2021 20th International Conference on Advanced Robotics (ICAR)
December 6-10, 2021. Ljubljana, Slovenia

Real-time temporal adaptation of dynamic movement primitives for moving targets

Akhil S Anand[1], Andreas Østvik[2], Esten Ingar Grøtli[3], Marialena Vagia[3], Jan Tommy Gravdahl[1]

Abstract— **This work is aimed at extending the standard dynamic movement primitives (DMP) framework to adapt to real-time changes in the task execution time while preserving its style characteristics. We propose an alternative polynomial canonical system and an adaptive law allowing a higher degree of control over the execution time. The extended framework has a potential application in robotic manipulation tasks that involve moving objects demanding real-time control over the task execution time. The existing methods require a computationally expensive forward simulation of DMP at every time step which makes it undesirable for integration in real-time control systems. To address this deficiency, the behaviour of the canonical system has been adapted according to the changes in the desired execution time of the task performed. An alternative polynomial canonical system is proposed to provide increased real-time control on the temporal scaling of DMP system compared to the standard exponential canonical system. The developed method was evaluated on scenarios of tracking a moving target where the desired tracking time is varied in real-time. The results presented show that the extended version of DMP provide better control over the temporal scaling during the execution of the task. We have evaluated our approach on a UR5 robotic manipulator for tracking a moving object.**

I. INTRODUCTION

The problem of targeting and manipulating a moving object by a robotic arm is of great importance in numerous industrial applications. Up until now, researchers mainly focused on the problem of static manipulation. However, in order to achieve higher levels of dexterity in robotic manipulators, moving target scenarios need to be addressed. For example, grasping moving objects is challenging as opposed to picking up a stationary object, such as the adaption of the motion plan, efficient trajectory tracking, modelling and estimating the motion of the object and so on. Learning from demonstration (LfD) has been widely deployed in many robotic manipulation tasks, but similarly it mainly involves static targets. Among the various LfD methods that exist, the Dynamic movement primitives (DMP) framework [1]–[5] is predominantly used for motion planning in manipulation tasks in order to learn from human demonstrations [6], [7].

The DMP framework offers a simple way to learn a complex motion trajectory from a single human demonstration without the need of any complex modelling. DMP framework encodes an arbitrary motion pattern using a second-order nonlinear system consisting of a linear point attractor modulated by a learned nonlinear forcing function. Additionally, the DMP system is capable of generalizing to different goal positions and task execution speeds using spatial and temporal scaling properties respectively. The DMP framework can be used for both point-to-point as well as rhythmic movements. The robustness to perturbations and collision avoidance capabilities can be incorporated into DMP [8]–[10], making it a highly useful framework for learning robotic manipulations skills. The DMP framework is further improved with the possibility of learning from multiple demonstrations [11]–[13]. To facilitate a singularity-free representation of orientation in Cartesian coordinates, DMP is represented using unit quaternions [14], [15].

There has been limited work conducted regarding DMP systems naturally adapting to a moving target. A bio-inspired formulation was developed for human-robot interaction by including a velocity feedback term into the DMP system in [16]. In [17], an interactive movement primitive is formulated to reach a moving target in a leader-follower configuration. Other approaches involve reaching a moving object by predicting the target trajectory in advance, as presented in [18], [19], based on a dynamic model of the moving target. However, in many robotic tasks, including human-robot interaction, it is difficult to model the movement of the target object and thereby generate an accurate predefined DMP. Such tasks, demand real-time adaptation of the DMP to continuously change the target/goal position and desired execution time. Consider the example of a human to robot object handover task, where the object's velocity depends on the human individual's movement. In such a scenario, the DMP system needs to slow down or speed up based on the human behaviour, while adapting to a moving target. Another example is a robotic grasping task where the target object is moving continuously, but the motion pattern of the object is unknown. In [20], an approach to achieve real-time control over the execution time by forward simulating the entire DMP execution at each time step was proposed. This approach is computationally expensive and thus less desirable for higher control frequencies.

In this paper, we propose an extension of the standard DMP framework for adaption to real-time changes in the task execution time. We define *real-time control of the execution time* as, how the DMP system is adapting to real-time changes in its desired execution time during the task. In order to achieve this, we only manipulate the temporal scaling of DMP system while preserving its spatial properties. We formulate two methods to achieve efficient real-

[1]Akhil S Anand and Jan Tommy Gravdahl are with Dept. of Engineering Cybernetics at Norwegian University of Science and Technology (NTNU), Trondheim, Norway. Contact: akhil.s.anand@ntnu.no

[2]Andreas Østvik is with Dept. of Health Research, SINTEF Digital and Dept. of Circulation and Medical Imaging, NTNU, Trondheim, Norway.

[3] Esten Ingar Grøtli and Marialena Vagia are with Dept. of Mathematics and Cybernetics, SINTEF Digital, Trondheim, Norway.

time temporal scaling, (i) a control law to vary the temporal scaling term of the standard exponential canonical system and (ii) an alternate polynomial based canonical system with a suitable control law for temporal scaling. This is useful in a manipulation task with an underlying DMP planner, where the task execution time needs to be changed during the execution phase. Additionally, in case of moving targets, a velocity feedback of the target and a simple estimate of the goal position based on the current target position and velocity are included into the DMP system. The proposed approach is tested on simulation and experimental setups with a moving object, with the use of a UR5 robotic manipulator.

