

Temporal Coupling of Dynamical Movement Primitives for Constrained Velocities and Accelerations

Albin Dahlin^{ID} and Yiannis Karayiannidis^{ID}

Abstract—The framework of Dynamical Movement Primitives (DMPs) has become a popular method for trajectory generation in robotics. Most robotic systems are subject to saturation and/or kinematic constraints on motion variables, but DMPs do not inherently encode constraints and this may lead to poor tracking performance. Temporal coupling (online temporal scaling) of DMPs represents a possible way for handling constrained systems. This letter presents a temporal coupling for DMPs to handle velocity and acceleration constraints for the generated trajectory. A novel filter is presented based on a potential function which proactively scales the trajectory before reaching the acceleration limits. In this way, the velocities and accelerations remain within the limits even for trajectories with aggressive accelerations and stricter bounds. The performance of the proposed method is demonstrated by means of simulations and experiments on a UR10 robot.

Index Terms—Motion control, reactive and sensor-based planning.

I. INTRODUCTION

DYNAMICAL Movement Primitives (DMPs), first introduced in [1] and presented in detail in [2], encode the desired trajectory using dynamical second-order systems. They are highly flexible in creating complex movements, and their structure allows for easy trajectory scaling in space and time by a simple modulation of a goal position or temporal scaling factor, respectively. One of the major advantages of the DMP framework is that it allows reactive planning by readily introducing sensory feedback in the planning process. In this way, the trajectory can adjust to unpredictable events and obstacles in the environment [3], be modified online to improve the task performance according to a given metric [4]–[6] or scale in order to adapt the final position to a moving object [2], [7]. Allowing online modifications of the trajectory, and thereby making it less predictable, can however raise problems when constraints on velocity and acceleration need to be accommodated in the motion planning. These constraints may arise due to actuator

limitations or safety limits, e.g. for robots in human environments. Maintaining the system trajectory within the limits can be guaranteed by introducing reference governors [8], [9] or by direct saturation in the controller, but this leads to divergence of the system trajectory from the target trajectory generated by the DMP and may distort the system path significantly. This letter deals with the problem of adapting a nominal trajectory online to respect velocity and acceleration constraints while preserving the original trajectory path.

In [10] a model predictive control approach inspired by the DMP framework was presented which offers the possibility to constrain the velocity and it may be expanded to handle acceleration constraints. This method of constrained optimization ensures that the velocity is kept within an allowed region for each degree of freedom but cannot ensure that the shape of the path remains unaltered. This is the general case for all types of spatial modulations on the dynamical system level since the temporal evolution for the rest of the system is not affected. To remain on the same path, it is necessary to simultaneously decrease the velocity in all dimensions. This motivates for temporal coupling that consists in modifications of the DMP temporal scaling factor rather than spatial coupling. In [11] temporal coupling is used to improve tracking performance of the controller by reducing the speed based on the tracking error. With this method, distortion of the path is necessary to slow down the trajectory and can therefore not be directly used to limit the velocity or acceleration. A method to counteract increased velocities due to moving goals was presented in [7] but this does not allow for explicit velocity limits and does not consider any other possible modification (obstacle avoidance term, modified kernel weights, etc.) that affect the velocity of the DMP system apart from a modified goal. In [12], we presented a temporal coupling to guarantee predefined velocity limits for the continuous-time DMP trajectory. However, for discrete implementation, this method encounters some stability issues depending on sampling rate and tuning parameters. To the best of our knowledge, no temporal coupling schemes have been presented to treat the problem of limited acceleration.

Temporal coupling for DMPs has a close relation to the path-velocity decomposition paradigm [13] since the temporal scaling factor in the DMP framework can be viewed as the path traversal time. In [14], online trajectory scaling using path-velocity decomposition was presented to handle torque constraints for robot joints and in [15], [16] the idea has been

Manuscript received October 7, 2020; accepted February 1, 2021. Date of publication February 11, 2021; date of current version March 11, 2021. This letter was recommended for publication by Associate Editor L. Jamone and Editor M. Vincze upon evaluation of the reviewers' comments. This work was supported by UNICORN, Vinnova, Effektiva och uppkopplade transportsystem. (Corresponding author: Albin Dahlin.)

The authors are with the Department of Electrical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden (e-mail: albin.dahlin@chalmers.se; yiannis@chalmers.se).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2021.3058874>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3058874

extended to consider both joint and end-effector velocities and accelerations as well as path velocity and acceleration. The aforementioned techniques scale the nominal trajectory online such that the specified limits are ensured *if* a solution exists. The problem may however become infeasible for some trajectories and a trade-off between path preservation and constraint satisfaction must be made. In [17] a condition for assuring feasibility over the entire trajectory is presented but is only treating the case of offline motion planning.

In this letter, we present a temporal coupling algorithm to constrain both the velocity and acceleration of the DMP in discrete time. We adopt the technique of transforming the differential constraints into constraints for the path traversal variable (temporal scaling factor) as in [13]–[17]. The feasibility condition in [17] is also applied but is here formalized for an online one-step prediction usage. In addition, a novel filter, inspired by the potential based update law in [12] used to constrain velocities, is introduced in a discrete-time setting to further facilitate feasibility for the online trajectory scaling over the complete movement. Compared to previous works on trajectory scaling [13]–[17], the proposed method can retain the feasibility for the differential constraints for a wider set of trajectories by also scaling in a proactive, instead of solely reactive, manner and can thereby satisfy the constraint to a greater extent. This comes at a cost of a slightly slower trajectory depending on a tuning parameter. We also demonstrate that satisfying the constraints via temporal coupling results in an accurate reproduction of the original path. In contrast to previous literature in DMP, we address simultaneously velocity and acceleration limits in the reactive planning process, with capability to adapt to online modification of the limits.

