Kinetic Euclidean Distance Matrices

Puoya Tabaghi, Ivan Dokmanić, and Martin Vetterli, *Fellow, IEEE*

Abstract—**Euclidean distance matrices (EDMs) are a major tool for localization from distances, with applications ranging from protein structure determination to global positioning and manifold learning. They are, however, static objects which serve to localize points from a snapshot of distances. If the objects move, one expects to do better by modeling the motion. In this paper, we introduce Kinetic Euclidean Distance Matrices (KEDMs)—a new kind of time-dependent distance matrices that incorporate motion. The entries of KEDMs become functions of time, the squared time-varying distances. We study two smooth trajectory models—polynomial and bandlimited trajectories—and show that these trajectories can be reconstructed from incomplete, noisy distance observations, scattered over multiple time instants. Our main contribution is a semidefinite relaxation, inspired by similar strategies for static EDMs. Similarly to the static case, the relaxation is followed by a spectral factorization step; however, because spectral factorization of polynomial matrices is more challenging than for constant matrices, we propose a new factorization method that uses anchor measurements. Extensive numerical experiments show that KEDMs and the new semidefinite relaxation accurately reconstruct trajectories from noisy, incomplete distance data and that, in fact, motion improves rather than degrades localization if properly modeled. This makes KEDMs a promising tool for problems in geometry of dynamic points sets.**

Index Terms—**Euclidean distance matrix, positive semidefinite programming, polynomial matrix factorization, trajectory, localization, spectral factorization.**

I. Introduction

THE famous distance geometry problem (DGP) [1] asks to reconstruct the geometry of a point set from a subset of interpoint distances. It models a wide gamut of practical problems, from sensor network localization [2]–[4] and microphone positioning [5]–[8] to clock synchronization [9], [10], to molecular geometry reconstruction from NMR data [11], [12]. Among the most successful vehicles for the design of DGP algorithms are the Euclidean distance matrices (EDM) [13].

EDMs model static objects. When things move, they characterize a snapshot of the interpoint distances and the point set geometry. It seems intuitive that with a good model for the trajectories one should be able to leverage the motion and improve trajectory estimation.

In this work we make a first step towards distance matrices for moving points, which we call Kinetic EDMs (KEDMs) inspired by the notion of kinetic data structures [14] for moving points. KEDMs are a generalization of EDMs whose entries now become functions of time. We show how by using KEDMs we can neatly address the kinetic distance geometry problem (KDGP), a natural generalization of the classical, static distance geometry problem (DGP) defined in Section II. Unlike with the static DGP, in order to make the kinetic version well posed, we must constrain the point trajectories to belong to a class of functions, for example polynomial trajectories of a bounded degree. Informally, we ask the following question: suppose a set of points move according to a known trajectory model. At given time instants we measure a subset of pairwise distances; the subset can change between measurements and it may be too small to allow localization at any time alone. Can we systematically localize the points and reconstruct trajectories by exploiting the trajectory model?

Localization of dynamic point sets from distance measurements finds applications whenever objects move. Robot swarms, for example, often must localize autonomously [15], especially in remote situations such as extraterrestrial exploration [16] or deep-water missions [17]. Related applications exist in environmental monitoring, for example for dynamic sensor networks composed of river-borne sensing nodes [18]. An important application of localization of moving objects is in global positioning with satellites where both the satellites and the users move. Applications are emerging where sensing is opportunistic and the positions of reference objects are not known [19]; in Section VI-D, we present a simulated example of global positioning with unknown satellite trajectories. This problem is further related to simultaneous localization and mapping (SLAM) [20], [21]. Kinetic distance geometry problems are common in computer vision. Examples are action recognition from dynamic interjoint distance skeleton data [22] and more generally data structures for describing kinetic point sets [14]. Applications in multi-robot coordination, crowd simulations, and motion retargeting are explored in [23], [24], where the authors introduce the *dynamical distance geometry problem* (dynDGP).[1] Even in applications to proteins and molecules, the atoms move (for example, proteins fold) in specific ways [25].

The study of distance geometry and EDMs goes back to the work of Menger [26], Schoenberg [27], Blumenthal [28], and

[1]Though related, the dynDGP is rather different from our KDGP.

Young and Householder [29]. Gower derived numerous results on EDMs [30], [31] including a complete rank characterization [31]. An extensive treatise on EDMs with many original results and an elegant characterization of the EDM cone was written by Dattorro [32]; [13] is a tutorial-style introduction to EDMs.

A large class of approaches to point set localization from distance measurements rely on semidefinite programming [33], [34]. Namely, the localization problem is written in terms of the Gram matrix of the point set which leads to a rank-constrained semidefinite program. The rank constraint is often relaxed to arrive at a semidefinite relaxation which is a convex optimization problem and can be solved using standard tools.

We take inspiration from these approaches and show how trajectory localization can also be formulated as a semidefinite program, thus answering the above question in the affirmative. Concretely, we show that the parameters of chosen trajectory models can be recovered by a semidefinite program and a tailor-made alignment procedure akin to Procrustes analysis. The latter can be interpreted as spectral factorization of semidefinite polynomial matrices with side information, and our developments rely on the related spectral factorization results [35].

As we will show, a major difference from the static case is that the (time-varying) distance information does not specify the point set up to a rigid transformation. The larger class of ambiguities can nevertheless be resolved through the use of anchor points.

We show through extensive computational experiments that our semidefinite relaxation indeed works as expected and that with an appropriate trajectory model we can reduce the number of measurements *per time instant* well below that minimally required for localization in the static case.

A. Paper Outline

In Section II we extend the definition of the distance geometry problem (DGP) to its kinetic version and review the essential facts about Euclidean distance matrices and associated semidefinite programs. Next, in Section III, we introduce the two trajectory models we will use throughout the paper—polynomial and bandlimited—and show how to write the corresponding polynomial Gram matrices in a form suitable for semidefinite programming. We use this form (given in terms of the so-called basis Gramians) to formulate a semidefinite relaxation in Section IV. The solution to the SDP, however, only gives us *distance* trajectories. To convert them to point trajectories, we need known and new results on spectral factorization of polynomial matrices developed in Section V. Finally, we provide simulation results and list several promising directions for future work.

B. Reproducible Research

We adhere to the philosophy of reproducible research: documented code and data to reproduce all experiments is available online.[2]

II. STATIC AND KINETIC DISTANCE GEOMETRY PROBLEMS

We begin by introducing the classical distance geometry problem (DGP) and then formulate its generalization to moving points. We also discuss an EDM-based approach to the DGP with noisy and incomplete distances.

The DGP can be informally stated as follows: find the d-dimensional locations $\{\boldsymbol{x}_n \in \mathbb{R}^d\}_{n=1}^N$ of a set of points, given a subset of possibly noisy pairwise distances $\{d_{mn} : 1 \leq m < n \leq N\}$. We will work only with Euclidean distances so that $d_{mn} = \|\boldsymbol{x}_m - \boldsymbol{x}_n\|$.

