

In the Cartesian space, n is equal to six. Three dimensions are for position, the other three for orientation. Considering only the follower's translation in the homogeneous coordinate, m is equal to four. Consequently, the homogeneous transformation matrix acts as a linear mapping in $\mathbb{R}^4$, which is determined by the leader's state in $\mathbb{R}^6$.

The formulation of CCDMP is based on this general idea, which is described in detail in the next section.

C. Coordinate Changed Dynamic Movement Primitive

CCDMP consists of two parts, leader's DMP and follower's DMP. The leader's DMP uses the basic formulation with a canonical system for discrete or periodic movement. The follower's DMP changes its pose by multiplying both sides of the equations with a coordinate transformation matrix.

$$\begin{aligned} \tau \cdot R_G^L \cdot \dot{v}^G &= R_G^L \cdot (K \cdot (g^G - y^G) - D \cdot v^G + \\ &\quad scale^G \cdot f^G) \\ \tau \cdot R_G^L \cdot \dot{y}^G &= R_G^L \cdot v^G, \end{aligned} \tag{5}$$

In equation 5, R_G^L is a coordinate transformation, which is a 4×4 homogeneous transformation matrix, and v^G is the current velocity in the global coordinate system. A homogeneous vector with fourth element zero is used to represent velocity or acceleration. The force term in the leader's coordinate system can be different from the force term in the global coordinate system, namely f^G. This difference is caused by the rotation of the leader's coordinate system in the global one. The global force term is scaled by the global scaling matrix $scale^G$. It is difficult to learn f^G in general. However, the force term in the leader's coordinate system f^L is easily learned from the follower's local movement. The goal in the global coordinate g^G is dependent on the leader's position and orientation, which is unknown to the follower at the beginning, while the follower's local goal $g^L = R_G^L \cdot g^G$ is already defined. Hence, with R_G^L and f^G the formulation original transformation system in Eq. 1 is changed as follows:

$$\begin{aligned} \tau \cdot R_G^L \cdot \dot{v}^G &= K \cdot (g^L - R_G^L \cdot y^G) - D \cdot R_G^L \cdot v^G \\ &\quad + scale^L \cdot f^L \\ \tau \cdot R_G^L \cdot \dot{y}^G &= R_G^L \cdot v^G. \end{aligned} \tag{6}$$

For the purpose of synchronization, the canonical system of the follower should coincide with the one of the leader. However, follower and leader are allowed to have different motion patterns: discrete or periodic. One solution is to have a multi-dimensional canonical system. Consequently, the ratio between the temporal factors governs the synchronization. Since the homogeneous matrix R_G^L is determined by the position and the orientation of the leader, the dimensions in this DMP formulation cannot be treated independently when the leader's rotation is not trivial. The leader's DMP formulation is given by the following equations:

$$\begin{aligned} \tau_l \cdot \dot{v}_l &= K_l \cdot (g_l - y_l) - D_l \cdot v_l + scale_l \cdot f_l \\ \tau_l \cdot \dot{y}_l &= v_l. \end{aligned} \tag{7}$$

This DMP encodes six dimensions, three for the leader's position and the remaining three for the leader's orientation. The leader's position determines the origin of the leader's coordinate system.

The solution to the follower's DMP is given by the Euler method:

$$\begin{aligned} R_{G,t+1}^L \cdot v_{t+1}^G &= \tfrac{\Delta t}{\tau} \cdot (K \cdot (g_t^L - R_{G,t}^L \cdot y_t^G) \\ &\quad - D \cdot R_{G,t}^L \cdot v_t^G + scale^L \cdot f_t^L) \\ &\quad + R_{G,t}^L \cdot v_t^G \\ R_{G,t+1}^L \cdot y_{t+1}^G &= \tfrac{\Delta t}{\tau} \cdot R_{G,t}^L \cdot v_t^G + R_{G,t}^L \cdot y_t^G. \end{aligned} \tag{8}$$