The letter is organized as follows. In Section II, a brief description of DMPs is provided. The proposed temporal coupling for constrained velocity and acceleration is presented in Section III. Evaluations are provided in terms of simulation in Section IV and experiments on a UR10 robot in Section V. Final conclusions are drawn in Section VI.

II. DYNAMICAL MOVEMENT PRIMITIVES

The framework of DMPs employs second-order linear dynamical systems with a nonlinear virtual forcing term along with a canonical dynamical system for a phase variable as follows:

$$\tau \dot{z} = K(g - y) - Dz + f(s) \doteq h(z, y, s) \quad (1)$$

$$\tau \dot{y} = z, \quad (2)$$

$$\tau \dot{s} = -\alpha s. \quad (3)$$

Here the position $y \in \mathbb{R}^n$, the scaled velocity $z \in \mathbb{R}^n$ and the phase variable $s \in \mathbb{R}$ are initialized to y_0 , $\mathbf{0}$ and 1 respectively, with n denoting the dimension and y_0 being the desired initial position of the target trajectory. $g \in \mathbb{R}^n$ is the attractor point (goal), $\tau \in \mathbb{R}$ is a temporal scaling term, K and D are constants determining the dynamics of the unforced system and α is a constant determining the rate of change for the phase variable. The forcing term, $f(s)$, is formulated using N weighted Gaussian basis functions and such that it converges to zero as

s converge to zero leading to a convergence of y to the goal. Since the virtual force is dependent on the phase variable rather than time, it is possible to couple the DMP with other dynamical systems and coordinate DMPs with several degrees-of-freedom. The selection of weights in the forcing term can be solved as an offline optimization problem such that the DMP evolution is a best fit to a target trajectory [18]. In this way, the DMPs can encode any trajectory along a sufficiently smooth path using the parameters τ , g and the kernel weights, with increasing precision as the number of kernels grows. See [2] for a more thorough description of DMPs.

Normally in control applications, the reference is kept constant for one time step, Δt , such as when addressing velocity constrained closed-loop inverse kinematics [19]. Thus, using DMP systems for online trajectory generation requires discretization that can be implemented by using Euler integration

$$z_{k+1} = z_k + \dot{z}_k \Delta t \quad (4)$$

$$y_{k+1} = y_k + \dot{y}_k \Delta t \quad (5)$$

$$s_{k+1} = s_k + \dot{s}_k \Delta t \quad (6)$$

where Δt is the sampling time and \dot{z}_k , \dot{y}_k and \dot{s}_k are defined as in (1)–(3). When temporal coupling is used, the scalar τ is modified online and we also have

$$\tau_{k+1} = \tau_k + \dot{\tau}_k \Delta t \quad (7)$$

where $\dot{\tau}_k$ is determined online by an update law. In this letter we refer to the DMP system without temporal coupling as the nominal system with corresponding nominal temporal scaling factor denoted by τ^* .

III. TEMPORAL COUPLING FOR CONSTRAINED VELOCITY AND ACCELERATION

The aim is to constrain the DMP system trajectory velocity and acceleration such that

$$|\dot{y}_{k,i}| \leq v_{\max,i}, \quad \forall k, i, \quad (8)$$

$$|\ddot{y}_{k,i}| \leq a_{\max,i}, \quad \forall k, i, \quad (9)$$

where $y_{k,i}$ indicate the i th element of y_k at time step k , while retaining the path shape. This should be done such that the path traversal speed is not increased compared to the nominal trajectory. Expressed in terms of temporal scaling factor this corresponds to

$$\tau_k \geq \tau^*, \quad \forall k. \quad (10)$$

As the trajectory velocity, \dot{y}_k , is directly proportional to the temporal scaling factor, task scaling could be applied to limit the velocity. However, the acceleration, given by differentiation of (2)

$$\ddot{y}_k = \frac{\dot{z}_k - \dot{y}_k \dot{\tau}_k}{\tau_k} = \frac{h(z_k, y_k, s_k) - z_k \dot{\tau}_k}{\tau_k^2}, \quad (11)$$

cannot be constrained in a similar straightforward manner. As seen in (11), the acceleration depends both on the current temporal scaling factor and its derivative, i.e. it depends explicitly on the decision variable for the temporal coupling, $\dot{\tau}_k$. The

constraints (8)-(9) can be stated as $4n$ inequality constraints on the one-dimensional variable $\dot{\tau}_k$. In this way, the trajectory velocity and acceleration can be limited while the trajectory path is preserved. However, due to this limitation, it is not certain that all limits result in a feasible solution, e.g. one dimension may yield an upper limit of $\dot{\tau}_k$ that is lower than the lower limit yielded from another dimension. Furthermore, the limits on $\dot{\tau}_k$ imposed from the acceleration constraints may very well be negative. Intuitively, the trajectory should slow down (τ should increase) in order to decrease the trajectory acceleration, but directly imposing these constraint would rather speed up the trajectory in such cases. In fact, as the velocity and acceleration are directly related to $\frac{1}{\tau_k}$, decreasing τ_k makes the constraints tighter and may lead to infeasibility in future steps.