II. Dynamic Movement Primitives

DMP framework provides an elegant way to encode any arbitrary spatial trajectory as a stable second-order nonlinear system, which is well suited and widely utilized controlling for robotic systems [21]. The standard DMP system consists of a point attractor formulated as a second-order ordinary differential equation (ODE) with a nonlinear forcing term. In the DMP framework this is called the transformation system, each degree of freedom (DOF) in the operation space will be denoted by a separate transformation system,

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

Here, $x, v \in \mathbb{R}$ are the position and velocity of the system at time t respectively. The initial position and the goal/target are given by $x_0, x_g \in \mathbb{R}$ respectively. $K, D \in \mathbb{R}^+$ represents the stiffness and damping coefficient terms of the second-order system. f denotes the nonlinear forcing term, which is a function of a phase variable s. If $f = 0$, these equations represent a globally stable second-order linear system with $(x, v) = (x_g, 0)$ being a point attractor. The forcing term is modelled to match the system output with any arbitrary trajectory demonstrated, which is then generalised to different goal and initial conditions. $\tau \in \mathbb{R}^+$ is a temporal scaling term, which is normally set equal to the task execution time or the motion duration while modelling/learning the forcing term. The temporal evolution of the DMP system can be modulated by varying τ. For generalization in time, the explicit time dependency of f is avoided by reparameterising time by a phase variable s guided by a first-order linear dynamical system, termed as the canonical system, thus making the system temporally scalable by varying τ,

$$\tau\dot{s} = -\alpha s\,. \tag{2}$$

The initial state of the canonical system is $s_0 = 1$ and $\alpha \in \mathbb{R}^+$ is the decay constant, where s decays exponentially from 1 to 0. The forcing term can be expressed as a function of the phase variable s using Gaussian kernel functions ψ_i with corresponding weights ω_i,

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

where,

$$\psi_i(s) = \exp\left(-h_i\,(s - c_i)^2\right)\,. \tag{4}$$

N denotes the number of Gaussian kernels used, which is a hyper-parameter decided based on the geometric complexity of the demonstration trajectory. The centres and widths of Gaussian kernels are given by c_i and h_i respectively for a demonstration trajectory of time duration, T,

$$c_i = \exp\left(-\alpha i \frac{T}{N}\right), \quad i = 1, 2, \ldots, N\,, \tag{5}$$

$$\begin{aligned} &h_i = \tfrac{1}{(c_{i+1} - c_i)^2}, \quad i = 1, 2, \ldots, N-1\,, \\ &h_N = h_{N-1}\,. \end{aligned} \tag{6}$$

The DMP system is easily translated to multidimensional problem as a separate transformation system is learned along each dimension, while having a single canonical system. The conventional transformation system representation in (1) has few drawbacks, such as the case when the goal position coincides or is too close to the initial position, i.e ($x_g = x_0$) or when ($x_g - x_0$) flips sign from the demonstrated trajectory. These problems are addressed in [8], [16], [18], [22] to formulate an improved version,

$$\begin{aligned} \tau\dot{x} &= v\,, \\ \tau\dot{v} &= K(x_g - x) - Dv - K\,(x_g - x_0)\, s + Kf(s)\,. \end{aligned} \tag{7}$$

The canonical system remains the same as (2). Unlike (1), generalisation to different goal positions is possible in (7) as f is independent of the spatial scaling. This generalisation is obtained by utilizing the invariance property of the transformation system in (7) and using the roto-dilation based transformation to find the transformation matrix [13]. Given a transformation matrix $\boldsymbol{S} \in \mathbb{R}^{n\times n}$, where n is number of DMPs or the dimension of trajectory, $X \in \mathbb{R}^n, V \in \mathbb{R}^n, X_0 \in \mathbb{R}^n, X_g \in \mathbb{R}^n, F \in \mathbb{R}^n$ respectively are the vector representation of x, v, x_0, x_g, f, and $\boldsymbol{K} \in \mathbb{R}^{n\times n}, \boldsymbol{D} \in \mathbb{R}^{n\times n}$ are the matrix representation of K, D. The transformed states and variables are:

$$\begin{aligned} &X' = \boldsymbol{S}X, \quad V' = \boldsymbol{S}V, \quad X_0' = \boldsymbol{S}X_0, \quad X_g' = \boldsymbol{S}X_g, \\ &F' = \boldsymbol{S}F, \quad \boldsymbol{K}' = \boldsymbol{S}K\boldsymbol{S}^{-1}, \quad \boldsymbol{D}' = \boldsymbol{S}D\boldsymbol{S}^{-1} \end{aligned} \tag{8}$$

The DMP formulation has some drawbacks for real-time control during execution. This becomes very pronounced if the target is non-stationary. In the standard DMP formulation, we can vary the execution time in real-time by modulating τ according to an adaptive law based on a complete forward simulation of the entire DMP, given by,

$$\begin{aligned} \tau' &= \lambda\tau\,, \\ \lambda^{t_{i+1}} &= \lambda^{t_i} + k_p\left(\hat{T}^{t_i} - T\right) - k_d\left(\hat{T}^{t_i} - \hat{T}^{t_i\ -1}\right)\,. \end{aligned} \tag{9}$$

where t_i is the time at the i^{th} control step, $\lambda^{t_{i+1}}$, is the temporal scaling factor at time t_i, $t_0 = 0$; $\lambda^{t_0} = 1$; k_p and k_d are the specified proportional and derivative gains. $\hat{T}^{t_i}$ is the total execution time (from start) as computed at time t_i. $\hat{T}^{t_i}$ is estimated by doing an entire forward simulation of the DMP at time t_i which is computationally inefficient for a real-time task. Also, sudden changes in τ could create unexpected behaviours in the DMP system and make it less robust in a moving target scenario.

