Mixture models for the analysis, edition, and synthesis of continuous time series

Sylvain Calinon

Abstract This chapter presents an overview of techniques used for the analysis, edition, and synthesis of continuous time series, with a particular emphasis on motion data. The use of mixture models allows the decomposition of time signals as a superposition of basis functions. It provides a compact representation that aims at keeping the essential characteristics of the signals. Various types of basis functions have been proposed, with developments originating from different fields of research, including computer graphics, human motion science, robotics, control, and neuroscience. Examples of applications with radial, Bernstein and Fourier basis functions are presented, with associated source codes to get familiar with these techniques.

1 Introduction

The development of techniques to process continuous time series is required in various domains of application, including computer graphics, human motion science, robotics, control, and neuroscience. These techniques need to cover various purposes, including the encoding, modeling, analysis, edition, and synthesis of time series (sometimes needed simultaneously). The development of these techniques is also often governed by additional important constraints such as interpretability and reproducibility. These heavy requirements motivate the use of mixture models, effectively leveraging the formalism and ubiquity of these models.

The first part of this chapter reviews decomposition techniques based on radial basis functions (RBFs) and locally weighted regression (LWR). The connections between LWR and Gaussian mixture regression (GMR) are dis-

Sylvain Calinon
Idiap Research Institute, Martigny, Switzerland, e-mail: sylvain.calinon@idiap.ch

cussed, based on the encoding of time series as Gaussian mixture models (GMMs). I will show how this mixture modeling principle can be extended to a weighted superposition of Bernstein basis functions, often known as Bézier curves. The aim is to examine the connections with mixture models and to highlight the generative aspects of these techniques. In particular, this link exposes the possibility of representing Bézier curves with higher order Bernstein polynomials. I then discuss the decomposition of time signals as Fourier basis functions, by showing how a mixture of Gaussians can leverage the multivariate Gaussian properties in the spatial and frequency domains. Finally, I show that these different decomposition techniques can be represented as time series distributions through a probabilistic movement primitives representation.

Pointers to various practical applications are provided for further readings, including the analysis of biological signals in the form of multivariate continuous time series, the development of computer graphics interfaces to edit trajectories and motion paths for manufacturing robots, the analysis and synthesis of periodic human gait data, or the generation of exploratory movements in mobile platforms with ergodic control.

The techniques presented in this chapter are described with a uniform notation that does not necessarily follow the original notation. The goal is to tie links between these different techniques, which are often presented in isolation of the more general context of mixture models. Matlab codes accompany the chapter [1], with full compatibility with GNU Octave.

2 Movement primitives

The term *movement primitives* refers to an organization of continuous motion signals in the form of a superposition in parallel and in series of simpler signals, which can be viewed as "building blocks" to create more complex movements, see Fig. 1. This principle, coined in the context of motor control [24], remains valid for a wide range of continuous time signals (for both analysis and synthesis). Next, I present three popular families of basis functions that can be employed for time series decomposition.

2.1 Radial basis functions (RBFs)

Radial basis functions (RBFs) are ubiquitous in continuous time series encoding [28], notably due to their simplicity and ease of implementation. Most algorithms exploiting this representation rely on some form of regression, often related to locally weighted regression (LWR), which was introduced by [9] in statistics and popularized by [4] in robotics. By representing, re-

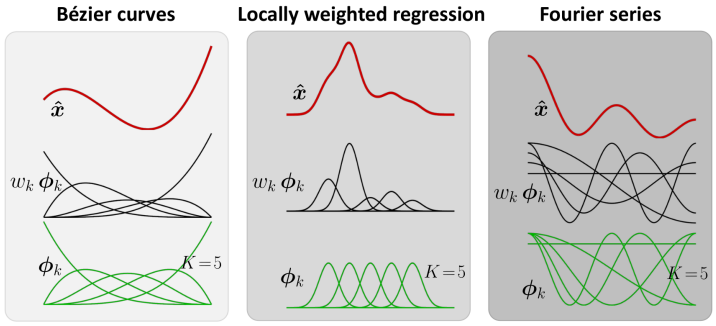

Fig. 1 Motion primitives with different basis functions ϕ_k, where a unidimensional time series $\hat{\boldsymbol{x}} = \sum_{k=1}^{K} w_k \phi_k$ is constructed as a weighted superposition of K signals ϕ_k.

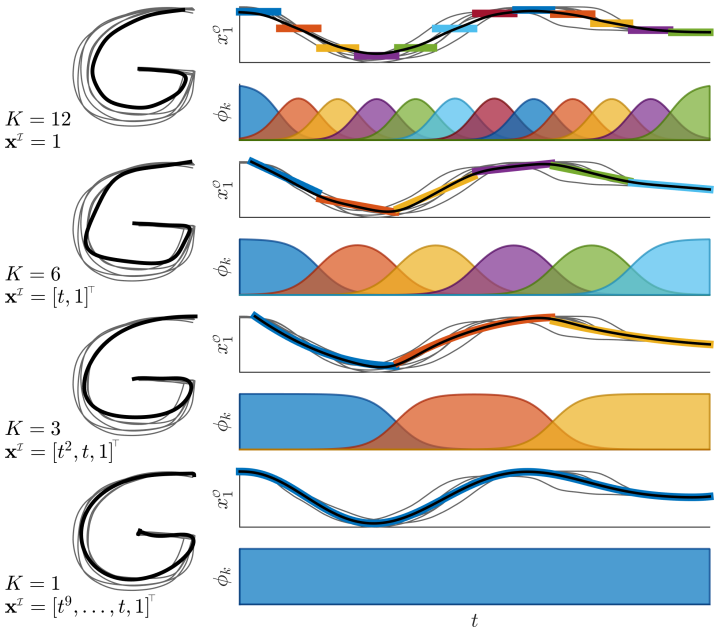

Fig. 2 Polynomial fitting with locally weighted regression (LWR), by considering different degrees of the polynomial and by adapting the number of basis functions accordingly. The top row shows a very localized encoding of the movement with constant values used in (1), thus requiring the use of many basis functions to represent the trajectory. The next rows show that a reduction of this number of basis functions typically needs to be compensated with more complex basis functions (polynomial of higher degrees). The bottom row depicts the limit case in which a global encoding of the movement would require a polynomial of high degree.

spectively, N input and output datapoints as $\boldsymbol{X}^I = [\boldsymbol{x}_1^I, \boldsymbol{x}_2^I, \dots, \boldsymbol{x}_N^I]^\top$ and $\boldsymbol{X}^O = [\boldsymbol{x}_1^O, \boldsymbol{x}_2^O, \dots, \boldsymbol{x}_N^O]^\top$, we are interested in the problem of finding a matrix $\boldsymbol{A}$ so that $\boldsymbol{X}^I\boldsymbol{A}$ would match $\boldsymbol{X}^O$ by considering different weights on the input–output datapoints $\{\boldsymbol{X}^I, \boldsymbol{X}^O\}$ (namely some datapoints are more informative than others for the estimation of $\boldsymbol{A}$). A weighted least squares estimate $\hat{\boldsymbol{A}}$ can be found by solving the objective