In this section, we present a temporal coupling algorithm for constrained velocity and acceleration. First, the velocity and acceleration constraints are transformed into constraints on $\dot{\tau}_k$. Lower bounds of $\dot{\tau}_k$ are then presented to guarantee feasibility of the acceleration constraints and limited path traversal speed. Next, an update rule for $\dot{\tau}_k$ before applying the constraints, referred to as the *base update law*, is introduced with intention to keep the acceleration away from the limits to improve feasibility of the saturation of $\dot{\tau}_k$. Finally, the complete algorithm combining all steps is presented.

A. Acceleration Constraints Imposed on $\dot{\tau}$

By use of (11), the acceleration constraints (9) can be formulated as

$$-a_{\max,i} \leq \frac{h(z_{k,i}, y_{k,i}, s_k) - \dot{\tau}_k z_{k,i}}{\tau_k^2} \leq a_{\max,i}, \quad (12)$$

or in vector notation after rearranging to one side

$$A_k \dot{\tau}_k + B \tau_k^2 + C_k \leq 0, \quad (13)$$

where

$$A_k = \begin{bmatrix} -z_k \\ z_k \end{bmatrix}, B = \begin{bmatrix} -a_{\max} \\ -a_{\max} \end{bmatrix}, C_k = \begin{bmatrix} h(z_k, y_k, s_k) \\ -h(z_k, y_k, s_k) \end{bmatrix}. \quad (14)$$

Inequality (13) can be reduced to scalar upper and lower bounds for $\dot{\tau}_k$ imposed by the acceleration constraints

$$\dot{\tau}_k^{\min,a} \leq \dot{\tau}_k \leq \dot{\tau}_k^{\max,a}, \quad (15)$$

where the bounds are defined as

$$\dot{\tau}_k^{\min,a} = \max_{i \in \{1, \dots, 2n\}} \left\{ -\frac{B_i \tau_k^2 + C_{k,i}}{A_{k,i}} \mid A_{k,i} < 0 \right\}, \quad (16)$$

$$\dot{\tau}_k^{\max,a} = \min_{i \in \{1, \dots, 2n\}} \left\{ -\frac{B_i \tau_k^2 + C_{k,i}}{A_{k,i}} \mid A_{k,i} > 0 \right\}, \quad (17)$$

and $A_{k,i}$ denote the i th element of A_k .

B. Velocity Constraints

As $\dot{\tau}_k$ does not directly impact the velocity but indirectly through integration, one-step prediction of the system velocity is possible and necessary to transform the velocity constraints into constraints on $\dot{\tau}_k$. From (2), we see that the velocity constraints

(8) can be reformulated as

$$-v_{\max,i} \leq \frac{z_{k+1,i}}{\tau_{k+1}} \leq v_{\max,i}, \quad (18)$$

or in vector notation after proper rearrangement

$$D \tau_{k+1} + A_{k+1} \leq 0, \quad (19)$$

where

$$D = \begin{bmatrix} -v_{\max} \\ -v_{\max} \end{bmatrix}. \quad (20)$$

As all elements of D are negative, (19) implies a scalar lower bound for τ_{k+1} defined as

$$\tau_{k+1}^{\min,v} = \max_{i \in \{1, \dots, 2n\}} \left\{ -\frac{A_{k+1,i}}{D_i} \right\}, \quad (21)$$

that, using (7), corresponds to

$$\dot{\tau}_k^{\min,v} = \frac{\tau_{k+1}^{\min,v} - \tau_k}{\Delta t}. \quad (22)$$

C. Feasibility of Acceleration Limits

Selecting $\dot{\tau}_k$ such that (15) is satisfied, the acceleration constraints (9) are ensured under the assumption that such a solution exists, i.e. $\dot{\tau}_k^{\min,a} \leq \dot{\tau}_k^{\max,a}$. Similar to [17], the feasibility of (15) can be guaranteed by requiring the inequality to hold for all combinations of the min and max functions. Let i and j denote all indices such that $A_{k+1,i} < 0$ and $A_{k+1,j} > 0$. Now, the condition for feasibility of (15) at next time step, i.e. $\dot{\tau}_{k+1}^{\min,a} \leq \dot{\tau}_{k+1}^{\max,a}$, can be written as

$$-\frac{B_i \tau_{k+1}^2 + C_{k+1,i}}{A_{k+1,i}} \leq -\frac{B_j \tau_{k+1}^2 + C_{k+1,j}}{A_{k+1,j}} \quad (23)$$

and extraction of τ_{k+1} leads to the inequality

$$\tau_{k+1}^2 \geq \frac{C_{k+1,i}|A_{k+1,j}| + C_{k+1,j}|A_{k+1,i}|}{|B_i A_{k+1,j}| + |B_j A_{k+1,i}|}. \quad (24)$$

Here we have made use of the fact that all elements of B are negative and the signs of all terms of $A_{k+1,i}$ and $A_{k+1,j}$ are known. The inequality (24) is satisfied for all $C_{k+1,i}|A_{k+1,j}| + C_{k+1,j}|A_{k+1,i}| < 0$ and the lower bound for τ_{k+1}

$$\tau_{k+1}^{\min,f} = \max_{\substack{i \in \{1, \dots, 2n\} \\ j \in \{1, \dots, 2n\}}} \left\{ \sqrt{\frac{C_{k+1,i}|A_{k+1,j}| + C_{k+1,j}|A_{k+1,i}|}{|B_i A_{k+1,j}| + |B_j A_{k+1,i}|}} \mid \begin{array}{l} C_{k+1,i}|A_{k+1,j}| + C_{k+1,j}|A_{k+1,i}| > 0 \\ A_{k+1,i} < 0, A_{k+1,j} > 0 \end{array} \right\}. \quad (25)$$

is used in (7) to get

$$\dot{\tau}_k^{\min,f} = \frac{\tau_{k+1}^{\min,f} - \tau_k}{\Delta t} \quad (26)$$

which ensures feasibility of (15) at next time step.