III. Extension of the DMP Model

A. Real-time Control of the DMP Execution Time

We propose an extension of the standard DMP framework to provide real-time control over the execution time T, while preserving the shape properties of the DMP system for tasks involving moving targets. Although we consider the moving target scenarios here, our approach is equally applicable to a stationary target scenario. In the standard DMP formulation, the execution time can be varied in real-time by varying τ according to a suitable adaptive law. However, such an adaptive law can not guarantee an exact final execution time. This error in the final execution could be higher when T is varied during the later stages of DMP execution. An adaptive law is formulated for τ based the value of the phase variable, s from the canonical system (2) and the fraction of DMP executed at s in each DOF in comparison to the demonstration trajectory. The dynamics of the temporal scaling term τ of the canonical system is represented using a first-order linear system modulated by the input u,

$$\dot{\tau} = -k_\tau(\tau - u)\,. \tag{10}$$

Here k_τ is a positive constant. An adaptive law is defined for control input u, such that the decay rate of the canonical system in (2) is slowed down if the desired execution time is increased or sped up if the desired execution time is reduced. The value of u is derived at each time step in order to have continuous control over the execution time. As the nonlinear forcing term f is a function of the phase variable s, the shape of the trajectory can be preserved while the execution time of the task is changed in real-time. The rate of change of s is always negative as it is a monotonically decreasing function. The control law for u is given by

$$u = \begin{cases} \frac{\hat{s}_t}{s_t}\frac{\|\overline{x}\|}{\|\overline{x}_d\|}\hat{\tau}_t & \text{if } s_t > \delta \\ \hat{\tau}_t & \text{otherwise}\ . \end{cases} \tag{11}$$

δ is chosen to be a very small value close to 0, to guarantee finite value for $u \forall\ t$. $\|\overline{x}_d\|$ denotes the norm of fraction of DMP executed in the demonstration trajectory for a specific value of phase variable at time t denoted by s_t. Similarly $\|\overline{x}\|$ denotes the norm of fraction of the current DMP corresponding to s_t. $\hat{s}_t$ and $\hat{\tau}_t$ are the desired value of s and τ at time t, which is described later. The mathematical expressions are given by,

$$\overline{x}_d(s) = \frac{g_d - x_d(s)}{g_d - x_d^0}\,, \quad \overline{x}(s) = \frac{\hat{x}_g - x(s)}{\hat{x}_g - x^0}\,. \tag{12}$$

$\overline{x}_d,\ \overline{x}\ \in\ [0,1]$, where $\overline{x}_d$ is derived from the transformed demonstration trajectory used for learning the DMP, whereas $\overline{x}(s)$ is calculated in real-time based on the value of s_t. In (12), it is assumed that the initial pose of the robot is different from the goal trajectory and needs to reach the goal point only once during a trajectory execution to avoid the division-by-zero condition. This is valid for general robotic manipulation tasks where the initial pose of the robot is different from the object's trajectory in the workspace and the robot needs to approach the object. In scenarios where the starting and goal positions coincide, rhythmic DMPs should be used instead of discrete DMP, which is not considered in this paper.

Furthermore, $\hat{s}_t$ is the desired value of the phase variable from the canonical system given the value τ, scaled linearly in the desired execution time $\hat{T}_t$ at time t,

$$\hat{\tau}_t\dot{\hat{s}}_t = -\alpha\hat{s}_t\,, \tag{13}$$

where,

$$\hat{\tau}_t = \tau_0\frac{\hat{T}_t}{T_0}\,. \tag{14}$$

Here τ_0 and T_0 are the initial value of τ and initial value of the task execution time respectively.

A similar approach of defining a first order dynamics on τ is provided in recent work [23]. Here, τ is adapted solely to maintain the demonstrated velocity levels in case of DMP execution for a task involving a moving goal. However, our work consider the aspect of real-time control of task execution time, which is more applicable in practical scenarios with strict timing requirements.

B. Polynomial Canonical System

The standard canonical system (2) in the DMP formulation can not guarantee complete control over the execution time with an adaptive law (10) because of its exponentially decaying nature. As the canonical system decays exponentially in time, the value of s approaches zero rapidly, with only very small changes in s in the later phases. This behaviour of the canonical system makes it impossible to have control over the time scaling by varying τ. For larger values of s during the early phase of DMP execution, a better level of real-time control over the task execution time is feasible. In order to improve this drawback of the DMP framework, we propose to use an alternative canonical system which could provide a higher degree of control over the execution time in later phases of the DMP. A polynomial function which decays slowly at the beginning and then rapidly converges to 0 at $t = T$, or any similar functions could offer such property. Considering this benefit, we choose a polynomial function which can be represented using an inverse gradient formulation of the standard canonical system in (2),

$$\tau\dot{s} = \begin{cases} -\alpha^{-1}s^{-1} & \text{if } s > \delta \\ -\alpha' s & \text{otherwise}\ . \end{cases} \tag{15}$$

To have s monotonically decreasing in the interval $[1, 0]$ and asymptotically converging to 0, the final stage of DMP where $s \leq \delta$ is represented using the standard canonical system (2). δ is a small positive constant close to 0. The value of τ for $s > \delta$ at any time instant t can be derived from the solution of the system in (15) based on the condition, $t = \hat{T} \Longrightarrow s = 0$ assuming the polynomial function for s,

$$\tau = \frac{2\alpha(\hat{T} - t)}{s_t^2}\,. \tag{16}$$

Here the value of τ needs to be updated only when there is a change in $\hat{T}$. From (15) and (16), $\tau > 0\ \forall\ t$ as $t < \hat{T}$