An elegant formalization can be made in graph-theoretic terms. Consider a graph $G = (V, E)$ whose vertex set V corresponds to the points $\{\boldsymbol{x}_n\}_{n=1}^N$. The edge set E tells us which distances are measured and which are not. Given two vertices $u, v \in V$ and the corresponding undirected edge $e = \{u, v\}$, we have $e \in E$ if and only if the distance between u and v is known. Let $f : E \to \mathbb{R}^+$ be the weight function that assigns those known, measured distances to edges. Then we can pose the following problem [1]:

Problem 1 (Distance Geometry Problem): Given an integer $d > 0$ (the ambient dimension) and an undirected graph $G = (V, E)$ whose edges are weighted by a non-negative function $f : E \to \mathbb{R}^+$ (distance), determine whether there is a function $x : V \to \mathbb{R}^d$ such that

$$\text{for all } \{u, v\} \in E \text{ we have } \|x(u) - x(v)\| = f(\{u, v\}).$$

The function x which assigns coordinates to vertices is called an embedding or a realization of the graph in $\mathbb{R}^d$. Of course, in practice the measurements are corrupted by measurement errors, and the goal is to minimize some notion of discrepancy between the measured distances and the distances induced by our estimate; for example:

$$\underset{x:V \to \mathbb{R}^d}{\text{minimize}} \sum_{\{u,v\} \in E} \left(\|x(u) - x(v)\| - f(\{u, v\}) \right)^2. \quad (1)$$

Section II-A explains how to use EDMs to proceed in this case. Figure 1 illustrates the DGP with an intermediate step of constructing an EDM. The EDM can be interpreted as a weighted adjacency matrix with weights being squared distances.

In this paper, we address distance geometry problems when the points move and the set of measured distance changes over time. Instead of localizing the points only at the measurement times, our goal will be to estimate entire trajectories for all times. To make this problem well posed we must introduce a class of admissible continuous trajectories $\mathcal{X}$. Then, we can formulate the following kinetic version of Problem 1:

Problem 2 (Kinetic Distance Geometry Problem): Given an embedding dimension $d > 0$, a set of T sampling times $\mathcal{T} = \{t_1, \ldots, t_T\} \subset \mathbb{R}$, and a sequence of undirected graphs $G_i = (V, E_i)$ whose edges are weighted by non-negative functions $f_i : E_i \to \mathbb{R}^+$, for $i \in \{1, \ldots, T\}$, determine whether there is a function $x : V \times \mathbb{R} \to \mathbb{R}^d$, where $x \in \mathcal{X}$ and for all $t_i \in \mathcal{T}$ we have:

$$\|x(u, t_i) - x(v, t_i)\| = f_i(\{u, v\}) \text{ for all } \{u, v\} \in E_i,$$

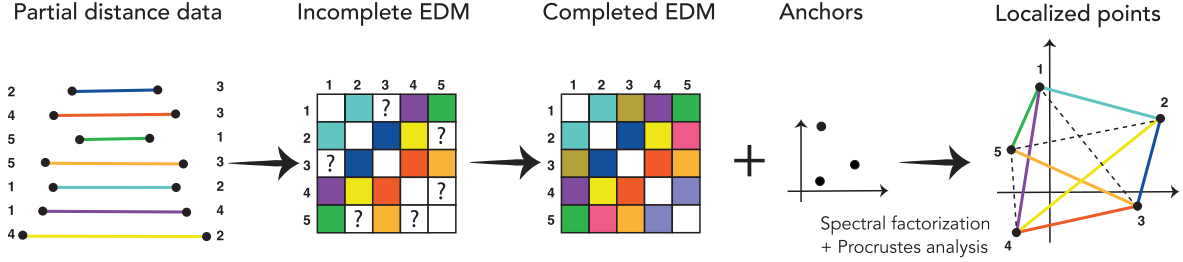where $\mathcal{X}$ is the set of admissible trajectories.

Fig. 1.   The objective of DGP is to find an embedding for a given partial pair-wise distance data. This can be done in two steps: a) Completing EDM associated with the measurements, i.e. estimating the missing measurements and b) Estimate an embedding and using anchor points to resolve the rigid transformation ambiguity, discussed in Section II-A.
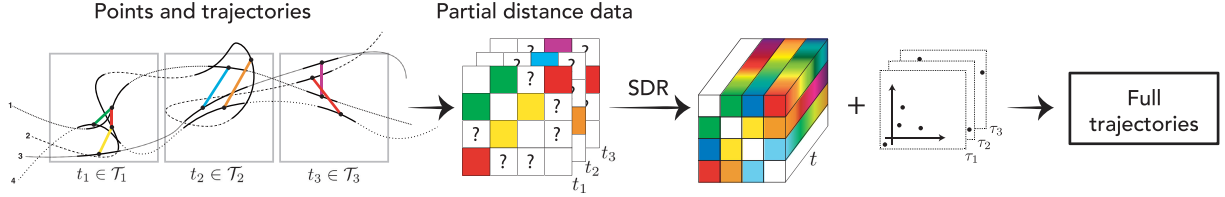

Fig. 2.   KDGP: Estimate an embedded trajectory for a given sequence of partial pair-wise distances at different times, $t_1, \ldots, t_T$. We estimate the corresponding KEDM with a semidefinite relaxation Algorithm 1, and then use anchors to estimate the trajectories.

Figure 2 illustrates the KDGP for four trajectories. One way to interpret KDGP is as a sequence of static DGPs with additional information about sampling times and trajectory model. Indeed, the KDGP can be seen a a nonlinear spatio–temporal sampling problem, with the nonlinear samples (distances) spread in space and time. A natural question is whether we can compensate for a reduction in the number of spatial samples by oversampling in time. We answer this question in affirmative in Section VI.

The first step is to estimate the continuous KEDM from samples distributed in space and time; this is discussed in Section IV. The second step is to use information about the absolute positions of a set of anchor points in order to assign absolute locations to trajectories; this is discussed in Section V. This step is more challenging than for the usual EDMs.

A. Solving the Distance Geometry Problem by EDMs

It is useful to recall the EDM-based approach to the DGP. Ascribe the coordinates of N points in a d-dimensional space to the columns of matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$. The squared distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is

$$d_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \|\boldsymbol{x}_i\|^2 - 2\boldsymbol{x}_i^\top \boldsymbol{x}_j + \|\boldsymbol{x}_j\|^2 ,$$

from which we can read out the equation for the EDM $\boldsymbol{D} = (d_{ij}^2)$ as

$$\boldsymbol{D} = \mathcal{K}(\boldsymbol{G}) \stackrel{\text{def}}{=} \operatorname{diag}(\boldsymbol{G})\boldsymbol{1}^\top - 2\boldsymbol{G} + \boldsymbol{1}\operatorname{diag}(\boldsymbol{G})^\top, \quad (2)$$

where $\boldsymbol{1}$ denotes the column vector of all ones, $\boldsymbol{G}$ is the Gram matrix $\boldsymbol{G} = \boldsymbol{X}^\top \boldsymbol{X}$, and $\operatorname{diag}(\boldsymbol{G})$ is a column vector of the diagonal entries of $\boldsymbol{G}$. We see that the EDM of a point set is a linear function of its Gram matrix. Reformulating the problem in terms of the Gram matrix is beneficial because it will lead to a semidefinite program. If we can find the Gram matrix, the point set can be obtained by an eigenvalue decomposition.