D. Path Traversal Speed

A final bound on $\dot{\tau}_k$ arises from constraint (10) to ensure that the path traversal speed is not faster than for the nominal

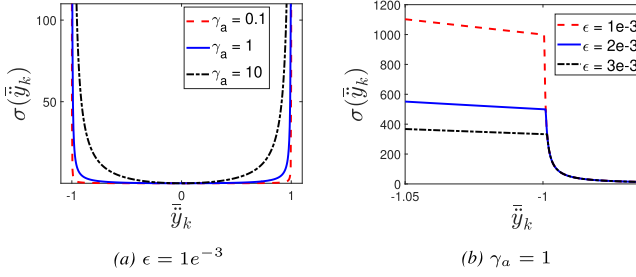


Fig. 1. The potential term for three values of γ_a and ϵ .

trajectory. Combining (7) and (10) we get

$$\dot{\tau}_k^{\min, \tau^*} = \frac{\tau^* - \tau_k}{\Delta t}. \quad (27)$$

E. Base Update Law

A first intuitive approach for the base update law is to select $\dot{\tau}_k$ such that $\tau_{k+1} = \tau^*$. The trajectory would then follow the nominal trajectory whenever no constraints are violated and slow down otherwise. However, this would in many cases lead to infeasibility due to extreme changes of τ when reaching the limits. Instead of drastically changing τ at the limits we propose an update law that increases τ gradually as the limits are approached. That is, instead of applying reactive task scaling, select $\dot{\tau}$ such that the trajectory is slowed down in a preventing way whenever approaching the limits. In [12] an update law for temporal coupling using repulsive potentials was developed which enforces the velocity to stay within prescribed bounds for the continuous-time case. Here we extend the idea to use acceleration constraints instead. However, as $\dot{\tau}_k$ directly affect the acceleration, the true acceleration cannot be used to calculate $\dot{\tau}_k$. Instead, the normalized acceleration evaluated statically with $\dot{\tau}_k = 0$, i.e. $\ddot{y}_{k,i} = \frac{h_i(z_k, y_k, s_k)}{\tau_k^2 a_{\max, i}}$, is used. The statically evaluated acceleration gives an indication of how close to the limits the acceleration components $\ddot{y}_{k,i}$ are assuming τ remains unaltered, i.e. $\tau_{k+1} = \tau_k$. The base update law is defined as follows:

$$\dot{\tau}_k = \gamma_* (\tau^* - \tau_k) + \tau_k \sigma(\ddot{y}_k) \quad (28)$$

where γ_* is a positive constant. The function σ is defined as

$$\sigma(\ddot{y}_k) = \gamma_a \sum_{i=1}^n \frac{\ddot{y}_{k,i}^2}{\max(1 - \ddot{y}_{k,i}^2, \gamma_a \epsilon)} \quad (29)$$

with ϵ being a small positive constant and γ_a a positive tuning parameter. The update law (28) consists of two terms. One acts as a potential function which aims at maintaining the acceleration within the limits by increasing τ whenever these are approached. The other drives τ back down to the nominal temporal scaling factor, τ^* . The potential function (29) in the one dimensional case is shown in Fig. 1. As seen, the function is rather flat when \ddot{y}_k is within some margin from the normalized limits, $[-1, 1]$, and increases drastically as it approaches the limits. The implication of γ_a is also obvious from Fig. 1(a): a smaller value of γ_a “flattens” the potential function in the region of allowed values and leads to a lower impact of the potential term when \ddot{y}_k

is not close to its limit. When all accelerations are sufficiently far away from the limits, the potential term in (28) is negligible and the base update law can be interpreted as a first-order filter of τ^* with time constant $\frac{1}{\gamma_*}$. Hence, a bigger γ_* leads to a faster convergence to the nominal temporal scaling factor. The proposed logarithmic-like function without the max term is properly defined in a region $(-1, 1)$. To continuously extend the domain of σ such that it is properly defined in \mathbb{R} we use the max function in the denominator of eq (29). This is equivalent to a quadratic extension for values $\ddot{y}_{k,i}^2 > 1 - \gamma_a \epsilon$, as illustrated in Fig. 1(b), and is preferred for compact representation purposes. Other extensions ensuring large positive values before leaving, and outside of, the bounds can be used. It should be noted that using the base update law (28), the trajectory is slowed down *before* a limit is reached and might therefore result in a slightly slower trajectory than necessary. This conservative way of treating the acceleration limits however gives more capacity for feedback terms in saturated controllers and can therefore lead to an improved tracking performance of the controller. The tuning parameter γ_a determines how cautious the trajectory scaling should be about approaching the acceleration limits. For selection of the tuning parameters, a good principle is to select such that the ratio $0.1 \leq \frac{\gamma_a}{\gamma_*} \leq 10$ to have a good balance between the return rate and potential term in (28).

