Margin requirements for first page
Paper size this page US Letter
72 pt
1 in
25.4 mm

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

From Movement Primitives to Distance Fields to Dynamical Systems

Yiming Li and Sylvain Calinon

***Abstract*—Developing autonomous robots capable of learning complex motions from demonstrations remains a fundamental challenge in robotics. On the one hand, movement primitives (MP) provide a compact and modular representation of continuous trajectories. On the other hand, dynamical systems (DS) provide control policies that are time-independent. In this paper, we propose a simple and flexible approach called MPDS that gathers the advantages of both representations by transforming MPs into autonomous systems. The key idea is to transform the explicit representation of a trajectory to an implicit shape encoded as a distance field. This conversion from a time-dependent motion to a spatial representation enables the construction of an autonomous dynamical system with modular reactions to perturbation. This approach bridges conventional MPs with distance fields, ensuring smooth and precise motion encoding, while maintaining a continuous spatial representation. We use Bernstein basis functions in the MPs to represent trajectories as concatenated quadratic Bézier curves, which provide an analytical method for computing distance fields. By simply leveraging the analytic gradients of the curve and its distance field, a stable dynamical system can be computed to reproduce the demonstrated trajectories while handling perturbations, without requiring a model of the dynamical system to be estimated. Numerical simulations and real-world robotic experiments validate our method's ability to encode complex motion patterns while ensuring trajectory stability, together with the flexibility of designing the desired reaction to perturbations. An interactive project page demonstrating our approach is available at https://idiap.github.io/mp-df-ds/.**

***Index Terms*—Movement Primitives, Splines, Distance Fields, Autonomous Systems, Dynamical Systems**

54 pt
0.75 in
19.1 mm

I. Introduction and Related Work

LEARNING complex motion skills by optimization or through demonstrations remains a fundamental challenge in robotics, which can computationally be addressed from different perspectives. Movement primitives (MPs) offer a compact and modular framework for encoding, generalizing, and reproducing demonstrated trajectories by superposition of basis functions, see [1] for a review. Various MP formulations have been proposed, including dynamical movement primitives (DMPs) [2], [3], Gaussian mixture regression (GMR) [4], probabilistic movement primitives (ProMPs) [5], kernelized movement primitives (KMPs) [6], or Fourier movement primitives (FMPs) [7].

Manuscript received: April, 10, 2025; June, 18, 2025; July, 19, 2025.

This paper was recommended for publication by Editor Aleksandra Faust upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the China Scholarship Council (No. 202204910113), and by the State Secretariat for Education, Research and Innovation in Switzerland for participation in the European Commission's Horizon Europe Program through the INTELLIMAN project (https://intelliman-project.eu/, HORIZON-CL4-Digital-Emerging Grant 101070136) and the SESTOSENSO project (http://sestosenso.eu/, HORIZON-CL4-Digital-Emerging Grant 101070310). We thank Guillaume Clivaz and Yan Zhang for assistance with the experiments and Martin Schonger for insightful discussions when preparing the paper.

The authors are with the Idiap Research Institute, Martigny, Switzerland and also with EPFL, Lausanne, Switzerland. (name.surname@idiap.ch)

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

Fig. 1: From movement primitive (*left*) to distance field (*middle*) to dynamical system (*right*). *Left:* A S-shape trajectory is parameterized using the concatenation of 2 quadratic curves. Control points are shown in blue. *Middle:* Distance field of the quadratic spline. The gradient of the distance field at a given point (in red) is perpendicular to the curve at its closest point (in orange). This property is used to compute the distance field analytically and build the stable dynamical system (*Right*). Red and black bullets are a random start point and the equilibrium point at the end of the trajectory.

Conventional MPs are often limited in their ability to robustly handle perturbations and dynamically adapt to environmental changes, due to their explicit representation of the signals based on a time or phase variable. This downside has motivated the development of alternative autonomous systems formulations to provide time-independent dynamical system representations, inherently robust to perturbations, at the expense of complexifying the problem of modeling long and/or high-dimensional movements. Examples include SEDS [8], CLF-DM [9], Imitation Flow [10], and Neural Contractive Dynamical Systems (NCDS) [11], which all ensure stability and adaptability by learning or imposing constraints on the system dynamics.

We propose MPDS, a simple and flexible approach that gathers the advantages of both representations by transforming MPs into autonomous systems (shown in Figure 1). Our work follows the same foundational principle of using autonomous systems for robust motion generation, but differs by avoiding learning-based modeling altogether. Instead, we leverage the geometric properties of a reference trajectory encoded as a distance field, allowing us to define an autonomous system with asymptotic stability guarantees through analytic computation. A related concept has been explored in contact-aware tasks such as robotic polishing, where surfaces and normal vectors guide the design of dynamical systems [12]. We apply the idea to full trajectory encoding beyond surface interaction.

The core idea is to reinterpret the demonstrated trajectories as implicit spatial representations encoded as distance fields. Rather than representing motion as a sequence of time-parameterized states, we encode the shape of the trajectory in space. The resulting field defines the shortest distance to the reference trajectory at each spatial location, enabling motion generation as gradient descent over this field.

Asymptotic stability is ensured by incorporating Bernstein

54 pt
0.75 in
19.1 mm

Margin requirements for the other pages
Paper size this page US Letter
54 pt
0.75 in
19.1 mm

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

basis functions into movement primitives (MPs), representing trajectories as concatenated quadratic Bézier curves. This formulation enables the analytical computation of distance fields, eliminating the need for point sampling or heuristic weighting [13]. By bridging conventional MPs with distance fields, this approach provides smooth and precise motion encoding within a continuous spatial representation. Leveraging the analytical gradients of both the curve and its distance field, a stable dynamical system can be directly derived to reproduce demonstrated trajectories and handle perturbations, without requiring explicit system modeling or learning techniques.