If $\Delta R_t = (R_{G,t+1}^L)^T \cdot R_{G,t}^L$, then the above equations system can be reformulated as:

$$\begin{aligned} v_{t+1}^G &= \tfrac{\Delta t}{\tau} \cdot (K \cdot ((R_{G,t+1}^L)^T \cdot g_t^L - \Delta R_t \cdot y_t^G) \\ &\quad - D \cdot \Delta R_t \cdot v_t^G + (R_{G,t+1}^L)^T \cdot scale^L \cdot f_t^L) \\ &\quad + \Delta R_t \cdot v_t^G \\ y_{t+1}^G &= \tfrac{\Delta t}{\tau} \cdot \Delta R_t \cdot v_t^G + \Delta R_t \cdot y_t^G. \end{aligned} \tag{9}$$

When the leader's movement includes only translation, each dimension can be treated separately. Thus, the coordinate transformation matrix R can be represented by adding a term to the original value, for example, that $R \cdot y = y + C$. After extracting and adding all the terms (Cs) given by the coordinate transformation, a coupling term appears in the CCDMP formulation. As a result, this translation CCDMP looks very similar to the one described in [9]. If the translation is a pure displacement of leader's position, CCDMP generates a new trajectory by the superposition of both trajectories.

Instead of trying to keep the dynamic features (shape and acceleration profiles) of a trajectory in the global frame after adapting it to a dynamic environment, CCDMP maintains the trajectories' features in the local frame, namely the leader's coordinate system. Hence, from the leader's view, the follower can always draw similar trajectories learned from the demonstration, no matter what position and orientation the leader takes. In Fig. 2, we show an example which benefits from this formulation.

Moreover, CCDMP allows the dynamic movement of the leader, which is used to encode a cooperative task.

D. Application

CCDMP meets a large range of applications including the human-robot interaction. Unlike coupled DMP, instead of directly learning the relation between two agents, CCDMP formulation makes the leader-follower relation more explicit. The leader's moving pattern, rhythmic or discrete, decides the follower's global trajectory. We can construct four different types of leader-follower systems based on two types of movements, discrete-discrete, discrete-periodic, periodic-discrete and periodic-periodic.

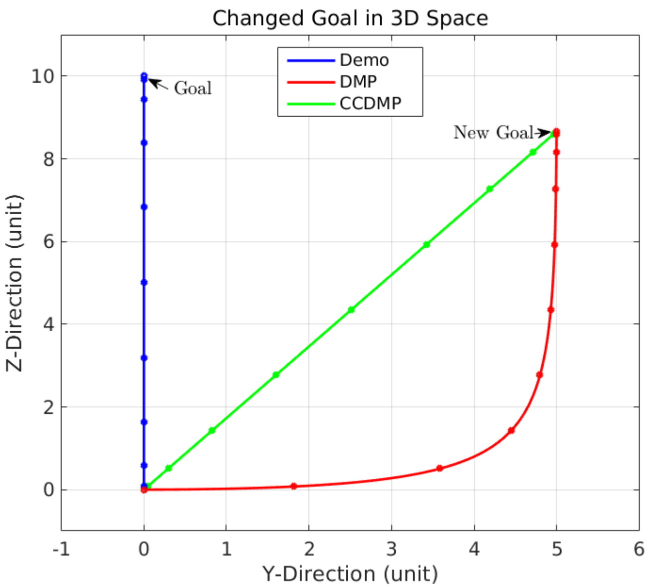

Fig. 2: If the changed goal is in the 3D space, DMP fails to generate a reasonable trajectory because of the independence of dimensions in the multi-dimensional DMP formulation. Conversely, CCDMP can easily solve the problem by considering the start position as a leader, whose orientation together with the follower's local goal constructs a polar coordinate system describing the changed goal of the follower's DMP in the global frame.

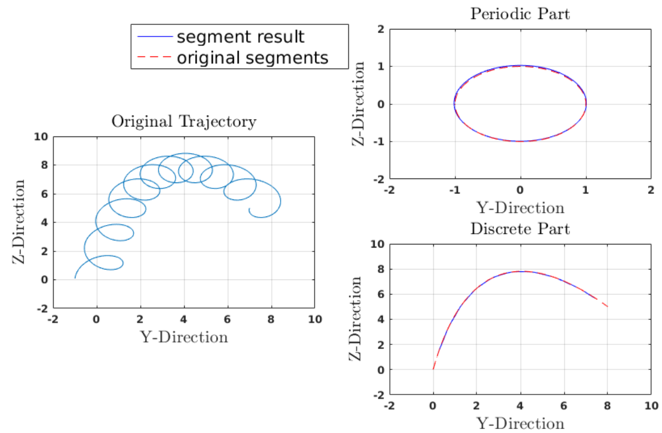

Fig. 3: Separate the discrete part and the periodic part from the original wiping pattern. The window's size is 101 for 1000 sample points. The original trajectory in this figure is created to test the moving average technique. In a human's demonstration (see Fig 8), however, it is difficult to keep the same periodic pattern. Hence, the perfect reproduction after trajectory separation is not possible. However, we do not focus on an exact reproduction, which can be easily done by a traditional discrete DMP. In contrast, we focus on the generalization and learning the skill.

