2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Dynamic Movement Primitives for moving goals with temporal scaling adaptation

Leonidas Koutras and Zoe Doulgeri

Abstract— **In this work, we propose an augmentation to the Dynamic Movement Primitives (DMP) framework which allows the system to generalize to moving goals without the use of any known or approximation model for estimating the goal's motion. We aim to maintain the demonstrated velocity levels during the execution to the moving goal, generating motion profiles appropriate for human robot collaboration. The proposed method employs a modified version of a DMP, learned by a demonstration to a static goal, with adaptive temporal scaling in order to achieve reaching of the moving goal with the learned kinematic pattern. Only the current position and velocity of the goal are required. The goal's reaching error and its derivative is proved to converge to zero via contraction analysis. The theoretical results are verified by simulations and experiments on a KUKA LWR4+ robot.**

I. INTRODUCTION

In recent years, Dynamic Movement Primitives (DMP) have provided a way of encoding arbitrary complex motions utilizing non - linear dynamics [1]–[5]. This encoding provides accuracy in the execution of the motion, as well as the ability to generalize to spatial and temporal scaling. DMP have been used for a wide range of tasks. Both point to point or discrete motions, as well as rhythmic, periodic trajectories are able to be encoded, resulting in the encoding of various behaviours and motion patterns [6], [7]. Modified DMP have also been proposed to extend their robustness to temporal and or spatial perturbations [8], [9]. Moreover, appropriately designed coupling terms have been used for DMP adaptation to external inputs (e.g. force adaptation) [10]–[12]. DMP are proven advantageous over typical trajectory encoding mechanisms as they are able to generalize the learned pattern both spatially and temporally as well as be used in conjunction with other dynamical fields to generate in real time complex behaviours in environments with goal perturbations and obstacles [9].

Movements encoded by DMP are designed to be executed towards a stationary goal known before the start of the motion although it can be perturbed to another stationary goal during motion. Few research efforts have been focused on DMP utilization with a moving goal. In [13], the authors are modifying the DMP, to follow a moving goal in a leader-follower framework. In [14] by augmenting a modified DMP formulation with velocity feedback the authors allow the system to follow moving goals. In both these works position and velocity trajectories are scaled towards the moving goal so that the preset time duration of the task is preserved. The latter may induce either higher or lower velocities depending on the current goal position. It is therefore possible that the generated velocity exceeds safety boundaries. The use of saturation on the robot's velocity does not resolve this problem as it generates a completely different velocity profile than that of the demonstration. However, in robot human coexistence, safety, trust and robot motion predictability is of outmost importance. It is therefore essential to maintain these human-like motion characteristics encoded during the demonstration.

A different solution to the problem of high velocities is to not require a fixed motion duration. This is possible in models that do not encode temporal characteristics of the motion [15]. Alternatively one could adapt the motion duration as in [16]; this adaptation is based on the utilization of a model for the goal dynamics which predicts the reaching location. However, such a model may not be available when for example the robot interacts with agents of unknown dynamic behaviour. Such agents could be other robots or humans whose motion patterns may not be necessarily known. In [17] a DMP formulation has been proposed in the context of hitting a moving ball with a specified velocity. This method has been extended in [18] utilizing the probabilistic DMP formulation proposed in [19]. Both formulations require again a predictor to estimate the reaching location and the moving goal's velocity at this point. Utilizing a predictor however causes the system to heavily depend on the prediction errors which can lead to execution failure.

The objective of this work is to enable a DMP, encoding a kinematic behaviour to a stationary target, to be utilized in the case of a moving target keeping velocities at levels similar to the demonstrated ones. Only the current goal's position and velocity is assumed to be provided by the perception system as opposed to the above works. To achieve this objective we propose a modified DMP for motion generation to a moving goal and prove its convergence to the goal using contraction analysis. Simulations and experiments are utilized to validate the theoretical findings. By alleviating the need of predictions and/or goal model dynamics the proposed approach does not depend on any prediction errors and addresses cases of unknown goal dynamics. Moreover, it addresses a wider range of DMP applications that involve collaboration with humans and work in shared environments in which maintaining a certain velocity level is important for the human safety and the robot's motion predictability. Such applications include collaborative tasks like handover

Authors are with the Automation & Robotics Lab, Dept. of Electrical & Computer Engineering, Aristotle University of Thessaloniki, Greece.{kleonidas@ece, doulgeri@eng}.auth.gr

The research leading to these results has received funding by the EU Horizon 2020 Research and Innovation Programme under grant agreement No 820767, project CoLLaboratE.

Authorized licensed use limited to: University of New South Wales. Downloaded on September 20,2020 at 10:52:05 UTC from IEEE Xplore. Restrictions apply.

or collaborative assembly in which the human (or another robot's) hand are considered as moving goals [13], [14].

