B. Third Order DMP with Proposed Accelerating Goal

Inspired by Kober et al.'s moving goal (2), we explicitly model an *accelerating* target. We start with a third order DMP formulation [14], reformulated to be analogous to (1):

$$\dot{p}_3 = \tau\alpha_p\left[\beta_p\left(\gamma_p\left(g_m - p_1\right) + \frac{(\dot{g} - \dot{p}_1)}{\tau}\right) + \frac{-\ddot{p}_1}{\tau^2}\right] + \tau A f(z) \tag{4a}$$

$$\dot{p}_2 = \tau p_3 \tag{4b}$$

$$\dot{p}_1 = \tau p_2, \tag{4c}$$

with scaled acceleration, p_3, and repeated eigenvalues set by α_p, β_p, and γ_p. We name the above formulation *Vel. goal* DMP as it employs a constant velocity goal (2). If DMPs are placed in sequence, the acceleration of joining points is assumed to be zero. While arbitrarily large θ_i can overcome modeling errors if the final state of the demonstration has non-zero acceleration, this can lead to large acceleration jumps when transitioning between DMPs in sequence.

We propose the *Acc. goal* DMP driven towards a constant acceleration, $\ddot{g}$. The evolution of the goal-driven system to the accelerating target is adapted by replacing (4a) with

$$\dot{p}_3 = \tau\alpha_p\left[\beta_p\left(\gamma_p\left(g_m - p_1\right) + \frac{(\dot{g}_m - \dot{p}_1)}{\tau}\right) + \frac{(\ddot{g} - \ddot{p}_1)}{\tau^2}\right] + \tau A f(z), \tag{5}$$

where the moving target in (2) is replaced by

$$g_m(z) = g_m^0 + 1/2\ddot{g}\left(\frac{\ln(z)}{\tau\alpha_h}\right)^2 - \dot{g}_m^0\left(\frac{\ln(z)}{\tau\alpha_h}\right). \tag{6}$$

The initial velocity $\dot{g}_m^0$ and position g_m^0 and parameter α_h are set so that the target reaches the desired goal position g and velocity $\dot{g}$ at $t = T$ while moving with constant acceleration $\ddot{g}$. In Section IV, we join DMPs in sequence by learning θ_i from imitation [15] and setting the goal state to the final state of the demonstration segment where it joins the subsequent segment; our *Acc. goal DMP* can model accelerations at these joining points. However, the *Acc. goal DMP* can be used with any DMP learning method to describe trajectories that end with non-zero acceleration.

IV. Real-Time Trajectory Generation and Control Algorithms

During control, we generate DMP trajectories, $P = \{p, \dot{p}, \ddot{p}\}$, starting at the observed state, x. An MPC [21] minimizes the predicted tracking error of the state, x, to the online trajectory P. The MPC's prediction horizon, T_C, is severely limited by real-time computation; if the MPC tracks the *offline* demonstration, it may be unable to find a reasonable solution in the short horizon. The DMPs generate trajectories of a longer duration T_G without significant computation time, thus allowing the MPC to track the *online* trajectories that gradually return to the reference over the longer horizon. The first term in (5) returns the trajectory to the reference goal state, and the transformation function (3) ensures the trajectory resembles demonstration dynamics.

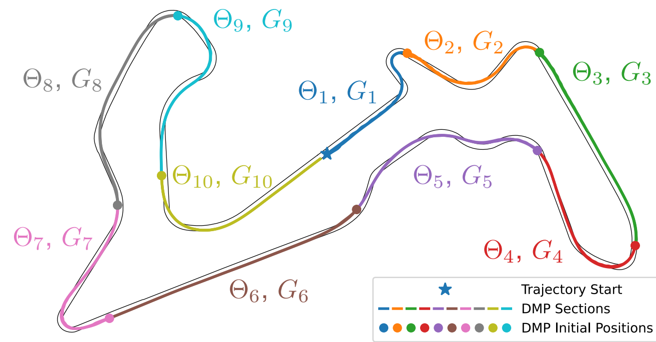

Fig. 1. DMP sections of duration T_S for a demonstration trajectory on Lago Maggiore racetrack; DMP weights, Θ_j, and goal states, G_j, for the jth segment are learned via imitation

Algorithm 1 Trajectory Generation from DMP Sequences

1: **Input**: observed state $P_i = \{p_i, \dot{p}_i, \ddot{p}_i\}$, demonstration $P_d = \{P_{d,0}, \ldots, P_{d,M}\}$, and time step Δt
2: **Output** $P_G = \{p_G, \dot{p}_G, \ddot{p}_G\}$ Generated trajectory
3: $P_G^{t=0} \leftarrow P_i$
4: $m = \text{argmin}_m \|p_{d,m} - p_i\|_2$
5: $j \leftarrow j(m)$, $G \leftarrow G_j$, $\Theta \leftarrow \Theta_j$
6: $t_{ref,0} \leftarrow t_m \in [0, T_S)$ ▷ Time of reference at m
7: **for** $t_i = 0, \ldots, T_G$ **do**
8: Calculate $\dot{P}_G^{t_i}(G, \Theta, z_i)$ at $z_i = z(t_{ref,0} + t_i)$ using DMP equations (i.e. Eqs. 1, 4, or 5)
9: Integrate trajectory $P_G^{t_{i+1}} = P_G^{t_i} + \dot{P}_G^{t_i}\Delta t$
10: **if** $t_i + t_m \geq T_G$ **then** ▷ Start next segment
11: $G \leftarrow G_{j(m)+1}$
12: $\Theta \leftarrow \Theta_{j[i]+1}$
13: $t_{ref,0} \leftarrow 0$
14: **end if**
15: **end for**

A. Trajectory Generation from Sequences of DMPs

We divide the demonstration into segments of equal duration T_S (Figure 1). Each segment is modeled offline by a single DMP. Namely, the goal state G_j of the j-th segment is the final state of the demonstrated segment $G = \{g, \dot{g}, \ddot{g}\} = \{p_{d,T_S}, \dot{p}_{d,T_S}, \ddot{p}_{d,T_S}\}$, and the weights $\Theta_j = ([\theta_0 \ldots \theta_i \ldots \theta_N]^T)_j$ are found via least weighted regression on the demonstrated segment [15]. During control, new trajectories, P_G, of duration T_G are generated starting at the last observed state using the offline-learned Θ_j and G_j of the nearest segment (Algorithm 1). Specifically, we find the index, m, of the closest demonstration waypoint to the observed position; this waypoint corresponds to the j-th segment and a reference time in the segment $t_m \in [0, T_S)$. The DMP equations, namely (1), (4), and (5), give the derivative of generated trajectory at the desired time steps, starting from reference time t_m (Line 8). If the generated trajectory is integrated to the end (in time) of the current segment (Line 10), it will switch to the DMP information of the following segment (G_{j+1}, Θ_{j+1}), and the reference time