The motivation of representing structure implicitly in continuous space is inspired by recent advances in Signed Distance Fields (SDFs) and Neural Radiance Fields (NeRFs) [14], [15], offering compact and continuous representations for shapes and scenes. These techniques have influenced robotics, particularly in motion planning and scene understanding [16], [17]. Our work in this paper instead focuses on representing 1D trajectories embedded in high-dimensional spaces. By exploiting this lower-dimensional structure, we derive closed-form distance fields from concatenated quadratic Bézier curves, enabling smooth, precise, and efficient computation of spatial gradients.

This spatial formulation offers several advantages. It eliminates the need for time synchronization, naturally supports modular responses to perturbations, and provides a continuous and robust encoding of motion by avoiding the discretization artifacts common in numerical methods.

The main contributions of this paper are:

- We propose a novel use of distance fields as an implicit trajectory representation for autonomous systems, shifting from conventional time-dependent trajectory encoding to a spatial formulation guided by distance gradients.
- We derive an analytical method for computing distance fields to reference trajectories represented by concatenated quadratic Bézier curves. This approach enables efficient and accurate computation of distance and gradient fields while preserving the ability to represent complex motion patterns.
- We show that combining movement primitives with their corresponding gradient fields naturally yields autonomous dynamical systems with provable asymptotic stability—without requiring system modeling or learning-based estimation.

We validate our approach through numerical simulations and real-world robot experiments, demonstrating its effectiveness in encoding complex motions, generating stable and smooth trajectories, and adapting to external perturbations. Additionally, we highlight the simplicity and flexibility of our framework in:

- integrating multiple demonstrations through simple field operations;
- scaling to high-dimensional settings such as robot joint spaces;
- enabling adaptive motions that achieve a balance between robustness to perturbations and generalization to new trajectories.

II. Representing Trajectories as Distance Fields

Given a reference trajectory $\mathbf{F}$ whose points are given by $\mathbf{f}(t) : [0, T] \to \mathbb{R}^D$ parameterized by $t \in [0, T]$, where T is the trajectory length and D is the spatial dimension, the distance field $d(\mathbf{x})$ at any point $\mathbf{x} \in \mathbb{R}^D$ is defined as:

$$d(\mathbf{x}) = \min_{t \in [0,T]} \|\mathbf{f}(t) - \mathbf{x}\|, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm. This function assigns each point in space a scalar value corresponding to its shortest distance to the trajectory. The gradient of the distance field, $\nabla \mathbf{d}(\mathbf{x})$, provides the direction of steepest ascent. Given a point $\mathbf{x}$, the closest point on the trajectory is given by

$$\mathbf{x}_{\text{proj}} = \mathbf{x} - d(\mathbf{x}) \nabla \mathbf{d}(\mathbf{x}). \tag{2}$$

In other words, one obtains the projection by subtracting from $\mathbf{x}$ the displacement $d(\mathbf{x})\nabla \mathbf{d}(\mathbf{x})$, which points from the trajectory toward $\mathbf{x}$. The gradient $\nabla \mathbf{d}(\mathbf{x})$ has the unit norm $\|\nabla \mathbf{d}(\mathbf{x})\| = 1$, which is known as the Eikonal equation [18].

While distance fields can be evaluated numerically by discretizing the trajectory and evaluating the minimum distance from sampled points, this introduces discretization errors and result in non-smooth transitions. Instead, we present an analytical distance computation by representing the trajectory as Bernstein basis functions.

A. Quadratic spline as Movement Primitives

Splines are widely used as movement primitives due to their ability to provide smooth and flexible trajectory representations. We use here the term *quadratic spline* to define the concatenation of quadratic Bézier curves, corresponding to piecewise Bernstein polynomial functions that ensures continuity in both position and velocity. A quadratic spline is composed of N segments. Each segment $\mathbf{f}_i$ is a quadratic Bézier curve parameterized by a local time $t_i \in [0, 1]$:

$$\mathbf{f}_i(t_i) = (1-t_i)^2, \mathbf{w}_i^1 + 2(1-t_i)t_i, \mathbf{w}_i^2 + {t_i}^2, \mathbf{w}_i^3, \quad t_i \in [0, 1], \tag{3}$$

where $\mathbf{w}_i^1$, $\mathbf{w}_i^2$, and $\mathbf{w}_i^3$ are the control points of the i-th segment. The full trajectory $\mathbf{f}(t)$ is formed by mapping a global time $t \in [0, T]$ to the correct segment i and its local time parameter t_i. This is achieved by assuming each of the N segments has an equal duration, allowing the global time t to be linearly scaled to identify the active segment and its corresponding local time in the [0,1] interval. Quadratic splines can approximate complex trajectories by decomposing the motion into localized polynomial segments. To ensure smooth concatenation between consecutive segments, continuity constraints are imposed with

$$\mathbf{w}_i^3 = \mathbf{w}_{i+1}^1, \quad \mathbf{w}_i^2 - \mathbf{w}_i^3 = \mathbf{w}_{i+1}^1 - \mathbf{w}_{i+1}^2, \quad \mathbf{w}_N^2 = \mathbf{w}_N^3. \tag{4}$$

The first constraint ensures positional continuity at the segment boundaries. The second guarantees consistent gradients, maintaining C^1 continuity across the trajectory. The last enforces zero velocity at the end of the trajectory by setting the last two control points of the final Bézier segment to be equal so that the derivative of the curve—and hence the velocity—vanishes at the endpoint.