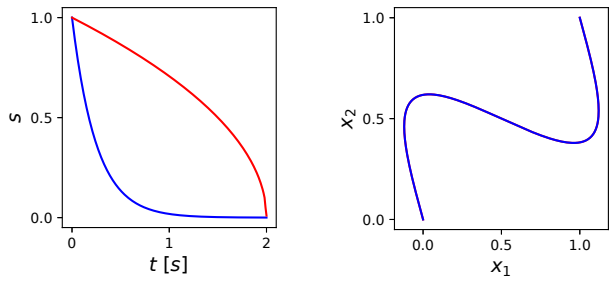
when $s > \delta$. Additionally the polynomial canonical system in (15) is continuous, which can be made smooth by choosing $\alpha' = 1/(\delta^2\alpha)$. An analytical solution for the polynomial canonical system when $s > \delta$ is derived from (15), (16) as,

$$s = \left(\frac{T-t}{T}\right)^{\frac{1}{2\alpha^2}}. \tag{17}$$

The value of α decides the nature of this polynomial canonical system, where $\alpha = 1$ defines a parabola. α is chosen as $\alpha \geq 1$, to make better use of the higher order polynomial behaviour motivating this approach. For $s \leq \delta$ the value of τ could be found using (10). The difference between systems (2) and (15) is shown in figure 1.a. The larger variation in s during the later phases of execution provides the option to have better control over the execution time. The intersection time t_{int} of the polynomial canonical system in (17) and the exponential canonical system from (2) (assuming $\tau/\alpha' = T/4$ for (2) for the sake of comparison) can be found to be,

$$t_{int} = T + \frac{TW\left(-8e^{-8\alpha^2}\alpha^2\right)}{8\alpha^2}, \tag{18}$$

where t_{int} is the time at which both the systems in figure 1.a intersects and W is the Lambert W-function [24]. As $\alpha \geq 1 \implies t_{int} \longrightarrow \hat{T}$ it follows that the polynomial canonical system maintains a higher value of s until s is close to 0.



(a) (b)

Fig. 1. (a) Evolution of the canonical systems, exponential canonical system is shown in blue and polynomial canonical system in red. (b) Behaviour of learned DMP systems corresponding to both canonical systems, both overlap as the same demonstration trajectory is used to learn both systems.

C. Stability Guarantee

For the adaptive law discussed in section III-A, from (10) and (11) it can be observed that $\tau > 0 \ \forall \ t$, which guarantees the asymptotic convergence of the exponential canonical system (2) to 0. From (17), the polynomial canonical system monotonically decreases to 0 as $t \to \hat{T}$ with $t = \hat{T} \implies s = 0$. Therefore, $\exists\, \Delta > 0$, such that, $(\hat{T} - t) \to \Delta \implies s \to \delta > 0$. The existence of δ guarantees the switching of polynomial canonical system to the exponential canonical system (15), which in turn guarantees asymptotic convergence of s to 0. The existence of $\delta > 0$ also guarantees a finite value of τ at switching (16) and τ is bounded $\forall\ t$.

In order to analyse the stability of the transformation system, the contraction analysis method is used [25], [26]. The stability analysis presented here is very similar to the one presented in [23]. Consider the transformation system (7) with $e = x - x_g$,

$$\begin{bmatrix} \dot{e} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau}v \\ \frac{1}{\tau}(-Ke - Dv - K(x_g - x_0)s + Kf(s)) \end{bmatrix}. \tag{19}$$

As s asymptotically converges to 0, for the fixed point of $s = 0$ with a finite value of $\tau = \tau_s$ and $f(s) = 0$, (19) can be written as,

$$\begin{bmatrix} \dot{e} \\ \dot{v} \end{bmatrix} = \frac{1}{\tau_s}\begin{bmatrix} 0 & 1 \\ -K & -D \end{bmatrix}\begin{bmatrix} e \\ v \end{bmatrix}. \tag{20}$$

This is a linear time varying system with a bounded time dependent parameter, τ_s.

Theorem 4.6 from [27]: Consider a linear time varying system of the form $\dot{x} = \boldsymbol{A}(t)x$ where $x \in \mathbb{R}_n$ and $\boldsymbol{A}(t) \in \mathbb{R}_{n\times n}$. The equilibrium state $x = 0$ 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 $-\boldsymbol{Q}(t) = \boldsymbol{P}(t)\boldsymbol{A}(t) + \boldsymbol{A}^T(t)\boldsymbol{P}(t) + \dot{\boldsymbol{P}}(t)$.

A similar approach is provided in Example 4.21 of [28] using Theorem 4.10 [28]. Choosing the matrix,

$$\boldsymbol{P} = \frac{1}{2}\begin{bmatrix} \frac{K}{D} + \frac{1}{D} + \frac{D}{K} & \frac{1}{K} \\ \frac{1}{K} & \frac{1}{D} + \frac{1}{KD} \end{bmatrix},$$

which is constant, bounded and positive definite, gives $\boldsymbol{Q}(t) = \frac{1}{\tau_s}\boldsymbol{I}$. Since τ_s is bounded, $\boldsymbol{Q}$ is also bounded $\forall\ t$. Based on Theorem 1 in [26], the entire DMP system is globally contracting. Therefore the DMP system with the proposed adaptive laws and polynomial canonical system asymptotically converges to a unique equilibrium point with $s = 0$, $e = 0$, $v = 0$.

