

Combining Movement Primitives with Contraction Theory

Moses C. Nah,¹ Johannes Lachner,^{1,2} Neville Hogan,^{1,2} and Jean-Jacques Slotine^{1,2}

Abstract—This paper presents a modular framework for motion planning using movement primitives. Central to the approach is Contraction Theory, a modular stability tool for nonlinear dynamical systems. The approach extends prior methods by achieving parallel and sequential combinations of both discrete and rhythmic movements, while enabling independent modulation of each movement. This modular framework enables a divide-and-conquer strategy to simplify the programming of complex robot motion planning. Simulation examples illustrate the flexibility and versatility of the framework, highlighting its potential to address diverse challenges in robot motion planning.

I. INTRODUCTION

Motion planning using movement primitives has been a popular approach in robotics [1]–[14]. Often referred to as a “dynamical systems approach” [11], it encodes trajectories (or “movement primitives”) as dynamical systems with specific attractor dynamics. For instance, a dynamical system with a stable fixed point attractor generates discrete point-to-point movement [15], [16], while one with a stable limit cycle is used to generate rhythmic, repetitive actions [1], [17].

Using dynamical systems for motion planning offers several advantages. The stability arising from the attractor dynamics provides robustness against errors and external disturbances [7], [12]. Fast online reactive control can be achieved in dynamically changing environments [11], e.g., real-time obstacle avoidance [18], [19]. The approach also facilitates robot learning. Methods such as Learning from Demonstration (LfD) [11] and Imitation Learning [7], [12], [13] allow robots to learn a desired attractor dynamics from few demonstrations.*

These favorable properties of using dynamical systems enabled success in a variety of applications [18]–[24]. Nevertheless, a key limitation is the restricted range of movements that can be generated. Research has been dominated by generating a (sequence of) discrete point-to-point movement(s) [11]. A method to combine both discrete and rhythmic movement primitives has not been thoroughly addressed

¹Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA.

²Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA.

MCN was supported in part by a Mathworks Fellowship. JL was supported in part by the MIT-Novo Nordisk Artificial Intelligence Postdoctoral Fellows Program.

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

*Although both methods are based on the same dynamical systems approach, they exhibit clear technical differences. For further details, readers may refer to Section 3.5 of [12].

[12], [13], despite its practical value for robot control (e.g., polishing or sanding a surface [9] or peg-in-hole assembly [25]). A fundamental reason is because the stability proof of prior approaches is based on (strictly) stable Lyapunov functions [10], [15], which precludes limit cycle attractors [26]. From Strogatz [26], “*If a (strictly stable) Lyapunov function exists, then closed orbits are forbidden.*” In fact, this issue also exists for other Lyapunov-based motion planning methods [27], [28]. While ways to combine both discrete and rhythmic movements have been proposed [9], [29], [30], they do not allow independent modification of each movement, since the combined movement is learned using a single dynamical system. This limitation reduces flexibility in motion planning, necessitating movements to be learned from scratch instead of being constructed by combining existing ones. One might circumvent this issue by separately constructing dynamical systems for discrete and rhythmic movements, integrating their results separately, and then combining the outcomes. However, this approach renounces the advantages of using dynamical system, such as robustness against errors and disturbances, adaptability and reactiveness against dynamically changing environments.

In this paper, we extend Dynamic Movement Primitives [7], [12] by proposing a method to combine both discrete and rhythmic movement primitives. The key is to use Contraction Theory [31]–[34], a modular stability tool to analyze nonlinear dynamical systems. We show that both discrete and rhythmic movements can be combined while preserving independent modification of each movement. Stability is proven by Contraction Theory. Using the fact that the condition of contraction is preserved (or closed) under various system combinations, we show that a combination of discrete and rhythmic movements is simply a combination of contracting and transverse-contracting systems, respectively. With stability analysis provided by Contraction Theory, the advantages of using dynamical systems are preserved.

The contributions of this paper are listed below:

- A definition of movement primitives for both discrete and rhythmic movements.
- A method for parallel and sequential combinations of *both* discrete and rhythmic movements, while allowing spatial and temporal scaling of each movement.
- A stability proof of the presented approach using Contraction Theory.

Simulation examples are presented with possible extensions for actual robot control.[†]

[†]The codes are available in Github: <https://github.com/mosesnah-shared/modularDMP>

II. CONTRACTION THEORY

In this Section, an overview of Contraction Theory is provided. The key theorems are presented; their proofs are omitted. For further details, see [31], [32], [35]–[37].

Throughout this paper, an n -dimensional nonlinear dynamical system is considered:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) \quad (1)$$

In this equation, $t \in \mathbb{R}_{\geq 0}$ is time; $\mathbf{x}(t) \equiv \mathbf{x} \in \mathbb{R}^n$ is the state (trajectory) of the system; $\mathbf{f} : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ is a vector field. All maps and quantities are assumed to be smooth unless specified. To avoid clutter, the time argument is often omitted.

Contraction Theory, introduced by Lohmiller and Slotine [31], considers whether trajectories of a nonlinear dynamical system converge toward a single, nominal trajectory. A system that exhibits such convergent behavior is referred to as contracting. For contracting systems, initial conditions or temporary disturbances are “forgotten” (exponentially fast), and the asymptotic behavior of the dynamical system converges to a nominal behavior. In contrast to Lyapunov Stability Theory, a key aspect of Contraction Theory is that *a prior* knowledge about the nominal trajectory is not required. This simplifies stability proofs for nonlinear dynamical systems, where an analytical solution of the nominal trajectory may be difficult to derive [36], [37].

A. Definitions

In this Section, definitions of contracting and transverse contracting systems are provided.

1) *Contraction*: A nonlinear dynamical system is contracting in region $\mathcal{C} \subseteq \mathbb{R}^n$, if $\forall t \in \mathbb{R}_{\geq 0}$ and $\forall \mathbf{x} \in \mathcal{C}$, there exists both a symmetric positive definite matrix $\mathbf{M}(\mathbf{x}, t) \equiv \mathbf{M} \in \mathbb{R}^{n \times n}$ and a positive constant $\lambda \in \mathbb{R}_{>0}$ that satisfy:

$$\dot{\mathbf{M}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \mathbf{M} + \mathbf{M} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \preceq -2\lambda \mathbf{M} \quad (2)$$

where $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, t) \in \mathbb{R}^{n \times n}$ is a Jacobian matrix of \mathbf{f} . Region \mathcal{C} , matrix \mathbf{M} , and positive constant λ are referred to as the “contraction region,” “contraction metric,” and “contraction rate” of the system, respectively.

2) *Transverse Contraction*: For autonomous dynamical systems with stable limit cycles, the system is not contracting along the phase variable [34]. In this case one can analyze contraction along a direction orthogonal (with respect to a contraction metric) to the flow $\mathbf{f}(\mathbf{x})$.