Margin requirements for the other pages
Paper size: this page: US Letter
54 pt
0.75 in
19.1 mm

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

This formulation can be compactly expressed in matrix form as

$$\mathbf{f}(t) = \boldsymbol{\phi}(t)\,\mathbf{w}, \quad \boldsymbol{\phi}(t) = \mathbf{T}(t)\,\mathbf{B}\,\mathbf{C}, \tag{5}$$

where $\mathbf{f}(t)$ represents the concatenated quadratic splines, $\mathbf{T}(t) = [1, t, t^2]$ encodes the quadratic polynomial basis, $\mathbf{B}$ is the corresponding coefficient matrix, and $\mathbf{C}$ enforces the continuity constraints. The basis function matrix $\boldsymbol{\phi}(t)$ is a piecewise function of the global time t that internally handles the selection and time-scaling for the appropriate segment. A detailed derivation of this formulation can be found in [19].

The parameters $\mathbf{w}$ determine the shape of the quadratic spline and are estimated to best fit a given reference trajectory. Given a set of trajectory points $\{\mathbf{p}_k\}_{k=1}^K$ sampled at time instances $\{t_k\}_{k=1}^K$, we formulate the parameter estimation as an optimization problem that minimizes the discrepancy between the spline representation and the reference trajectory $\boldsymbol{f}(t)$. Specifically, we solve

$$\min_{\mathbf{w}} \sum_{k=1}^{K} \|\mathbf{f}(t_k) - \mathbf{p}_k\|^2, \tag{6}$$

where $\mathbf{f}(t)$ is the spline-based reconstruction of the trajectory at t, and $\|\cdot\|$ denotes the Euclidean norm. The optimization problem can be solved efficiently using least squares. By expressing the trajectory in the matrix form

$$\mathbf{P} = \boldsymbol{\Phi}\,\mathbf{w}, \tag{7}$$

where $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_M]^\top$ is the matrix of trajectory points and $\boldsymbol{\Phi}$ is the corresponding basis function matrix evaluated at $\{t_j\}$, the optimal parameters are obtained by solving

$$\mathbf{w} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{P}, \tag{8}$$

where $(\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top$ is the Moore-Penrose pseudoinverse of $\boldsymbol{\Phi}$. The quadratic spline is then reconstructed using (5) for a smooth, continuous-time trajectory representation.

B. Distance Fields of Quadratic Splines

By minimizing $c(t_i) = \frac{1}{2}(\mathbf{f}_i(t_i) - \mathbf{x})^\top (\mathbf{f}_i(t_i) - \mathbf{x})$ w.r.t. t_i, we get the closest point $\mathbf{f}_i(t_i)$ to $\mathbf{x}$ on the curve. By differentiating $c(t_i)$ and equating to zero, we get

$$(\mathbf{f}_i(t_i) - \mathbf{x})^\top \, \dot{\mathbf{f}}_i(t_i) = 0, \tag{9}$$

where $\dot{\mathbf{f}}_i(t_i)$ denotes the derivative of the spline with respect to t_i. We can observe in the above equation that the closest point on the curve satisfies the orthogonality condition between the residual vector $\mathbf{f}_i(t_i) - \mathbf{x}$ and the tangent vector of the spline $\dot{\mathbf{f}}_i(t_i)$. Substituting the quadratic spline formulation

$$\begin{aligned} \mathbf{f}_i(t_i) &= (1 - t_i)^2\,\mathbf{w}_i^1 + 2(1 - t_i)t_i\,\mathbf{w}_i^2 + t_i^2\,\mathbf{w}_i^3 \\ \dot{\mathbf{f}}_i(t_i) &= -2(1 - t_i)\,\mathbf{w}_i^1 + (2 - t_i)\,\mathbf{w}_i^2 + t_i\,\mathbf{w}_i^3, \end{aligned} \tag{10}$$

in (9) results in a cubic equation

$$\boldsymbol{\alpha}_i^{\mathbf{3}}\, t_i^3 + \boldsymbol{\alpha}_i^{\mathbf{2}}\, t_i^2 + \boldsymbol{\alpha}_i^{\mathbf{1}}\, t_i + \boldsymbol{\alpha}_i^{\mathbf{0}} = 0, \tag{11}$$

where the coefficients $\boldsymbol{\alpha}_i^{\mathbf{3}}, \boldsymbol{\alpha}_i^{\mathbf{2}}, \boldsymbol{\alpha}_i^{\mathbf{1}}, \boldsymbol{\alpha}_i^{\mathbf{0}}$ are determined by $\mathbf{w}_i^1$, $\mathbf{w}_i^2$, $\mathbf{w}_i^3$, and $\mathbf{x}$. Therefore, the roots of this cubic equation within the interval $[0, 1]$ correspond to the candidate closest points on the spline. The minimal distance is then computed as

$$d_i(\mathbf{x}) = \min\big\{\|\mathbf{f}_i(t_i^*) - \mathbf{x}\| : t_i^* \in [0, 1]\big\}. \tag{12}$$

If no roots satisfy $t_i^* \in [0, 1]$, the closest point occurs at one of the segment boundaries, $t_i = 0$ or $t_i = 1$. The overall minimal distance to a trajectory composed of multiple segments is given by

$$d(\mathbf{x}) = \min\big\{d_i(\mathbf{x}) : i = 1, \ldots, N\big\}, \tag{13}$$

where d_i is the minimal distance for the i-th segment and N is the toal number of segments.