To see how, let $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ with all eigenvalues λ_i non-negative, and $\boldsymbol{U}$ orthonormal, as $\boldsymbol{G}$ is a symmetric positive semidefinite matrix. Assume that the eigenvalues are sorted in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. Then we can estimate the point set as $\widehat{\boldsymbol{X}} \stackrel{\text{def}}{=} \left[\operatorname{diag}\left(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_d}\right), \boldsymbol{0}_{d \times (N-d)}\right]\boldsymbol{U}^\top$. Since the EDM only specifies the points up to a rigid transform, $\widehat{\boldsymbol{X}}$ will be a rotated, reflected and translated version of $\boldsymbol{X}$.

One way to estimate $\boldsymbol{D}$ from noisy, incomplete distance data is by semidefinite programming. This hinges on the one-to-one equivalence between EDMs with embedding dimension d and centered Gram matrices of rank d. Define the geometric centering matrix $\boldsymbol{J}$ as

$$\boldsymbol{J}_N \stackrel{\text{def}}{=} \boldsymbol{I} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^\top.$$

Then $\mathcal{K}(\boldsymbol{G})$ is an invertible map on the set of Gram matrices which correspond to centered point sets (implying $\boldsymbol{G}\boldsymbol{1} = \boldsymbol{0}$) with the inverse given by

$$-\frac{1}{2}\boldsymbol{J}_N \mathcal{K}(\boldsymbol{G})\boldsymbol{J}_N = \boldsymbol{G}.$$

In particular, we have the following equivalence that holds for matrices $\boldsymbol{D}$ with a zero diagonal:

$$\left.\begin{array}{r} \boldsymbol{D} = \mathcal{D}(\boldsymbol{X}) \\ \operatorname{affdim}(\boldsymbol{X}) \leq d \end{array}\right\} \iff \begin{cases} -\frac{1}{2}\boldsymbol{J}_N \boldsymbol{D}\boldsymbol{J}_N \succeq 0 \\ \operatorname{rank}(\boldsymbol{J}_N \boldsymbol{D}\boldsymbol{J}_N) \leq d. \end{cases} \quad (3)$$

where $\mathcal{D}(\boldsymbol{X}) = \mathcal{K}(\boldsymbol{X}^\top \boldsymbol{X})$ and affdim denotes the dimension of the smallest affine space that can hold $\boldsymbol{X}$. In other words, instead of directly searching for the points $\boldsymbol{X}$ given distance data, we can search for the suitable Gram matrix.

Let $\widetilde{\boldsymbol{D}}$ be the noisy, incomplete EDM from which we want to estimate the point locations, with unknown entries replaced by

zeroes. Define the mask matrix $W = (w_{ij})$ as

$$w_{ij} \overset{\text{def}}{=} \begin{cases} 1, & (i,j) \in E \\ 0, & \text{otherwise} \end{cases}.$$

This mask matrix will let us compute the loss only on those entries that were actually measured. Note that W is precisely the adjacency matrix of the undirected graph from Problem 1.

Then the above discussion is summarized in the following rank-constrained semidefinite program:

$$\begin{aligned} \underset{G}{\text{minimize}} \quad & \|\widetilde{D} - W \circ \mathcal{K}(G)\|_F^2 \\ \text{subject to} \quad & G \succeq 0 \\ & G\mathbf{1} = \mathbf{0} \\ & \text{rank}(G) \leq d. \end{aligned} \tag{4}$$

Since the Gram matrix (Gramian) is linearly related to the EDM, the objective function is convex. However, the rank constraint, $\text{rank}(G) \leq d$, makes the feasible set in (4) non-convex. Note that (1) is also a non-convex program.

The value of reformulation in terms of G is that it admits a simple convexification strategy—a semidefinite relaxation—which was repeatedly shown to perform well in the context of distance geometry [13], [33]. An intuitive explanation is that while the rank condition ensures the correct embedding dimension, the constraint that G be positive semidefinite (in other words, that it be a Gramian) enforces a number of geometric constraints. For instance, it ensures that the entries of the EDM verify triangle inequalities, as well as other subtle properties (see, for example, the Cayley-Menger conditions [36]).

We should add that a semidefinite relaxation is by no means the only way to convexify (1) or (4). The mathematical optimization literature knows a number of others, many of which could also be applied in the X-domain (1). One may, for example, replace non-linear terms in (4) by suitable convex over- and under-estimators. A well-known example is McCormick's convexification for bilinear terms [37], and similar strategies for quadratic and higher-order terms [38]. In this paper we limit ourselves to semidefinite relaxation.

Once the rank constraint is discarded, the embedding dimension of the reconstructed point set is dictated by the measurements. One often looks for a solution with the lowest possible embedding dimension via various rank-minimization heuristics [13]. In general, especially with noisy measurements, we can at best hope that the reconstructed points will lie close to a linear variety of the desired dimension.[3] To ensure the right embedding dimension, a suboptimal solution can be computed by replacing the estimated Gramian with its best rank-d approximation; see Section V for the kinetic case.

The constraint $G\mathbf{1} = \mathbf{0}$ serves to set the centroid of the recovered point set at the origin of the coordinate system as it implies $X\mathbf{1} = \mathbf{0}$. This resolves the translational invariance of the problem. The remaining rotational (and reflection) invariance

---

[3]An empirical study of the number of required measurements is available in [13]

must be resolved once the points are estimated from the Gramian. The Gramian itself is of course invariant to the rotations of the point set since $G = X^\top X = (UX)^\top UX$ for any orthonormal matrix $U \in \mathbb{R}^{d \times d}$.

B. Orthogonal Procrustes Problem

As mentioned before, the EDM only specifies the point set up to a rigid transformation (rotation, translation, and reflection). If the task requires determining absolute positions of points, the standard method is to designate a subset of points as *anchors* whose positions are known, and use anchors to align the reconstructed point set.

Let $X_a \in \mathbb{R}^{d \times N_a}$ be the submatrix (a selection of columns) of X that should be aligned with the anchors listed as columns of $Y \in \mathbb{R}^{d \times N_a}$. We note that the number of anchors—columns in X_a—is typically small compared with the total number of points—columns in X.

We first center the columns of Y and X_a by subtracting the corresponding column means y_c and $x_{a,c}$, obtaining matrices $\overline{Y}$ and $\overline{X}_a$. Next, we perform the orthogonal Procrustes analysis—we search for the rotation and reflection that best maps $\overline{X}_a$ onto $\overline{Y}$:

$$R = \underset{Q: QQ^\top = I}{\arg\min} \|Q\overline{X}_a - \overline{Y}\|_F^2. \tag{5}$$

The solution to (5) is given by the singular value decomposition (SVD) [39] as follows. Let $U \Sigma V^\top$ be the SVD of $\overline{X}_a \overline{Y}^\top$; then $R = VU^\top$. The best alignement is applied to the reconstructed point set as

$$X_{\text{aligned}} = R(X - x_{a,c}\mathbf{1}^\top) + y_c\mathbf{1}^\top.$$

III. TRAJECTORY MODELS AND BASIS GRAMIANS

In order to extend the EDM-based tools to the KDGP, we must define the class of trajectories $\mathcal{X}$. We introduce two trajectory models—polynomial and bandlimited—and show how they can be parameterized in terms of the so-called basis Gramians.