Consider an autonomous dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. The system is said to be transverse contracting in region $\mathcal{C}_T \subseteq \mathbb{R}^n$, if $\forall \mathbf{x} \in \mathcal{C}_T$, there both exist a time-invariant symmetric positive definite matrix $\mathbf{M}_T(\mathbf{x}) \equiv \mathbf{M}_T \in \mathbb{R}^{n \times n}$ and a positive constant $\lambda_T \in \mathbb{R}_{>0}$ that satisfy:

$$\delta \mathbf{x}^\top \left\{ \dot{\mathbf{M}}_T + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \mathbf{M}_T + \mathbf{M}_T \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + 2\lambda_T \mathbf{M}_T \right\} \delta \mathbf{x} \leq 0 \quad (3)$$

for all $\delta \mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ satisfying the orthogonality condition $\delta \mathbf{x}^\top \mathbf{M}_T(\mathbf{x}) \mathbf{f}(\mathbf{x})$. Region \mathcal{C}_T , matrix \mathbf{M}_T , and positive constant λ_T are referred to as the “transverse contraction

region,” “transverse contraction metric,” and “transverse contraction rate” of the system, respectively.

B. Modular Properties of Contracting Systems

Contraction (or transverse contraction) are automatically preserved through various system combinations [31], [32], [35]. All results are global unless otherwise specified.

1) *Parallel Combination* [31]: Consider two dynamical systems with identical dimensions, that are contracting *under the same contraction metric* $\mathbf{M}(\mathbf{x})$:

$$\dot{\mathbf{x}}(t) = \mathbf{f}_1(\mathbf{x}, t), \quad \dot{\mathbf{x}}(t) = \mathbf{f}_2(\mathbf{x}, t)$$

Then $\forall t \in \mathbb{R}_{\geq 0}$, a parallel combination of these two contracting systems also contracts under the same contraction metric $\mathbf{M}(\mathbf{x})$:

$$\alpha_1(t), \alpha_2(t) \in \mathbb{R}_{>0} : \dot{\mathbf{x}} = \alpha_1(t)\mathbf{f}_1(\mathbf{x}, t) + \alpha_2(t)\mathbf{f}_2(\mathbf{x}, t) \quad (4)$$

The contraction rate of the combined dynamical system is $\alpha_1(t)\lambda_1 + \alpha_2(t)\lambda_2$, where λ_1 and λ_2 are the contraction rates of \mathbf{f}_1 and \mathbf{f}_2 , respectively.

(*Remark 1*): As stated by Slotine [32], one example is a parallel combination of movement primitives. In detail, consider a movement primitive input $\mathbf{p}_i(\mathbf{x}, t) \in \mathbb{R}^p$, where for each $i \in \{1, 2, \dots, N_p\}$, the following dynamical system contracts under the same contraction metric $\mathbf{M}(\mathbf{x})$:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t) + \mathbf{B}(\mathbf{x}, t)\mathbf{p}_i(\mathbf{x}, t) \quad (5)$$

In this equation, $\mathbf{B}(\mathbf{x}, t) \in \mathbb{R}^{n \times p}$ is the input weighting matrix. Then, a combination of these N_p primitives:

$$\begin{aligned} \forall i \in \{1, 2, \dots, N_p\} \quad \alpha_i(t) \geq 0 \quad \sum_{i=1}^{N_p} \alpha_i(t) = 1 \\ \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t) + \sum_{i=1}^{N_p} \mathbf{B}(\mathbf{x}, t) \left(\alpha_i(t) \mathbf{p}_i(\mathbf{x}, t) \right) \end{aligned} \quad (6)$$

also contracts under the same contraction metric $\mathbf{M}(\mathbf{x})$ with contraction rate $\sum_{i=1}^{N_p} \alpha_i(t)\lambda_i$, where λ_i is the contraction rate of the i^{th} primitive input.

Moreover, given a duration $T \in \mathbb{R}_{>0}$, if a T -periodic primitive input—potentially derived from the phase output of a transverse contracting system—is included, $\mathbf{x}(t)$ converges to a *unique* T -periodic solution [37].

2) *Hierarchical Combination* [31], [38]: Consider two dynamical systems, of possibly different dimensions and contraction metrics, and their hierarchical combination, i.e., the system dynamics $\dot{\mathbf{x}}_1(t) = \mathbf{f}_1(\mathbf{x}_1, t)$ and $\dot{\mathbf{x}}_2(t) = \mathbf{f}_2(\mathbf{x}_1, \mathbf{x}_2, t)$, leading to virtual dynamics of the form:

$$\begin{bmatrix} \dot{\mathbf{x}}_1(t) \\ \dot{\mathbf{x}}_2(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{11}(\mathbf{x}_1) & \mathbf{0} \\ \mathbf{F}_{12}(\mathbf{x}_1, \mathbf{x}_2) & \mathbf{F}_{22}(\mathbf{x}_1, \mathbf{x}_2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} \quad (7)$$

If \mathbf{f}_1 is contracting, \mathbf{f}_2 is contracting (for each fixed \mathbf{x}_1), and \mathbf{F}_{12} is bounded, then the overall system is contracting, i.e., $\delta \mathbf{x}_1(t), \delta \mathbf{x}_2(t) \rightarrow \mathbf{0}$. The contraction rate of the overall system is the slowest of the individual contraction rates.

(Remark 2): If either of the vector fields \mathbf{f}_1 or \mathbf{f}_2 is transverse contracting (possibly with different transverse contraction metrics), then the overall system is transverse contracting [39].

3) *Sequential Combination* [40]: Consider a dynamical system $\dot{\mathbf{x}}_1(t) = \mathbf{f}(\mathbf{x}_1, t)$ which is one-way coupled to another dynamical system with identical dimension via input \mathbf{u} :

$$\begin{aligned}\dot{\mathbf{x}}_1(t) &= \mathbf{f}(\mathbf{x}_1, t) \\ \dot{\mathbf{x}}_2(t) &= \mathbf{f}(\mathbf{x}_2, t) + \mathbf{u}(\mathbf{x}_1) - \mathbf{u}(\mathbf{x}_2)\end{aligned}\quad (8)$$

If vector field $\mathbf{f} - \mathbf{u}$ is contracting, then $\mathbf{x}_2(t) \rightarrow \mathbf{x}_1(t)$.

III. DYNAMIC MOVEMENT PRIMITIVES

This Section presents an overview of Dynamic Movement Primitives (DMP). For further details see [7], [12], [13].

DMP, introduced by Ijspeert and Schaal [41], [42], generates trajectories of arbitrary complexity by using dynamical systems with specific attractors. DMP consists of three components: a canonical system, a nonlinear forcing term and a transformation system. Each component plays a distinct role in trajectory generation, as will be detailed in this Section.

Several types of DMP exist, depending on the space in which the trajectory resides. For instance, to generate trajectories for spatial orientation (i.e., on the $\text{SO}(3)$ manifold), DMP which accounts for the manifold structure of $\text{SO}(3)$ should be used [8], [14]. While the approach we present can be generalized to these cases, in this paper, we focus on DMP which generate trajectories on \mathbb{R}^n .

A. Canonical System

A canonical system serves as an “internal clock” of movement primitives. It provides a notion of phase without an explicit time parameterization [7]. Different types of canonical systems are used for discrete and rhythmic movements, as discrete movement requires a temporal phase, unlike the periodic phase used in rhythmic movement.

1) *For Discrete Movement*: A canonical system for discrete movement, $s_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is defined by a first-order stable linear system:

$$\tau_d \dot{s}_d(t) = -\alpha_s s_d(t) \quad (9)$$

In this equation, $\alpha_s, \tau_d \in \mathbb{R}_{>0}$ determine the time constant; the analytical solution is given by $s_d(t) = \exp(-\frac{\alpha_s}{\tau_d}t)s_d(0)$; while any initial conditions $s_d(0)$ can be used, the value is typically chosen to be $s_d(0) = 1$ [12].