C. Fusing Multiple Trajectories

An advantage of representing motion using distance fields is the simple formulation to handle multiple demonstrations, achieved through the union operation of distance fields. Specifically, given a set of reference trajectories $\{\mathbf{f}^m(t)\}_{m=1}^M$, each represented as an individual distance field $d^m(\mathbf{x})$, the combined distance field is computed as

$$d(\mathbf{x}) = \min_{m \in \{1, \ldots, M\}} d^m(\mathbf{x}), \tag{14}$$

ensuring that at each point $\mathbf{x}$, the distance field retains the minimal distance to any of the given trajectories.

D. From Distance Fields to Dynamical Systems

In this section, we demonstrate how the distance field representation can be utilized to design stable dynamical systems in a structured and efficient manner. Given a reference trajectory encoded by quadratic splines, we use the approach described in Section II-B to calculate the distance field $d(\mathbf{x})$, the closest point $\mathbf{x}_{\text{proj}}$ on the trajectory, and the corresponding phase variable $t_{\text{phase}} = \mathbf{f}^{-1}(\mathbf{x}_{\text{proj}})$. We define the dynamical system

$$\dot{\mathbf{x}} = -\alpha\,\nabla \mathbf{d}(\mathbf{x}) + \beta\,\dot{\mathbf{f}}(t_{\text{phase}}), \tag{15}$$

where $\nabla \mathbf{d}(\mathbf{x})$ is the gradient of the distance field, pointing away from the trajectory. The term $\dot{\mathbf{f}}(t_{\text{phase}})$ is the velocity vector of the reference trajectory evaluated at t_{phase}, obtained from the analytic derivatives of the quadratic spline. $\alpha > 0$, $\beta > 0$ are gain parameters that regulate attraction to the trajectory and movement along it, respectively. The first term in the system ensures attraction toward the reference trajectory (if a perturbation occurs), while the second term produces smooth movement along the reference trajectory. To analyze the stability of this system, we introduce the following Lyapunov function

$$V(\mathbf{x}) = \frac{1}{2} d^2(\mathbf{x}). \tag{16}$$

Taking the time derivative and using the property $\dot{d}(\mathbf{x}) = \nabla \mathbf{d}(\mathbf{x})^\top \dot{\mathbf{x}}$, we obtain:

$$\begin{aligned} \dot{V}(\mathbf{x}) &= d(\mathbf{x})\dot{d}(\mathbf{x}) \\ &= d(\mathbf{x})\nabla \mathbf{d}(\mathbf{x})^\top(-\alpha\,\nabla \mathbf{d}(\mathbf{x}) + \beta\,\dot{\mathbf{f}}(t_{\text{phase}})) \\ &= -\alpha\, d(\mathbf{x})\|\nabla \mathbf{d}(\mathbf{x})\|^2 + \beta\, d(\mathbf{x})\nabla \mathbf{d}(\mathbf{x})^\top \dot{\mathbf{f}}(t_{\text{phase}}) \\ &= -\alpha\, d(\mathbf{x}) + \beta\, d(\mathbf{x})\nabla \mathbf{d}(\mathbf{x})^\top \dot{\mathbf{f}}(t_{\text{phase}}). \end{aligned} \tag{17}$$

54 pt
0.75 in
19.1 mm
54 pt
0.75 in
19.1 mm
54 pt
0.75 in
19.1 mm

Margin requirements for the other pages
Paper size this page US Letter
54 pt
0.75 in
19.1 mm

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Reconstruction error for different basis functions.

Method	3	7	12	17	22
Piecewise	7.97 ± 2.76	3.40 ± 1.10	1.97 ± 0.63	1.39 ± 0.44	1.08 ± 0.35
B.P.	6.18 ± 3.65	0.96 ± 0.60	0.28 ± 0.13	0.14 ± 0.06	0.10 ± 0.04
RBF	16.51 ± 4.35	1.79 ± 0.61	0.26 ± 0.08	0.12 ± 0.04	$\mathbf{0.05 \pm 0.02}$
Fourier	$\mathbf{5.18 \pm 3.52}$	$\mathbf{0.56 \pm 0.33}$	$\mathbf{0.19 \pm 0.08}$	$\mathbf{0.09 \pm 0.04}$	$\mathbf{0.05 \pm 0.02}$
Q.S. (ours)	6.18 ± 3.65	0.88 ± 0.53	0.23 ± 0.10	0.11 ± 0.04	0.06 ± 0.02

Since $d(\mathbf{x}) \geq 0$, the first term in $\dot{V}(\mathbf{x})$ is always non-positive. To ensure $\dot{V}(\mathbf{x}) \leq 0$, we analyze the second term $d(\mathbf{x})\nabla\mathbf{d}(\mathbf{x})^\top \dot{\mathbf{f}}(t_{\text{phase}})$ in two cases:

Case 1: Interior Point ($0 < t_{\text{phase}} < T$). This means the projection point $\mathbf{x}_{\text{proj}}$ lies in the interior of the trajectory, the orthogonality condition from (9) holds. This ensures that the gradient vector is perpendicular to the trajectory velocity vector, making their dot product zero:

$$\nabla\mathbf{d}(\mathbf{x})^\top \dot{\mathbf{f}}(t_{\text{phase}}) = 0.$$

Case 2: Boundary Points ($t_{\text{phase}} = 0$ or $t_{\text{phase}} = T$).