D. Moving Target DMP with Velocity Feedback

In order to better preserve the shape properties of DMP while following a moving target, a velocity feedback of the moving target is incorporated into the DMP formulation similar to [16].

$$\begin{aligned} \tau\dot{x} &= v\,, \\ \tau\dot{v} &= K(\hat{x}_g - x) - D(v - \dot{x}_g) - K\,(\hat{x}_g - x_0)\,s + f(s)\,. \end{aligned} \tag{21}$$

Here $\hat{x}_g$ is an estimate of the final goal position. A fair assumption is that the goal is moving slow enough such that $v \geq \dot{x}_g$ the convergence properties holds for the system in (21). The estimate of the final goal position at time t is updated with a simple weighted average of the current goal position and the position estimated using the goal velocity at time t,

$$\hat{x}_g(t) = x_g(t) + \frac{\dot{x}_g(t)(\hat{T}_t - t)t}{\hat{T}_t}\,. \tag{22}$$

At any time instant t, $x_g(t)$ is the measured current goal position , $\dot{x}_g(t)$ is the velocity of the goal trajectory and $\hat{T}_t$ is the desired time for the entire execution of the DMP.

IV. RESULTS

A. Simulations

Two separate sets of simulations were conducted to evaluate the performance of our approach. In the first simulation setup, a 2 DOF DMP system was set up with a moving target. The control law for τ, for both the approaches are evaluated based on how accurately we can control the execution time of the DMP system. For both the approaches, a single demonstration trajectory of a rotated sinusoidal pattern with $x_0 = (0, 0)$ and $x_g = (1, 1)$ is used. The execution time of the demonstrated trajectory is set as 2s. The sampling time for all the simulations is $dt = 0.01$s, the position tolerance for DMP convergence to the goal/target point is $\mu = 0.01$, i.e $\|x_g - x\| < \mu \|x_g - x_0\|$. The other system parameters are $K = 1000, \alpha = 4, N = 100, k_\tau = 1$, $D = 2\sqrt{K}$ and $\delta = 0.01$. For the purpose of analysis the initial pose is always kept constant at origin (0,0).

The system is analysed for the following four scenarios in simulation, regarding its adaptation to real-time changes in the desired execution time $\hat{T}$ as shown in figure 2 and summarised in the following,

- S-1: $\hat{T}_t$ gradually increases from 2s to 3.6s.
- S-2: $\hat{T}_t$ gradually decreases from 2s to 1.5s.
- S-3: at $t = 1$s $\hat{T}_t$ switches from 2s to 3s.
- S-4: at $t = 1$s $\hat{T}_t$ switches from 2s to 1.5s.

All four scenarios are simulated with a moving target with a random positive velocity $\dot{x}_g \in [0, 0.5]$ at every time instant along both the DOFs. The goal position is updated in real-time based on a simple estimate from equation (22). The DMP system is simulated with a velocity feedback from the moving goal based on the transformation system in (21). For all the scenarios, the real-time behaviour of DMP systems with both exponential and polynomial canonical systems described by (2) and (15) respectively are shown in figure 4. The behavior of their corresponding adapted canonical systems is shown in figure 3.

In all four scenarios, the DMP trajectories converge smoothly to a moving goal as shown in figure 4. The trajectories generated by both DMP systems with standard

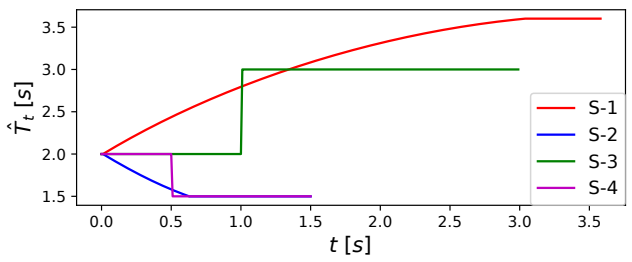


Fig. 2. The change in desired execution time $\hat{T}_t$ for (S-1 to S-4).

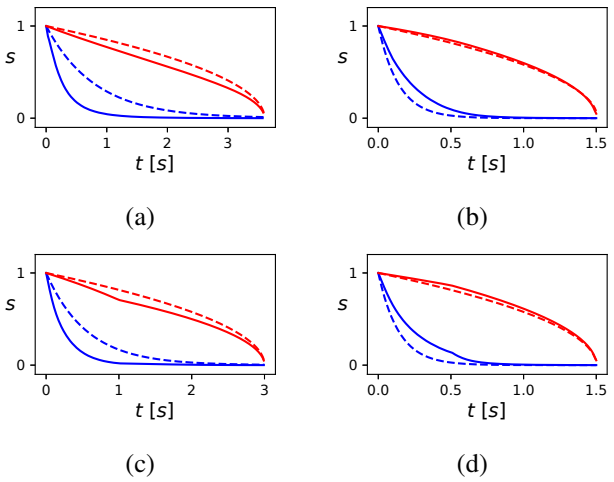


Fig. 3. (a) - (d) represent behaviour of the canonical systems for S-1 to S-4 respectively. The dashed lines represents the ideal DMP profile when the final desired execution time T is known at $t = 0$. The solid lines represent the real-time behaviour of the canonical system based on the control law u given only the current $\hat{T}$. The red and blue lines corresponds to the standard exponential and the proposed polynomial canonical systems respectively.