The chosen trajectory models are standard; they model many interesting trajectories, as shown in Figure 3. The polynomial model is common in simultaneous localization and mapping as well as tracking, where it appears as constant velocity or constant acceleration model [40], [41]. The bandlimited model describes periodic trajectories of varying degrees of smoothness which are locally well-approximated by polynomials.

We use similar notation as in the static case. Let $X(t) = [x_1(t), \ldots, x_N(t)]$ be the trajectory matrix of N points in $\mathbb{R}^d$ where $x_n(t)$ is the position of n-th point at time t. We define the KEDM in a natural way:

Definition 1 (KEDM): Given a set of trajectories $X(t) \in \mathbb{R}^{d \times N}[t]$, the corresponding KEDM is the time-dependent matrix $D(t) \in \mathbb{R}^{N \times N}[t]$ of time-varying squared distances between the points:

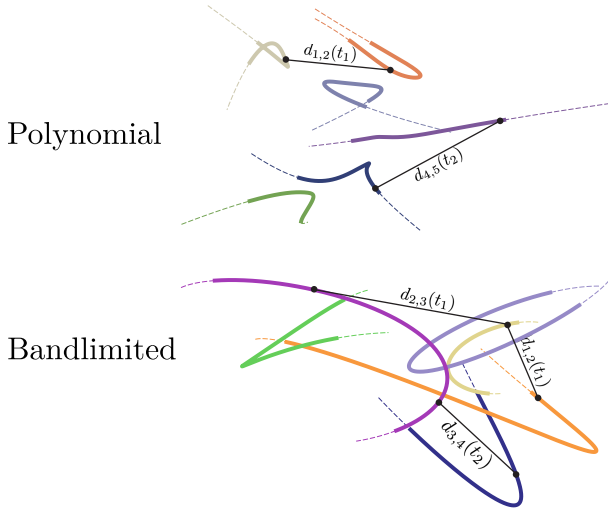$$D(t) \overset{\text{def}}{=} \mathcal{D}(X(t)). \tag{6}$$

Fig. 3. Example polynomial ($P = 5$) and bandlimited ($P = 3, \omega = \pi/4$) trajectories. Solid parts correspond to the sampling window $[T_1, T_2] = [-0.8, 0.8]$. The goal is to retrieve the trajectories of all points by measuring $d_{i,j}(t) = \|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|_2$ at different times and for different (i,j)-pairs.

A. Polynomial Trajectories

For a set of N points in $\mathbb{R}^d$, we define the set of polynomial trajectories of degree P as

$$\mathcal{X}_{\text{poly}} = \left\{ \sum_{p=0}^{P} t^p \boldsymbol{A}_p \ \middle| \ \boldsymbol{A}_p \in \mathbb{R}^{d \times N}, \ p \in \{0, \dots, P\} \right\}. \quad (7)$$

For $\boldsymbol{X}(t) \in \mathcal{X}_{\text{poly}}$, the Gramian at time t can be written as

$$\boldsymbol{G}(t) = \sum_{k=0}^{K} \boldsymbol{B}_k t^k, \quad (8)$$

where $\boldsymbol{B}_k = \sum_{i=\max\{0, k-p\}}^{\min\{k,p\}} \boldsymbol{A}_i^\top \boldsymbol{A}_{k-i}$ and $K = 2P$.[4] Similar to the static case, our goal is to cast the trajectory retrieval problem as a semidefinite program; we do so via the time-dependent Gramian in Section IV.

The key step is to parameterize the problem entirely in terms of (constant) positive semidefinite matrices, instead of the parameterization in terms of $\boldsymbol{A}_p$ or $\boldsymbol{B}_k$. To do so, we fix $K+1$ time instants $\tau_0, \dots, \tau_K$ and define $\boldsymbol{G}_k \overset{\text{def}}{=} \boldsymbol{G}(\tau_k)$. The matrices $\boldsymbol{G}_k$ should be interpreted as *elementary*, or *basis* Gramians in the sense that the Gramian $\boldsymbol{G}(t)$ can be written as a linear combination of $\boldsymbol{G}_0, \dots, \boldsymbol{G}_K$ as elaborated in the following proposition.

Proposition 1: Consider the polynomial trajectory in (7). Let $\boldsymbol{G}_k, \ k \in \{0, 1, \dots, K\}, \ K = 2P$ be given as above with τ_k all distinct. Then we have

$$\boldsymbol{G}(t) = \sum_{k=0}^{K} w_k(t) \boldsymbol{G}_k, \quad$$

[4]Simplified from $\boldsymbol{G}(t) = \left(\sum_{p=0}^{P} t^p \boldsymbol{A}_p \right)^\top \left(\sum_{p=0}^{P} t^p \boldsymbol{A}_p \right)$.

with the weights $\boldsymbol{w}(t) = [w_0(t), \dots, w_K(t)]^\top$ given as

$$\boldsymbol{w}(t) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \tau_0^K & \tau_1^K & \cdots & \tau_K^K \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ t \\ \vdots \\ t^K \end{pmatrix}.$$

Proof: The Gramians can be written as linear combinations of a set of monomial terms (cf. (8)), which gives

$$\boldsymbol{G}_0 = \boldsymbol{B}_0 + \tau_0 \boldsymbol{B}_1 + \cdots + \tau_0^K \boldsymbol{B}_K$$

$$\vdots \quad (9)$$

$$\boldsymbol{G}_K = \boldsymbol{B}_0 + \tau_K \boldsymbol{B}_1 + \cdots + \tau_K^K \boldsymbol{B}_K.$$

Each matrix equation in (9) consists of $N \times N$ scalar equations for entries of $\boldsymbol{G}_k$. Focusing on a particular entry (i, j) gives a linear system $\boldsymbol{g} = \boldsymbol{M} \boldsymbol{b}$ with column vector $\boldsymbol{g} = [g_0, \dots, g_K]^\top$ where g_k is (i, j)-th element of $\boldsymbol{G}_k$, the matrix $\boldsymbol{M} \overset{\text{def}}{=} [\tau_k^{k'}]_{0 \le k, k' \le K}$, and $\boldsymbol{b} = [b_0, \dots, b_K]^\top$ where b_k is (i, j)-th element of $\boldsymbol{B}_k$. We also have from (8) that $[\boldsymbol{G}(t)]_{ij} = (1, t, t^2, \dots, t^K) \boldsymbol{b} \overset{\text{def}}{=} \boldsymbol{t}^\top \boldsymbol{b}$. Since τ_k are distinct, the square Vandermonde matrix $\boldsymbol{M}$ is invertible. We have $\boldsymbol{b} = \boldsymbol{M}^{-1} \boldsymbol{g}$, which gives $[\boldsymbol{G}(t)]_{ij} = \boldsymbol{t}^\top \boldsymbol{M}^{-1} \boldsymbol{g}$. Denoting $\boldsymbol{w}(t) = (\boldsymbol{M}^\top)^{-1} \boldsymbol{t}$ we have that $[\boldsymbol{G}(t)]_{ij} = \boldsymbol{w}(t)^\top \boldsymbol{g} = \sum_{k=0}^{K} w_k(t) [\boldsymbol{G}_k]_{ij}$ which proves the claim. ∎