$$\begin{aligned}\hat{\boldsymbol{A}} &= \arg\min_{\boldsymbol{A}} \operatorname{tr}\Big((\boldsymbol{X}^O - \boldsymbol{X}^I\boldsymbol{A})^\top \boldsymbol{W}(\boldsymbol{X}^O - \boldsymbol{X}^I\boldsymbol{A})\Big)\\ &= (\boldsymbol{X}^{I\top}\boldsymbol{W}\boldsymbol{X}^I)^{-1}\boldsymbol{X}^{I\top}\boldsymbol{W}\,\boldsymbol{X}^O,\end{aligned} \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is a weighting matrix. Locally weighted regression (LWR) is a direct extension of the weighted least squares formulation in which K weighted regressions are performed on the same dataset $\{\boldsymbol{X}^I, \boldsymbol{X}^O\}$. It aims at splitting a nonlinear problem so that it can be solved locally by linear regression. LWR computes K estimates $\hat{\boldsymbol{A}}_k$, each with a different function $\phi_k(\boldsymbol{x}_n^I)$, classically defined as the radial basis functions

$$\tilde{\phi}_k(\boldsymbol{x}_n^I) = \exp\Big(-\frac{1}{2}(\boldsymbol{x}_n^I - \boldsymbol{\mu}_k^I)^\top {\boldsymbol{\Sigma}_k^I}^{-1}(\boldsymbol{x}_n^I - \boldsymbol{\mu}_k^I)\Big), \tag{2}$$

where $\boldsymbol{\mu}_k^I$ and $\boldsymbol{\Sigma}_k^I$ are the parameters of the k-th RBF, or in its rescaled form[1]

$$\phi_k(\boldsymbol{x}_n^I) = \frac{\tilde{\phi}_k(\boldsymbol{x}_n^I)}{\sum_{i=1}^K \tilde{\phi}_i(\boldsymbol{x}_n^I)}. \tag{3}$$

An associated diagonal matrix

$$\boldsymbol{W}_k = \operatorname{diag}\Big(\phi_k(\boldsymbol{x}_1^I), \phi_k(\boldsymbol{x}_2^I), \dots, \phi_k(\boldsymbol{x}_N^I)\Big) \tag{4}$$

can be used with (1) to evaluate $\hat{\boldsymbol{A}}_k$. The result can then be employed to compute

$$\hat{\boldsymbol{X}}^O = \sum_{k=1}^K \boldsymbol{W}_k\,\boldsymbol{X}^I\hat{\boldsymbol{A}}_k. \tag{5}$$

The centroids $\boldsymbol{\mu}_k^I$ in (2) are usually set to uniformly cover the input space, and $\boldsymbol{\Sigma}_k^I = \boldsymbol{I}\sigma^2$ is used as a common bandwidth shared by all basis functions. Figure 2 shows an example of LWR to encode planar trajectories.

LWR can be directly extended to local least squares polynomial fitting by changing the definition of the inputs. Multiple variants of the above formulation exist, including online estimation with a recursive formulation [27], Bayesian treatments of LWR [31], or extensions such as locally weighted pro-

[1] We will see later that the rescaled form is required for some techniques, but for locally weighted regression, it can be omitted to enforce the independence of the local function approximators.

jection regression (LWPR) that exploit partial least squares to cope with redundant or irrelevant inputs [33].

Examples of application range from inverse dynamics modeling [33] to the skillful control of a devil-stick juggling robot [5]. A Matlab code example `demo_LWR01.m` can be found in [1].

2.1.1 Gaussian mixture regression (GMR)

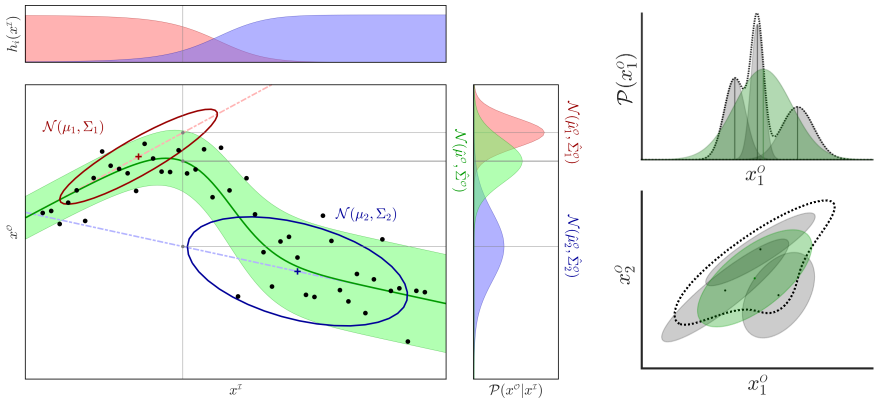

Fig. 3 *Left:* Gaussian mixture regression (GMR) for 1D input x^I and 1D output x^O. *Right:* Gaussian that best approximates a mixture of Gaussians. The multimodal distributions in dashed line depict the probability density functions for the mixtures of three Gaussians in gray color (examples in 1D and 2D are depicted). The Gaussians in green color approximate these multimodal distributions.

Gaussian mixture regression (GMR) is a another popular technique for time series and motion representations [13, 8]. It relies on linear transformation and conditioning properties of multivariate Gaussian distributions. GMR provides a synthesis mechanism to compute output distributions with a computation time independent of the number of datapoints used to train the model. A characteristic of GMR is that it does not model the regression function directly. Instead, it first models the joint probability density of the data in the form of a Gaussian mixture model (GMM). It can then compute the regression function from the learned joint density model, resulting in very fast computation of a conditional distribution.

In GMR, both input and output variables can be multidimensional. Any subset of input–output dimensions can be selected, which can change, if required, at each time step. Thus, any combination of input–output mappings can be considered, where expectations on the remaining dimensions are computed as a multivariate distribution. In the following, we will denote the block decomposition of a datapoint $\boldsymbol{x}_t \in \mathbb{R}^D$ at time step t, and the center $\boldsymbol{\mu}_k$ and

covariance $\boldsymbol{\Sigma}_k$ of the k-th Gaussian in the GMM as

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{x}_t^I \\ \boldsymbol{x}_t^O \end{bmatrix}, \quad \boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_k^I \\ \boldsymbol{\mu}_k^O \end{bmatrix}, \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k^I & \boldsymbol{\Sigma}_k^{IO} \\ \boldsymbol{\Sigma}_k^{OI} & \boldsymbol{\Sigma}_k^O \end{bmatrix}. \tag{6}$$

We first consider the example of time-based trajectories by using $\boldsymbol{x}_t^I$ as a time variables. At each time step t, $\mathcal{P}(\boldsymbol{x}_t^O|\boldsymbol{x}_t^I)$ can be computed as the multimodal conditional distribution

$$\begin{aligned} \mathcal{P}(\boldsymbol{x}_t^O|\boldsymbol{x}_t^I) &= \sum_{k=1}^{K} h_k(\boldsymbol{x}_t^I)\, \mathcal{N}\Big(\hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I), \hat{\boldsymbol{\Sigma}}_k^O\Big), \\ \text{with} \quad \hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I) &= \boldsymbol{\mu}_k^O + \boldsymbol{\Sigma}_k^{OI} {\boldsymbol{\Sigma}_k^I}^{-1}(\boldsymbol{x}_t^I - \boldsymbol{\mu}_k^I), \\ \hat{\boldsymbol{\Sigma}}_k^O &= \boldsymbol{\Sigma}_k^O - \boldsymbol{\Sigma}_k^{OI} {\boldsymbol{\Sigma}_k^I}^{-1} \boldsymbol{\Sigma}_k^{IO}, \\ \text{and} \quad h_k(\boldsymbol{x}_t^I) &= \frac{\pi_k\, \mathcal{N}(\boldsymbol{x}_t^I|\, \boldsymbol{\mu}_k^I, \boldsymbol{\Sigma}_k^I)}{\sum_{i=1}^{K} \pi_i\, \mathcal{N}(\boldsymbol{x}_t^I|\, \boldsymbol{\mu}_i^I, \boldsymbol{\Sigma}_i^I)}, \end{aligned} \tag{7}$$

computed with

$$\mathcal{N}(\boldsymbol{x}_t^I|\, \boldsymbol{\mu}_k^I, \boldsymbol{\Sigma}_k^I) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}_k^I|^{-\frac{1}{2}} \exp\Big(-\frac{1}{2}(\boldsymbol{x}_t^I - \boldsymbol{\mu}_k^I)^\top {\boldsymbol{\Sigma}_k^I}^{-1}(\boldsymbol{x}_t^I - \boldsymbol{\mu}_k^I)\Big).$$