Discrete-discrete system can be used in a situation concerning multiple end-effectors with various discrete movements, such as handover task and bimanual manipulation. The purpose of this system is to reach the common attractor in the task space. CCDMP can also be used to represent complex discrete-periodic tasks of physically coupled agents such as the arm and the hand. For example, a wiping task is led by the arm movement which sets the anchor point of a following, periodic wiping movement executed by the hand. A further example is a bimanual task with one hand performing a periodic movement such as stirring, where the hand holding the bowl represents the leader and specifies the position and orientation of the other hand.

E. Learning Strategy

The learning strategy of CCDMP is straightforward, if the CCDMP is used to encode the movements of two independent agents. A single demonstration of the interaction between the leader and the follower is sufficient for the learning of the two DMPs. The DMP for the leader is learned directly from the leader's trajectory, while the follower's local DMP is obtained by transforming follower's state into the leader's coordinate system. The force term profile of the local DMP is kept in the CCDMP.

To train a CCDMP for complex discrete-periodic task such as wiping, a demonstration of this task is given by a global trajectory. The trajectory has to be decomposed in a global leader movement and local periodic motion pattern. One solution is to use moving-average technique to extract the approximated discrete part from the original movement. Then the periodic part can be obtained by subtracting the discrete part from the trajectory. A discrete DMP for the leader and a periodic DMP for the follower are separately learned from the discrete and periodic parts.

The problem of the moving average approach is that it is difficult to decide the reasonable window's size. If the window's size is too small, then the discrete part might contain undesired jerks. If the window's size is too large, then the discrete part might be much shorter than the original trajectory. In section IV, we address this problem by analysing the frequency of the given wiping movement. The concept of movement's decomposition is depicted in Fig. 3.

IV. Experiments

In this section, two different examples are illustrated to demonstrate the strength of CCDMP formulation: the handover and the wiping task. The first one is realized by a discrete-discrete system. Different from other DMP formulations, CCDMPs can not only consider the translation of the other agent's hand, but also its orientation. For the wiping task, CCDMP facilitates the adaptation of a wiping pattern to a changing and moving surface, e.g. a service robot wiping human being's back, whose surface is not only complex, but also moving all the times.

A. Handover Task

One of the important applications in human-robot interaction is the handover task. The handover task is difficult, since the robot must be able to predict the exchange position

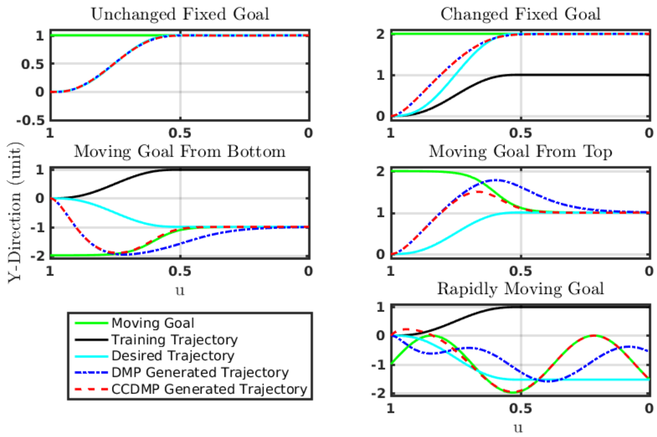

Fig. 4: Handover task in different cases. The green line represents the moving goal. The training trajectory, the black solid line is a minimum jerk trajectory. The desired trajectory, the cyan one, is also a minimum-jerk trajectory connecting new start position and goal. The blue dotted line is generated by the original DMP and the red dotted one is the result given by the CCDMP. As diagrams show, CCDMP is better than DMP according to the criterion stated in the text. (u is the canonical value)

when the partner hands over an object. At the same time, the movement of the robot's arm should look smooth and naturally.