This result is a matrix version of Lagrange interpolation. Since entries of $\boldsymbol{G}(t)$ are polynomials of degree $2P$ in t, they are completely determined by their values at $2P + 1$ points. However, in this Gram matrix version it gives us something rather useful: a way to write a positive semidefinite $\boldsymbol{G}(t)$ as a linear combination of positive semidefinite $\boldsymbol{G}_k$, which lends itself nicely to convex optimization. We note that the question of how to choose the sampling times τ_k is beyond the scope of this article, though we give empirical results in Section VI.

B. Bandlimited Trajectories

The second model we consider are periodic bandlimited trajectories. For a set of N points in $\mathbb{R}^d$, the set of periodic bandlimited trajectory of degree P can be written as

$$\mathcal{X}_{\text{BL}} = \left\{ \boldsymbol{B}_0 + \sum_{p=1}^{P} \{\boldsymbol{A}_p \sin(p\omega t) + \boldsymbol{B}_p \cos(p\omega t)\} \ \middle| \right.$$

$$\left. \boldsymbol{A}_p, \boldsymbol{B}_0, \boldsymbol{B}_p \in \mathbb{R}^{d \times N}, p \in \{1, \dots, P\}, \omega \in \mathbb{R}^+ \right\}. \quad (10)$$

Similar to the polynomial case, we represent the Gramian $\boldsymbol{G}(t)$ as a linear combination of some Gramian basis.

Proposition 2: Consider the bandlimited trajectory in (10). Let $\boldsymbol{G}_k, \ k \in \{0, 1, \dots, K\}, \ K = 4P$ be given as above with τ_k all distinct (modulo $\frac{2\pi}{\omega}$). We have

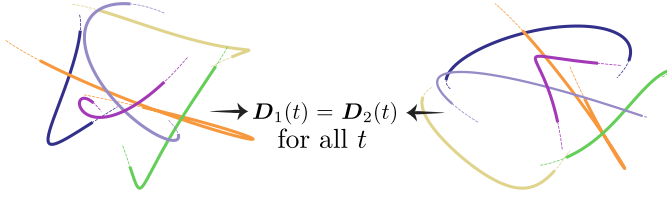$$\boldsymbol{G}(t) = \sum_{k=0}^{K} w_k(t) \boldsymbol{G}_k$$

Fig. 4. Two trajectory sets which are not rigid transforms of each other, but which generate the same KEDM. Corresponding points have the same color.

with the weights $\boldsymbol{w}(t) = [w_0(t), \dots, w_K(t)]^\top$ given as

$$\boldsymbol{w}(t) = \begin{pmatrix} 1 & \cdots & 1 \\ \sin(\omega\tau_0) & \cdots & \sin(\omega\tau_K) \\ \cos(\omega\tau_0) & \cdots & \cos(\omega\tau_K) \\ \vdots & \ddots & \vdots \\ \cos(2P\omega\tau_0) & \cdots & \cos(2P\omega\tau_K) \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \sin(\omega t) \\ \cos(\omega t) \\ \vdots \\ \sin(2P\omega t) \\ \cos(2P\omega t) \end{pmatrix}.$$

Proof: The proof is analogous to the polynomial case. We only need to show that the system matrix is full rank which is a standard result [42]. ∎

We have thus developed a way to write a time-dependent Gramian in terms of a linear combination of positive semidefinite (constant) basis Gramians.

C. Ambiguities Beyond Rigid Transformations in KDGP

Same as the static DGP, the KDGP suffers from rigid transformation ambiguity. Namely, the rotated and translated trajectory sets cannot be distinguished from pairwise distance data. However, since at every time instant we can apply a different rigid transform, the set of ambiguities that arise in the KDGP is much larger than just the rigid transforms.

In particular, trajectory sets which look rather differently (nothing like rotations and translations of each other) could generate the same KEDM. We give an example in Figure 4. The following straightforward result characterizes trajectories that lead to the same KEDM.

Definition 2: We say that the two trajectories $\boldsymbol{X}(t), \boldsymbol{Y}(t) \in \mathbb{R}^d[t]$ defined over some time interval T are distance-equivalent and write $\boldsymbol{X} \stackrel{\mathcal{D}}{\sim} \boldsymbol{Y}$ if and only if $\mathcal{D}(\boldsymbol{X}(t)) = \mathcal{D}(\boldsymbol{Y}(t))$, for all $t \in T$.

Proposition 3: Let $\boldsymbol{X}(t), \boldsymbol{Y}(t)$ be arbitrary trajectories in $\mathbb{R}^d[t]$. Then, the following statements are equivalent
1) $\boldsymbol{X} \stackrel{\mathcal{D}}{\sim} \boldsymbol{Y}$,
2) $\boldsymbol{Y}(t) = \boldsymbol{U}(t)\boldsymbol{X}(t) + \boldsymbol{c}(t)\boldsymbol{1}^\top$ where $\boldsymbol{U}(t)^\top\boldsymbol{U}(t) = \boldsymbol{I}$ and $\boldsymbol{c}(t)$ is a d-dimensional time-varying vector.

Proof: $(1) \Rightarrow (2)$: From $\boldsymbol{X} \stackrel{\mathcal{D}}{\sim} \boldsymbol{Y}$, by definition we have $\mathcal{K}(\boldsymbol{X}(t)^\top\boldsymbol{X}(t)) = \mathcal{K}(\boldsymbol{Y}(t)^\top\boldsymbol{Y}(t)), \forall t \in T$. Then, from (2) there is an orthogonal matrix $\boldsymbol{U}(t)$ such that

$$\boldsymbol{J}\boldsymbol{Y}(t) = \boldsymbol{U}(t)\boldsymbol{J}\boldsymbol{X}(t)$$

for all $t \in T$. On the other hand, $\boldsymbol{X}(t) = \boldsymbol{J}\boldsymbol{X}(t) + \boldsymbol{x}(t)\boldsymbol{1}^\top$, and $\boldsymbol{Y}(t) = \boldsymbol{J}\boldsymbol{Y}(t) + \boldsymbol{y}(t)\boldsymbol{1}^\top$ for $\boldsymbol{x}(t), \boldsymbol{y}(t) \in \mathbb{R}^d$. Finally,

$$\begin{aligned} \boldsymbol{Y}(t) &= \boldsymbol{J}\boldsymbol{Y}(t) + \boldsymbol{y}(t)\boldsymbol{1}^\top \\ &= \boldsymbol{U}(t)\boldsymbol{J}\boldsymbol{X}(t) + \boldsymbol{y}(t)\boldsymbol{1}^\top \\ &= \boldsymbol{U}(t)(\boldsymbol{X}(t) - \boldsymbol{x}(t)\boldsymbol{1}^\top) + \boldsymbol{y}(t)\boldsymbol{1}^\top \\ &= \boldsymbol{U}(t)\boldsymbol{X}(t) + (\boldsymbol{y}(t) - \boldsymbol{U}(t)\boldsymbol{x}(t))\boldsymbol{1}^\top. \end{aligned}$$