When a unimodal output distribution is required, the law of total mean and variance (see Fig. 3-*right*) can be used to approximate the distribution with the Gaussian

$$\begin{aligned} \mathcal{P}(\boldsymbol{x}_t^O|\boldsymbol{x}_t^I) &= \mathcal{N}\Big(\boldsymbol{x}_t^O|\, \hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I), \hat{\boldsymbol{\Sigma}}^O(\boldsymbol{x}_t^I)\Big), \\ \text{with} \quad \hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I) &= \sum_{k=1}^{K} h_k(\boldsymbol{x}_t^I)\, \hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I), \\ \text{and} \quad \hat{\boldsymbol{\Sigma}}^O(\boldsymbol{x}_t^I) &= \sum_{k=1}^{K} h_k(\boldsymbol{x}_t^I)\Big(\hat{\boldsymbol{\Sigma}}_k^O + \hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I)\, \hat{\boldsymbol{\mu}}_k^O(\boldsymbol{x}_t^I)^\top\Big) - \hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I)\, \hat{\boldsymbol{\mu}}^O(\boldsymbol{x}_t^I)^\top. \end{aligned} \tag{8}$$

Figure 3 presents an example of GMR with 1D input and 1D output. With the GMR representation, LWR corresponds to a GMM with diagonal covariances. Expressing LWR in the more general form of GMR has several advantages: (1) it allows the encoding of local correlations between the motion variables by extending the diagonal covariances to full covariances; (2) it provides a principled approach to estimate the parameters of the RBFs, similar to a GMM parameters fitting problem; (3) it often allows a significant reduction of the number of RBFs, because the position and spread of each RBF are also estimated; and (4) the (online) estimation of the mixture model parameters and the model selection problem (automatically estimating the number of basis functions) can readily exploit techniques compatible with

GMM (Bayesian nonparametrics with Dirichlet processes, spectral clustering, small variance asymptotics, expectation-maximization procedures, etc.).

Another approach to encode and synthesize a movement is to rely on time-invariant autonomous systems. GMR can also be employed in this context to retrieve an autonomous system $\mathcal{P}(\dot{\boldsymbol{x}}|\boldsymbol{x})$ from the joint distribution $\mathcal{P}(\boldsymbol{x},\dot{\boldsymbol{x}})$ encoded in a GMM, where $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are position and velocity, respectively (see [14] for details). Similarly, it can be used in an autoregressive context by retrieving $\mathcal{P}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_{t-2},\dots,\boldsymbol{x}_{t-T})$ at each time step t, from the joint encoding of the positions on a time window of size T.

Practical applications of GMR include the analysis of speech signals [32, 16], electromyography signals [18], vision and MoCap data [30], and cancer prognosis [11]. A Matlab code example `demo_GMR01.m` can be found in [1].

2.2 Bernstein basis functions

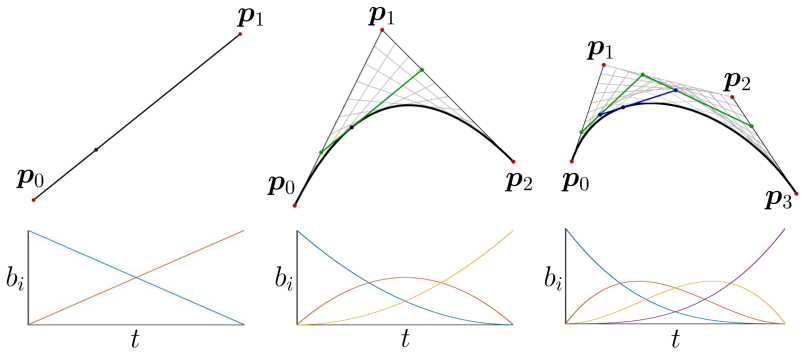

Fig. 4 Linear (*left*), quadratic (*center*) and cubic (*right*) Bézier curves constructed as a weighted superposition of Bernstein basis functions.

Bézier curves are well-known representations of trajectories [12]. Their underlying representation is a superposition of basis functions, which is overlooked in many applications. For $0 \leqslant t \leqslant 1$, a linear Bézier curve is the line traced by the function $\boldsymbol{x}_{\boldsymbol{p}_0,\boldsymbol{p}_1}(t)$, from $\boldsymbol{p}_0$ to $\boldsymbol{p}_1$,

$$\boldsymbol{x}_{\boldsymbol{p}_0,\boldsymbol{p}_1}(t) = (1-t)\boldsymbol{p}_0 + t\,\boldsymbol{p}_1. \tag{9}$$

For $0 \leqslant t \leqslant 1$, a quadratic Bézier curve is the path traced by the function

$$\begin{aligned}\boldsymbol{x}_{\boldsymbol{p}_0,\boldsymbol{p}_1,\boldsymbol{p}_2}(t) &= (1-t)\,\boldsymbol{x}_{\boldsymbol{p}_0,\boldsymbol{p}_1}(t) + t\,\boldsymbol{x}_{\boldsymbol{p}_1,\boldsymbol{p}_2}(t)\\ &= (1-t)\Big((1-t)\boldsymbol{p}_0 + t\,\boldsymbol{p}_1\Big) + t\Big((1-t)\boldsymbol{p}_1 + t\,\boldsymbol{p}_2\Big)\\ &= (1-t)^2\boldsymbol{p}_0 + 2(1-t)t\,\boldsymbol{p}_1 + t^2\boldsymbol{p}_2. \end{aligned} \tag{10}$$

For $0 \leqslant t \leqslant 1$, a cubic Bézier curve is the path traced by the function

$$\begin{aligned}\boldsymbol{x}_{\boldsymbol{p}_0,\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3}(t) &= (1-t)\,\boldsymbol{x}_{\boldsymbol{p}_0,\boldsymbol{p}_1,\boldsymbol{p}_2}(t) + t\,\boldsymbol{x}_{\boldsymbol{p}_1,\boldsymbol{p}_2,\boldsymbol{p}_3}(t)\\ &= (1-t)^3\boldsymbol{p}_0 + 3(1-t)^2t\boldsymbol{p}_1 + 3(1-t)t^2\boldsymbol{p}_2 + t^3\boldsymbol{p}_3. \end{aligned} \tag{11}$$

For $0 \leqslant t \leqslant 1$, a recursive definition for a Bézier curve of degree n can be expressed as a linear interpolation of a pair of corresponding points in two Bézier curves of degree $n-1$, namely

$$\boldsymbol{x}(t) = \sum_{i=0}^{n} b_{i,n}(t)\,\boldsymbol{p}_i, \quad \text{with} \quad b_{i,n}(t) = \binom{n}{i}\,(1-t)^{n-i}\,t^i, \tag{12}$$

with $b_{i,n}(t)$ the Bernstein basis polynomials of degree n, where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ are binomial coefficients.