II. PRELIMINARIES

A. Dynamic Movement Primitives

Dynamic movement primitives (DMP) are used to encode trajectories, through the augmentation of a unique point attractor linear dynamical system with a non-linear term that encapsulates arbitrary complex shape modulations known as the transformation system of the DMP. For encoding an n degrees of freedom task, the transformation system is given by [5]:

$$\begin{gathered} \tau \dot{\boldsymbol{z}} = \alpha_z(\beta_z(\boldsymbol{g}-\boldsymbol{y})-\boldsymbol{z}) + \operatorname{diag}(\boldsymbol{g}-\boldsymbol{y}_0)\boldsymbol{F}(x) \\ \tau \dot{\boldsymbol{y}} = \boldsymbol{z} \end{gathered} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the position, $\boldsymbol{z} \in \mathbb{R}^n$ is the scaled velocity, $\boldsymbol{g} \in \mathbb{R}^n$ is the attractor of the linear system, $\boldsymbol{y}_0 \in \mathbb{R}^n$ its initial position, $\operatorname{diag}(\boldsymbol{g}-\boldsymbol{y}_0) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with its elements equal to the coordinates of $\boldsymbol{g}-\boldsymbol{y}_0$, τ is a scalar temporal scaling term, which traditionally is set equal to the duration of the motion and $\boldsymbol{F}(x) \in \mathbb{R}^n$ is the forcing term which determines the motion pattern:

$$\boldsymbol{F}(x) = \frac{\sum_i \boldsymbol{w}_i \Psi_i(x) x}{\sum_i \Psi_i(x)} \tag{2}$$

with $\boldsymbol{w}_i \in \mathbb{R}^n$ being the weights of the N Gaussian kernel functions Ψ_i:

$$\Psi_i(x) = exp[-h_i(x-c_i)^2] \tag{3}$$

and c_i the kernels' centers distributed to the interval $[0,1]$ and h_i their inverse widths. Variable x is a monotonically decreasing phase variable, whose dynamics are known as the canonical system and are given by a first order system with its initial value set to 1:

$$\tau \dot{x} = -\alpha_x x \tag{4}$$

DMP are traditionally used for movements towards a constant goal $\boldsymbol{g}$.

As mentioned earlier, the DMP framework is able to generalize to new goals. When a new stationary goal is selected, the demonstrated trajectory is scaled accordingly to the new goal's position. Therefore the generated position, velocity and acceleration will be the scaled demonstrated motion to the new goal. For example a goal which is twice as far from the initial position as compared to the demonstrated goal, will produce twice as large velocity and acceleration. As discussed in Section I this property may not be suitable for collaborative environments.

In case the new goal is not stationary but moving, the system may be eventually able to track the goal but the trajectory does not follow the encoded motion pattern after a preset time duration since the forcing term goes to zero when the phase variable vanishes. This is happening as the phase variable evolution (4) does not depend on $\boldsymbol{g}$. For example when tracking a goal that moves away from the initial position, the phase variable will reach zero before reaching the goal, which implies a zero forcing term, thus removing all encoded motion patterns from the trajectory.

B. Contraction Analysis

Contraction Analysis was introduced in [20] and used in conjunction with DMP in [21] with the objective to provide a simpler method for the stability analysis of non linear dynamical systems. Instead of trying to verify point equilibrium stability, this framework tries to prove that if neighbouring state trajectories converge to each other, then all trajectories should exponentially converge to a single trajectory. This methodology can be expressed mathematically as follows:

Definition 1 [20]: Given the system equations $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x})$ a region of the state space is called a contraction region if the Jacobian $\frac{\partial \boldsymbol{f}}{\partial t}$ is uniformly negative definite in that region.

Theorem 1 [20]: Given the system equations $\dot{\boldsymbol{x}} = f(t, \boldsymbol{x})$, any trajectory, which starts in a ball of constant radius centered about a given trajectory and contained at all times in a contraction region, remains in that ball and converges exponentially to this trajectory. Furthermore, global exponential convergence to the given trajectory is guaranteed if the whole state space is a contraction region.

Proposition 1 [21]: If $\boldsymbol{f}_1(t, \boldsymbol{x}_1)$ is contracting and $\boldsymbol{f}_2(t, \boldsymbol{x}_1, \boldsymbol{x}_2)$ is contracting for each fixed $\boldsymbol{x}_1$, then the hierarchy $\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(t, \boldsymbol{x}_1)$, $\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(t, \boldsymbol{x}_1, \boldsymbol{x}_2)$ is contracting.

III. PROPOSED MODIFIED DMP FOR REACHING MOVING GOALS

Consider a moving goal $\boldsymbol{g}(t)$, $\dot{\boldsymbol{g}}(t)$. We assume that a perception system provides the goal's position and an estimate of its velocity on-line. The latter can be calculated by numerically differentiating the position measurements.

We propose the following DMP transformation system for the case of a moving goal:

$$\begin{gathered} \tau \dot{\boldsymbol{z}} = -\alpha_z(\beta_z \boldsymbol{e} + \boldsymbol{z}) + \operatorname{diag}(\boldsymbol{g}-\boldsymbol{y}_0)\boldsymbol{F}(x) \\ \tau \dot{\boldsymbol{e}} = \boldsymbol{z} \end{gathered} \tag{5}$$

where $\boldsymbol{e} = \boldsymbol{g} - \boldsymbol{y} \in \mathbb{R}^n$ is the position error and $\boldsymbol{z} \in \mathbb{R}^n$ is the scaled velocity error. Notice that (5) is equivalent to (1) for the case of a stationary goal with the forcing term expressing the negative of that in (1). The system is trained as the traditional DMP framework, using a demonstration from the initial position $\boldsymbol{y}_{0,d}$ to the stationary goal $\boldsymbol{g}_d$ with temporal scaling τ_d.