2) *For Rhythmic Movement*: A canonical system for rhythmic movement, $s_r : \mathbb{R}_{\geq 0} \rightarrow [0, 2\pi)$ is a periodic function, defined by the phase variable of the Andronov-Hopf oscillator—a nonlinear dynamical system with a stable limit cycle attractor:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \gamma^2 - (x_1^2 + x_2^2) & -1/\tau_r \\ 1/\tau_r & \gamma^2 - (x_1^2 + x_2^2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (10)$$

In this equation, $\gamma, \tau_r \in \mathbb{R}_{>0}$; $x_1(t), x_2(t) \in \mathbb{R}$ are the state-variables. For non-zero initial conditions, all trajectories converge to a stable limit cycle with radius γ rotating with

a constant angular velocity $1/\tau_r$ [43]. Hence, the system is transverse contracting in region $\mathcal{C}_T = \mathbb{R}^2 \setminus \{0\}$ [39].

The phase variable of the Andronov-Hopf oscillator is used as the canonical system for rhythmic movement, i.e., $s_r(t) = \arctan(x_2(t)/x_1(t))$. To generate a T -periodic function using s_r , τ_r is chosen to be $\tau_r = T/(2\pi)$. Any positive values can be used for γ ; a common choice is $\gamma = 1$ to yield a unit circle limit cycle [44].

(Remark 3): The Andronov-Hopf oscillator is transverse contracting. Its dynamics (Eq. (10)) in polar coordinates, $r \equiv \sqrt{x_1^2 + x_2^2}$, $\theta \equiv \arctan(x_1/x_2)$ are given by:

$$\begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r(\gamma^2 - r^2) \\ 1/\tau_r \end{bmatrix} \equiv \mathbf{f}_{r\theta}(r) \quad (11)$$

In this equation, $\mathbf{f}_{r\theta}$ is the vector field of the Andronov-Hopf oscillator in polar coordinates. Given an arbitrary positive constant $\epsilon \in \mathbb{R}_{>0}$, consider region $r \geq \epsilon$. In this region, define a transverse contraction metric $\mathbf{M}_T(r, \theta) \in \mathbb{R}^{2 \times 2}$ which is symmetric and positive definite:

$$\mathbf{M}_T(r, \theta) = \begin{bmatrix} 1/r^2 & -\tau_r(\gamma^2 - r^2)/r \\ -\tau_r(\gamma^2 - r^2)/r & m_{\theta\theta}(r, \theta) \end{bmatrix} \quad (12)$$

The quantity $m_{\theta\theta}(r, \theta) \in \mathbb{R}_{>0}$ is chosen so that $m_{\theta\theta}(r, \theta) > (\gamma^2 - r^2)\tau_r^2$ has a symmetric positive definite transverse contraction metric \mathbf{M}_T . Consequently, for $\delta \mathbf{x} = c[1, 0]^\top$ with $c \in \mathbb{R}$, (1) $\delta \mathbf{x}^\top \mathbf{M}_T \mathbf{f}_{r\theta} = 0$, and (2) the transverse contraction condition (Eq. (3)) is satisfied with transverse contraction rate $\lambda_T = 4c^2$. The transverse contraction region is $\mathcal{C}_T = \{(r, \theta) \in \mathbb{R}^2 \mid \epsilon \in \mathbb{R}_{>0}, r \geq \epsilon, \theta \in [0, 2\pi)\}$.

B. Nonlinear Forcing Term

A nonlinear forcing term, which takes the canonical system as a function argument, consists of a weighted sum of nonlinear basis functions. As with the canonical system, different types of nonlinear forcing terms are used to generate discrete and rhythmic movements.

1) *For Discrete Movement*: A nonlinear forcing term for discrete movement, $\mathbf{F}_d : \mathbb{R}_{>0} \rightarrow \mathbb{R}^n$, is defined by a weighted sum of Gaussian functions:[‡]

$$\begin{aligned}\mathbf{F}_d(s_d(t)) &= \mathbf{W}\boldsymbol{\sigma}(s_d(t)) \\ \boldsymbol{\sigma}(s_d(t)) &= \frac{s_d(t)}{\sum_{i=1}^N \sigma_i(s_d(t))} [\sigma_1(s_d(t)) \cdots \sigma_N(s_d(t))]^\top \\ \sigma_i(s_d(t)) &= \exp\{-h_i(s_d(t) - c_i)^2\}\end{aligned}\quad (13)$$

$\mathbf{W} \in \mathbb{R}^{n \times N}$ is a weight matrix; $c_i, h_i \in \mathbb{R}$ are the center location and inverse width of the i -th Gaussian basis function σ_i , respectively; N is the number of basis functions.

To define \mathbf{F}_d , parameters N, c_i, h_i are manually chosen [7]; a common choice for these parameters is $c_i = \exp(-\alpha_s(i-1)/(N-1))$ for $i \in \{1, 2, \dots, N\}$, $h_i = 1/(c_{i+1} - c_i)^2$ for $i \in \{1, 2, \dots, N-1\}$ and $h_{N-1} = h_N$ [12]. On the other hand, the weight matrix \mathbf{W} can be learned through various methods [7], [20], [21], [45]. Details for choosing \mathbf{W} are deferred to Section III-D.

[‡]While capital letters denote vectors, tuples, or arrays in this paper, an exception is made for the nonlinear forcing term to avoid confusion with the notation of nonlinear vector fields.