Figure 4 illustrates the construction of Bézier curves of different orders. Practical applications are diverse but include most notably trajectories in computer graphics [12] and path planning [10]. A Matlab code example `demo_Bezier01.m` can be found in [1].

2.3 Fourier basis functions

In time series encoding, the use of Fourier basis functions provides useful connections between the spatial domain and the frequency domain. In the context of Gaussian mixture models, several Fourier series properties can be exploited, notably regarding zero-centered Gaussians, shift, symmetry, and linear combination. For the 1D case, these properties are:

- If $\phi(x) = \mathcal{N}(x \,|\, 0, \sigma^2) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp(-\frac{x^2}{2\sigma^2})$ is used to create a periodic function with period $L \gg \sigma$, the corresponding Fourier series coefficients are of the form $\phi_k = \exp(-\frac{2\pi^2k^2\sigma^2}{L^2})$;
- If ϕ_k are the Fourier series coefficients of a function $\phi(x)$, $\exp(-i\frac{2\pi k\mu}{L})\phi_k$ are the Fourier coefficients of $\phi(x-\mu)$, with i the imaginary unit ($i^2 = -1$);
- If $\phi_{k,1}$ (resp. $\phi_{k,2}$) are the Fourier series coefficients of a function $\phi_1(x)$ (resp. $\phi_2(x)$), then $\alpha_1\phi_{k,1} + \alpha_2\phi_{k,2}$ are the Fourier coefficients of $\alpha_1\phi_1(x) + \alpha_2\phi_2(x)$.

Well-known applications of Fourier basis functions in the context of time series include speech processing [32, 16] and the analysis of periodic motions

such as gaits [3]. Such decompositions also have a wider scope of applications, as illustrated next with ergodic control.

2.4 Ergodic control

In ergodic control, the aim is to find a series of control commands $\boldsymbol{u}(t)$ so that the retrieved trajectory $\boldsymbol{x}(t) \in \mathbb{R}^D$ covers a bounded space $\mathcal{X}$ in proportion of a given spatial distribution $\phi(\boldsymbol{x})$. As proposed in [22], this can be achieved by defining a metric in the spectral domain, by decomposing in Fourier series coefficients both the spatial distribution $\phi(\boldsymbol{x})$ and the (partially) retrieved trajectory $\boldsymbol{x}(t)$.[2] The goal of ergodic control is to minimize

$$\epsilon\big(\boldsymbol{x}(t)\big) = \frac{1}{2} \sum_{\boldsymbol{k} \in \mathcal{K}} \Lambda_{\boldsymbol{k}} \Big(c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big) - \phi_{\boldsymbol{k}} \Big)^2 \tag{13}$$

$$= \frac{1}{2} \Big(\boldsymbol{c}\big(\boldsymbol{x}(t)\big) - \boldsymbol{\phi} \Big)^\top \boldsymbol{\Lambda} \Big(\boldsymbol{c}\big(\boldsymbol{x}(t)\big) - \boldsymbol{\phi} \Big), \tag{14}$$

where $\Lambda_{\boldsymbol{k}}$ are weights, $\phi_{\boldsymbol{k}}$ are the Fourier series coefficients of $\phi(\boldsymbol{x})$, and $c_{\boldsymbol{k}}$ are the Fourier series coefficients along the trajectory $\boldsymbol{x}(t)$. $\mathcal{K}$ is a set of index vectors in $\mathbb{N}^D$ covering the D-dimensional array $\boldsymbol{k} = \boldsymbol{r} \times \boldsymbol{r} \times \cdots \times \boldsymbol{r}$, with $\boldsymbol{r} = [0, 1, \ldots, K]$ and K the resolution of the array. $\boldsymbol{c} \in \mathbb{R}^{K^D}$ and $\boldsymbol{\phi} \in \mathbb{R}^{K^D}$ are vectors composed of elements $c_{\boldsymbol{k}}$ and $\phi_{\boldsymbol{k}}$, respectively. $\boldsymbol{\Lambda} \in \mathbb{R}^{K^D \times K^D}$ is a diagonal weighting matrix with elements $\Lambda_{\boldsymbol{k}}$. In (13), the weights

$$\Lambda_{\boldsymbol{k}} = \big(1 + \|\boldsymbol{k}\|^2\big)^{-\frac{D+1}{2}} \tag{15}$$

assign more importance on matching low frequency components (related to a metric for Sobolev spaces of negative order). The Fourier series coefficients $c_{\boldsymbol{k}}$ along a trajectory $\boldsymbol{x}(t)$ of duration t are defined as

$$c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big) = \frac{1}{t} \int_{s=0}^{t} f_{\boldsymbol{k}}\big(\boldsymbol{x}(s)\big)\, ds, \tag{16}$$

whose discretized version can be computed recursively at each time step t to build

$$c_{\boldsymbol{k}}(\boldsymbol{x}_t) = \frac{1}{t} \sum_{s=1}^{t} f_{\boldsymbol{k}}(\boldsymbol{x}_s), \tag{17}$$

or equivalently in vector form $\boldsymbol{c}(\boldsymbol{x}_t) = \frac{1}{t} \sum_{s=1}^{t} \boldsymbol{f}(\boldsymbol{x}_s)$.

For a spatial signal $\boldsymbol{x} \in \mathbb{R}^D$, where x_d is on the interval $[-\frac{L}{2}, \frac{L}{2}]$ of period L, $\forall d \in \{1, \ldots, D\}$, the basis functions of the Fourier series with complex

[2] In [22], cosine basis functions are employed but the approach can be extended to other basis functions.