- At the **end point** ($t_{\text{phase}} = T$), the trajectory represented by the spline ends with zero velocity by design ($\dot{\mathbf{f}}(T) = 0$), which further ensures the autonomous system halts when reaching this last point. Therefore, the dot product term $\nabla\mathbf{d}(\mathbf{x})^\top \dot{\mathbf{f}}(T)$ is zero.
- At the **start point** ($t_{\text{phase}} = 0$), we always have $\nabla\mathbf{d}(\mathbf{x})^\top \dot{\mathbf{f}}(0) \leq 0$. Otherwise, it would mean the trajectory initially moves into the half-space containing $\mathbf{x}$ and the start point $\mathbf{f}(0)$ is not the closest point of $\mathbf{x}$ on the entire trajectory.

Since the second term is non-positive in all cases, we confirm that $\dot{V}(\mathbf{x}) \leq 0$, ensuring global stability. Furthermore, $\dot{V}(\mathbf{x}) = 0$ only when $d(\mathbf{x}) = 0$, which implies that the system asymptotically converges to the reference trajectory.

To achieve a balance between guiding the system toward the desired trajectory and ensuring smooth motion along it, we introduce an inverse barrier function to modulate the influence of the gradient term with

$$\beta = \frac{1}{1 + \lambda\, d(\mathbf{x})}, \quad \alpha = 1 - \beta, \tag{18}$$

where $\lambda > 0$ is a parameter to balance the two terms. In essence, larger values of the distance field $d(\mathbf{x})$ correspond to regions farther from the trajectory, prompting the dynamical system to exert a stronger influence to steer the state back toward the trajectory. Conversely, smaller distance values indicate proximity to the trajectory, causing the system to reduce its corrective force and allow for smoother motion along the path. The parameters, α and β in balancing attraction and progression terms can also be integrated with the phase variable t_{phase} for more complex behaviors.

III. Experiments

We evaluated the proposed approach through numerical simulations and real-world robotic experiments. Our evaluation is guided by the following key questions: **Q1**: How efficiently can quadratic splines capture and reproduce complex motion patterns? **Q2**: How do the analytical distance and gradient fields contribute to motion generation in autonomous systems? **Q3**: How robust and generalizable is the approach in handling perturbations and enabling adaptive motion responses? We address each of these questions through detailed analysis and discussion of our experimental results.

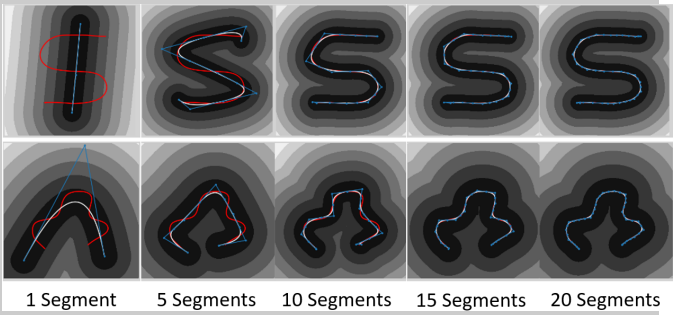

Fig. 2: The accuracy of quadratic splines in modeling trajectories improves as the number of segments increases. The red curve represents raw data from the LASA dataset, while the white curve shows trajectories encoded using quadratic splines. The blue components illustrate the superposition weights, and the colormap in the background represents the retrieved distance field.

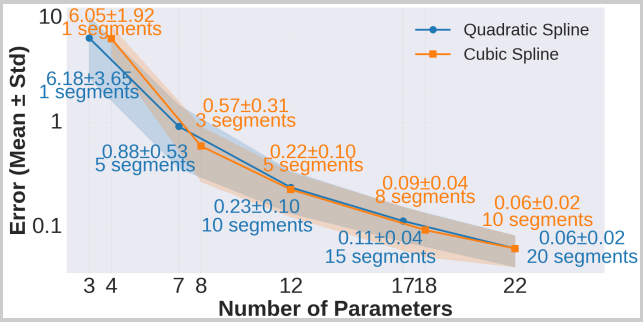

Fig. 3: Comparison between quadratic and cubic splines in terms of trajectory encoding accuracy.

A. Numerical Experiments

We first assess how well quadratic splines encode trajectories through numerical experiments on the LASA hand-written dataset.

Metrics. The quality of reconstruction is measured using the average ℓ_2 norm error

$$e = \frac{1}{N}\|\boldsymbol{\Phi w} - \boldsymbol{P}\|, \tag{19}$$

where $\boldsymbol{\Phi w}$ represents reconstructed points on the spline, $\boldsymbol{p}$ is a vector containing the original points of the trajectory and N is the number of points of $\boldsymbol{P}$.

Accuracy. To answer **Q1**, we compare the quadratic splines (Q.S.) with other basis functions, including piecewise constant basis functions, a single high-order Bernstein polynomial (B.P.), radial basis functions (RBF), and Fourier basis functions. For a fair comparison, we encode the trajectory using the

Margin requirements for the other pages
Paper size this page: US Letter
54 pt
0.75 in
19.1 mm

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

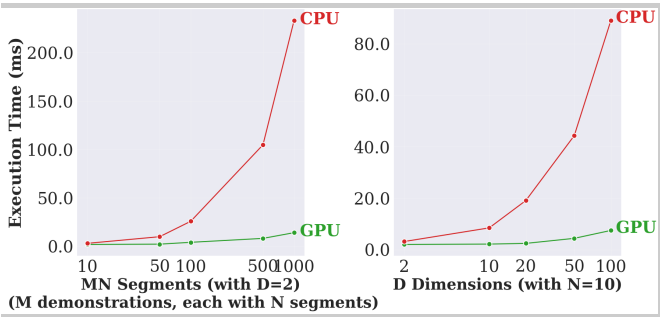

Fig. 4: Computation time for various numbers of segments (or demonstrations) and dimensions.

same number of parameters. A piecewise quadratic spline with N segments has $N+2$ parameters in total, so we use $N+2$ basis functions for each comparison method. Experimental results are reported in Table I.