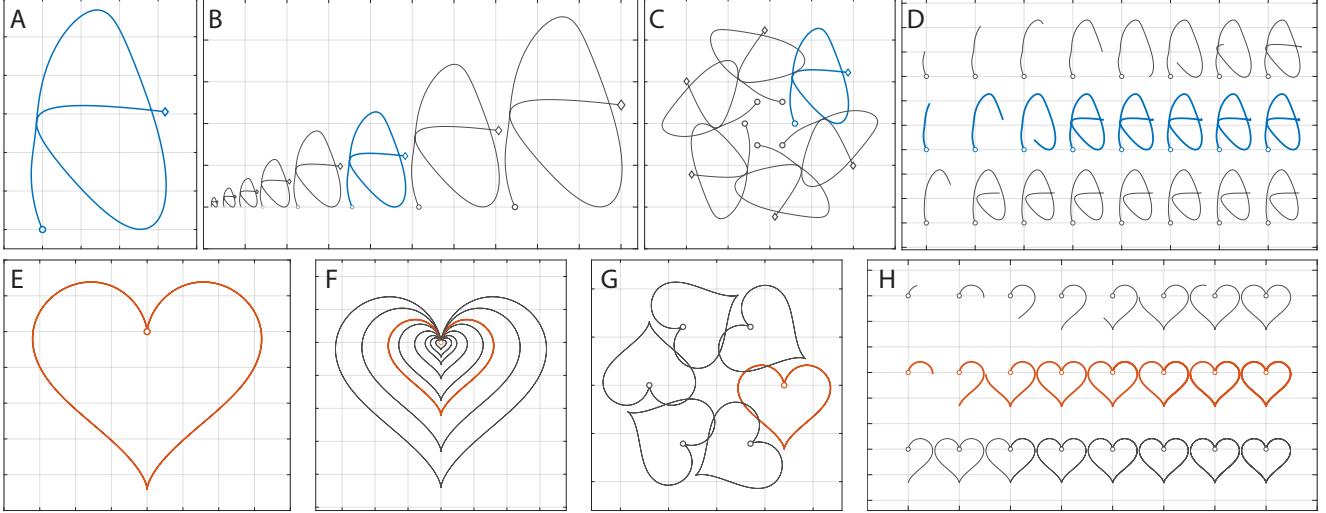


Fig. 1. Dynamic movement primitives (DMP) for (A-D) discrete and (E-F) rhythmic movements and their (B, F) scaling, (C, G) rotational, and (D, H) temporal invariance properties. Discrete and rhythmic movements are depicted with blue and orange colors, respectively. (A, E) Discrete and rhythmic DMPs for letter A and heart, respectively. DMP parameters used for Imitation Learning: $\alpha_z = 100$, $\beta_z = 25$, $\alpha_s = 1.0$, $N = 50$, $P = 1200$, $\tau_d^{(d)} = \tau_r^{(d)} = 1.0s$, $T = 2\pi$, $\mathbf{S} = \mathbb{I}_2$. (B, F) Spatial invariance property for discrete and rhythmic DMPs (Section III-E.1). Values used for spatial scaling: $\kappa_s = 0.1, 0.2, 0.3, 0.5, 0.8, 1.0, 1.5, 2.0$, ordered from left to right for discrete DMP (B), ordered from inside to outside for rhythmic DMP (F). (C, G) Rotational invariance property for discrete and rhythmic DMPs (Section III-E.2). Spatial rotation matrices used for rotation: $\mathbf{R} = \text{rotz}(0^\circ), \text{rotz}(60^\circ), \text{rotz}(120^\circ), \text{rotz}(180^\circ), \text{rotz}(240^\circ), \text{rotz}(300^\circ)$, ordered counterclockwise from the colored trajectories; rotz denotes the rotation along $+Z$, which is positive out of the page. (D, H) Temporal invariance property for discrete and rhythmic DMPs (Section III-E.1). Trajectories on top, middle, bottom rows depict trajectories generated by temporal scaling values of $\kappa_t = 0.5, 1.0, 2.0$, respectively. Ordered from left to right columns, trajectories at time $t = [0.25, 0.50, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0]\tau_d^{(d)}$ and $t = [0.25, 0.50, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0]T$ are depicted for figure (D) and (H), respectively. For all figures, circle markers depict the start position for discrete and rhythmic DMPs; diamond markers depict the goal position of discrete DMP. For figures (B-D) and (E-G), blue and orange trajectories depict the trajectories in (A) and (B), respectively.

2) *For Rhythmic Movement:* A nonlinear forcing term for rhythmic movement, $\mathbf{F}_r : [0, 2\pi) \rightarrow \mathbb{R}^n$, is defined by a weighted sum of von Mises functions:

$$\begin{aligned} \mathbf{F}_r(s_r(t)) &= \mathbf{W}\psi(s_r(t)) \\ \psi(s_r(t)) &= \frac{1}{\sum_{i=1}^N \psi_i(s_r(t))} [\psi_1(s_r(t)) \cdots \psi_N(s_r(t))]^\top \\ \psi_i(s_r(t)) &= \exp \{h_i(\cos(s_r(t) - c_i) - 1)\} \end{aligned} \quad (14)$$

As with the nonlinear forcing term for discrete movement, parameters N , c_i , h_i for \mathbf{F}_r are manually chosen [7]; a common choice is $c_i = 2\pi(i-1)/(N-1)$ and $h_i = 2.5N$ for $i \in \{1, 2, \dots, N\}$ [12], [46]. Details for choosing \mathbf{W} are deferred to Section III-D.

C. Transformation System

The nonlinear forcing term \mathbf{F}_d (respectively \mathbf{F}_r) with canonical system s_d (respectively s_r) as its function argument, is used as an input to the transformation system to generate discrete (respectively rhythmic) trajectories. The transformation system for \mathbb{R}^n consists of n second-order linear systems with a scaled nonlinear forcing term \mathbf{F} :

$$\begin{aligned} \tau \dot{\mathbf{y}}(t) &= \mathbf{z}(t) \\ \tau \dot{\mathbf{z}}(t) &= \alpha_z \{ \beta_z (\mathbf{g} - \mathbf{y}(t)) - \mathbf{z}(t) \} + \mathbf{S} \mathbf{F}(s(t)) \end{aligned} \quad (15)$$

In these equations, \mathbf{F} and its argument $s(t)$ denote (\mathbf{F}_d, s_d) or (\mathbf{F}_r, s_r) to generate discrete or rhythmic movements, respectively; $\mathbf{y}(t), \mathbf{z}(t) \in \mathbb{R}^n$ are the position and time-scaled velocity; $\alpha_z, \beta_z \in \mathbb{R}_{>0}$ are positive coefficients of

the transformation system; $\tau \in \mathbb{R}_{>0}$ is the value used for the canonical system (τ_d for discrete movement, τ_r for rhythmic movement) (Section III-A); $\mathbf{g} \in \mathbb{R}^n$ is a constant array to which $\mathbf{y}(t)$ converges when $\mathbf{F} \equiv \mathbf{0}$; $\mathbf{S} \in \mathbb{R}^{n \times n}$ is the scaling matrix for the nonlinear forcing term.

While any positive values of α_z, β_z can be used, usually the values of α_z, β_z are chosen such that if $\mathbf{F} = \mathbf{0}$, the transformation system is critically damped (i.e., has repeated eigenvalues) for $\tau = 1$ (i.e., $\beta_z = \alpha_z/4$) [7], [13].

To generate trajectories $\mathbf{y}(t), \mathbf{z}(t)$, the initial conditions $\mathbf{y}(t=0), \mathbf{z}(t=0)$ should be chosen and forward integrated. For this, appropriate choices of $\mathbf{S}, \mathbf{g}, \tau$ must be made. Details of choosing these parameters are discussed in Section III-E.

D. Imitation Learning

Without the nonlinear forcing term, the trajectory for the transformation system $\mathbf{y}(t)$ follows the response of a stable second-order linear system which (exponentially) converges to \mathbf{g} . To generate a wider range of movements for $\mathbf{y}(t)$ (e.g., rhythmic movements), the weight matrix \mathbf{W} term can be learned through various methods [21], [47].

One prominent approach is Imitation Learning [7], [12], [48], where the best-fit weights are learned from a demonstrated trajectory $\mathbf{y}^{(d)}(t)$. To conduct Imitation Learning, the position, velocity and acceleration of the demonstrated trajectory must be collected [7], [12], [13]. Let the P data points of these values be denoted by $\mathbf{y}^{(d)}(t_i), \dot{\mathbf{y}}^{(d)}(t_i), \ddot{\mathbf{y}}^{(d)}(t_i) \in \mathbb{R}^n$ for $i \in \{1, 2, \dots, P\}$; for the position data, offset is added such that $\mathbf{y}^{(d)}(t_1) = \mathbf{0}$, i.e., the data points start at the origin.