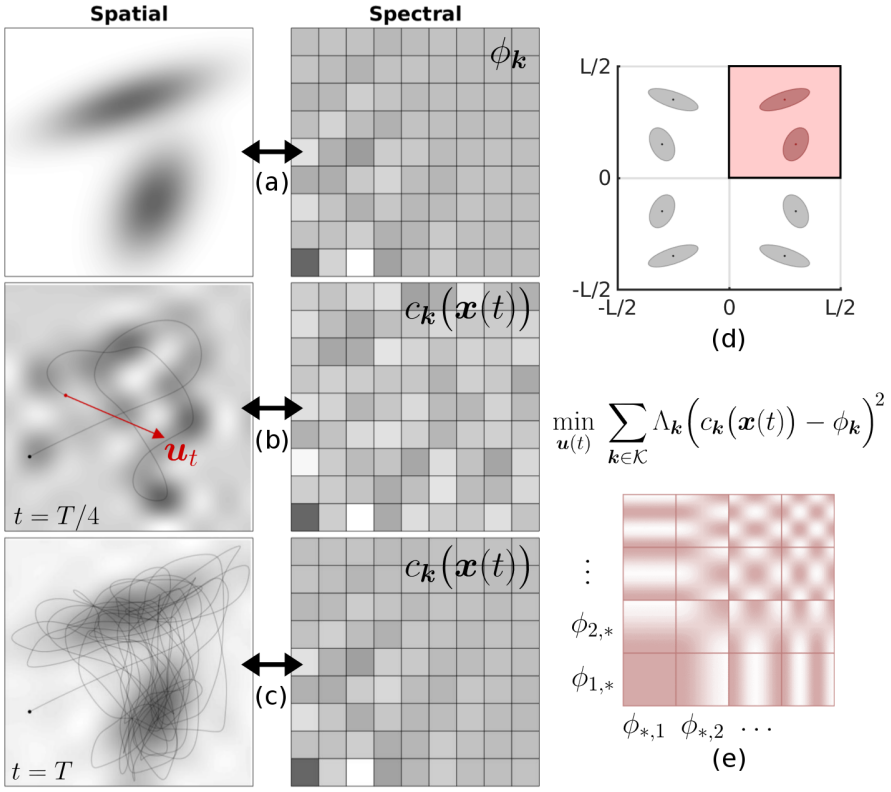

Fig. 5 2D ergodic control problem. In **(a)**–**(c)**, the left graphs show the spatial distribution (gray colormap) that the agent has to explore, encoded here as a mixture of two Gaussians. The right graphs show the corresponding Fourier series coefficients $\phi_{\boldsymbol{k}}$ in the frequency domain ($K = 9$ coefficients per dimension), which can be computed analytically by exploiting the shift, symmetry and linear combination properties of Gaussians. **(b)** shows the evolution of the reconstructed spatial distribution (left graph) and the computation of the next control command $\boldsymbol{u}$ (red arrow) after one fourth of the movement. The corresponding Fourier series coefficients $c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$ are shown in the right graph. **(c)** shows that after T iterations, the agent covers the space in proportion to the desired spatial distribution, with a good match of coefficients in the frequency domain ($\phi_{\boldsymbol{k}}$ in **(a)** and $c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$ in **(c)** are nearly the same). **(d)** shows how a periodic signal $\phi(\boldsymbol{x})$ (with range $[-L/2, L/2]$ for each dimension) can be constructed from the original mixture of two Gaussians $\phi_0(\boldsymbol{x})$ (red area). The constructed signal $\phi(\boldsymbol{x})$ is composed of eight Gaussians in this 2D example (mirroring the Gaussians along horizontal and vertical axes to construct an even signal of period L). **(e)** depicts the spatial reconstruction of each Fourier series coefficient (for the first four coefficients in each dimension), corresponding to periodic signals at different frequencies along the two axes.

exponential functions are defined as

$$f_{\boldsymbol{k}}(\boldsymbol{x}) = \frac{1}{L^D}\prod_{d=1}^{D}\exp\!\left(-i\frac{2\pi k_d x_d}{L}\right)$$
$$= \frac{1}{L^D}\prod_{d=1}^{D}\cos\!\left(\frac{2\pi k_d x_d}{L}\right) - i\,\sin\!\left(\frac{2\pi k_d x_d}{L}\right), \quad \forall \boldsymbol{k}\in\mathcal{K}. \tag{18}$$

Computation of Fourier series coefficients $\phi_{\boldsymbol{k}}$ for a spatial distribution represented as a Gaussian mixture model

We consider a desired spatial distribution $\phi_0(\boldsymbol{x})$ represented as a mixture of J Gaussians with centers $\boldsymbol{\mu}_j$, covariance matrices $\boldsymbol{\Sigma}_j$, and mixing coefficients α_j (with $\sum_{j=1}^{J}\alpha_j = 1$ and $\alpha_j \geqslant 0$),

$$\phi_0(\boldsymbol{x}) = \sum_{j=1}^{J}\alpha_j\,\mathcal{N}\big(\boldsymbol{x}\,|\,\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\big) \tag{19}$$
$$= \sum_{j=1}^{J}\alpha_j\,(2\pi)^{-\frac{D}{2}}\,|\boldsymbol{\Sigma}_j|^{-\frac{1}{2}}\,\exp\!\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\Big),$$

with each dimension on the interval $[0, \frac{L}{2}]$. $\phi_0(\boldsymbol{x})$ is extended to a periodized function by constructing an even function on the interval $\mathcal{X}$, where each dimension x_d is on the interval $\mathcal{X} = [-\frac{L}{2}, \frac{L}{2}]$ of period L. This is achieved with mirror symmetries of the Gaussians around all zero axes, see Fig. 5-*(d)*. The resulting spatial distribution can be expressed as a mixture of $2^D J$ Gaussians

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{J}\sum_{m=1}^{2^D}\frac{\alpha_j}{2^D}\,\mathcal{N}\big(\boldsymbol{x}\,\big|\,\boldsymbol{A}_m\boldsymbol{\mu}_j, \boldsymbol{A}_m\boldsymbol{\Sigma}_j\boldsymbol{A}_m^\top\big), \tag{20}$$

with linear transformation matrices $\boldsymbol{A}_m$.[3] By exploiting the symmetries and Gaussian distribution properties presented in Section 2.3, the Fourier series coefficients $\phi_{\boldsymbol{k}}$ can be analytically computed as

[3] $\boldsymbol{A}_m = \mathrm{diag}(\boldsymbol{H}_{2^D-D+1:2^D,m})$, where $\boldsymbol{H}_{2^D-D+1:2^D,m}$ is a vector composed of the last D elements in the column m of the Hadamard matrix $\boldsymbol{H}$ of size 2^D. Alternatively, $\boldsymbol{A}_m = \mathrm{diag}\big(\mathrm{vec}(\boldsymbol{\ell}_m)\big)$ can be constructed with the array $\boldsymbol{\ell}_m$, with m indexing the first dimension of the array $\boldsymbol{\ell} = \boldsymbol{s}\times\boldsymbol{s}\times\cdots\times\boldsymbol{s} \in \mathbb{Z}^{2\times 2\times\ldots\times 2}$ with $\boldsymbol{s} = [-1, 1]$. In 2D, we have $\boldsymbol{A}_1 = \left[\begin{smallmatrix}-1 & 0\\ 0 & -1\end{smallmatrix}\right]$, $\boldsymbol{A}_2 = \left[\begin{smallmatrix}-1 & 0\\ 0 & 1\end{smallmatrix}\right]$, $\boldsymbol{A}_3 = \left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right]$ and $\boldsymbol{A}_2 = \left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right]$, see Fig. 5-*(d)*.

$$\begin{aligned}\phi_{\boldsymbol{k}} &= \int_{\boldsymbol{x}\in\mathcal{X}} \phi(\boldsymbol{x})\ f_{\boldsymbol{k}}(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{L^D}\sum_{j=1}^{J}\sum_{m=1}^{2^D}\frac{\alpha_j}{2^D}\exp\!\left(-i\frac{2\pi\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\mu}_j}{L}\right)\exp\!\left(-\frac{2\pi^2\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\Sigma}_j\boldsymbol{A}_m^\top\boldsymbol{k}}{L^2}\right) \\ &= \frac{1}{L^D}\sum_{j=1}^{J}\sum_{m=1}^{2^{D-1}}\frac{\alpha_j}{2^{D-1}}\cos\!\left(\frac{2\pi\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\mu}_j}{L}\right)\exp\!\left(-\frac{2\pi^2\boldsymbol{k}^\top\boldsymbol{A}_m\boldsymbol{\Sigma}_j\boldsymbol{A}_m^\top\boldsymbol{k}}{L^2}\right).\end{aligned} \tag{21}$$

With this mirroring, we can see that $\phi_{\boldsymbol{k}}$ are real and even, where an evaluation over $\boldsymbol{k}\in\mathcal{K}$, $j\in\{1,2,\dots,J\}$ and $m\in\{1,2,\dots,2^{D-1}\}$ in (21) is sufficient to fully characterize the signal.