F. Algorithm

As opposed to [12], where the trajectory is ensured to stay within the velocity limits, the update law (28) itself cannot guarantee that the acceleration limits are not violated. This is due to the direct impact of the decision variable, $\dot{\tau}_k$, on the acceleration, \ddot{y}_k . Rather, it is used as a base update law before applying the derived constraints on $\dot{\tau}_k$. In Algorithm 1, the complete temporal coupling for constrained velocity and acceleration is presented. The order of saturation of $\dot{\tau}_k$ in step 7 is important as there might be no feasible solution, i.e. $\dot{\tau}_k^{\min} > \dot{\tau}_k^{\max, a}$. This can happen even though the feasibility of acceleration limits are ensured by $\dot{\tau}_{k-1}^{\min, f}$ since, for instance, we may still have $\dot{\tau}_k^{\min, f} > \dot{\tau}_k^{\max, a}$. In these cases, the lower bound has priority. The reasoning for this prioritization is that the infeasibility indicates that the current path traversal is too fast and the trajectory should slow down (τ should increase) to enable feasibility in future states at the expense of a momentary violation of the acceleration limits. This order of prioritization also guarantees that the velocity limits are satisfied at all time since we get $\dot{\tau}_k \geq \dot{\tau}_k^{\min, v}$.

Remark: The use of the statically evaluated acceleration based update law (28) is essential for satisfying the acceleration constraints and cannot be replaced by a velocity based update law such as in [12]. The latter would only yield redundant velocity bounds, given that $\dot{\tau}$ is saturated by the limit in (22). Note that resolving this redundancy by removing (22) would also jeopardize the velocity limits because of the max saturation in step 4.

IV. SIMULATIONS

A two-dimensional DMP system was implemented with the nominal trajectory fitted to a handwritten trajectory with a

TABLE I
PARAMETER SELECTION OF TEMPORAL COUPLING

	Simulation	Experiment
γ_a	0.5	0.5
γ_*	1	1
ϵ	$1e^{-3}$	$1e^{-3}$
v_{\max}	-	$[30 \ 30]^\circ/\text{s}$
a_{\max}	$[0.454 \ 0.416]$	$[100 \ 100]^\circ/\text{s}^2$

Algorithm 1: Temporal Coupling for Constrained Velocity and Acceleration.

Input: y_k, z_k, s_k, τ_k

Output: $\hat{\tau}_k$

Parameter: $\gamma_*, \gamma_a, \epsilon, v_{\max}, a_{\max}$

- 1 Calculate DMP prediction z_{k+1}, y_{k+1} and s_{k+1} .
- 2 Formulate matrices $A_k, A_{k+1}, B, C_k, C_{k+1}$ and D .
- 3 Compute bounds $\dot{\tau}_k^{\min,a}, \dot{\tau}_k^{\max,a}, \dot{\tau}_k^{\min,v}, \dot{\tau}_k^{\min,f}$ and $\dot{\tau}_k^{\min,\tau^*}$ using (16), (17), (22), (26) and (27) respectively.
Let $\hat{\tau}_k^{\min} = \max\{\dot{\tau}_k^{\min,a}, \dot{\tau}_k^{\min,v}, \dot{\tau}_k^{\min,f}, \dot{\tau}_k^{\min,\tau^*}\}$.
- 4 Calculate $\hat{\tau}_k$ with a saturated base update law

$$\hat{\tau}_k = \gamma_*(\tau^* - \tau_k) + \tau_k \sigma(\ddot{y}_k)$$

$$\hat{\tau}_k = \max(\min(\hat{\tau}_k, \dot{\tau}_k^{\max,a}), \dot{\tau}_k^{\min})$$

G-shaped path and nominal total time duration of 6.4 seconds. The DMP was used as reference generator for a saturated PD controller with a feedforward term, $u = \text{sat}(\ddot{y} + k_v(\dot{\xi} - \dot{y}) + k_p(\xi - y), u_{\max})$, controlling a second-order system $\ddot{\xi} = u$ at 1 kHz. Here, sat is a saturating function defined as $\text{sat}(x, \bar{x}) = \min(\max(x, -\bar{x}), \bar{x})$. The controller was designed to achieve a critically damped control loop with parameters $k_p = 25$ and $k_v = 10$. The acceleration limit was set to half the peak acceleration of the nominal trajectory and no velocity limit was initially set. The parameters for the temporal coupling was set according to Table I. When approximately half of the path had been traversed ($s = 0.65$) a limit for the velocity in the second dimension was set to 0.15 and the controller saturation was online adapted to this. The generation of the trajectories and the system updates were implemented in Matlab.¹ In this simulation, we compare the generated target trajectories and resulting tracking performance for three cases of DMP systems; without temporal coupling (referred to as *nominal*), using Algorithm 1 as temporal coupling (referred to as *proposed*) and using Algorithm 1 as temporal coupling but without potential term (referred to as *proposed w/o σ*).