Although the quadratic spline only uses piecewise quadratic polynomials for trajectory encoding, it achieves comparable performance with other commonly used movement primitives. This is a benefit from the flexibility provided by the spline, which allows local adjustments to be made without affecting the entire trajectory (see Figure 2). The piecewise line segments also enable distance field computation, however, it is limited by the low degree of the polynomials and the discontinuity of the derivatives. Therefore, the quadratic spline is preferable as it can encode trajectories expressively while providing an analytical form for distance computation. We also compare the results with cubic splines in Figure 3, a concatenation of higher-order Bernstein polynomials which is also widely used in trajectory encoding. The cubic spline with N segments has $2N+2$ parameters in total in the case of C^1 continuity. While cubic splines perform better with the same number of segments, they offer no significant advantage over quadratic splines for an equivalent number of parameters.

Computation Time. We evaluate in Figure 4 the computational cost of our approach across varying numbers of segments and dimensions. The computational cost of fusing M demonstrations, each with N segments, is equivalent to processing a single trajectory composed of MN segments. We measure the time required to compute distances, gradients, and time phases for 2,500 points relative to a quadratic curve in parallel using PyTorch. Results are shown in Figure 4. Notably, for typical robotic settings (e.g., $MN \leq 100$, $D \leq 10$), our method is highly efficient—requiring less than 5ms on a single NVIDIA GeForce RTX 3060 laptop GPU and less than 10 ms on an AMD Ryzen 7 6800H CPU. Even with significantly larger MN and D, the method remains efficient, particularly on the GPU. Further performance gains are possible due to the method's simplicity and ease of implementation.

Flexibility and Stability. To answer **Q2**, we evaluate the dynamical system derived from the distance field by analyzing its behavior with varying λ values (Figure 5). A larger λ increases the influence of the distance field, enhancing resistance to disturbances. In contrast, smaller λ values allow for smoother curvature following, resulting in less rigid trajectory adherence. Adjusting λ directly impacts the system's responsiveness, stability, and convergence. The results demonstrate

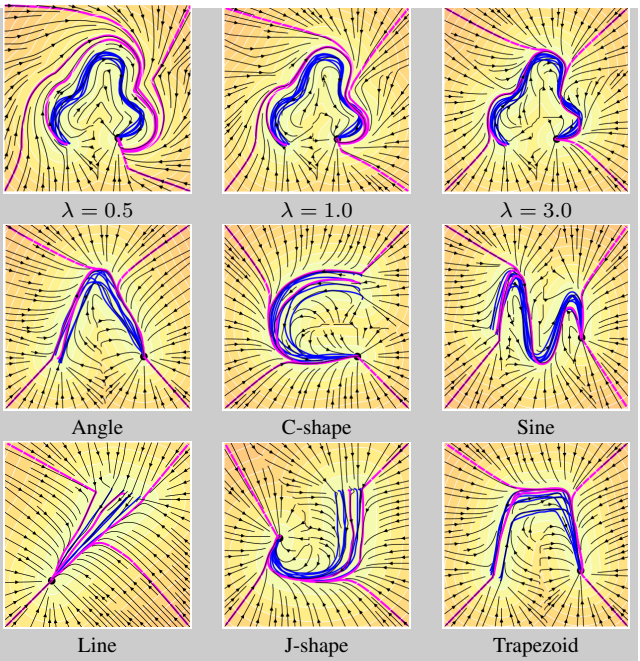

Fig. 5: Dynamical systems derived from the distance field. Blue curves represent demonstrations. Purple curves depict generated paths starting from the 4 corners of the image. *Top row:* Effect of varying λ. *Other rows:* Results for different trajectories from the LASA dataset.

the flexibility of the distance field representation in dynamically shaping autonomous system behaviors. Figure 5 shows results on various shapes from the LASA dataset with $\lambda = 3.0$.

To demonstrate the importance of using an analytical distance gradient in the design of globally stable dynamical systems, we compare our method with an alternative that uses a numerically computed distance field. The numerical field is obtained by uniformly sampling points along the trajectory and finding the closest point for each query. The results are shown in Figure 6. The comparison reveals that numerical distance fields can introduce instability, as their gradients may not remain perpendicular to the trajectory, causing the dynamical system to exhibit jerky motion or become stuck. In contrast, our analytically derived gradients ensure smooth, stable behavior throughout the trajectory.