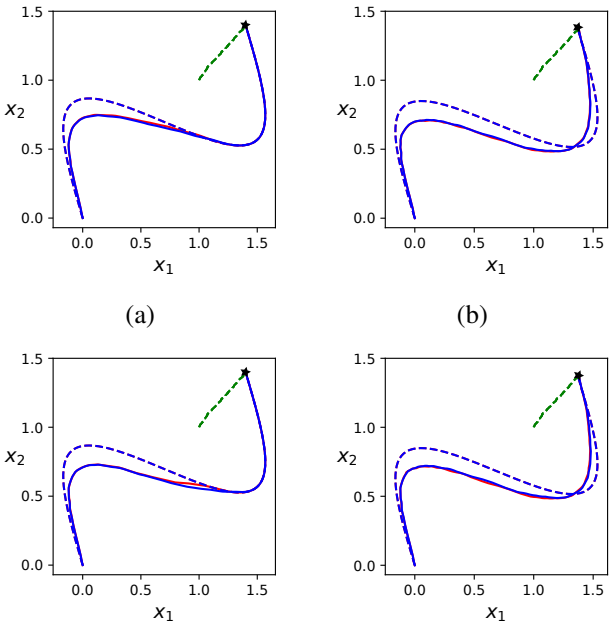


Fig. 4. (a) - (d) represent the trajectory generated by their corresponding transformation systems for S-1 to S-4 respectively. The dashed lines represents the ideal DMP profile when the final desired execution time T and exact end position of goal $x_g(T)$ are known at $t = 0$. The solid lines represent the real-time behaviour of DMP based on the adapted canonical system and the green dotted lines represents the trajectory of the moving target with star shape denoting its final position. The red and blue lines correspond to the standard exponential and the proposed polynomial canonical systems respectively.

exponential and the proposed polynomial canonical systems in figure 4 are very similar in its shape characteristic. The adaptation to real-time changes in $\hat{T}$ can be seen in figure 3 from the evolution of canonical systems. For gradual changes in $\hat{T}$ as shown in figures 3.a and 3.b, both the exponential

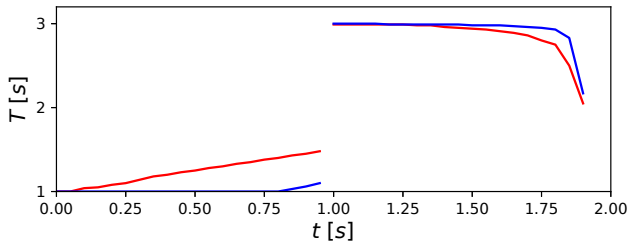


Fig. 5. Performance comparison of DMP system with the two canonical systems, the standard exponential system (in red) and the proposed polynomial canonical system (in blue). The lines denotes the final execution time T for the DMP system when perturbed with a step change $\hat{T}$ from 2s to 1s and from 2s to 3s evaluated every 0.05 seconds for $t \in [0, 1)$s and $t \in [1, 2)$s respectively.

and polynomial canonical systems adapts smoothly with the desired values of τ and converges to the moving goal exactly at the final desired execution time $T = 3.6$s and $T = 1.5$s for S-1 and S-2 respectively. In response to a negative step change in $\hat{T}$ in S-3 as shown in figure 3.c , the exponential canonical system converges to the moving goal $t = 2.98$s with an error of 0.02s, whereas the proposed polynomial based canonical system converges exactly at the final desired execution time $T = 3$s. But in S-4 in response to a negative step change in $\hat{T}$ as shown in figure 3.d, both canonical systems adapts with the desired values of τ and converges to the moving goal exactly at final desired execution time, $T = 1.5$s.

However, these simulations do not reflect the effect of larger changes in the execution time during the later phases of execution. As the difference is expected to become more pronounced when changes are made on $\hat{T}$ towards the later stages of trajectory execution, a second set of simulations are conducted to evaluate the real-time performance of the approach during the DMP execution. Two scenarios of step changes in $\hat{T}$ are considered in the simulation with a stationary target, $x_g = (1, 1)$, all other system parameters remain unchanged.

Negative step change in T from 2s to 1s: This step change in T is simulated with time step of 0.05s for $t \in [0, 1)$. The resulting performance and the corresponding error are shown in figure 5 for $t \in [0, 1]$. The proposed polynomial canonical system outperforms the standard exponential canonical system, as seen by the increased difference towards the later phases of DMP execution as shown in figure 5. Take $t = 0.75$s, on changing T from 2s to 1s the adaptive version of standard exponential canonical system takes 1.38s to finish execution which is a 0.38s delay over the desired time. But with the proposed polynomial canonical system the DMP system converges to the goal at exactly 1s.

Positive step change in T from 3s to 2s: This step change in T is simulated at every 0.05s for $t \in [1, 2)$: The resulting performance and the corresponding error are shown in figure 5 for $t \in [1, 2]$. Here the proposed polynomial canonical system outperforms the standard exponential canonical system. Again, this difference increases towards the later phases of DMP execution as shown in 5. Take $t = 1.45$s, on changing T from 2s to 3s the adaptive version of standard exponential canonical system takes 2.95s to finish execution which is 0.05s earlier that the desired time. But with the proposed polynomial canonical system the DMP system converges to the goal at 2.99s. At $t = 1.75$s, this is 2.8s and 2.95s respectively.