In order for the system to generate a trajectory with the encoded motion pattern while reaching a moving goal, the temporal scaling parameter τ must be continuously adjusted. If the robot keeps the motion pattern of the demonstrated motion without adjusting its time scaling [13], [14] then, a faraway target would induce high velocities that could exceed the robot's capabilities. Hence, in the case of a moving target, the temporal scaling parameter should be modified in real time. We propose to augment the DMP system with the following temporal scaling adaption law:

$$\dot{\tau} = -\kappa_\tau(\tau - \tau_g) + \dot{\tau}_g \tag{6}$$

where κ_τ is a positive design parameter and τ_g is given by:

$$\tau_g = \frac{\|\boldsymbol{g}-\boldsymbol{y}_0\|}{\|\boldsymbol{g}_d - \boldsymbol{y}_{0,d}\|}\tau_d \tag{7}$$

Authorized licensed use limited to: University of New South Wales. Downloaded on September 20,2020 at 10:52:05 UTC from IEEE Xplore. Restrictions apply.

Equation (6) ensures the tracking of τ_g which can be interpreted as the temporal scaling related to the motion towards the current goal location $\boldsymbol{g}(t)$ considered stationary. When the goal moves away from the initial position $\boldsymbol{y}_0$ the temporal scaling increases, leading to slower system evolution. This slow down counteracts the speed-up caused by the goal's position. When the goal approaches the initial position, τ decreases leading to faster execution. This again counteracts the slowing down caused by the goal's position. In this way, the produced trajectory maintains the demonstrated velocity characteristics. Notice that τ_g is bounded for $\boldsymbol{g}_d \neq \boldsymbol{y}_{0,d}$ which is generally true for discrete motions. We also assume that the goal will not coincide with the initial position $\boldsymbol{y}_0$, $\|\boldsymbol{g}(t) - \boldsymbol{y}_0\| \neq 0, \forall t$, which implies $\tau_g > 0$. The same assumption holds for (1) in order to enable motion encoding by ensuring a non-zero value for the term $\operatorname{diag}(\boldsymbol{g} - \boldsymbol{y}_0)\boldsymbol{F}(x)$. The derivative $\dot{\tau}_g$ in (6) can be computed analytically from the goal's position and velocity. Notice that the proposed system does not require a model for the goal's motion, but only measurements of the goal's position and velocity. Further notice that after the moving goal is reached the forcing term vanishes; if then a disturbance temporarily displaces the arm away from the goal trajectory the DMP system may be re-initialized if a kinematic behavior similar to the demonstrated is desired.

We claim that the system composed of (4), (5), and (6) with $\tau(0) = \tau_g(0)$ can achieve the objective of reproducing a desired kinematic behaviour while reaching a moving target.

A. Stability Analysis

The complete system can be written as:

$$\begin{aligned} \dot{\tau} &= -\kappa_\tau(\tau - \tau_g) + \dot{\tau}_g \\ \tau\dot{x} &= -\alpha_x x \\ \tau\dot{\boldsymbol{z}} &= -\alpha_z(\beta_z \boldsymbol{e} + \boldsymbol{z}) + \operatorname{diag}(\boldsymbol{g} - \boldsymbol{y}_0)\boldsymbol{F}(x) \\ \tau\dot{\boldsymbol{e}} &= \boldsymbol{z} \end{aligned} \tag{8}$$

Stability analysis concerns the asymptotic convergence of the goal tracking error $\boldsymbol{e} = \boldsymbol{g} - \boldsymbol{y}$ and its velocity, $\dot{\boldsymbol{e}} = \dot{\boldsymbol{g}} - \dot{\boldsymbol{y}}$ to zero. In addition, τ must follow the desired temporal scaling τ_g, therefore the temporal scaling tracking error $e_\tau = \tau - \tau_g$ must also converge to zero.

We define the state vector $\boldsymbol{s}$ as follows:

$$\boldsymbol{s} = \begin{bmatrix} e_\tau & x & \boldsymbol{e}^T & \boldsymbol{z}^T \end{bmatrix}^T \tag{9}$$

and write the system (8) in the state space form:

$$\dot{\boldsymbol{s}} = \begin{bmatrix} \dot{e}_\tau \\ \dot{x} \\ \dot{\boldsymbol{e}} \\ \dot{\boldsymbol{z}} \end{bmatrix} = \begin{bmatrix} -\kappa_\tau e_\tau \\ -\frac{1}{\tau}\alpha_x x \\ \frac{1}{\tau}\boldsymbol{z} \\ \frac{1}{\tau}\left(-\alpha_z(\beta_z \boldsymbol{e} + \boldsymbol{z}) + \operatorname{diag}(\boldsymbol{g} - \boldsymbol{y}_0)\boldsymbol{F}(x)\right) \end{bmatrix} \tag{10}$$

The system consisting of the first two equations of (10) is a hierarchical combination of the linear first order system:

$$\dot{e}_\tau = -\kappa_\tau e_\tau \tag{11}$$

and of $\dot{x} = -\frac{1}{\tau}\alpha_x x$. System (11) is globally exponentially stable, with a unique equilibrium at $e_\tau = 0$, as $-\kappa_\tau < 0$. Therefore the system is globally contracting. For $e_\tau = 0$, we have $\tau = \tau_g$ and $\dot{x} = -\frac{1}{\tau_g}\alpha_x x$. As $-\frac{1}{\tau_g}\alpha_x < 0$, $\forall t$ and the time depended parameter τ_g is bounded, $\dot{x} = -\frac{1}{\tau_g}\alpha_x x$ is also contracting. Therefore, using *Proposition 1* we proved that the hierarchical combination of (11) and of $\dot{x} = -\frac{1}{\tau}\alpha_x x$ is contracting. Finally, in order to prove that system (10) is contracting we will prove that the hierarchical combination of the system comprised of the last two equations of (10) is also contracting. In particular, the system:

$$\begin{bmatrix} \dot{\boldsymbol{e}} \\ \dot{\boldsymbol{z}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau}\boldsymbol{z} \\ \frac{1}{\tau}(-\alpha_z(\beta_z \boldsymbol{e} + \boldsymbol{z}) + \operatorname{diag}(\boldsymbol{g} - \boldsymbol{y}_0)\boldsymbol{F}(x)) \end{bmatrix} \tag{12}$$

for the fixed point $[e_\tau \; x]^T = [0 \; 0]^T$, which implies $\tau = \tau_g$, $\boldsymbol{F}(x) = \boldsymbol{F}(0) = \boldsymbol{0}$, can be written as:

$$\begin{bmatrix} \dot{\boldsymbol{e}} \\ \dot{\boldsymbol{z}} \end{bmatrix} = \frac{1}{\tau_g} \begin{bmatrix} \boldsymbol{0}_n & \boldsymbol{I}_n \\ -\alpha_z\beta_z \boldsymbol{I}_n & -\alpha_z \boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{z} \end{bmatrix} \tag{13}$$

which is a linear time varying system, with strictly negative eigenvalues for all values of the bounded time dependent parameter τ_g. We denote $\boldsymbol{I}_n$ the $n \times n$ identity matrix and $\boldsymbol{0}_n$ the $n \times n$ zero matrix. Notice that τ_g is bounded by the minimum and maximum distances between $\boldsymbol{y}_0$ and $\boldsymbol{g}(t)$.

Theorem II [22]: Consider a linear non autonomous system $\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x}, \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{A}(t) \in \mathbb{R}^{n \times n}$. The equilibrium state $\boldsymbol{x} = \boldsymbol{0}$ is exponentially stable if and only if for any given symmetric, positive definite, continuous and bounded matrix $\boldsymbol{Q}(t)$, there exists a symmetric, positive definite, continuously differentiable and bounded matrix $\boldsymbol{P}(t)$ such that:

$$-\boldsymbol{Q}(t) = \boldsymbol{P}(t)\boldsymbol{A}(t) + \boldsymbol{A}^T(t)\boldsymbol{P}(t) + \dot{\boldsymbol{P}}(t) \tag{14}$$

We select matrix $\boldsymbol{P}$:

$$\boldsymbol{P} = \frac{1}{2} \begin{bmatrix} (\beta_z + \frac{1}{\alpha_z} + \frac{1}{\beta_z})\boldsymbol{I}_n & \frac{1}{\alpha_z\beta_z}\boldsymbol{I}_n \\ \frac{1}{\alpha_z\beta_z}\boldsymbol{I}_n & (\frac{1}{\alpha_z} + \frac{1}{\alpha_z^2\beta_z})\boldsymbol{I}_n \end{bmatrix} \tag{15}$$

$\boldsymbol{P}$ does not depend on time, therefore it is by construct bounded, and continuously differentiable, with $\dot{\boldsymbol{P}} = \boldsymbol{0}$. $\boldsymbol{P}$ is also positive definite (See Appendix). With this selection, we obtain:

$$\boldsymbol{Q}(t) = \frac{1}{\tau_g}\boldsymbol{I}_{2n} \tag{16}$$

$\boldsymbol{Q}(t)$ is bounded by the maximum and minimum values of its eigenvalues:

$$\frac{1}{\tau_{g,max}}\boldsymbol{x}^T\boldsymbol{x} \leq \boldsymbol{x}^T\boldsymbol{Q}(t)\boldsymbol{x} \leq \frac{1}{\tau_{g,min}}\boldsymbol{x}^T\boldsymbol{x} \tag{17}$$

$\forall \boldsymbol{x} \in \mathbb{R}^{2n}$. Therefore $\boldsymbol{Q}(t)$ is bounded $\forall t$ and using *Theorem II* the system of (13) is exponentially stable.

Hence, the system is contracting. Thus using *Proposition 1* we conclude that this hierarchy is also globally contracting. Consequently, system (10) is contracting in the whole state space. As a result, all trajectories converge exponentially to the unique equilibrium point $\boldsymbol{s} = \boldsymbol{0}$, where $\boldsymbol{e} = \boldsymbol{0}$, $\dot{\boldsymbol{e}} = \boldsymbol{0}$ and $e_\tau = 0$ reaching the moving goal. The rate of this exponential convergence will be bounded between the rates corresponding to the largest, from above and smallest, from below value of τ.

Authorized licensed use limited to: University of New South Wales. Downloaded on September 20,2020 at 10:52:05 UTC from IEEE Xplore. Restrictions apply.

TABLE I: SYSTEM PARAMETERS

Parameter	Description	Value
α_z	Gain of DMP	40
β_z	Gain of DMP	10
α_x	Gain of phase system	4.6052
N	Number of Gaussian Kernels	60
κ_τ	Gain of temporal scaling controller	100