In Figs. 2 and 3 the DMP velocity and DMP acceleration are shown respectively for the three cases. Note that the phase variable, and not time, is used in the x-axis to simplify comparison. It is clear that for the nominal system, both velocity and acceleration limits are exceeded. When the proposed temporal coupling is used, the acceleration stays within the bounds for the entire simulation and the trajectory adapts to the online modified velocity limit and respects the new limits for the rest of the

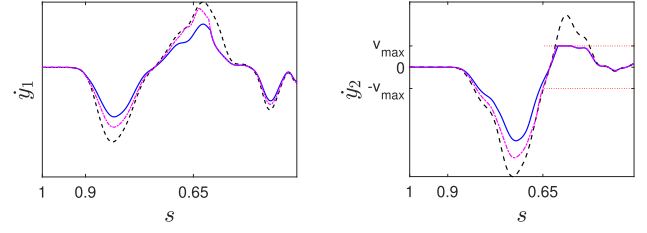


Fig. 2. Target trajectory velocity for the cases: nominal (dashed black line), proposed (solid blue line) and proposed w/o σ (dash-dotted magenta line).

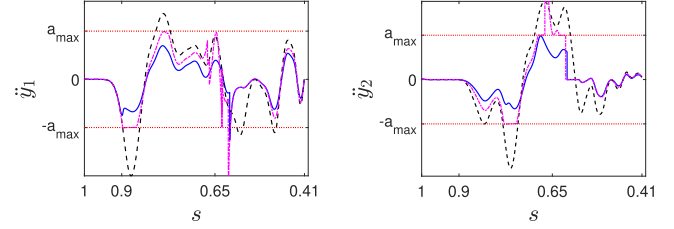


Fig. 3. Target trajectory acceleration for the cases: nominal (dashed black line), proposed (solid blue line) and proposed w/o σ (dash-dotted magenta line).

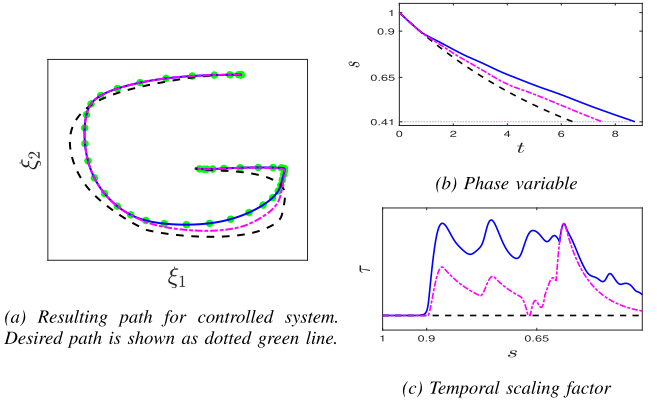


Fig. 4. Path and temporal evolution for the cases: nominal (dashed black line), proposed (solid blue line) and proposed w/o σ (dash-dotted magenta line).

simulation. The resulting system path is shown in Fig. 4(a) for the three cases along with the desired path. As seen, the control saturation results in a significantly altered system path when the nominal DMP system is used as reference generator while the path is followed closely when Algorithm 1 is applied.

The significance of the potential term in the update law becomes clear when comparing the tracking performance of the controller using Algorithm 1 with and without potential. The worse performance when no potential is used can be attributed partly to exceeded acceleration limits in some regions due to infeasibility, and partly to lack of capacity in the control feedback when the target trajectory acceleration is fully saturated. As seen in Figs. 4(b) and 4(c), when the base update law includes the potential term, τ is increased (trajectory is slowed down) slightly before the acceleration limit is reached (at $s = 0.9$) and the imposed constraints on $\dot{\tau}$ are thus relaxed which makes the saturation feasible over the whole trajectory. With potential, a

¹Code can be found in https://github.com/albindgit/TC_DMP_constrainedVelAcc.git

TABLE II
TIME TO REACH FINAL POSITION, T , TOTAL TIME OF EXCEEDED
ACCELERATION LIMITS, T_a , AND MEAN SQUARED ERROR (MSE) AS A
FRACTION OF MSE WHEN $\gamma_a = 0.5$

$\frac{\gamma_a}{\gamma_*}$	5.0	1.0	0.5	0.1	0.05	0.01	0
T	11.6	9.27	8.73	7.94	7.82	7.71	7.51
T_a	0	0	0	0.739	0.735	0.733	0.995
$\frac{MSE}{MSE_{\gamma_a=0.5}}$	0.476	0.842	1.0	3.68	465.0	2010.0	7030.0

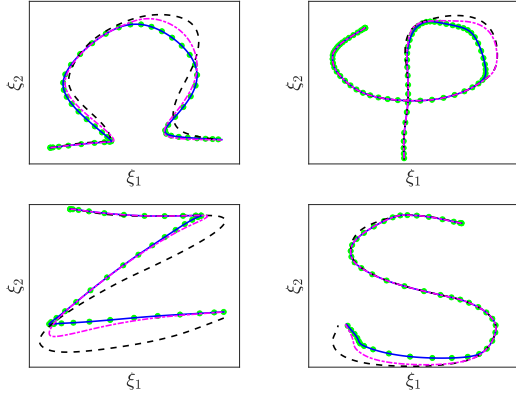


Fig. 5. Resulting path for controlled system for different handwritten trajectories for the cases: nominal (dashed black line), proposed (solid blue line) and proposed w/o σ (dash-dotted magenta line). Desired path is shown as dotted green line.

smoother acceleration is also achieved compared to the more nervous behavior when no potential is used.

As mentioned, it is not guaranteed that the DMP acceleration stays within the bounds for the full trajectory even with the potential. In particular, when small potential gains, γ_a , compared to the nominal gain, γ_* , are chosen, the risk for momentary exceeding the limits is higher. This would also lead to less capacity for feedback compensation in the controller. Both of these aspects influence the tracking error of the desired path. In Table II, a comparison of the resulting mean squared error, total time of exceeded acceleration limits and the time to reach the final position is shown for several simulations with various γ_a but with maintained $\gamma_* = 1$. It is clear that reducing the potential gain results in a slightly faster trajectory but with an increased tracking error.