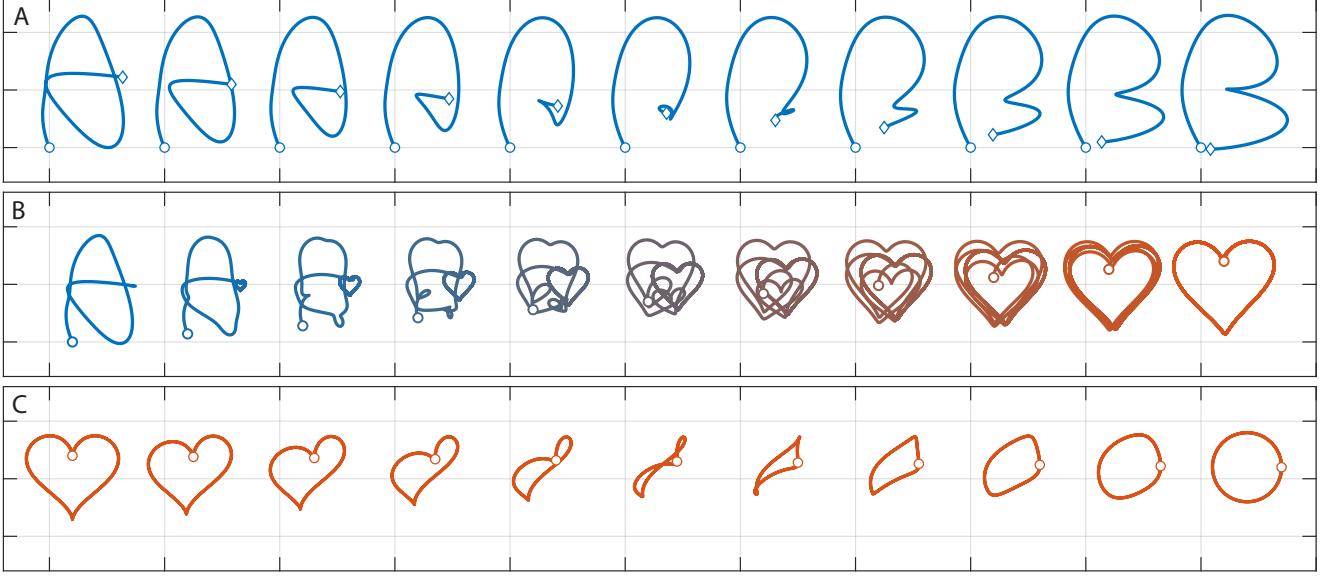


Fig. 2. Parallel combination of both discrete and rhythmic DMPs. (A) Parallel combination of discrete DMPs. For the combination, primitive inputs to generate the letters A and B were used. (B) Parallel combination of discrete and rhythmic DMPs. For the combination, primitive inputs to generate letter A (discrete) and heart (rhythmic) were used. (C) Parallel combination of rhythmic DMPs. For the combination, primitive inputs to generate heart and circle were used. For the 11 trajectories of each figure, the parameters used for parallel combination were $\alpha_1 = [0.0, 0.1, \dots, 1.0]$ and $\alpha_2 = 1 - \alpha_1$, ordered from left to right. DMP parameters used for Imitation Learning were identical to those in Figure 1. For all DMPs, the parameters for spatial scaling and rotation were $\kappa_s = 1.0$, $\mathbf{R} = \mathbb{I}_2$. For temporal scaling, $\kappa_t = 1.0$ was used for the trajectories in figure (A) and (C); for figure (B), $\kappa_t = 0.25, 6.00$ were used for letter A (discrete) and heart (rhythmic), respectively. For all figures, circle markers depict the start position for discrete and rhythmic DMPs; diamond markers depict the goal position of discrete DMP.

With these P data points, the weight matrix \mathbf{W} is calculated by linear least-square regression [12]:

$$\mathbf{W} = \mathbf{B}\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1} \quad (16)$$

For matrices $\mathbf{A} \in \mathbb{R}^{N \times P}$ and $\mathbf{B} \in \mathbb{R}^{n \times P}$, different values are used for discrete and rhythmic movements.

1) *For Discrete Movement:* For discrete movements, matrices \mathbf{A} and \mathbf{B} are defined by:

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}_d(t_1) \quad \mathbf{a}_d(t_2) \quad \dots \quad \mathbf{a}_d(t_P)] \\ \mathbf{B} &= [\mathbf{b}_d(t_1) \quad \mathbf{b}_d(t_2) \quad \dots \quad \mathbf{b}_d(t_P)] \\ \mathbf{a}_d(t) &= s_d^{(d)}(t)\boldsymbol{\sigma}(s_d^{(d)}(t)) \quad s_d^{(d)}(t) = \exp\left(-\frac{\alpha_s}{\tau_d^{(d)}}t\right) \\ \mathbf{b}_d(t) &= (\tau_d^{(d)})^2 \ddot{\mathbf{y}}^{(d)}(t) + \alpha_z \tau_d^{(d)} \dot{\mathbf{y}}^{(d)}(t) \\ &\quad + \alpha_z \beta_z (\mathbf{y}^{(d)}(t) - \mathbf{g}_d^{(d)}) \end{aligned} \quad (17)$$

In these equations, $\tau_d^{(d)} \in \mathbb{R}_{>0}$ and $\mathbf{g}_d^{(d)} \in \mathbb{R}^n$ are the duration and goal location of the demonstrated trajectory, respectively, i.e., $\tau_d^{(d)} = t_P - t_1$ and $\mathbf{g}_d^{(d)} = \mathbf{y}^{(d)}(t_P)$.

2) *For Rhythmic Movement:* For rhythmic movements, matrices \mathbf{A} and \mathbf{B} are defined by:

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}_r(t_1) \quad \mathbf{a}_r(t_2) \quad \dots \quad \mathbf{a}_r(t_P)] \\ \mathbf{B} &= [\mathbf{b}_r(t_1) \quad \mathbf{b}_r(t_2) \quad \dots \quad \mathbf{b}_r(t_P)] \\ \mathbf{a}_r(t) &= \boldsymbol{\psi}(s_r^{(d)}(t)) \quad s_r^{(d)}(t) = \frac{t}{\tau_r^{(d)}} \mod 2\pi \\ \mathbf{b}_r(t) &= (\tau_r^{(d)})^2 \ddot{\mathbf{y}}^{(d)}(t) + \alpha_z \tau_r^{(d)} \dot{\mathbf{y}}^{(d)}(t) \\ &\quad + \alpha_z \beta_z (\mathbf{y}^{(d)}(t) - \mathbf{g}_r^{(d)}) \end{aligned} \quad (18)$$

In these equations, $\tau_r^{(d)} \in \mathbb{R}_{>0}$ is $\tau_r^{(d)} = \frac{T^{(d)}}{2\pi}$ where $T^{(d)} = t_P - t_1$ is the period of the demonstrated trajectory. We assume the P data points $\mathbf{y}^{(d)}(t_1) = \mathbf{0}, \mathbf{y}^{(d)}(t_2), \dots, \mathbf{y}^{(d)}(t_P)$ comprise a closed-loop trajectory with an average position $\mathbf{g}_r^{(d)} = (\sum_{i=1}^P \mathbf{y}^{(d)}(t_i))/P$.