B. Experiments

To validate the performance of the proposed canonical system, experiments were conducted on a UR5 robot manipulator for the task of approaching a moving object. The 3D-cartesian position of the object is tracked in real-time using a Polaris Vicra optical tracking system. A PID trajectory tracking controller is utilized to track the trajectories generated by the DMP. In the experiments, the proposed polynomial canonical system is evaluated for its performance on reaching a moving object within a specified time which is varied during the DMP execution similar to the simulations conducted. The experiment consists of an object moved around by a human operator and the UR5 robot manipulator reaching it from a fixed starting pose. The object is moved around slowly in order to keep the robot velocity within safe limits. An open-source robot control framework developed for UR robots is used for controlling and communicating with the UR5 robot [29]. A demonstrated robot trajectory is collected using the optical tracking system with a static goal as shown in figure 7. The time duration of the demonstration trajectory is $T_d = 10$s. The sampling time for all the experiments is 60Hz, the position tolerance for DMP convergence in Cartesian space is defined to be $\mu = 0.01$, other system parameters are $K = 800, \alpha = 1/2T_d, N = 100, k_\tau = 1$, $D = 2\sqrt{K}$ and $\delta = 0.01$. Four sets of experiments (E-1 to E-4) were conducted similar to the simulations (figure 8),

- E-1: $\hat{T}_t$ gradually increases from 35s to 50s.
- E-2: $\hat{T}_t$ gradually decreases from 50s to 35s.
- E-3: at $t = 15$s, $\hat{T}_t$ switches from 25s to 40s.
- E-4: at $t = 20$s, $\hat{T}_t$ switches from 50s to 40s.

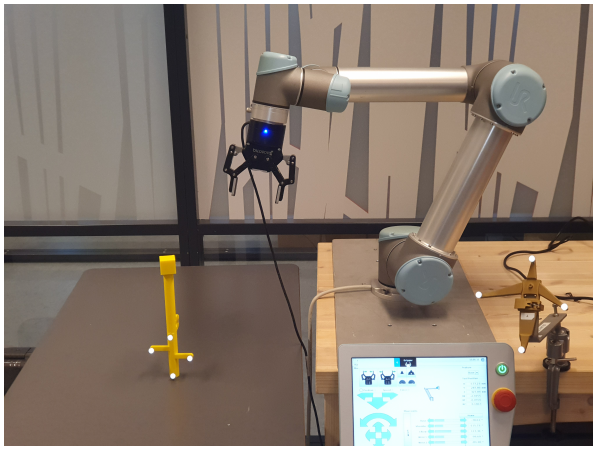


Fig. 6. The experimental setup with UR5 robot and an object fitted with motion capture markers, which is moved around manually by hand.

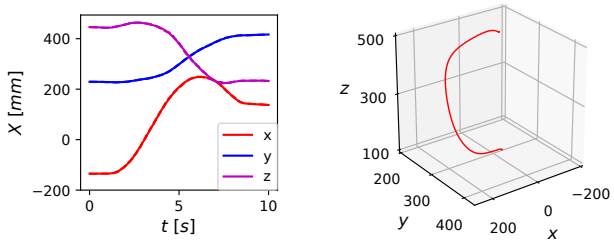


Fig. 7. Demonstration trajectory used for learning the DMP. In (a) time evolution of the Cartesian position (x, y, z) is shown and in (b) the corresponding 3D trajectory in the Cartesian space is shown.

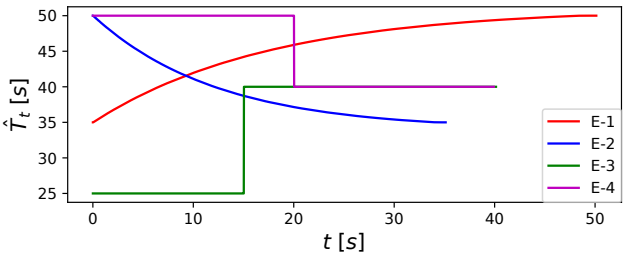


Fig. 8. The change in desired execution time $\hat{T}_t$ for E-1 to E-4

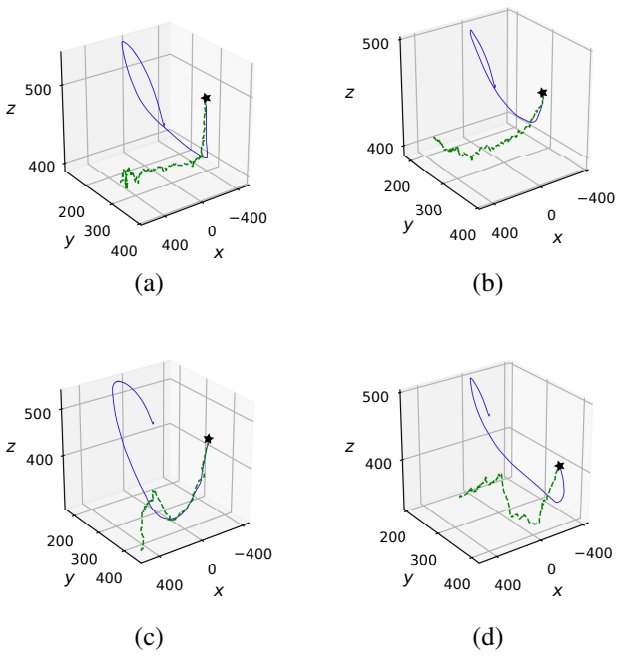


Fig. 9. (a) - (d) represents the 3D Cartesian trajectory generated by the DMP in red and trajectory followed by the UR5 robot in blue for E-1 to E-4 (the blue and red trajectories approximately overlap). The trajectory of the target object is shown in green with star shape denoting its final position. The target trajectory is noisy as the target object is moved around by hand.