The handover task is similar to the docking problem, where the robot should manipulate its hand to find a correct or suitable position and orientation to receive the object from the passer or pass the object to the receiver. Not only the exchange position but also the exchange orientation plays an important role during handing over the object.

Instead of predicting the exchange position, observing the movement of the leader's hand is a much simpler way to accomplish the task. It is not novel to generate a trajectory for a moving goal using DMP. In [12], the authors argued that the original DMP formulation is very focused on reaching the goal which prevents the generation of a shape-preserving trajectory. Hence, they separated the shape attractor from the goal attractor in the original formulation and used a weight to control both parts. However, the shape-preserving trajectory might not be the best choice for the handover task, e.g. if leader and follower exchange their standing positions. As a result, an appropriate handover task trajectory should fulfill these two criteria:

1) The generated trajectory should finally reach the goal. This should be the dominant requirement;
2) The shape of the generated trajectory should be close to the shape of the adapted training trajectory for a new start and goal position. For example, if the training trajectory is a minimum-jerk trajectory, the good generated shape should be close to the shape of a minimum-jerk trajectory with new start position and goal.

Instead of considering the leader's hand as a moving goal, we consider the leader's hand as the origin of a local coordinate. The follower executes its own DMP in this local coordinate system. Fig. 4 shows that CCDMP outperforms the original DMP formulation with regard to the two criteria above. The CCDMP generated trajectories that are closer to the desired trajectory compared to the ones generated by the original DMP in the first four cases. In the last case, DMP cannot adapt to the rapidly moving goal, while CCDMP reaches the leader in the early stage and then follows it in the rest of the time. In fact, CCDMP guarantees to reach the leader's state theoretically, because the follower must finish its local movement after a period, whose length is dependent on the follower's temporal factor. In contrast, other DMP formulations in the papers mentioned before cannot ensure the reachability of the unexpected moving goal because of the global force term. The stable property is only guaranteed when the canonical system runs out and the canonical value is close to zero. The DMP without the force term is a PD controller when the derivative of goal movement is also added into the original formulation (Eq 10).

$$\begin{aligned} \tau \cdot \dot{v} &= K_p \cdot (g - y) - K_d \cdot (\dot{g} - \dot{y}) + scale \cdot f \\ \tau \cdot \dot{y} &= v. \end{aligned} \tag{10}$$

The other problem of the handover task is the leader's orientation, which has been given less consideration in previous approaches. As mentioned before, handover task is similar as the docking problem. The robot must be able to choose a correct position for different leader's orientation to receive or pass the object. With CCDMP, it is easily solved, since the change of the leader's orientation will rotate the follower's trajectory and enforce the follower reach the leader from the similar direction as in the demonstration. Fig. 5 shows the docking problem solved by CCDMP.

CCDMP mentioned here only concerns the follower's position change, which is dependent on the leader's position and orientation. The follower's orientation is not covered in the above CCDMP formulation. In fact, it is difficult to decide the follower's orientation without losing the capability of generalization. For example, in a handover task, the follower's orientation is sometimes dependent on the target object, even when its position is determined. One solution to this problem is to learn another CCDMP related to the follower's orientation change in the leader's local coordinate system. The learning can be done easily by calculating the orientation difference between follower and leader at each time stamp and learning a DMP on this orientation difference trajectory. The CCDMP's formulation should be also changed in this case. Instead of only concerning the position vector, a more general version of CCDMP directly manipulates the transformation matrix including translation and rotation. This generalized CCDMP is not concerned in this paper and it might be our future work.

In Fig 5, we show that the follower's trajectory is altered by the leader's orientation change when using CCDMP. The follower's orientation change is designed to be always against the leader's orientation.