IV. Simulations

For the demonstration with simulations of the system's performance the following setup was used. We have initially created a 3-D trajectory in the Cartesian Space to a stationary goal from $\boldsymbol{y}_{0,d} = [0.2\ 0.3\ 0.85]^T$ to $\boldsymbol{g}_d = [0.6\ 0.6\ 0.65]^T$ with duration $\tau_d = 1.124s$. Then we trained the proposed DMP (5) utilizing these data. The DMP parameters used in the simulation are depicted in Table I. The value of α_x was chosen so that x reaches 0.01 at the end of the demonstrated motion. To demonstrate the DMP capabilities in the case of a moving goal, we simulated the motion of a moving agent, from $\boldsymbol{g}_1 = [0.6\ 0.8\ 0.75]^T$ to $\boldsymbol{g}_2 = [2\ 1.5\ 0.18]^T$. We then executed both the goal motion and the proposed DMP for moving goals (4) - 6), from the initial position $\boldsymbol{y}_0 = \boldsymbol{y}_{0,d}$, feeding back the DMP with the current goal position and velocity. Notice that the system reaches the goal at the point $\boldsymbol{g}_c = [1.92\ 1.44\ 0.33]$, at time $t = 3.37s$ before the goal completes its execution and reaches $\boldsymbol{g}_2$. After reaching the goal the system tracks the goal's motion to $\boldsymbol{g}_2$. The simulation results are presented in Figures 1 - 5. In particular, Figures 1, 2 and 3 show the path and trajectories of the $\boldsymbol{y}$ coordinates and its derivatives as well as those of the moving goal. We see the goal moving from a position $\boldsymbol{g}_1$ near the stationary goal $\boldsymbol{g}_d$ of the demonstration (red line path in Figure 1) following its own trajectory (black dashed line in Figures 2 and 3) towards $\boldsymbol{g}_2$ which is further away from $\boldsymbol{y}_0$ as compared to $\boldsymbol{g}_1$. The relative position and velocity errors converge to zero validating the theoretical analysis as shown in Figure 4. As shown in Figure 5, τ increases as the goal moves away from the initial position $\boldsymbol{y}_0$. The adaptation of τ prevents the appearance of high velocities as seen in Figure 3 keeping the velocity at levels similar to the demonstration. Notice in Figure 3, the discontinuity arising initially since $\dot{\boldsymbol{g}} \neq \boldsymbol{0}$ at $t = 0$. In practice, smoothness techniques should be employed at the beginning of motion. To demonstrate the advantages of the temporal adaptation law (6), Figure 6 shows the proposed DMP velocities together with those produced in a DMP execution without temporal adaptation (4), (5) with $\tau = \tau_d$. Notice the higher velocities induced by the lack of temporal adaptation. This may not be acceptable as it may exceed the robot's capabilities, be dangerous for humans and ruin the human-likeness of the demonstrated motion.

Overall, simulation results show that the proposed DMP framework successfully reaches the moving goal, scaling the demonstrated motion's position pattern and subsequently tracks the goal motion, while maintaining velocity levels close to the demonstrated ones.

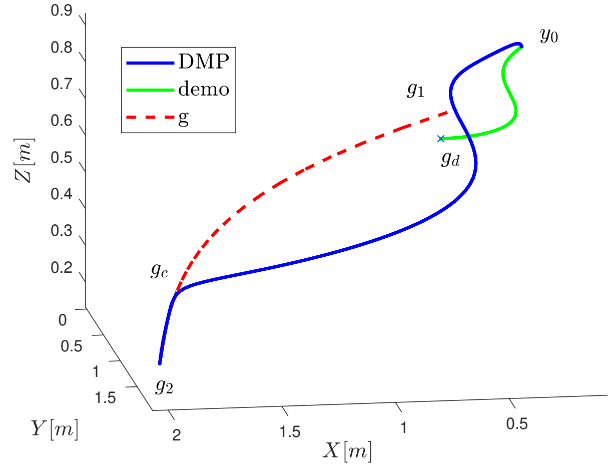

Fig. 1: Path of the DMP, the demonstration and the moving goal.

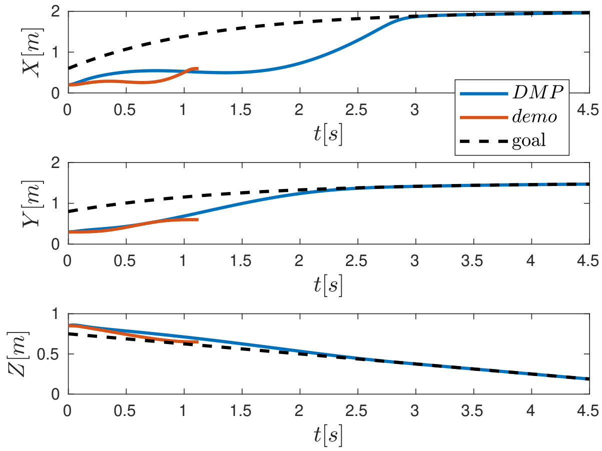

Fig. 2: Evolution of the position when reaching the moving goal. Also depicting the goal trajectory as well as the demonstration trajectory.

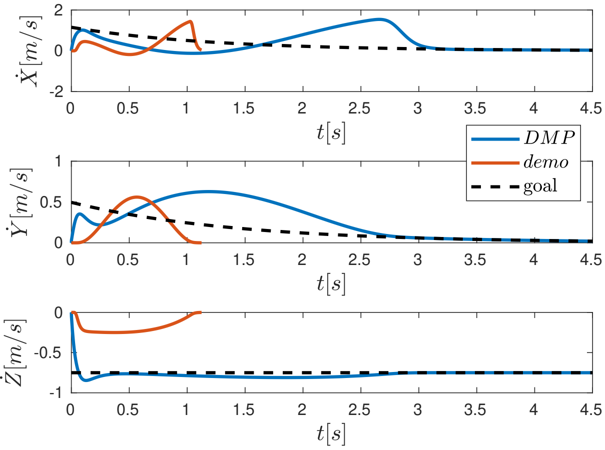

Fig. 3: Evolution of the velocity when reaching the moving goal. Also depicting the goal trajectory as well as the demonstration trajectory.