$(2) \Rightarrow (1)$ is trivial. ∎

Requiring that the trajectories follow a particular model (for example polynomial or bandlimited) limits possible choices of the time-varying rigid transform parameters $\boldsymbol{U}(t)$ and $\boldsymbol{c}(t)$. In particular, known results on spectral factorization of polynomial matrices imply that the orthogonal $\boldsymbol{U}(t)$ must be a constant matrix [43], [44]. On the other hand, as long as $\boldsymbol{c}(t)$ is polynomial (or bandlimited) of the same degree as $\boldsymbol{X}(t)$, it is a legal choice in the sense that the trajectories remain polynomial or bandlimited. But even with a fixed $\boldsymbol{U}(t) = \boldsymbol{U}$, varying $\boldsymbol{c}(t)$ can produce trajectories of rather different shapes which are indistinguishable from their time-varying distances.

In Section V, we propose a method for spectral factorization of kinetic Gramians based on anchor points and show how it resolves the described ambiguities. In our algorithms we will choose $\boldsymbol{c}(t)$ so that the centroid of the point set is kept fixed at the origin at all times, and then recover the correct centroid using anchor points. The following proposition will be useful.

Proposition 4: For trajectories of the form (7) or (10), the following statements are equivalent:
1) All coefficient matrices $\{\boldsymbol{A}_p\}_{p=0}^P$ have zero-mean columns;
2) $\boldsymbol{X}(t)\boldsymbol{1} = \boldsymbol{0}, \forall t \in \mathbb{R}$;
3) $\boldsymbol{G}_k\boldsymbol{1} = \boldsymbol{0}, \forall k \in \{0, \dots, K\}$.

Proof: We establish $(1) \Leftrightarrow (2)$ and $(2) \Leftrightarrow (3)$ for the polynomial trajectories and leave the straightforward extension to bandlimited trajectories to the reader. It is obvious that (1) implies (2) and (2) implies (3).

(2) implies (1): We have $\boldsymbol{X}(t)\boldsymbol{1} = \sum_{p=0}^P (\boldsymbol{A}_p\boldsymbol{1})t^p = \boldsymbol{0}$. Since the monomials $\{t \mapsto t^p\}_{p=0}^P$ form a linearly independent set, the coefficients $\boldsymbol{A}_p\boldsymbol{1}$ must all be zero. (In other words, the column centroid of all $\boldsymbol{A}_p$ must be at the origin.)

(3) implies (2): Since $\boldsymbol{G}(t)\boldsymbol{1} = \sum_{k=0}^K w_k(t)\boldsymbol{G}_k\boldsymbol{1} = \boldsymbol{0}$, we have

$$\|\boldsymbol{X}(t)\boldsymbol{1}\|_2^2 = \boldsymbol{1}^\top\boldsymbol{X}(t)^\top\boldsymbol{X}(t)\boldsymbol{1} = \boldsymbol{1}^\top\boldsymbol{G}(t)\boldsymbol{1} = \boldsymbol{0}.$$

Hence $\boldsymbol{X}(t)\boldsymbol{1} = \boldsymbol{0}$ for all $t \in \mathbb{R}$. ∎

IV. COMPUTING THE KEDM FROM NOISY, INCOMPLETE DATA BY SEMIDEFINITE PROGRAMMING

In this section we use the basis Gramian representation to derive an algorithm that solves the KDGP. Just as in the static case, we can either search directly for the set of trajectories $\boldsymbol{X}(t)$ which reproduces the measured distances, or we can search for the time-varying Gramian $\boldsymbol{G}(t)$ and use spectral factorization to estimate $\boldsymbol{X}(t)$. In the static case, the two formulations are equivalent (they produce the same solution up to a rigid transform), but the formulation in terms of the Gram matrix led to

a convenient semidefinite relaxation. In the time-varying case, we again state both formulations, and argue that the difference is now more significant.

To treat polynomial and bandlimited trajectories at once, we define the symbol $\boldsymbol{\Theta}$ to mean either $\boldsymbol{\Theta} = \{\boldsymbol{A}_p\}_{p=0}^{P}$ for the polynomial model or $\boldsymbol{\Theta} = \{\boldsymbol{A}_p, \boldsymbol{B}_p\}_{p=1}^{P} \cup \{\boldsymbol{B}_0\}$ for the bandlimited model, and similarly let $\boldsymbol{X}_{\boldsymbol{\Theta}}(t) = \sum_{p=0}^{P} \boldsymbol{A}_p t^p$ (resp. $\boldsymbol{X}_{\boldsymbol{\Theta}}(t) = \boldsymbol{B}_0 + \sum_{p=1}^{P} \boldsymbol{A}_p \cos(p\omega t) + \boldsymbol{B}_p \sin(p\omega t))$.

a) Formalizing in $\boldsymbol{X}$ domain: In this case, trajectory retrieval is written as

$$\underset{\boldsymbol{\Theta} \in \mathcal{A}}{\text{minimize}} \quad \sum_{i=1}^{T} \alpha_i \left\| \widetilde{\boldsymbol{D}}_{t_i} - \boldsymbol{W}_i \circ \mathcal{D}\left(\boldsymbol{X}_{\boldsymbol{\Theta}}(t_i)\right) \right\|_F^2 \tag{11}$$

$$\text{subject to} \quad \boldsymbol{X}_{\boldsymbol{\Theta}}(t)\mathbf{1} = \mathbf{0}, \forall t \in \mathbb{R},$$

where $\mathcal{D}(\boldsymbol{X}) = \mathcal{K}(\boldsymbol{X}^\top \boldsymbol{X})$, $\widetilde{\boldsymbol{D}}_{t_i}$ is the matrix of partial measured distances at time t_i, $\boldsymbol{W}_i$ is the adjacency matrix corresponding to measurements, $\alpha_i \geq 0$ are non-negative weights, and $\mathcal{A}$ is the set of all feasible parameters. It is not hard to see that the objective in (11) is nonconvex in $\boldsymbol{\Theta}$ (even though the constraint set is convex by Proposition 4). Hence, this problem involves minimizing a nonconvex functional which is in general difficult.

b) Formalizing in $\boldsymbol{G}$ Domain: Next, we derive a semidefinite program inspired by (4) for the trajectory recovery problem. The key ingredient is the basis Gramian representation of $\boldsymbol{G}(t)$ from Section III. Since the actual kinetic Gramian is linear in basis Gramians, the overall objective will be convex as long as the data fidelity metric is convex. The latter holds true since we use the squared Frobenius norm:

$$\underset{(\boldsymbol{G}_k:\boldsymbol{G}_k \succeq 0)_{k=0}^{K}}{\text{minimize}} \quad \sum_{i=1}^{T} \alpha_i \left\| \widetilde{\boldsymbol{D}}_{t_i} - \boldsymbol{W}_i \circ \mathcal{K}\left(\sum_{k=0}^{K} w_k(t_i)\boldsymbol{G}_k\right) \right\|_F^2$$

$$\text{subject to} \quad \boldsymbol{G}(t)\mathbf{1} = \mathbf{0}, \forall t \in \mathbb{R}$$

$$\boldsymbol{G}(t) \succeq 0, \forall t \in \mathbb{R}$$

$$\underset{t \in \mathbb{R}}{\max} \ \text{rank} \ \boldsymbol{G}(t) = d. \tag{12}$$

The constraints ensure that the solution corresponds to a time-varying Gramian $\boldsymbol{G}(t)$ with correct rank.