Comparison with Baselines. We compare our method against two widely-used approaches: Dynamic Movement Primitives (DMP) [2] and Stable Estimator of Dynamical Systems (SEDS) [8]. Our approach offers distinct advantages in terms of model capability and computational efficiency. Specifically, DMP and SEDS rely on a learned model of the system's dynamics (e.g., a forcing term profile encoded with basis functions for DMP, or a Gaussian mixture model for SEDS). SEDS formulates the autonomous system as a constrained optimization problem. The constructed vector fields predominantly guide trajectories toward a single attractor, making it unsuitable for modeling and reproducing elaborate paths that would temporarily move away from the goal. It also becomes computationally challenging to apply this approach to high-dimensional spaces. In contrast, our method constructs the dynamics directly from the geometry of the reference path

54 pt
0.75 in
19.1 mm
54 pt
0.75 in
19.1 mm
54 pt
0.75 in
19.1 mm

Margin requirements for the other pages
Paper size this page US Letter
54 pt
0.75 in
19.1 mm

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

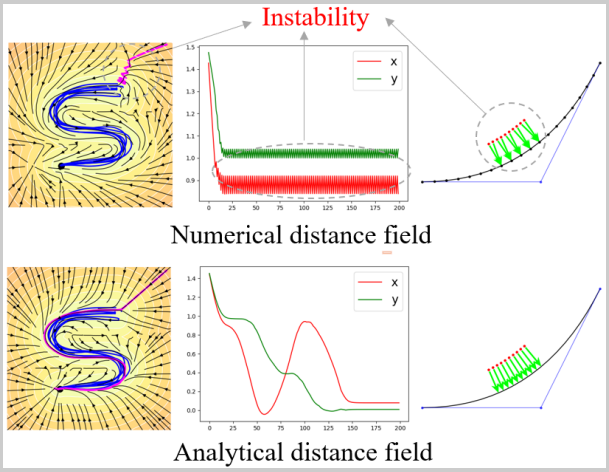

Fig. 6: Dynamical systems using either numerical or analytical distance fields. The left plots show the produced trajectory. The right plots show the instability caused by discontinuities in the numerical gradient, leading to an unstable dynamical system.

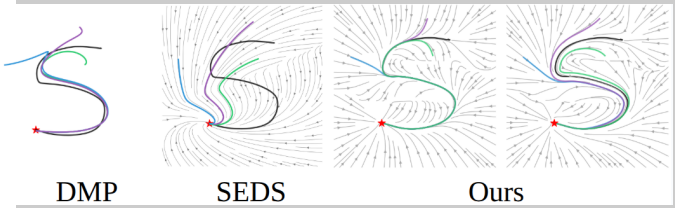

Fig. 7: Qualitative comparison with baseline approaches. Black curves represent the original demonstration, while colored curves show the reproductions.

(without learning a model) by superposing normal attraction and tangential progression vectors. DMP is based on the advancement of a phase variable (reparameterization of time), instead of an autonomous system. It is also typically designed to learn from a single demonstration. In contrast, our approach constructs an autonomous system that naturally accommodates multiple demonstrations by fusing their respective distance fields through a simple and efficient minimum operation. Additionally, our method offers highly intuitive and modular control over the robot's behavior. A single parameter directly balances the influence of the attraction term (how strongly we need to come back to the path if a perturbation occurs) and the progression term (how to move along the path), which can be easily balanced (see the last two plots of Figure 7). In contrast, the parameters in DMP and SEDS are often embedded within the learned models, making them less modular.

B. Robot Experiments

To answer **Q3**, we carry out three experiments with a 7-axis Franka robot:

1) Disturbance Handling: In this experiment, the robot was tasked with maintaining two demonstrated trajectories: S-shaped and L-shaped, which correspond to the dynamical system with $\alpha = 1$ and $\beta = 0$, in a task space of dimension $D = 3$, as shown in Figure 8-*top*. Disturbances are introduced by a person physically moving the robot. In the left panel, disturbances were introduced along the trajectory, and the robot

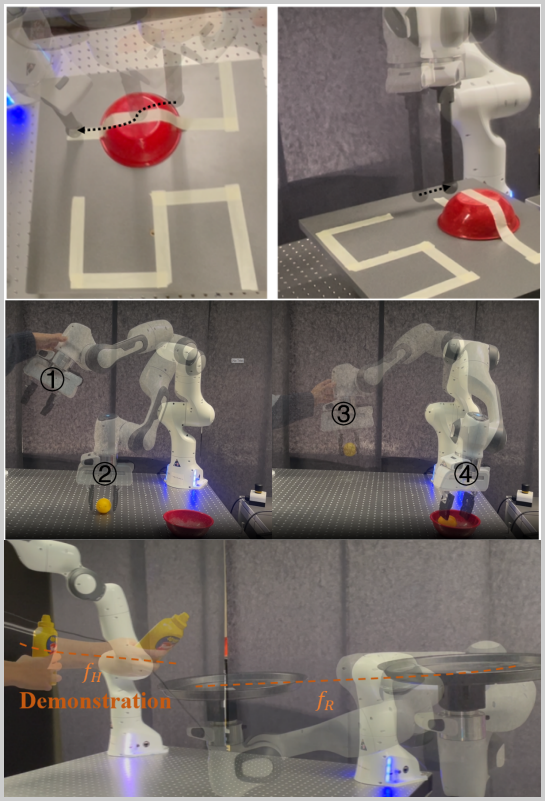

Fig. 8: Real-world experiments. *Top:* Disturbance handling. *Middle:* Pick and place. *Bottom:* Human-robot collaboration.

remains stationary after the disturbance. In the right panel, disturbances caused displacement along the normal direction of the reference trajectory; In response, the manipulator returned to the closest point on the reference trajectory, demonstrating its adaptability to unexpected deviations, facilitated by the distance field.

2) Pick and Place: In this task, the robot was required to pick up an object from the table and place it into a bowl (shown in Figure 8-*middle*). The demonstrated trajectory is defined in joint space with dimension $D = 7$. During execution, we manually disturbed the robot by physically dragging it from the reference path. At each iteration of the control loop, the robot estimates its task phase and the closest point on the demonstrated trajectory, which defines the dynamical system to return to the desirable positions. The robot is controlled by a torque controller in a high-compliance mode, allowing us to apply external disturbances easily. In the experiment, the gripper actions were triggered at predefined segments of the trajectory; however, this process could be integrated more dynamically into the overall framework by estimating the task phase online.