Authorized licensed use limited to: University of New South Wales. Downloaded on September 20,2020 at 10:52:05 UTC from IEEE Xplore. Restrictions apply.

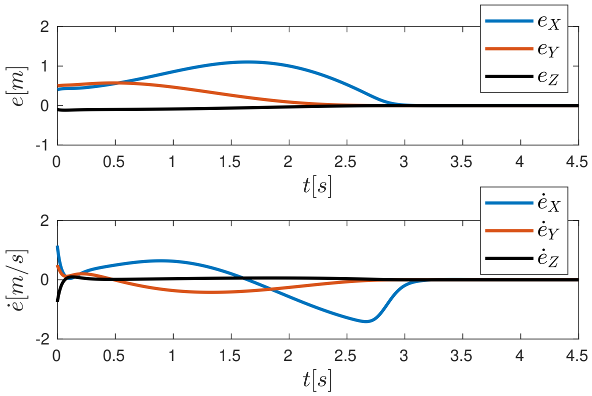

Fig. 4: Position and velocity errors in each coordinate.

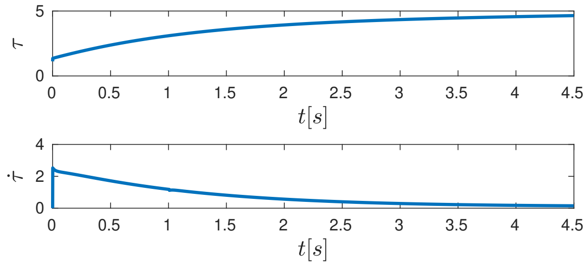

Fig. 5: Temporal scaling τ and its derivative $\dot{\tau}$.

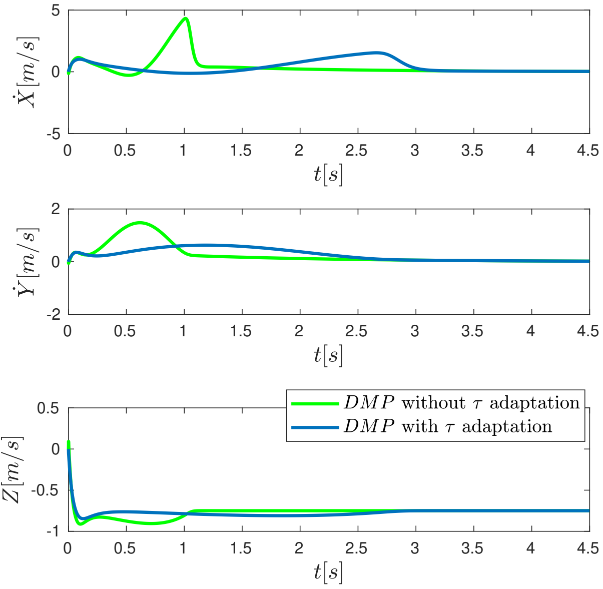

Fig. 6: Evolution of the velocity when reaching the moving goal with and without temporal scaling adaptation.

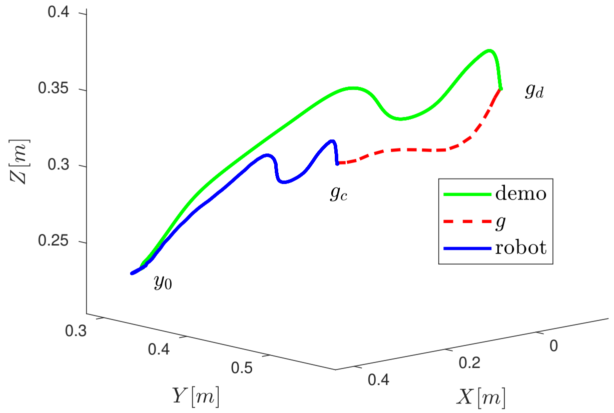

Fig. 7: Experimental Result: Path of the robot, the demonstration and the moving goal.

V. EXPERIMENTS

To demonstrate the proposed method on real world conditions, an implementation was made on a 7 degree of freedom KUKA LWR4+ robotic manipulator. The methodology was implemented in C++, on a real-time Linux PC. For the communication with robot, the FRI library [23] was used, with control frequency $500Hz$. Robot kinesthetic guidance was used to record two trajectories. The recorded tool position data from the first demonstration was used to train the proposed DMP (4), (5) while data from the second demonstration was used to emulate a moving target. During both demonstrations the robot was under gravity compensation and high stiffness orientation control for maintaining its initial tool orientation.

The demonstrated trajectory started from initial position $\boldsymbol{y}_{0,d} = [0.39\ \ 0.32\ \ 0.24]^T$ towards the goal position $\boldsymbol{g}_d = [0.03\ \ 0.56\ \ 0.36]^T$ in $6.54s$. The moving goal's demonstrated trajectory started from the stationary goal of the demonstration $\boldsymbol{g}_d$, moving closer to the initial position and reaching $\boldsymbol{g}_c = [0.19\ 0.44\ 0.31]^T$. During execution the DMP with the parameters of Table I generated the tool reference trajectory which was mapped to the joint space by the Jacobian pseudoinverse. Closed loop inverse kinematics [24] was utilized to ensure accurate following of the reference trajectory. The robot's initial position was set equal to the demonstrated one. When the moving goal was reached in $4.7s$, the execution was terminated.