Controller for a spatial distribution represented as a Gaussian mixture model

In [22], ergodic control is set as the constrained problem of computing a control command $\hat{\boldsymbol{u}}(t)$ at each time step t with

$$\hat{\boldsymbol{u}}(t) = \arg\min_{\boldsymbol{u}(t)}\ \epsilon\big(\boldsymbol{x}(t)+\Delta t\big),\quad \text{s.t.}\quad \dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t),\boldsymbol{u}(t)\big),\quad \|\boldsymbol{u}(t)\|\leqslant u^{\max}, \tag{22}$$

where the simple system $\dot{\boldsymbol{x}}(t)=\boldsymbol{u}(t)$ is considered (control with velocity commands), and where the error term is approximated with the Taylor series

$$\epsilon\big(\boldsymbol{x}(t)+\Delta t\big)\ \approx\ \epsilon\big(\boldsymbol{x}(t)\big)\ +\ \dot{\epsilon}\big(\boldsymbol{x}(t)\big)\Delta t\ +\ \frac{1}{2}\ddot{\epsilon}\big(\boldsymbol{x}(t)\big)\Delta t^2. \tag{23}$$

By using (13), (16), (18) and the chain rule $\frac{\partial f}{\partial t}=\frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial t}$, the Taylor series is composed of the control term $\boldsymbol{u}(t)$ and $\nabla_{\boldsymbol{x}} f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)\in\mathbb{R}^{1\times D}$, the gradient of $f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$ with respect to $\boldsymbol{x}(t)$. Solving the constrained objective in (22) then results in the analytical solution (see [22] for the complete derivation)

$$\begin{aligned}\boldsymbol{u} = \tilde{\boldsymbol{u}}(t)\frac{u^{\max}}{\|\tilde{\boldsymbol{u}}(t)\|},\quad \text{with}\quad \tilde{\boldsymbol{u}} &= -\sum_{\boldsymbol{k}\in\mathcal{K}}\Lambda_{\boldsymbol{k}}\Big(c_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)-\phi_{\boldsymbol{k}}\Big)\nabla_{\boldsymbol{x}} f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)^\top \\ &= -\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{f}\big(\boldsymbol{x}(t)\big)\,\boldsymbol{\Lambda}\Big(\boldsymbol{c}\big(\boldsymbol{x}(t)\big)-\boldsymbol{\phi}\Big),\end{aligned} \tag{24}$$

where $\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{f}\big(\boldsymbol{x}(t)\big)\in\mathbb{R}^{D\times K^D}$ is a concatenation of the vectors $\nabla_{\boldsymbol{x}} f_{\boldsymbol{k}}\big(\boldsymbol{x}(t)\big)$. Figure 5 shows a 2D example of ergodic control to create a motion approximating the distribution given by a mixture of two Gaussians. A remarkable characteristic of such approach is that the controller produces natural exploration behaviors (see Fig. 5-*(c)*) without relying on stochastic noise in the formulation. In the limit case, if the distribution $\phi(\boldsymbol{x})$ is a single Gaussian

with a very small isotropic covariance, the controller results in a standard tracking behavior.

Examples of application include surveillance with multi-agent systems [22], active shape estimation [2], and localization for fish-like robots [23]. A Matlab code example `demo_ergodicControl_2D01.m` can be found in [1].

3 Probabilistic movement primitives

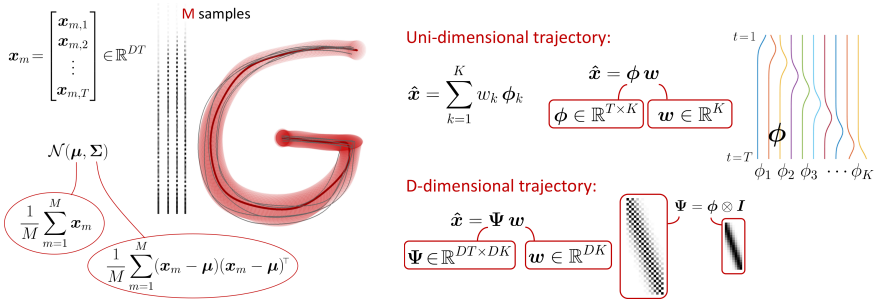

Fig. 6 *Left:* Raw trajectory distribution as a Gaussian of size DT by organizing each of the M samples as a trajectory vector, where each trajectory has T time steps and each point has D dimensions ($T = 100$ and $D = 2$ in this example). *Right:* Trajectory distribution encoded with probabilistic movement primitives (superposition of K basis functions). The right part of the figure depicts the linear mapping functions $\boldsymbol{\phi}$ and $\boldsymbol{\Psi}$ created by a decomposition with radial basis functions.

The representation of time series as a superposition of basis functions can also be exploited to construct trajectory distributions. Representing a collection of trajectories in the form of a multivariate distribution has several advantages. First, new trajectories can be stochastically generated. Then, the conditional probability property (see (7)) can be exploited to generate trajectories passing through via-points (including starting and/or ending points). This is simply achieved by specifying as inputs $\boldsymbol{x}^I$ in (7) the datapoints that the system needs to pass through (with corresponding dimensions in the hyperdimensional vector) and by retrieving as output $\boldsymbol{x}^O$ the remaining parts of the trajectory.

A naive approach to represent a collection of M trajectories in a probabilistic form is to reorganize each trajectory as a hyperdimensional datapoint $\boldsymbol{x}_m = [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \dots, \boldsymbol{x}_T^\top]^\top \in \mathbb{R}^{DT}$, and fitting a Gaussian $\mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{x}}, \boldsymbol{\Sigma}^{\boldsymbol{x}})$ to these datapoints, see Fig. 6-*left*. Since the dimension DT might be much larger than the number of datapoints M, a potential solution to this issue could be to consider an eigendecomposition of the covariance (ordered by decreasing eigenvalues)

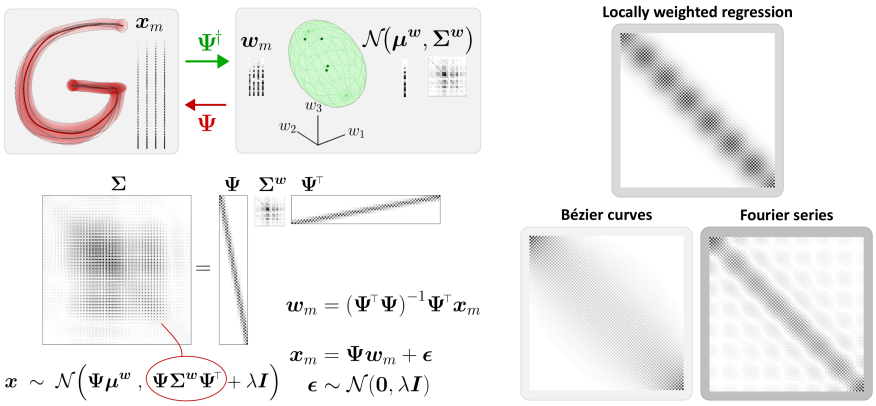

Fig. 7 *Left:* Illustration of probabilistic movement primitives as a linear mapping between the original space of trajectories and a subspace of reduced dimensionality. After projecting each trajectory sample in this subspace (with linear map $\boldsymbol{\Psi}^\dagger$ computed as the pseudoinverse of $\boldsymbol{\Psi}$), a Gaussian is evaluated, which is then projected back to the original trajectory space by exploiting the linear transformation property of multivariate Gaussians (with linear map $\boldsymbol{\Psi}$). Such decomposition results in a low rank structure of the covariance matrix, which is depicted in the bottom part of the figure. *Right:* Representation of the covariance matrix $\boldsymbol{\Psi}\boldsymbol{\Psi}^\top$ for various basis functions, all showing some form of sparsity.

$$\boldsymbol{\Sigma}^{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top = \sum_{j=1}^{DT} \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^\top, \tag{25}$$

with $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{DT}]$ and $\boldsymbol{D} = \text{diag}(\lambda_1^2, \lambda_2^2, \ldots, \lambda_{DT}^2)$. This can be exploited to project the data in a subspace of reduced dimensionality through principal component analysis. By keeping the first KT components, such approach provides a Gaussian distribution of the trajectories with the structure $\mathcal{N}(\boldsymbol{\Psi}\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Psi}\boldsymbol{\Psi}^\top)$, where $\boldsymbol{\Psi} = [\boldsymbol{v}_1\lambda_1, \boldsymbol{v}_2\lambda_2, \ldots, \boldsymbol{v}_{DK}\lambda_{DK}]$.