E. Generating the Trajectory

Once the weight matrix \mathbf{W} of the nonlinear forcing term is learned via Imitation Learning, the transformation system is integrated using the values of \mathbf{S} , \mathbf{g} , and τ to determine the shape of $\mathbf{y}(t)$. For both discrete and rhythmic DMPs, if one uses $\mathbf{S} = \mathbb{I}_n$, $\mathbf{g} = \mathbf{g}^{(d)}$ (i.e., $\mathbf{g}_d^{(d)}$ for discrete, $\mathbf{g}_r^{(d)}$ for rhythmic), $\tau = \tau^{(d)}$ (i.e., $\tau_d^{(d)}$ for discrete, $\tau_r^{(d)}$ for rhythmic), and initial conditions $\mathbf{y}(t=0) = \mathbf{0}$, $\mathbf{z}(t=0) = \mathbf{0}$ then $\mathbf{y}(t) \rightarrow \mathbf{y}^{(d)}(t)$, i.e., the transformation system generates a trajectory provided by demonstration (Figure 1A, 1E).

1) *Spatial and Temporal Invariance Properties:* To spatially and temporally scale the learned trajectory $\mathbf{y}^{(d)}(t)$, the following parameters are chosen: $\mathbf{S} = \kappa_s \mathbb{I}_n$, $\mathbf{y}(0) = \mathbf{y}_0$, $\mathbf{z}(0) = \mathbf{0}$, $\mathbf{g} = \mathbf{y}_0 + \kappa_s \mathbf{g}^{(d)}$, $\tau = \kappa_t \tau^{(d)}$ where $\kappa_s, \kappa_t \in \mathbb{R}_{>0}$ are the spatial and temporal scaling factors, respectively. With these parameters, $\mathbf{y}(t) \rightarrow \mathbf{y}_0 + \kappa_s \mathbf{y}^{(d)}(\kappa_t t)$ (Figure 1B, 1D, 1F, 1H).

2) *Rotational Invariance Property:* For $n = 3$, one can also spatially rotate the learned trajectory. Given an arbitrary spatial rotation matrix $\mathbf{R} \in \text{SO}(3)$, the following parameters are chosen: $\mathbf{S} = \kappa_s \mathbf{R}$, $\mathbf{y}(0) = \mathbf{y}_0$, $\mathbf{z}(0) = \mathbf{0}$, $\mathbf{g} = \mathbf{y}_0 + \kappa_s \mathbf{R} \mathbf{g}^{(d)}$, $\tau = \kappa_t \tau^{(d)}$. With these parameters, $\mathbf{y}(t) \rightarrow \mathbf{y}_0 + \kappa_s \mathbf{R} \mathbf{y}^{(d)}(\kappa_t t)$ (Figure 1C, 1G).

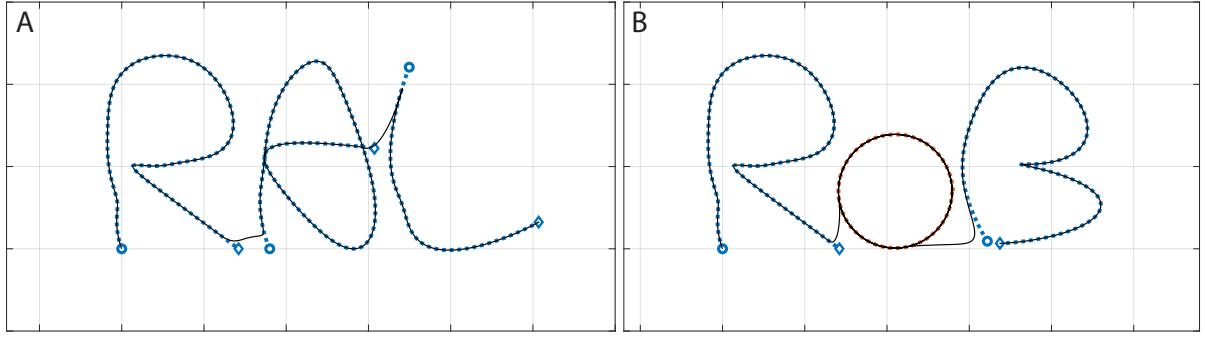


Fig. 3. Sequential combination of both discrete and rhythmic DMPs. (A) Sequential combination of three discrete DMPs, which generate Alphabets “R”, “A” and “L”. (B) Sequential combination of two discrete DMPs, which generate Alphabets “R” and “B”, and one rhythmic DMP which generates a circular trajectory. DMP parameters used for Imitation Learning were identical to those in Figure 1. Black filled trajectories depict the result of the sequenced DMP; blue and orange dotted trajectories depict the trajectory of each individual discrete and rhythmic DMPs, respectively; circle and diamond markers depict the start and goal positions of discrete DMP, respectively. Time offsets for each movements (Eq. (23)): (A) $t_{off,1} = 0.0s$, $t_{off,2} = 0.969s$, $t_{off,3} = 1.939s$; (B) $t_{off,1} = 0.0s$, $t_{off,2} = 0.970s$, $t_{off,3} = 4.499s$. The corresponding parameters for the activation function (Eq. (24)) were selected to achieve a smooth transition to the subsequent trajectory.

IV. CONTRIBUTION: COMBINING MOVEMENT PRIMITIVES WITH CONTRACTION THEORY

In this Section, we show how both discrete and rhythmic DMPs can be combined to generate parallel and sequential combinations of movement primitives.

A. Definition of Movement Primitive Input

Consider a nonlinear forcing term $\mathbf{F}(s(t))$ (either for discrete or rhythmic DMP), where the weights \mathbf{W} are learned via Imitation Learning (Section III-D) with parameters $\alpha_z, \beta_z, \tau^{(d)}, \mathbf{g}^{(d)}$ (and additionally α_s for discrete DMP).

A movement primitive input, $\mathbf{p}(\mathbf{x}(t)) \in \mathbb{R}^n$ is defined by:

$$\mathbf{p}(\mathbf{x}(t)) = \mathbf{F}(s^{(d)}(t)) + \alpha_z \beta_z \mathbf{g}^{(d)} \quad (19)$$

where:

$$\mathbf{F}(s^{(d)}(t)) = \begin{cases} \mathbf{W}\sigma(s_d^{(d)}(t)) & \text{for Discrete DMP} \\ \mathbf{W}\psi(s_r^{(d)}(t)) & \text{for Rhythmic DMP} \end{cases} \quad (20)$$

With this definition for discrete and rhythmic movements, $\mathbf{p}(\mathbf{x})$ is used as an input to a transformation system with $\mathbf{g} = \mathbf{0}$ (Eq. (15)):

$$(\tau^{(d)})^2 \ddot{\mathbf{y}}(t) + \tau^{(d)} \alpha_z \dot{\mathbf{y}}(t) + \alpha_z \beta_z \mathbf{y}(t) = \mathbf{p}(\mathbf{x}) \quad (21)$$

Given initial conditions $\mathbf{y}(t = 0) = \mathbf{0}$, $\mathbf{z}(t = 0) = \mathbf{0}$, $\mathbf{y}(t) \rightarrow \mathbf{y}^{(d)}(t)$ (Section III-D) (Figure 1A, 1E). Primitive $\mathbf{p}(\mathbf{x})$ generates $\mathbf{y}^{(d)}(t)$ as provided by demonstration.

(Remark 4): The stability of this approach is proven using Contraction Theory [33], [34]. A transformation system itself is a (globally) contracting system. Movement primitives $\mathbf{p}(\mathbf{x})$ for discrete and rhythmic movements are contracting and transverse contracting systems, respectively. For discrete (respectively rhythmic) DMP, the transformation system with input $\mathbf{p}(\mathbf{x})$ is a hierarchical combination of contracting systems (respectively contracting and transverse contracting systems), which thereby results in a contracting (respectively transverse contracting) system.