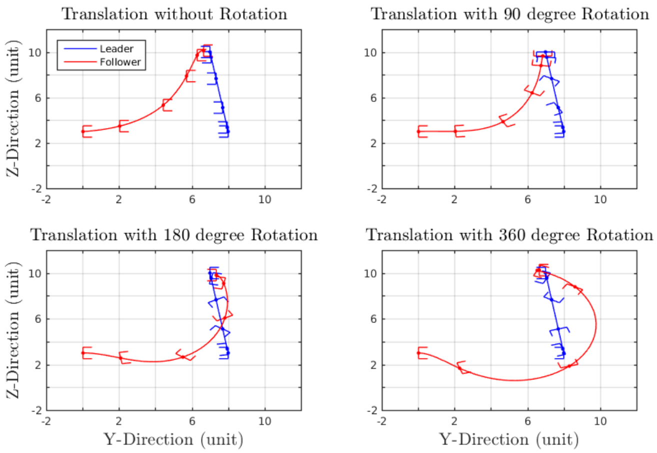

Fig. 5: The red dots represent the follower's positions at different time points, while the blue dots are for the leader's positions. Two rectangles with one missing edge are separately used to represent the orientation of both agents. The direction of the CCDMP generated follower's trajectory is always against the direction of the leader's open mouth. The final trajectory is projected onto the YZ-plane. The follower's orientation is designed to be always against the leader's orientation.

B. Wiping Task

As mentioned before, CCDMP can be used to generate different wiping patterns. The wiping movement can be considered as a simple periodic pattern moving along a trajectory, which might be represented by a discrete motion primitive. In CCDMP formulation, this discrete DMP is regarded as the leader and the follower generates the periodic part. Compared with a periodic DMP with a moving anchor point, the movement of the anchor point in CCDMP is also described by a movement primitive, which means that it can also be learned by observing and segmenting the wiping movement of a demonstrator. One solution mentioned before is to use the moving average technique to extract the discrete part from the original wiping movement and learn a DMP based on it. After learning both DMPs, different wiping pattern can be generated by adjusting the parameters in both DMPs. Fig. 6 shows the generated trajectory according to different parameter settings. After synchronizing the canonical systems, the temporal factor ratio ($\frac{\tau_{follower}}{\tau_{leader}}$) decides the final wiping movement.

The discrete and periodic parts of the CCDMPs can be replaced with more suitable discrete respectively periodic DMPs. For example, for a given surface, a discrete DMP can be trained which follows the object surface. This way, CCDMPs can be constructed with are immediately adapted to registered surface. Analogously, the periodic part can replaced with the one from another CCDMPs in order to meet changing task constraints.

Another application of CCDMP in wiping or washing tasks is the adaptation to the movement of the other agent. This is similar to the above handover task. For example,

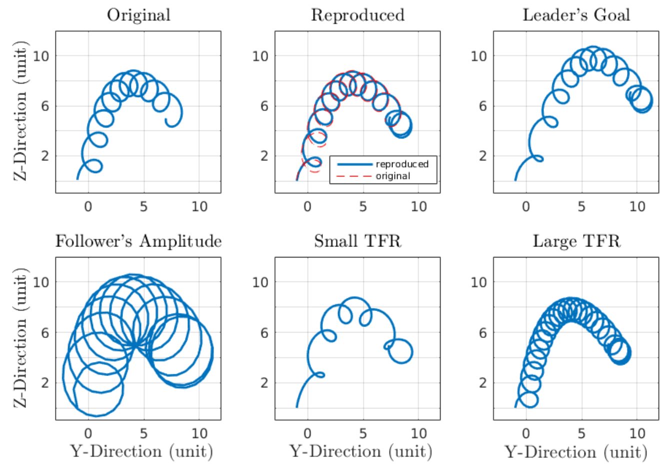

Fig. 6: Different trajectories generated by the same CCDMP with different parameters. The most top left diagram shows the original trajectory. The second diagram in the first line is the reproduction of the original trajectory by CCDMP. In the third diagram, we change the goal to a new position. In the bottom left diagram, the follower's amplitude is the triple of the original amplitude. The last two diagrams show the effect of the temporal factors' ratio (TFR). The small temporal factor ratio ($\frac{\tau_f}{\tau_l} = 0.5$, where the character f is for the follower and l is for the leader) generates a trajectory with sparse periodic patterns, while the large temporal factor ratio ($\frac{\tau_f}{\tau_l} = 3.0$) is corresponding to a dense trajectory.

the robot must be able to adapt its wiping movement to the movement of the object surface, e.g. a robot washing a human's back. The movement of the human's back is represented by a discrete DMP in CCDMP formulation and considered as a leader. Fig. 7 shows that a periodic movement is adapted to a moving surface represented by an another discrete DMP.