The ProMP (probabilistic movement primitive) model proposed in [25] also encodes the trajectory distribution in a subspace of reduced dimensionality, but provides a RBF structure to this decomposition instead of the eigendecomposition as in the above. It assumes that each sample trajectory $m \in \{1, \ldots, M\}$ can be approximated by a weighted sum of K normalized RBFs with

$$\boldsymbol{x}_m = \boldsymbol{\Psi}\boldsymbol{w}_m + \boldsymbol{\epsilon}, \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \lambda\boldsymbol{I}), \tag{26}$$

and basis functions organized as

$$\boldsymbol{\Psi} = \boldsymbol{\phi} \otimes \boldsymbol{I} = \begin{bmatrix} \boldsymbol{I}\phi_1(t_1) & \boldsymbol{I}\phi_2(t_1) & \cdots & \boldsymbol{I}\phi_K(t_1) \\ \boldsymbol{I}\phi_1(t_2) & \boldsymbol{I}\phi_2(t_2) & \cdots & \boldsymbol{I}\phi_K(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}\phi_1(t_T) & \boldsymbol{I}\phi_2(t_T) & \cdots & \boldsymbol{I}\phi_K(t_T) \end{bmatrix}, \tag{27}$$

with $\boldsymbol{\Psi} \in \mathbb{R}^{DT \times DK}$, identity matrix $\boldsymbol{I} \in \mathbb{R}^{D \times D}$, and $\otimes$ the Kronecker product operator. A vector $\boldsymbol{w}_m \in \mathbb{R}^{DK}$ can be estimated for each of the M sample trajectories by the least squares estimate

$$\boldsymbol{w}_m = (\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \boldsymbol{x}_m. \tag{28}$$

By assuming that $\{\boldsymbol{w}_m\}_{m=1}^M$ can be represented with a Gaussian $\mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Sigma}^{\boldsymbol{w}})$ characterized by a center $\boldsymbol{\mu}^{\boldsymbol{w}} \in \mathbb{R}^{DK}$ and a covariance $\boldsymbol{\Sigma}^{\boldsymbol{w}} \in \mathbb{R}^{DK \times DK}$, a trajectory distribution $\mathcal{P}(\boldsymbol{x})$ can then be computed as

$$\boldsymbol{x} \sim \mathcal{N}\Big(\boldsymbol{\Psi}\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Psi}\boldsymbol{\Sigma}^{\boldsymbol{w}}\boldsymbol{\Psi}^\top + \sigma^2 \boldsymbol{I}\Big), \tag{29}$$

with $\boldsymbol{x} \in \mathbb{R}^{DT}$ a trajectory of T datapoints of D dimensions organized in a vector form and $\boldsymbol{I} \in \mathbb{R}^{DT \times DT}$, see Figures 6 and 7.

The parameters of the ProMP model are σ^2, μ_k^I, Σ_k^I, $\boldsymbol{\mu}^{\boldsymbol{w}}$, and $\boldsymbol{\Sigma}^{\boldsymbol{w}}$. A Gaussian of DK dimensions is estimated, providing a compact representation of the movement, separating the temporal components $\boldsymbol{\Psi}$ and spatial components $\mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{w}}, \boldsymbol{\Sigma}^{\boldsymbol{w}})$. Similarly to LWR, ProMP can be coupled with GMM/GMR to automatically estimate the location and bandwidth of the basis functions as a joint distribution problem, instead of specifying them manually. A mixture of ProMPs can be efficiently estimated by fitting a GMM to the datapoints $\boldsymbol{w}_m$, and using the linear transformation property of Gaussians to convert this mixture into a mixture at the trajectory level. Moreover, such representation can be extended to other basis functions, including Bernstein and Fourier basis functions, see Fig. 7-*right*.

ProMP has been demonstrated in various robotic tasks requiring human-like motion capabilities such as playing the maracas and using a hockey stick [25], or for collaborative object handover and assistance in box assembly [21]. A Matlab code example `demo_proMP01.m` can be found in [1].

4 Further challenges and conclusion

This chapter presented various forms of superposition for time signals analysis and synthesis, by emphasizing the connections to Gaussian mixture models. The connections between these decomposition techniques are often underexploited, mainly due to the fact that these techniques were developed separately in various fields of research. The framework of mixture models provides a unified view that is inspirational to make links between these models. Such links also stimulate future developments and extensions.

Future challenges include a better exploitation of the joint roles that mixture of experts (MoE) and product of experts (PoE) can offer in the treatment of time series and control policies [26]. While MoE can decompose a complex signal by superposing a set of simpler signals, PoE can fuse information by

considering more elaborated forms of superposition (with full precision matrices instead of scalar weights). Often, either one or the other approach is considered in practice, but many applications would leverage the joint use of these two techniques.

There are also many further challenges specific to each basis function categories presented in this chapter. For Gaussian mixture regression (GMR), a relevant extension is to include a Bayesian perspective to the approach. This can take the form of a model selection problem, such as an automatic estimation of the number of Gaussians and rank of the covariance matrices [29]. This can also take the form of a more general Bayesian modeling perspective by considering the variations of the mixture model parameters (including means and covariances) [26]. Such extension brings new perspectives to GMR, by providing a representation that allows uncertainty quantification and multimodal conditional estimates to be considered. Other techniques like Gaussian processes also provide uncertainty quantification, but they are typically much slower. A Bayesian treatment of mixture model conditioning offers new perspectives for an efficient and robust treatment of wide-ranging data. Namely, models that can be trained with only few datapoints but that are rich enough to scale when more training data are available.

Another important challenge in GMR is to extend the techniques to more diverse forms of data. Such regression problem can be investigated from a geometrical perspective (e.g., by considering data lying on Riemannian manifolds [18]) or from a topological perspective (e.g., by considering relative distance space representations [17]). It can also be investigated from a structural perspective by exploiting tensor methods [20]. When data are organized in matrices or arrays of higher dimensions (tensors), classical regression methods first transform these data into vectors, therefore ignoring the underlying structure of the data and increasing the dimensionality of the problem. This flattening operation typically leads to overfitting when only few training data are available. Tensor representations instead exploit the intrinsic structure of multidimensional arrays. Mixtures of experts can be extended to tensorial representations for regression of tensor-valued data [19], which could potentially be employed to extend GMR representations to arrays of higher dimensions.