Additional trajectories have been evaluated with the same simulation setup without any added online velocity constraints, these are shown in Fig. 5.

V. EXPERIMENTS

The proposed method was tested on a 6 ° of freedom UR10 robotic manipulator in real world conditions. The robot was controlled by a saturated joint velocity controller with position feedback, $u = \text{sat}(\dot{y} + k_p(y - q), v_{\max})$, at a frequency of 125 Hz with $v_{\max} = 30^\circ/\text{s}$ and $k_p = 2$, tuned approximately to minimize tracking error for the nominal case. This velocity command was sent, using ROS¹, to the low-level controller of the UR10 that further saturated the acceleration to a level $a_{\max} = 100^\circ/\text{s}^2$. A DMP was used as reference generator, fitted with a nominal 6D trajectory in the joint space resulting in

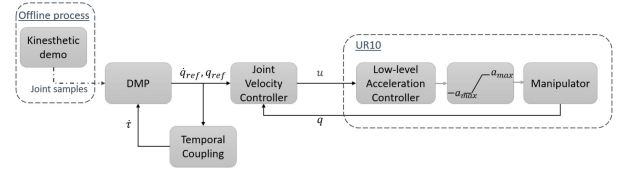


Fig. 6. Experiment setup.

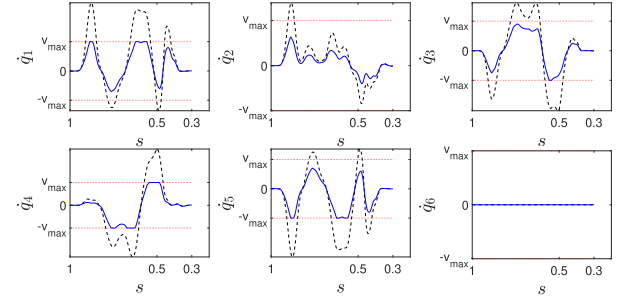


Fig. 7. Target joint velocities with proposed temporal coupling (solid blue line) and without (dashed black line).

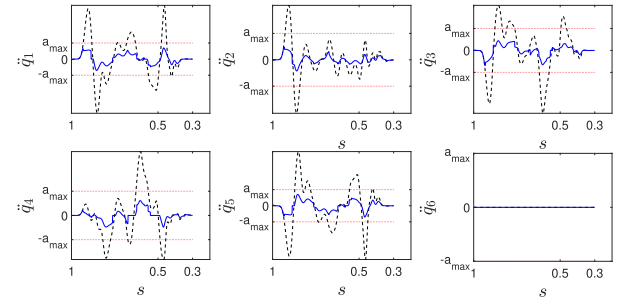


Fig. 8. Target joint accelerations with proposed temporal coupling (solid blue line) and without (dashed black line).

the Omega-shaped path in the xz -plane for the end-effector. Experiments were performed both with the proposed temporal coupling, parameterized according to Table I, and without. A schematic diagram of the experiment setup is shown in Fig. 6.

The DMP velocities and DMP accelerations are shown in Figs. 7 and 8, respectively. In Figs. 9 and 10 the resulting robot end-effector path is shown for the case without and with the proposed temporal coupling, respectively. As seen, the limits are exceeded at several time instances for the nominal trajectory resulting in an altered end-effector path compared to the desired shape. For the case when the proposed temporal coupling is used, no limits are violated leading to close following of the desired path. The same result in terms of path preservation was achieved when running the algorithm with a kinematic simulator of the UR10 assuming perfect acceleration tracking with saturation, i.e. $\ddot{q} = \text{sat}(u, a_{\max})$. When DMP without the proposed temporal coupling is used there is a discrepancy between the resulting distorted path in simulation and real robot. The coherency of simulation and real system when the proposed algorithm is used can be attributed to the consistency of the DMP imposed and the real robot constraints.

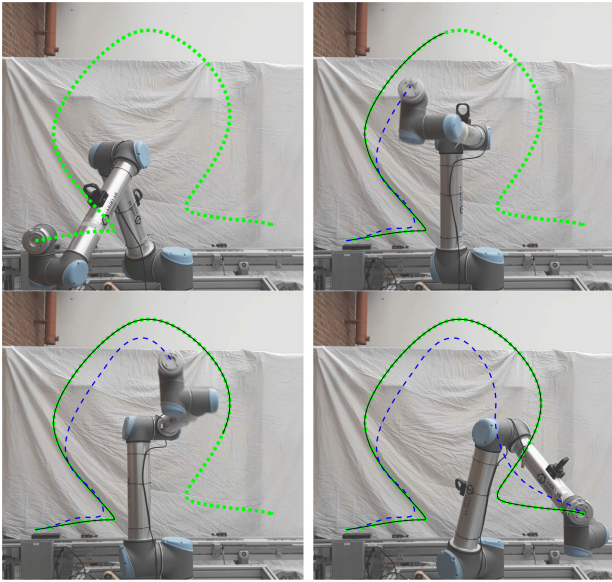


Fig. 9. End-effector path for nominal trajectory (green dotted line), traversed DMP trajectory (black solid line) and traversed actual robot position (blue dashed line) at four time instances (for DMP phase $s = 1, 0.6, 0.5$ and 0.3) when no temporal coupling is used.