(Remark 5): As discussed in Section III-E, spatial and temporal scaling and rotation can be achieved by scaling the primitive inputs. In detail, using $\kappa_s \mathbf{R} \mathbf{p}(\mathbf{x}(\kappa_t t)) \equiv \kappa_s \mathbf{R} \{ \mathbf{F}(s^{(d)}(\kappa_t t)) + \alpha_z \beta_z \mathbf{g}^{(d)} \}$, as an input to Eq. (21), given initial conditions $\mathbf{y}(t = 0) = \mathbf{y}_0$, $\mathbf{z}(t = 0) = \mathbf{0}$, $\mathbf{y}(t) \rightarrow \mathbf{y}_0 + \kappa_s \mathbf{R} \mathbf{y}^{(d)}(\kappa_t t)$ (Figure 1B-D, 1F-H). Moreover, an offset $\mathbf{y}_{off} \in \mathbb{R}^n$ can be added to the trajectories by adding $\alpha_z \beta_z \mathbf{y}_{off}$ to the movement primitive input. Then, for initial conditions $\mathbf{y}(t = 0) = \mathbf{y}_0 + \mathbf{y}_{off}$, $\mathbf{z}(t = 0) = \mathbf{0}$, $\mathbf{y}(t) \rightarrow (\mathbf{y}_0 + \mathbf{y}_{off}) + \kappa_s \mathbf{R} \mathbf{y}^{(d)}(\kappa_t t)$.

B. Combining Movement Primitives

Parallel and sequential combinations of both discrete and rhythmic DMPs can be achieved.

1) *Application 1: Parallel Combination:* Consider two primitive inputs, $\mathbf{p}_1(\mathbf{x}_1(t))$ and $\mathbf{p}_2(\mathbf{x}_2(t))$, which can represent either discrete or rhythmic DMPs and may be spatially, temporally scaled, or rotated. A parallel combination of these two movement primitives can be achieved by a parallel combination of the two primitive inputs (Section II-B.1):

$$\alpha_1, \alpha_2 \geq 0 : \mathbf{p}(\mathbf{x}) = \alpha_1 \mathbf{p}_1(\mathbf{x}_1(t)) + \alpha_2 \mathbf{p}_2(\mathbf{x}_2(t)) \quad (22)$$

And recursively, any number of primitives can be parallel combined.

Figure 2 illustrates examples of parallel combination of discrete DMPs (Figure 2A), discrete and rhythmic DMPs (Figure 2B), and rhythmic DMPs (Figure 2C).

(Remark 6): One promising robotic application of the presented method is motion planning for tasks which involve combinations of both discrete and rhythmic movements, e.g., polishing, grinding, peg-in-hole assembly tasks [25], [49]. As each movement can be spatially and temporally scaled or rotated, flexible and versatile motion planning can be achieved. For instance, in a polishing task, temporal scaling can adjust the speed of a rhythmic polishing motion, slowing it down for thorough polishing or speeding it up for rough surface treatment. Similarly, spatial scaling and rotation can modify the range of movement to adapt to varying surface

sizes and orientations. Note that spatial or temporal scaling can be achieved by changing a single scalar variable, thereby simplifying robot motion programming.

2) *Application 2: Sequential Combination:* Consider two primitive inputs, $\mathbf{p}_1(\mathbf{x}_1(t))$ and $\mathbf{p}_2(\mathbf{x}_2(t))$, which can represent either discrete or rhythmic DMPs and may be spatially, temporally scaled, or rotated with a position offset. A sequential combination of these two movement primitives can be achieved by using time-varying weights, $\alpha_1(t)$, $\alpha_2(t)$ with time offsets of each primitive input (Section II-B.3):

$$\begin{aligned} \alpha_1(t), \alpha_2(t) &\geq 0 : \mathbf{p}(\mathbf{x}) = \\ \alpha_1(t)\mathbf{p}_1(\mathbf{x}_1(t - t_{1,off})) + \alpha_2(t)\mathbf{p}_2(\mathbf{x}_2(t - t_{2,off})) \end{aligned} \quad (23)$$

In this equation, $t_{1,off}, t_{2,off} \in \mathbb{R}_{>0}$ are the time offsets. Recursively, any number of primitives can be sequentially combined.

Figure 3 illustrates examples of sequential combination of both discrete and rhythmic DMPs. The time-varying weighting function used for this example was:

$$\alpha_i(t) = \begin{cases} 0 & 0 \leq t < t_1 \\ 3\left(\frac{t-t_1}{t_2-t_1}\right)^2 - 2\left(\frac{t-t_1}{t_2-t_1}\right)^3 & t_1 \leq t < t_2 \\ 1 & t_2 \leq t < t_3 \\ 1 - 3\left(\frac{t-t_3}{t_4-t_3}\right)^2 + 2\left(\frac{t-t_3}{t_4-t_3}\right)^3 & t_3 \leq t < t_4 \\ 0 & t_4 \leq t \end{cases} \quad (24)$$

where $i \in \{1, 2, \dots, N_p\}$ and $t_1, t_2, t_3, t_4 \in \mathbb{R}_{>0}$.

Compared to prior approaches [10], [11], [15], [27], [28] which account for sequential combinations of discrete movements, the presented approach achieves a sequential combination of both discrete and rhythmic movements while preserving the spatial, temporal, and rotational invariance properties of DMP (Section III-E).

(Remark 7): This approach can be used for any robotic application involving a sequence of trajectories, e.g., planning of robot end-effector motion [27], [50], or unmanned aerial vehicle trajectories (UAV) [28]. The approach expands prior methods by incorporating rhythmic movements.

3) *Application 3: Parallel and Sequential Combinations:* Parallel and sequential combinations of discrete and rhythmic DMPs can be achieved. Figure 4 illustrates an example.

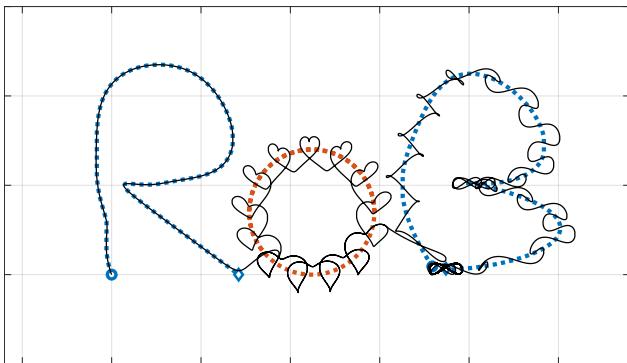


Fig. 4. Parallel and sequential combinations of both discrete and rhythmic DMPs. On the sequenced trajectory used in Figure 3B, additional rhythmic DMPs were parallel combined. For the circular trajectory and letter B, rhythmic DMPs for heart (Figure 2B) and infinity symbol were parallel combined, respectively.

(Remark 8): The presented approach enables both parallel and sequential combinations of discrete and rhythmic movements, with independent modulation of each movement. This provides a divide-and-conquer strategy for robot programming—programmers can break down complex robot applications into simpler, manageable sub-tasks, each associated with a (combination of) movement primitive(s), which can later be combined. The ability to independently modulate each movement further simplifies programming for motion planning, allowing robot programmers to refine individual movements without the need to relearn the whole movement from scratch. This modular framework not only improves flexibility but also accelerates development by allowing the reuse and refinement of existing components, making the approach well-suited for diverse and complex motion planning challenges in robotics.