3) Human-Robot Collaboration: We demonstrate our approach in a dynamic human-robot handover task. A single demonstration is first recorded, encoding both the human's hand trajectory in task space, $f_H(t)$, and the robot's synchronized joint-space motion, $f_R(t)$, as quadratic splines. During interaction, a camera tracks the human's current hand position, $\mathbf{x}_H$. Our method is used here to project this position

54 pt
0.75 in
19.1 mm
54 pt
0.75 in
19.1 mm
54 pt
0.75 in
19.1 mm

Margin requirements for the other pages
Paper size: this page: US Letter
54 pt
0.75 in
19.1 mm

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

onto the learned human path f_H to continuously infer the task's current phase, t_{phase}. This phase variable then directly drives the robot, which is commanded to the corresponding point on its own trajectory, $\mathbf{f}_R(t_{\text{phase}})$. This coupling creates a fluid and anticipatory interaction where the robot dynamically synchronizes its timing with the human. It robustly adapts to unpredictable movements and sensor noise, ensuring a successful and efficient handover, as the system robustly handles unpredictable human timing and noisy sensor data. This adaptive behavior is illustrated in Figure 8-*bottom* and in the supplementary video.

IV. Discussion and Conclusion

We presented an approach called MPDS in which trajectories encoded using quadratic splines are converted to implicit distance fields, which can further be converted to stable autonomous systems. The key approaches described in this paper involve the least-squares method to represent a trajectory as a quadratic spline, and solving a cubic equation for the corresponding distance field, which are very simple, straightforward and intuitive. The approach can handle disturbances, dynamic environments and high-dimensional problems. While our primary focus has been on the adaptive and robust reproduction of demonstrations, the integration of movement primitives with distance fields opens new avenues for broader applications.

Limitations. In the approach presented in this paper, we re-estimate a phase/time variable t at each step without taking into account the previous estimates of t, in order to obtain a time-independent autonomous system. This design introduces sensitivity to initial conditions: small variations in the starting point may cause the system to converge to different segments of the trajectory. While relevant in some scenarios, this behavior may be undesirable in applications requiring more cautious behavior. Future work could investigate how the trace of intermediary phase/time variable estimates could be exploited to provide control commands as a trade-off between a fully myopic and a time-dependent system, similarly to model predictive control. Another possible limitation is that the encoded and retrieved path only has C^1 continuity. While this is good enough in many applications, further extensions could also consider the derivatives of the trajectories (also analytic with the proposed representation), so that states composed of both positions and velocities can be considered, together with acceleration commands to drive the system. This extension would also enable the handling of trajectories that cross in position space but that do not cross in the full state space.

Future work. One promising extension for future work is the formulation of optimal control problems to learn the control points of the curve, whose results are then re-interpreted as distance fields. Beyond deterministic representations, distance fields could also be extended into a probabilistic framework capturing uncertainty in trajectory execution. By modeling the distribution of distances, this approach could be used to enhance motion planning under uncertainty, enabling safe and adaptive behaviors in unstructured environments. Furthermore, distance fields could be learned directly from multimodal inputs, such as images, videos, or vision language models, allowing robots to infer motion trajectories from high-level visual or textual descriptions. These extensions illustrate the versatility of distance fields as a motion representation module that can be integrated into many learning, planning, control, and optimization problems.

54 pt
0.75 in
19.1 mm

References

[1] S. Calinon, "Mixture models for the analysis, edition, and synthesis of continuous time series," in *Mixture Models and Applications* (N. Bouguila and W. Fan, eds.), pp. 39–57, Springer, Cham, 2019.
[2] 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.
[3] 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.
[4] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.
[5] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems (NeurIPS)* (C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.
[6] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.
[7] T. Kulak, J. Silvério, and S. Calinon, "Fourier movement primitives: an approach for learning rhythmic robot skills from demonstrations," in *Proc. Robotics: Science and Systems (RSS)*, pp. 1–9, 2020.
[8] 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.
[9] S. M. Khansari-Zadeh and A. Billard, "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 752–765, 2014.
[10] J. Urain, M. Ginesi, D. Tateo, and J. Peters, "Imitationflow: Learning deep stable stochastic dynamic systems by normalizing flows," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5231–5237, 2020.
[11] H. Beik-Mohammadi, S. Hauberg, G. Arvanitidis, N. Figueroa, G. Neumann, and L. Rozo, "Neural contractive dynamical systems," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
[12] W. Amanhoud, M. Khoramshahi, and A. Billard, "A dynamical system approach to motion and force generation in contact tasks," Robotics: Science and Systems (RSS), 2019.
[13] X. Chen, Y. Michel, and D. Lee, "Closed-loop variable stiffness control of dynamical systems," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, pp. 163–169, 2021.
[14] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174, 2019.
[15] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
[16] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pp. 489–494, 2009.
[17] Y. Li, Y. Zhang, A. Razmjoo, and S. Calinon, "Representing robot geometry as distance fields: Applications to whole-body manipulation," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pp. 15351–15357, 2024.
[18] Y. Li, X. Chi, A. Razmjoo, and S. Calinon, "Configuration space distance fields for manipulation planning," in *Proc. Robotics: Science and Systems (RSS)*, 2024.
[19] "Robotics codes from scratch (RCFS)." https://rcfs.ch/, Accessed: 2025.

54 pt
0.75 in
19.1 mm