The moving object scenario is created by moving the target by hand approximately within a Cartesian space of 1.0m, 0.3m and 0.3m along x, y and z directions respectively. The movement of the object is tracked at approximately 60 Hz using the optical tracking system. The corresponding velocity and estimate of the goal position are fed back to the DMP system. Figure 9 and 10 show the generated DMP trajectories and the behaviour of the polynomial canonical system respectively for E-1 to E-4. We found the experimental results to be very consistent with the simulation results. The experimental results validates the usefulness of the proposed polynomial canonical system in adapting the DMP to the changes in desired task execution time. In all the four trials, the difference between the desired execution time and final execution time was within ± 0.1s which is acceptable compared to the total execution time of the task.

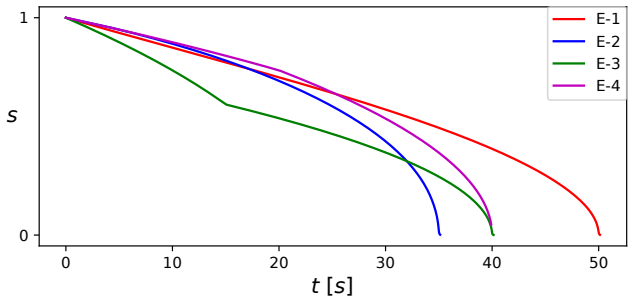


Fig. 10. Behaviour of the polynomial canonical system for E-1 to E-4.

V. CONCLUSIONS

In this study, we extended the DMP framework with real-time control over the execution time while preserving the shape characteristics. We used a DMP formulation focused on a moving target scenario, incorporating the target's velocity feedback and a simple estimation of the target's final position based on its current position and velocity. We formulated an adaptive law for the standard exponential canonical system and an alternative polynomial canonical system to have better real-time control on the task execution time even during the later phases of DMP execution. We compared the proposed polynomial canonical system with the standard exponential canonical system for their relative performance. On comparing both the canonical systems, the proposed polynomial canonical system was found to perform better in all the simulations conducted. The proposed polynomial canonical system was evaluated on an experiment using a UR5 robotic manipulator with a moving target. The extended DMP framework is useful in various robotic tasks with strict task execution time requirements, when the desired task execution time is unknown prior to performing the task itself. This requirement is very relevant to robotic tasks involving moving targets such as industrial handling of moving parts, and human-robot collaborative tasks.

ACKNOWLEDGMENT

This work was part of the project "Dynamic Robot Interaction and Motion Compensation" funded by the Research Council of Norway under contract number 270941.

REFERENCES

[1] S. Schaal, S. Kotosaka, and D. Sternad, "Nonlinear dynamical systems as movement primitives," in *IEEE international conference on humanoid robotics*, 2000, pp. 1–11.

[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. IEEE, 2002, pp. 1398–1403.

[3] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics research. the eleventh international symposium*. Springer, 2005, pp. 561–572.

[4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in neural information processing systems*, 2003, 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, 2013.

[6] A. Kramberger, I. Iturrate, M. Deniša, S. Mathiesen, and C. Sloth, "Adapting learning by demonstration for robot based part feeding applications," in *2020 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2020, pp. 954–959.

[7] R. S. Sharma, S. Shukla, H. Karki, A. Shukla, L. Behera, and K. Venkatesh, "Dmp based trajectory tracking for a nonholonomic mobile robot with automatic goal adaptation and obstacle avoidance," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8613–8619.

[8] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2008, pp. 91–98.

[9] A. Rai, F. Meier, A. Ijspeert, and S. Schaal, "Learning coupling terms for obstacle avoidance," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 512–518.

[10] M. Chi, Y. Yao, Y. Liu, and M. Zhong, "Learning, generalization, and obstacle avoidance with dynamic movement primitives and dynamic potential fields," *Applied Sciences*, vol. 9, no. 8, p. 1535, 2019.

[11] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning stylistic dynamic movement primitives from multiple demonstrations," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1277–1283.

[12] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural networks*, vol. 24, no. 5, pp. 493–500, 2011.

[13] M. Ginesi, N. Sansonetto, and P. Fiorini, "Overcoming some drawbacks of dynamic movement primitives," *arXiv*, pp. arXiv–1908, 2019.

[14] A. Ude, B. Nemec, T. Petrić, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2997–3004.

[15] L. Koutras and Z. Doulgeri, "A correct formulation for the orientation dynamic movement primitives for robot control in the cartesian space," in *Conference on Robot Learning*, 2020, pp. 293–302.

[16] M. Prada and A. Remazeilles, "Dynamic movement primitives for human robot interaction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, workshop on Robot Motion Planning: online, reactive and in Real-time, Algarve, Portugal*, 2012.

[17] 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)*. IEEE, 2016, pp. 5481–5488.

[18] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 763–768.

[19] 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*. IEEE, 2010, pp. 853–858.

[20] S. Kim, E. Gribovskaya, and A. Billard, "Learning motion dynamics to catch a moving object," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2010, pp. 106–111.

[21] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *arXiv preprint arXiv:2102.03861*, 2021.

[22] H. Hoffmann, P. Pastor, D.-H. 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*. IEEE, 2009, pp. 2587–2592.

[23] L. Koutras and Z. Doulgeri, "Dynamic movement primitives for moving goals with temporal scaling adaptation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 144–150.

[24] D. E. Knuth, "On the lambert w function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

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

[26] 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.

[27] H. J. Marquez, *Nonlinear control systems: analysis and design*. John Wiley Hoboken^ eN. JNJ, 2003, vol. 161.

[28] H. K. Khalil, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[29] A. Østvik, L. E. Bø, and E. Smistad, "Echobot: An open-source robotic ultrasound system," *Proc. IPCAI*, pp. 1–4, 2019.