V. DISCUSSION, FUTURE WORK AND CONCLUSION

This paper introduced a modular framework to combine discrete and rhythmic DMPs. The method preserves the spatial, temporal, and rotational invariance properties of DMP. Methods for parallel and/or sequential combinations of both discrete and rhythmic DMPs were presented and demonstrated through simulation examples. Stability analysis was provided using Contraction Theory, which enabled incorporation of rhythmic movements with stability guarantees. As a result, robust and adaptable motion planning is achieved, allowing complex movements to be constructed by combining simpler movement primitives.

Note that the combination of movement primitive inputs is achieved under the same contraction metric (Section II-B.1). The weights of each movement primitive must be learned under identical DMP parameters. While this constraint may appear to limit the method's flexibility, it facilitates additional applications, e.g., movement identification [7].

As the approach achieves modular motion planning based on linear superposition of movement primitive inputs, the system can be reactive (or adaptive) in changing environments. For instance, obstacle avoidance methods using an additional coupling term [18] can be seamlessly incorporated. Temporal coupling terms can be added to the canonical system to adapt to moving goals for discrete DMP [7], [51], or to switch between different rhythmic patterns for rhythmic DMP (e.g., changing gait patterns in different terrain).

Future work could explore actual robot implementation or an extension to different manifolds, e.g., the SO(3) manifold for spatial orientation.

REFERENCES

- [1] M. M. Williamson, “Robot arm control exploiting natural dynamics,” Ph.D. dissertation, Massachusetts Institute of Technology, 1999.
- [2] A. Ijspeert, J. Nakanishi, and S. Schaal, “Learning attractor landscapes for learning motor primitives,” *Advances in neural information processing systems*, vol. 15, 2002.
- [3] S. Schaal, P. Mohajerian, and A. Ijspeert, “Dynamics systems vs. optimal control—a unifying view,” *Progress in brain research*, vol. 165, pp. 425–445, 2007.

- [4] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 819–824.
- [5] 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.
- [6] N. Hogan and D. Sternad, "Dynamic primitives of motor behavior," *Biological Cybernetics*, vol. 106, no. 11-12, pp. 727–739, 2012.
- [7] 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.
- [8] 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*. PMLR, 2020, pp. 293–302.
- [9] F. Khadivar, I. Lauzana, and A. Billard, "Learning dynamical systems with bifurcations," *Robotics and Autonomous Systems*, vol. 136, p. 103700, 2021.
- [10] N. Figueroa and A. Billard, "Locally active globally stable dynamical systems: Theory, learning, and experiments," *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 312–347, 2022.
- [11] A. Billard, S. Mirazavi, and N. Figueroa, *Learning for adaptive and reactive robot control: a dynamical systems approach*. Mit Press, 2022.
- [12] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023.
- [13] M. C. Nah, J. Lachner, and N. Hogan, "Robot control based on motor primitives: A comparison of two approaches," *The International Journal of Robotics Research*, p. 02783649241258782, 2024.
- [14] F. J. Abu-Dakka, M. Saveriano, and V. Kyriki, "A unified formulation of geometry-aware discrete dynamic movement primitives," *Neurocomputing*, p. 128056, 2024.
- [15] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [16] L. Koutras and Z. Doulgeri, "A novel dmp formulation for global and frame independent spatial scaling in the task space," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 727–732.
- [17] F. Abu-Dakka, M. Saveriano, and L. Peternel, "Learning periodic skills for robotic manipulation: Insights on orientation and impedance," *Robotics and Autonomous Systems*, p. 104763, 2024.
- [18] 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.
- [19] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, pp. 433–454, 2012.
- [20] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7-9, pp. 1180–1190, 2008.
- [21] J. Kober and J. Peters, "Learning motor primitives for robotics," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2112–2118.
- [22] Y. Zhou and T. Asfour, "Task-oriented generalization of dynamic movement primitive," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3202–3209.
- [23] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.
- [24] L. Huber, J.-J. Slotine, and A. Billard, "Fast obstacle avoidance based on real-time sensing," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1375–1382, 2022.
- [25] J. Lachner, F. Tessari, A. M. West Jr, M. C. Nah, and N. Hogan, "Divide et impera: Learning impedance families for peg-in-hole assembly," *arXiv preprint arXiv:2410.01054*, 2024.
- [26] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018.
- [27] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.
- [28] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.
- [29] S. Degallier, C. P. Santos, L. Righetti, and A. Ijspeert, "Movement generation using dynamical systems: a humanoid robot performing a drumming task," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 512–517.
- [30] J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal, "Encoding of periodic and their transient motions by a single dynamic movement primitive," in *2012 12th IEEE-RAS international conference on humanoid robots (humanoids 2012)*. IEEE, 2012, pp. 57–64.
- [31] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.
- [32] J.-J. E. Slotine, "Modular stability tools for distributed computation and control," *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 6, pp. 397–416, 2003.
- [33] B. E. Perk and J.-J. E. Slotine, "Motion primitives for robotic flight control," *arXiv preprint cs/0609140*, 2006.
- [34] P. M. Wensing and J.-J. Slotine, "Sparse control for dynamic movement primitives," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10114–10121, 2017.
- [35] J.-J. Slotine and W. Lohmiller, "Modularity, evolution, and the binding problem: a view from stability theory," *Neural networks*, vol. 14, no. 2, pp. 137–145, 2001.
- [36] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, "Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview," *Annual Reviews in Control*, vol. 52, pp. 135–169, 2021.
- [37] F. Bullo, *Contraction theory for dynamical systems*. Francesco Bullo, 2022.
- [38] J. W. Simpson-Porco and F. Bullo, "Contraction theory on riemannian manifolds," *Systems & Control Letters*, vol. 65, pp. 74–80, 2014.
- [39] I. R. Manchester and J.-J. E. Slotine, "Transverse contraction criteria for existence, stability, and robustness of a limit cycle," *Systems & Control Letters*, vol. 63, pp. 32–38, 2014.
- [40] W. Wang and J.-J. E. Slotine, "On partial contraction analysis for coupled nonlinear oscillators," *Biological cybernetics*, vol. 92, no. 1, pp. 38–53, 2005.
- [41] 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.
- [42] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Control, planning, learning, and imitation with dynamic movement primitives," in *Workshop on Bilateral Paradigms on Humans and Humanoids, IEEE International Conference on Intelligent Robots and Systems*, 2003, pp. 1–21.
- [43] K. Ren, *Synthesis of Linear Distributed Control and Coupled Oscillators with Multiple Limit Cycles*. University of California, Los Angeles, 2021.
- [44] Q.-C. Pham, N. Tabareau, and J.-J. Slotine, "A contraction theory approach to stochastic incremental stability," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 816–820, 2009.
- [45] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.
- [46] L. Peternel, T. Noda, T. Petrić, A. Ude, J. Morimoto, and J. Babić, "Adaptive control of exoskeleton robots for periodic assistive behaviours based on emg feedback minimisation," *PloS one*, vol. 11, no. 2, p. e0148942, 2016.
- [47] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [48] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.
- [49] M. C. Nah, J. Lachner, F. Tessari, and N. Hogan, "On the modularity of elementary dynamic actions," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024.
- [50] M. Saveriano, F. Franzel, and D. Lee, "Merging position and orientation motion primitives," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7041–7047.
- [51] 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.