Recall that any trajectory generates a Gramian with the form $\boldsymbol{G}(t) = \sum_{k=0}^{K} w_k(t)\boldsymbol{G}_k$. Hence, it is clear that the set $\mathcal{G} = \{(\boldsymbol{G}_k : \boldsymbol{G}_k \succeq 0)_{k=0}^{K} : \sum_{k=0}^{K} w_k(t)\boldsymbol{G}_k \succeq 0 \text{ for all } t\}$ is nonempty. Further, $\mathcal{G}$ is convex as an (infinite) intersection of convex sets, $\mathcal{G}_t = \bigcap_{t \in \mathbb{R}} \{(\boldsymbol{G}_k : \boldsymbol{G}_k \succeq 0)_{k=0}^{K} : \sum_{k=0}^{K} w_k(t)\boldsymbol{G}_k \succeq 0\}$.

Let us emphasize that even though the objective is convex, the problem (12) is not easy to solve: it is still non-convex (due to the rank constraint) and in fact uncountably infinite (due to the continuous-time constraints).

There is no rotation ambiguity associated with this formulation because the Gramian is invariant to rotation and reflection of the points. Translation ambiguity has been resolved by requiring that $\boldsymbol{G}(t)\mathbf{1} = \mathbf{0}$ which implies that the recovered point set shall be centered at all times.

A. Equivalence Between (11) and (12)

The two formulations are equivalent if for every possible set of measurements, the solution sets produce the same KEDM. Denoting the optimizers (which could be sets) by $\boldsymbol{\Theta}^*$ and $(\boldsymbol{G}_k^*)_{k=0}^{K}$, it should hold that

$$\mathcal{D}\left(\boldsymbol{X}_{\boldsymbol{\Theta}^*}(t)\right) = \mathcal{K}\left(\sum_{k=0}^{K} w_k(t)\boldsymbol{G}_k^*\right), \ t \in \mathbb{R}.$$

By Propositions 1 and 2, for any optimizer $\boldsymbol{\Theta}^*$ of (11) and the corresponding trajectory $\boldsymbol{X}_{\boldsymbol{\Theta}^*}(t)$, we can find a Gramian basis $(\widetilde{\boldsymbol{G}}_k)_{k=0}^{K}$ such that $\mathcal{D}(\boldsymbol{X}_{\boldsymbol{\Theta}^*}(t)) = \mathcal{K}(\sum_{k=0}^{K} w_k(t)\widetilde{\boldsymbol{G}}_k)$. Therefore,

$$J_1(\boldsymbol{\Theta}^*) = J_2((\widetilde{\boldsymbol{G}}_k)_{k=0}^{K}) \geq J_2((\boldsymbol{G}_k^*)_{k=0}^{K}),$$

where J_1 denotes the loss in (11), and J_2 denotes the loss in (12). The question is whether this inequality can be made strict. Could the solution to (12) lead to a Gramian $\boldsymbol{G}(t)$ with no corresponding trajectory in $\mathcal{A}$? An in-depth study of this question is beyond the scope of this paper, but to see that this is indeed possible consider a contrived case of no measurements at all, that is to say, a feasibility search.

Trivially, any $\boldsymbol{\Theta} \in \mathcal{A}$ is a solution to (11) and any set of basis Gramians $(\boldsymbol{G}_k : \boldsymbol{G}_k \succeq \mathbf{0})_{k=0}^{K}$ is a solution to (12). By Lemma 1 every Gramian $\boldsymbol{G}(t)$ produced by its basis $(\boldsymbol{G}_k)_{k=0}^{K}$ has a polynomial spectral factor, that is, it corresponds to a polynomial trajectory $\boldsymbol{X}(t)$ such that $\boldsymbol{G}(t) = \boldsymbol{X}(t)^\top \boldsymbol{X}(t)$. Even though $\boldsymbol{G}(t)$ is real, its spectral factor, however, need not be; see [35] for a characterization of rank-deficient polynomial Gramians $\boldsymbol{G}(t)$ without real spectral factors. This situation is fundamentally different from what we had in the static case. Hence, we can construct feasible "complex trajectories" which are outside of $\mathcal{A}$. Consequently, the constraints in (11) are necessary but not sufficient for the two formulations to be equivalent. Nonetheless, they become equivalent with sufficient measurements:

Proposition 5: Suppose that $\widetilde{\boldsymbol{D}}_i = \boldsymbol{W}_i \circ \mathcal{D}(\boldsymbol{X}_{\boldsymbol{\Theta}}(t_i))$ and (12) has a unique optimizer $\boldsymbol{G}^*(t) = \sum_{k=0}^{K} w_k(t)\boldsymbol{G}_k^*$. Then

$$\boldsymbol{G}^*(t) = \boldsymbol{X}_{\boldsymbol{\Theta}^*}(t)^\top \boldsymbol{X}_{\boldsymbol{\Theta}^*}(t) \tag{13}$$

where $\boldsymbol{X}_{\boldsymbol{\Theta}^*}(t) = \boldsymbol{U}\boldsymbol{X}_{\boldsymbol{\Theta}}(t)\boldsymbol{J}_N$ for some orthogonal matrix $\boldsymbol{U} \in \mathbb{R}^{d \times d}$ (that is, it is a centered, rotated version of the true geometry).