Results of the path, the position and velocity trajectory and the goal's trajectory are shown in Figures 7 - 9. The robot's position and velocity coincide with the reference generated by the DMP. Position and velocity error convergence is shown in Figure 10. The temporal scaling parameter τ is shown to decrease in Figure 11 as the goal is approaching the initial position, from it initial value of $\tau_d = 6.54$ to $\tau = 3.63$ at the end of motion. As the goal movement was simulated by a demonstrated motion, measurements of the velocity were available. However, we chose to numerically differentiate

Authorized licensed use limited to: University of New South Wales. Downloaded on September 20,2020 at 10:52:05 UTC from IEEE Xplore. Restrictions apply.

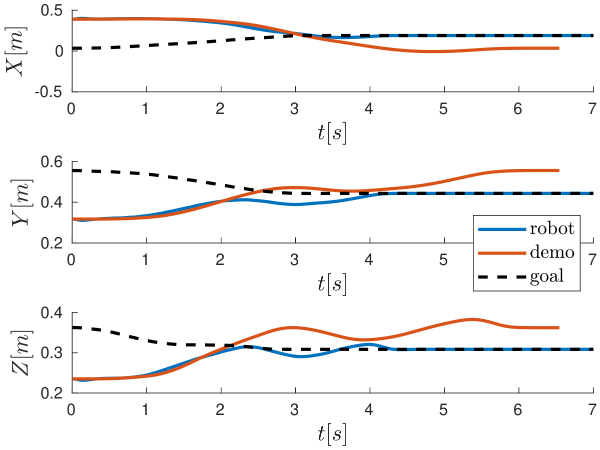

Fig. 8: Experimental Result: Evolution of the position of the robot when reaching the moving goal. Also depicting the goal trajectory as well as the demonstration trajectory.

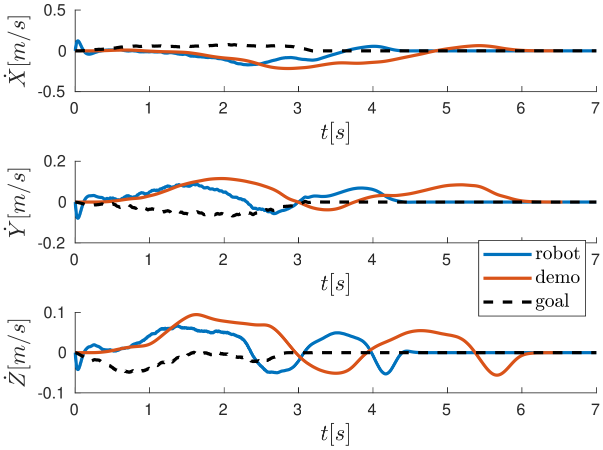

Fig. 9: Experimental Result: Evolution of the velocity the robot when reaching the moving goal. Also depicting the goal trajectory as well as the demonstration trajectory.

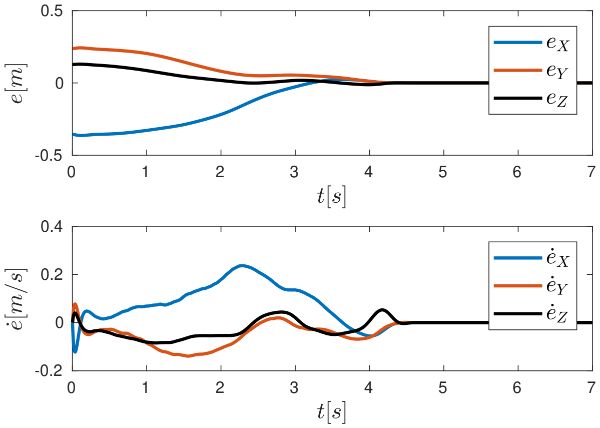

Fig. 10: Experimental Result: Position and velocity errors in each coordinate at the robot experiment.

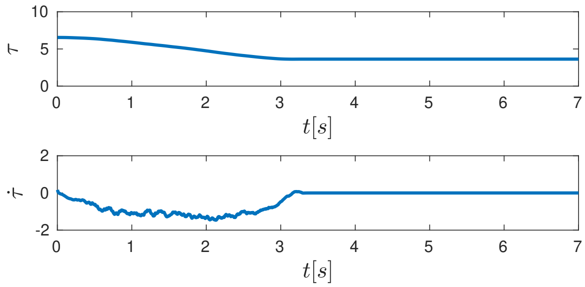

Fig. 11: Experimental Result: Temporal scaling τ and its derivative $\dot{\tau}$ at the robot experiment.

the values of $\boldsymbol{g}$, as it would happen if the goal's state was provided by a perception system. This differentiation explains the noise appearing in the $\dot{\boldsymbol{g}}$ and $\dot{\tau}$ in Figures 9 and 11. The system however is not affected as it is evident from Figures 7 - 11.

VI. CONCLUSIONS

In this paper a modified DMP framework is proposed for the case of reaching moving goals maintaining the demonstrated velocity characteristics. The framework only requires the goal's current position and velocity that can be in general provided by a perception system. Thus, it avoids prediction and model uncertainty errors that arise in predictor or model based approaches. Temporal adaptation provides predictable velocity behaviours that remain closer to the demonstrated as the moving goal scales the trajectory spatially. This generalization allows DMP to be used for more complex tasks and to operate effectively in workplaces shared with humans. The reaching of the moving goal is theoretically proved and validated by both simulations and experiments.

APPENDIX

Theorem III [25]: Consider a matrix that can be written as:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix} \tag{18}$$

with $\boldsymbol{X} \in \mathbb{R}^{m\times m}$, $\boldsymbol{A} \in \mathbb{R}^{p\times p}$, $\boldsymbol{B} \in \mathbb{R}^{p\times q}$ and $\boldsymbol{C} \in \mathbb{R}^{q\times q}$, where $m = p + q$.