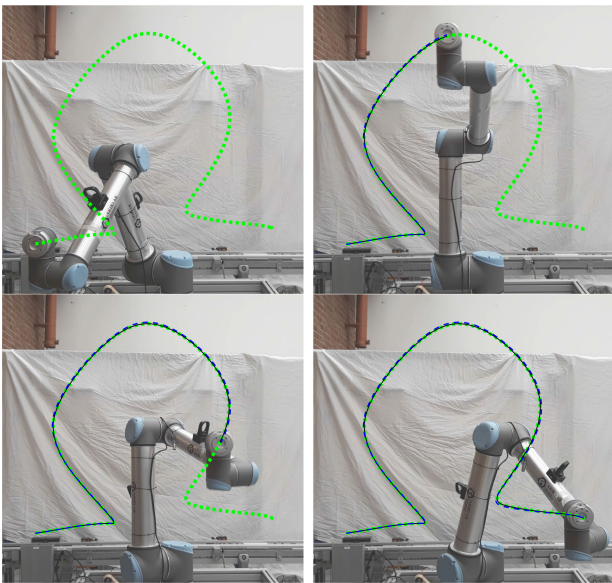


Fig. 10. End-effector path for nominal trajectory (green dotted line), traversed DMP trajectory (black solid line) and traversed actual robot position (blue dashed line) at four time instances (for DMP phase $s = 1, 0.6, 0.5$ and 0.3) when the proposed temporal coupling is used.

VI. CONCLUSION

In this letter an algorithm for temporal coupling was developed to handle velocity and acceleration constraints for DMP trajectories. The proposed method allows for, and directly adapts to, online modifications of the limits. Both simulations and experiments on a UR10 robot have been used to evaluate the performance. The resulting DMP trajectory is guaranteed to stay within the velocity limits, but for scenarios with aggressive accelerations and strict bounds momentary violations of the

acceleration limits may occur. A tuning parameter is used to determine how conservative the acceleration bounds should be treated to allow constraint satisfaction for more extreme scenarios.

The presented temporal coupling solely handles velocity and acceleration constraints on the DMP trajectory. In the future, we aim to extend the algorithm to also treat forward kinematic constraints enabling end-effector limitations when the DMP trajectory is defined in the robot joint space. Moreover, we intend to generalize the algorithm to deal with real-time trajectory scaling for any description of the target trajectory, not limited to DMPs.

REFERENCES

- [1] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2002, vol. 2, pp. 1398–1403.
- [2] 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, 2013.
- [3] M. Chi, Y. Yao, Y. Liu, and M. Zhong, "Learning, generalization, and obstacle avoidance with dynamic movement primitives and dynamic potential fields," *Appl. Sci.*, vol. 9, no. 8, 2019, Art. no. 1535.
- [4] J. Kober, E. Oztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," *Robot.: Sci. Syst. VI*, vol. 6, pp. 33–40, 2011.
- [5] F. Stulp, E. A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1360–1370, Dec. 2012.
- [6] Z. Li, T. Zhao, F. Chen, Y. Hu, C. Su, and T. Fukuda, "Reinforcement learning of manipulation and grasping using dynamical movement primitives for a humanoidlike mobile manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 121–131, Feb. 2018.
- [7] L. Koutras and Z. Doulgeri, "Dynamic movement primitives for moving goals with temporal scaling adaptation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 144–150.
- [8] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *Proc. Amer. Control Conf.*, 2014, vol. 6, pp. 226–241.
- [9] M. M. Nicotra and E. Garone, "Explicit reference governor for continuous time nonlinear systems subject to convex constraints," in *Proc. Amer. Control Conf.*, 2015, pp. 4561–4566.
- [10] R. Krug and D. Dimitrov, "Model predictive motion control based on generalized dynamical movement primitives," *J. Intell. Robot. Syst.*, vol. 77, no. 1, p. 17–35, 2015.
- [11] M. Karlsson, F. B. Carlson, A. Robertsson, and R. Johansson, "Two-degree-of-freedom control for trajectory tracking and perturbation recovery during execution of dynamical movement primitives," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1923–1930, 2017.
- [12] A. Dahlin and Y. Karayiannidis, "Adaptive trajectory generation under velocity constraints using dynamical movement primitives," *IEEE Contr. Syst. Lett.*, vol. 4, no. 2, pp. 438–443, Apr. 2020.
- [13] K. Kant and S. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, pp. 72–89, Sep. 1986.
- [14] O. Dahl and L. Nielsen, "Torque-limited path following by online trajectory time scaling," *IEEE Trans. Robot. Automat.*, vol. 6, no. 5, pp. 554–561, Oct. 1990.
- [15] C. Guarino Lo Bianco and F. M. Wahl, "A novel second order filter for the real-time trajectory scaling," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 5813–5818.
- [16] C. Guarino Lo Bianco and F. Ghilardelli, "Real-time planner in the operational space for the automatic handling of kinematic constraints," *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 3, pp. 730–739, Jul. 2014.
- [17] T. Kunz and M. Stilman, "Time-optimal trajectory generation for path following with bounded acceleration and velocity," *Robot.: Sci. Syst.*, Jul. 2012.
- [18] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 800–815, Oct. 2010.
- [19] F. Flacco, A. De Luca, and O. Khatib, "Motion control of redundant robots under joint constraints: Saturation in the null space," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 285–292.