Regarding Bézier curves, even if the technique is well established, there is still room for further perspectives, in particular with the links to other techniques that such approach has to offer. For example, Bézier curves can be reframed as a model predictive control (MPC) problem [10, 6], a widespread optimal control technique used to generate movements with the capability of anticipating future events. Formulating Bézier curves as a superposition of Bernstein polynomials also leaves space for probabilistic interpretations, including Bayesian treatments.

The consideration of Fourier series for the superposition of basis functions might be the approach with the widest range of possible developments. Indeed, the representation of continuous time signals in the frequency domain

is omnipresent in many fields of research, and, as exemplified with ergodic control, there are many opportunities to exploit the Gaussian properties in mixture models by taking into account their dual representation in spatial and frequency domains.

With the specific application of ergodic control, the dimensionality issue requires further consideration. In the basic formulation, by keeping K basis functions to encode time series composed of datapoints of dimension D, K^D Fourier series components are required. Such formulation has the advantage of taking into account all possible correlations across dimensions, but it slows down the process when D is large. A potential direction to cope with such scaling issue would be to rely on Gaussian mixture models (GMMs) with low-rank structures on the covariances [29], such as in mixtures of factor analyzers (MFA) or mixtures of probabilistic principal component analyzers (MPPCA) [7]. Such subspaces of reduced dimensionality could potentially be exploited to reduce the number of Fourier basis coefficients to be computed.

Finally, the probabilistic representation of movements primitives in the form of trajectory distributions also offers a wide range of new perspectives. Such models classically employ radial basis functions, but can be extended to a richer family of basis functions (including a combination of those). This was exemplified in the chapter with the use of Bernstein and Fourier bases to build probabilistic movement primitives, see Fig. 7-*right*. More generally, links to kernel methods can be created by extension of this representation [15]. Other extensions include the use of mixture models and associated Bayesian methods to encode the weights $\boldsymbol{w}_m$ in the subspace of reduced dimensionality.

Acknowledgements I would like to thank Prof. Michael Liebling for his help in the development of the ergodic control formulation applied to Gaussian mixture models and for his recommendations on the preliminary version of this chapter.
The research leading to these results has received funding from the European Commission's Horizon 2020 Programme (H2020/2018-20) under the MEMMO Project (Memory of Motion, http://www.memmo-project.eu/), grant agreement 780684.

References

[1] (Accessed: 2019/04/18) PbDlib robot programming by demonstration software library. `http://www.idiap.ch/software/pbdlib/`

[2] Abraham I, Prabhakar A, Hartmann MJZ, Murphey TD (2017) Ergodic exploration using binary sensing for nonparametric shape estimation. IEEE Robotics and Automation Letters 2(2):827–834

[3] Antonsson EK, Mann RW (1985) The frequency content of gait. Journal of Biomechanics 18(1):39–47

[4] Atkeson CG (1989) Using local models to control movement. In: Advances in Neural Information Processing Systems (NIPS), vol 2, pp 316–323

[5] Atkeson CG, Moore AW, Schaal S (1997) Locally weighted learning for control. Artificial Intelligence Review 11(1-5):75–113
[6] Berio D, Calinon S, Fol Leymarie F (2017) Generating calligraphic trajectories with model predictive control. In: Proc. 43rd Conf. on Graphics Interface, Edmonton, AL, Canada, pp 132–139
[7] Bouveyron C, Brunet C (2014) Model-based clustering of high-dimensional data: A review. Computational Statistics and Data Analysis 71:52–78
[8] Calinon S, Lee D (2019) Learning control. In: Vadakkepat P, Goswami A (eds) Humanoid Robotics: a Reference, Springer, pp 1261–1312, (in press)
[9] Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. American Statistical Association 74(368):829–836
[10] Egerstedt M, Martin C (2010) Control Theoretic Splines: Optimal Control, Statistics, and Path Planning. Princeton University Press
[11] Falk TH, Shatkay H, C WY (2006) Breast cancer prognosis via Gaussian mixture regression. In: Conference on Electrical and Computer Engineering, pp 987–990
[12] Farouki RT (2012) The Bernstein polynomial basis: A centennial retrospective. Computer Aided Geometric Design 29(6):379–419
[13] Ghahramani Z, Jordan MI (1994) Supervised learning from incomplete data via an EM approach. In: Cowan JD, Tesauro G, Alspector J (eds) Advances in Neural Information Processing Systems (NIPS), Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, vol 6, pp 120–127
[14] Hersch M, Guenter F, Calinon S, Billard AG (2008) Dynamical system modulation for robot learning via kinesthetic demonstrations. IEEE Trans on Robotics 24(6):1463–1467
[15] Huang Y, Rozo L, Silvério J, Caldwell DG (2019) Kernelized movement primitives. International Journal of Robotics Research (IJRR) (in press)
[16] Hueber T, Bailly G (2016) Statistical conversion of silent articulation into audible speech using full-covariance HMM. Comput Speech Lang 36(C):274–293
[17] Ivan V, Zarubin D, Toussaint M, Komura T, Vijayakumar S (2013) Topology-based representations for motion planning and generalization in dynamic environments with interactions. Intl Journal of Robotics Research 32(9-10):1151–1163
[18] Jaquier N, Calinon S (2017) Gaussian mixture regression on symmetric positive definite matrices manifolds: Application to wrist motion estimation with sEMG. In: Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS), Vancouver, Canada, pp 59–64
[19] Jaquier N, Haschke R, Calinon S (2019) Tensor-variate mixture of experts. arXiv:190211104 pp 1–11
[20] Kolda T, Bader B (2009) Tensor decompositions and applications. SIAM Review 51(3):455–500

[21] Maeda GJ, Neumann G, Ewerton M, Lioutikov R, Kroemer O, Peters J (2017) Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks. Autonomous Robots 41(3):593–612
[22] Mathew G, Mezic I (2011) Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. Physica D: Nonlinear Phenomena 240(4):432–442
[23] Miller LM, Silverman Y, MacIver MA, Murphey TD (2016) Ergodic exploration of distributed information. IEEE Trans on Robotics 32(1):36–52
[24] Mussa-Ivaldi FA, Giszter SF, Bizzi E (1994) Linear combinations of primitives in vertebrate motor control. Proc National Academy of Sciences 91:7534–7538
[25] Paraschos A, Daniel C, Peters J, Neumann G (2013) Probabilistic movement primitives. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc., USA, pp 2616–2624
[26] Pignat E, Calinon S (2019) Bayesian gaussian mixture model for robotic policy imitation. arXiv:190410716 pp 1–7
[27] Schaal S, Atkeson CG (1998) Constructive incremental learning from only local information. Neural Computation 10(8):2047–2084
[28] Stulp F, Sigaud O (2015) Many regression algorithms, one unified model — a review. Neural Networks 69:60–79
[29] Tanwani AK, Calinon S (2019) Small variance asymptotics for non-parametric online robot learning. International Journal of Robotics Research (IJRR) 38(1):3–22
[30] Tian Y, Sigal L, De la Torre F, Jia Y (2013) Canonical locality preserving latent variable model for discriminative pose inference. Image and Vision Computing 31(3):223–230
[31] Ting J, Kalakrishnan M, Vijayakumar S, Schaal S (2008) Bayesian kernel shaping for learning control. In: Advances in Neural Information Processing Systems (NIPS), pp 1673–1680
[32] Toda T, Black AW, Tokuda K (2007) Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Transactions on Audio, Speech, and Language Processing 15(8):2222–2235
[33] Vijayakumar S, D'souza A, Schaal S (2005) Incremental online learning in high dimensions. Neural Computation 17(12):2602–2634