The surface movement can be represented by a discrete DMP, while the surface itself can be represented by an another discrete DMP. They are both the leaders of the simple periodic pattern. Furthermore, the surface's movement should lead the movement on the surface. Hence, the CCDMP formulation can be extended to a hierarchical leader-follower system. The leader in one CCDMP formulation is the follower in another formulation.

In the following, we will use CCDMP in a simulator to adapt a wiping pattern to a ball surface. In this experiment, the discrete trajectory for the ball surface is manually designed and learned by the robot, while the periodic part is learned by extracting it from the original wash back movement in our motion database.

As mentioned before, we use mean average technique to extract the discrete movement. The rest part is obtained by subtracting discrete part from the original trajectory. Because the wiping pattern is repeated several times in the demonstration, it is necessary to decide its frequency. In order to get the frequency, we observe the Fourier transformation of the rest part and find the frequency with the maximal spectrum.

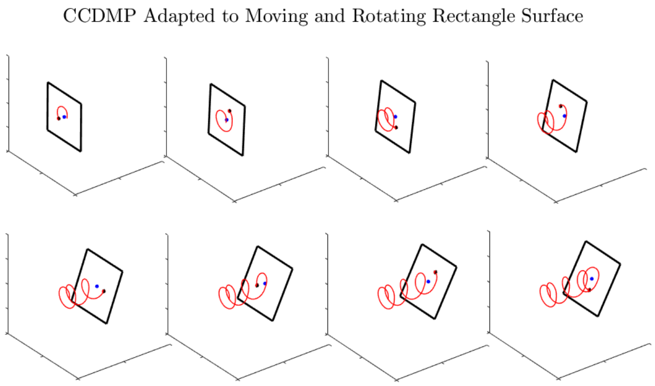

Fig. 7: The periodic movement is adapted to a moving and rotating surface described by an another discrete DMP. The red line is the global trajectory of the follower, whose responsibility is to generate a wiping pattern on the rectangular surface. The blue point is the origin of this circular pattern.

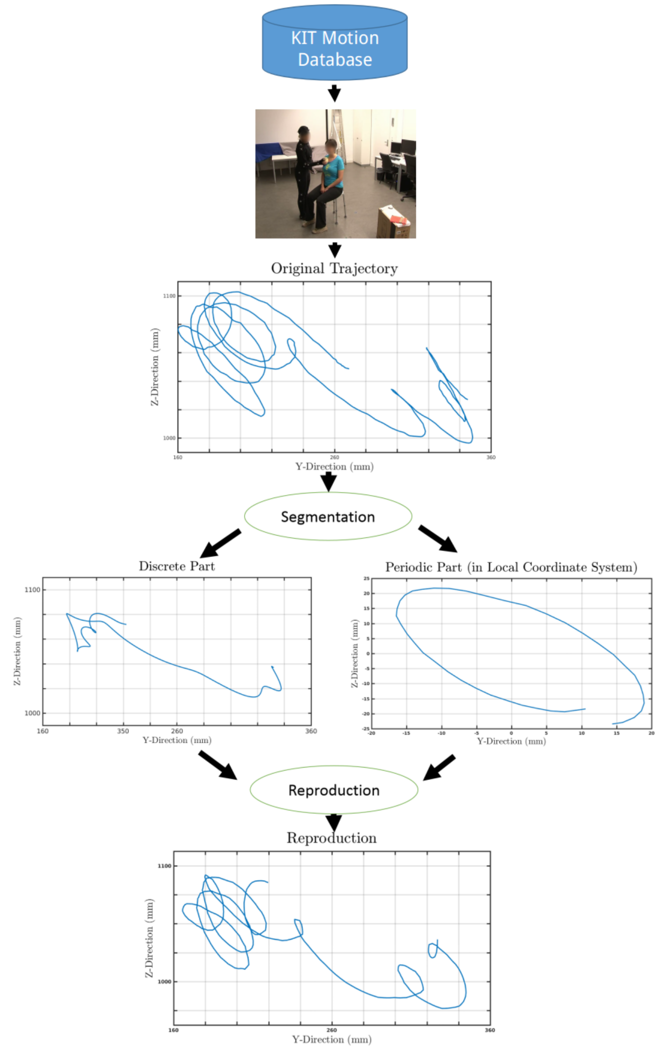

Fig. 8: Trajectory segmentation of a real wash back movement. The top most diagram is the original wiping trajectory, which is extracted from the wash-front motion saved in KIT motion database [13]. It is segmented into two different parts: periodic part and discrete part. The periodic part is learned and stored in the wiping database. The discrete part is used to record the surface's situation. The bottom image shows the result of the reproduction using CCDMP.