The Schur complement of $\boldsymbol{A}$ in $\boldsymbol{X}$ is denoted $\boldsymbol{X}/\boldsymbol{A}$ and defined as:

$$\boldsymbol{X}/\boldsymbol{A} = \boldsymbol{C} - \boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{B} \tag{19}$$

If $\boldsymbol{A}$ and $\boldsymbol{X}/\boldsymbol{A}$ are both positive definite, then $\boldsymbol{X}$ is also positive definite.

Notice that matrix $\boldsymbol{P}$ of (15) has the form described in (18) with $\boldsymbol{A} = \frac{1}{2}(\beta_z + \frac{1}{\alpha_z} + \frac{1}{\beta_z})\boldsymbol{I}_n$, $\boldsymbol{B} = \frac{1}{2}\frac{1}{\alpha_z\beta_z}\boldsymbol{I}_n$, $\boldsymbol{C} = \frac{1}{2}(\frac{1}{\alpha_z} + \frac{1}{\alpha_z^2\beta_z})\boldsymbol{I}_n$, $m = 2n$ and $p = q = n$. The Schur complement of $\boldsymbol{A}$ in $\boldsymbol{P}$ is:

$$\boldsymbol{P}/\boldsymbol{A} = \frac{1}{2}\frac{(\alpha_z\beta_z + 1)^2 + \alpha_z^2}{\alpha_z^3\beta_z^2 + \alpha_z^2\beta_z + \alpha_z^3}\boldsymbol{I}_n \tag{20}$$

Both $\boldsymbol{A}$ and $\boldsymbol{P}/\boldsymbol{A}$ are positive definite, thus $\boldsymbol{P}$ is also positive definite.

Authorized licensed use limited to: University of New South Wales. Downloaded on September 20,2020 at 10:52:05 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] S. Schaal, S. Kotosaka, and D. Sternad, “Nonlinear dynamical systems as movement primitives,” in *Humanoids2000, First IEEE-RAS International Conference on Humanoid Robots*. Cambridge, MA: CD-Proceedings, Sept. 2000, clmc.

[2] A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Movement imitation with nonlinear dynamical systems in humanoid robots,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, May 2002, pp. 1398–1403 vol.2.

[3] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, “Learning movement primitives,” in *Robotics Research. The Eleventh International Symposium*, P. Dario and R. Chatila, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 561–572.

[4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Learning attractor landscapes for learning motor primitives,” in *Proceedings of the 15th International Conference on Neural Information Processing Systems*, ser. NIPS’02. MIT Press, 2002, pp. 1547–1554.

[5] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Computation*, vol. 25, no. 2, pp. 328–373, Feb 2013.

[6] S. Schaal, *Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics*. Tokyo: Springer Tokyo, 2006, pp. 261–280.

[7] J. Rosado, F. Silva, and V. Santos, “Adaptation of robot locomotion patterns with dynamic movement primitives,” in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, April 2015, pp. 23–28.

[8] M. Karlsson, A. Robertsson, and R. Johansson, “Convergence of dynamical movement primitives with temporal coupling,” in *2018 European Control Conference (ECC)*. IEEE–Institute of Electrical and Electronics Engineers Inc., 6 2018.

[9] 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 *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 2587–2592.

[10] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, “Online movement adaptation based on previous sensor experiences,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 365–371.

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

[12] B. Nemec, A. Gams, M. Deniša, and A. Ude, “Human-robot cooperation through force adaptation using dynamic motion primitives and iterative learning,” in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Dec 2014, pp. 1439–1444.

[13] Y. Zhou, M. Do, and T. Asfour, “Coordinate change dynamic movement primitives — a leader-follower approach,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5481–5488.

[14] M. Prada, A. Remazeilles, A. Koene, and S. Endo, “Dynamic movement primitives for human-robot interaction: Comparison with human behavioral observation,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1168–1175.

[15] E. Gribovskaya and A. Billard, “Learning nonlinear multi-variate motion dynamics for real-time position and orientation control of robotic manipulators,” in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, Dec 2009, pp. 472–477.

[16] S. Kim, A. Shukla, and A. Billard, “Catching objects in flight,” *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1049–1065, Oct 2014.

[17] J. Kober, K. Mülling, O. Krömer, C. H. Lampert, B. Schölkopf, and J. Peters, “Movement templates for learning of hitting and batting,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 853–858.

[18] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters, “Using probabilistic movement primitives for striking movements,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 502–508.

[19] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, “Probabilistic movement primitives,” in *Advances in NeuralInformation Processing Systems*, 2013, p. 2616–2624.

[20] W. Lohmiller and J.-J. E. Slotine, “On contraction analysis for nonlinear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, June 1998.

[21] P. M. Wensing and J.-J. Slotine, “Sparse control for dynamic movement primitives,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 114 – 10 121, 2017, 20th IFAC World Congress.

[22] H. Marquez, *Nonlinear Control Systems: Analysis and Design*, 11 2002, ch. 4.

[23] G. Schreiber, A. Stemmer, and R. Bischoff, “The fast research interface for the kuka lightweight robot,” in *IEEE ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, 2010.

[24] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer-Verlag, 2007, ch. 11.

[25] F. Z. Zhang, *The Schur Complement and Its Applications*, 01 2005.

Authorized licensed use limited to: University of New South Wales. Downloaded on September 20,2020 at 10:52:05 UTC from IEEE Xplore. Restrictions apply.