After getting the frequency, we cut the rest part into several small segments according to the inverse of the frequency and average them to get the approximated periodic pattern. Fig. 8 shows the result. In order to reproduce the original wiping trajectory, an amplitude profile is kept when extracting the periodic pattern. Because the real wiping movement is not strictly periodic, it changes with different amplitudes during the demonstration.

The discrete DMP for wiping a ball is manually designed. It is a trajectory along the ball's surface. The benefit of using CCDMP is that we do not need to learn another DMP when changing the ball's size. If the ball's size is enlarged, we can change the trajectory by only modifying the goal of the discrete DMP. Once CCDMP is learned, the robot is able to wash balls of different sizes (see Fig. 9).

V. Conclusion and Future Work

In this work, we introduce the Coordinate Changed Dynamic Movement Primitive which is a generalization of the original Dynamic Movement Primitive. Its flexibility, extensibility, and the encoding of the leader-follower configuration allows the representation of complex coordinated tasks of independent agents such as handover tasks and of physically coupled agents such as a wiping task of a hand-arm system. In our experiments, we showed the suitability of CCDMP to represent tasks which describe interaction and coordination between multiple agents. Furthermore, we showed that environmental elements such as a surface to be wiped can be encoded as a DMP and used to generate a well-adapted wiping movement.

However, the learning strategy and trajectory segmentation approach are designed according to the examples that have been presented in the paper. We will approach this problem by integrating sensorimotor feedback of the robot in order to explore an unknown object and task properties using vision, haptics and proprioceptive information. The result of the exploration can be processed and reduced to a trajectory, which can be used to learn a CCDMP. Furthermore, we will integrate sensorimotor feedback in an iterative CCDMP learning process in order to refine learned task representations. In hand over task, the robot should also adjust its hand's orientation according to the position and the orientation of the partner. We will extend the CCDMP, in order to consider the follower's orientation as well.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, May 2009. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2008.10.024

[2] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning*

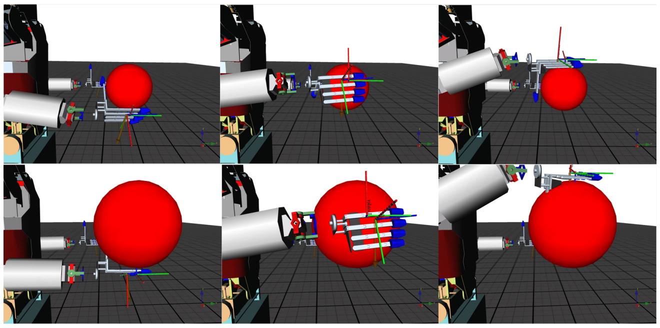

Fig. 9: The robot can wash the balls of different sizes with the same CCDMP. There are two coordinate systems in the images. One coordinate system is for the leader, and the other one is for the follower. The robot's hand executes the learned wiping pattern along the ball's surface.

by observation, demonstration and imitation, vol. 37, no. 2, pp. 286–298, 2007.

[3] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 183–202, December 2008.

[4] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *The International Journal of Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008.

[5] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems 15*, 2003, pp. 1547–1554.

[6] S. Schaal, "Dynamic movement primitives - a framework for motor control in humans and humanoid robots," in *The International Symposium on Adaptive Motion of Animals and Machines*, 2003.

[7] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, Feb. 2013. [Online]. Available: http://dx.doi.org/10.1162/NECO_a_00393

[8] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1450–1459, 2013.

[9] 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, 2014.

[10] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[11] M. Ewerton, G. Neumann, R. Lioutikov, H. B. Amor, J. Peters, and G. Maeda, "Learning multiple collaborative tasks with a mixture of interaction primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*.

[12] M. Prada, A. Remazeilles, A. Koene, and S. Endo, "Dynamic movement primitives for human-robot interaction: comparison with human behavioral observation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1168–1175.

[13] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "The kit whole-body human motion database," in *International Conference on Advanced Robotics (ICAR)*, 2015, pp. 329–336.