Proof: We prove this proposition by construction. Let us define $\boldsymbol{G}_k^* = \boldsymbol{J}_N \boldsymbol{X}_{\boldsymbol{\Theta}}(\tau_k)^\top \boldsymbol{X}_{\boldsymbol{\Theta}}(\tau_k)\boldsymbol{J}_N$ for $k \in \{0, \ldots, K\}$ and $\boldsymbol{G}^*(t) = \sum_{k=0}^{K} w_k(t)\boldsymbol{G}_k^*$. From Propositions 1 and 2, we deduce that $\boldsymbol{G}^*(t) = \boldsymbol{J}_N \boldsymbol{X}_{\boldsymbol{\Theta}}(t)^\top \boldsymbol{X}_{\boldsymbol{\Theta}}(t)\boldsymbol{J}_N$. Hence, $\boldsymbol{G}^*(t)$ belongs to the feasible set of (12) as $\boldsymbol{J}_N \boldsymbol{X}_{\boldsymbol{\Theta}}(t)^\top \boldsymbol{X}_{\boldsymbol{\Theta}}(t)\boldsymbol{J}_N$ is a zero-mean positive semidefinite matrix for all $t \in \mathbb{R}$, with rank at most d. On the other hand, since $\mathcal{D}(\boldsymbol{X}_{\boldsymbol{\Theta}}(t)) = \mathcal{K}(\boldsymbol{G}^*(t))$, we have $J_2(\boldsymbol{G}^*) = 0$. Finally, since (12) has a unique solution, the minimizer of (12) must have the form (13). ∎

It is useful to interpret the two approaches in (11) and (12) in terms of graph-based definition of the KDGP (Problem 2). The sequence of incomplete and noisy distances, $\widetilde{\boldsymbol{D}}_{t_1}, \ldots, \widetilde{\boldsymbol{D}}_{t_T}$ is modeled as a series of incomplete graphs whose edge weights correspond to the measured distances. The goal of KDGP is

to find a node function $x(u, t)$ that maps vertices of measurement graphs to points in $\mathbb{R}^d$ whose pairwise distances match the measured distances at sampling times $t_k \in \mathcal{T}$. From this perspective, the formulation (11) aims to directly estimate the node function $x(u, t)$ from distance measurements, while in formulation (12), we break the KDGP into two subproblems:

1) **Completing the measurement graphs:** This amounts to estimating the edge function, $f(e, t)$ for every $e \in E_t$ instead of only for $e \in E_i$, with E_i being the edges measured at time $t_i \in \mathcal{T}$;
2) **Estimating the node function,** $x(u, t)$: This is equivalent to spectral factorization of the time-dependent Gramian.

The formulation (12) solves the first subproblem since it outputs a time-varying Gramian $G(t)$ from which we easily get the KEDM as $\mathcal{K}(G(t))$. The second problem is addressed in Section V.

Finally, we note that the KEDM formulation in (12) is a generalization of the static EDM formulation in (4). To see the equivalence, note that static points are modeled by a polynomial of degree zero, $P = 0$, in which case the Gramian becomes $G(t) = G_0$ since $w_0(t) = 1$.

B. Practical Considerations: Relax and Sample

To get a practical algorithm for (12), we sample the continuous-time semidefiniteness constraint, $G(t) \succeq 0, \forall t \in \mathbb{R}$, and relax the non-convex rank constraint. In Algorithm 1, we denote the set of sampling times for this constraint by $\mathcal{T}_{\text{psd}}$.

In relaxations for static EDMs, instead of simply removing the rank constraint, it is often replaced by a regularizer. Perhaps counterintuitively (see [13] for a longer discussion), a strategy that works well is to *maximize* the rank of the Gram matrix, as this corresponds to pushing the points apart and minimizing the embedding dimension. We use a similar strategy in our KEDM semidefinite relaxation (Algorithm 1).

One issue with the semidefinite relaxation for the standard DGP is that there are often no strictly feasible points; the feasible Gram matrices lie on the low-rank faces of the positive semidefinite cone. This is troublesome for the primal–dual interior point solvers since it precludes strong duality (Slater's constraint qualification fails). On the other hand, Krislock and Wolkowicz skillfully exploit it by noting that the degeneracy is due to the existence of cliques in the DGP graph. They characterize faces of the positive semidefinite cone associated with individual cliques, and design fast, accurate solvers for noiseless instances [45].

Whether their ideas can be applied to the KDGP remains an open question. At a glance, it seems challenging: not only does the connectivity graph in the KDGP change between the sampling instants, but we work with a non-unique decomposition of the time-varying Gramian into basis Gramians.

Thus, even with low-rank-promoting regularization, the recovered Gram matrices will rarely be *exactly* rank-d due noise and numerical issues of the off-the-shelf semidefinite solvers. To address this, we apply a standard rank projection to the retrieved Gramians by setting the least significant $N - d$ singular values to 0.

Algorithm 1: Semidefinite Relaxation for KEDM.

1: **procedure** SDR $(\{t_i\}_{i=1}^T, \{\widetilde{D}_{t_i}\}_{i=1}^T, \{W_i\}_{i=1}^T)$
2: Solve for $\{G_k\}$:

$$\text{minimize} \sum_{i=1}^T \alpha_i \left\| \tilde{D}_{t_i} - W_i \circ \mathcal{K}\left(\sum_{k=0}^K w_k(t_i)G_k\right) \right\|_F^2$$

$$- \lambda \sum_{k=0}^K \text{Tr}(G_k)$$

w.r.t $\quad G_0, \ldots, G_K \succeq 0$

such that $\quad G_k \mathbf{1} = \mathbf{0} \qquad k \in \{0, \ldots, K\},$

$$\sum_{k=0}^K w_k(t)G_k \succeq \mathbf{0} \quad t \in \mathcal{T}_{\text{psd}}.$$

3: $\quad G_k \leftarrow$ RankProjection$(G_k, d), k \in \{0, \ldots, K\}$
4: **return** $\widehat{D}(t) = \mathcal{K}(\sum_{k=0}^K w_k(t)G_k)$
5: **end procedure**

V. SPECTRAL FACTORIZATION OF THE GRAMIAN

Algorithm 1 produces a time-varying Gramian whose KEDM best represents the measured distance sequence. In this section, we show how to estimate the corresponding trajectory by factoring the Gramian as $G(t) = X(t)^\top X(t)$ where $X(t)$ is the set of point trajectories. We know that the trajectory can only be estimated up to a time-invariant rotation (and possibly reflection) [43] and a time-varying translation. To resolve this uncertainty, we introduce a set of anchors—points whose absolute positions are known.

In practice, anchors might correspond to nodes whose position is fixed such as buoys and beacons, or nodes equipped with a positioning technology such as GPS. Because in the KDGP the anchors can move (unlike in the usual DGP), we have more possibilities for anchor measurements than in the static case. For our trajectory models, we only need to know the positions of the anchor points at some fixed, finite set of times, but we could measure positions of different sets of points at different times.

Given a spectral factor[5] $\overline{X}(t)$, of the time-varying Gramian, the true trajectory, $X(t)$, can be found as

$$X(t) = U\overline{X}(t) + x(t)\mathbf{1}^\top + N(t). \tag{14}$$

where U is a $d \times d$ orthogonal matrix, $x(t)$ is a $d \times 1$ time-varying vector and $N(t)$ represents the net effect of model mismatch and measurement noise. The matrix U is constant (by the spectral factorization theorem) whereas the translation factor $x(t)$ is a function of time. On the other hand, the translation factor $x(t)$ must belong to the same trajectory model as $X(t)$ (polynomial or bandlimited). Hence, $x(t)$ can be written as

$$x(t) = Mz(t),$$

---

[5]One out of infinitely many possible.

where for the polynomial model, $\mathcal{X}_{\text{poly}}$ we have

$$z(t) = [1,\ t,\ \cdots,\ t^P]^\top$$

and $\boldsymbol{M} \in \mathbb{R}^{d \times (P+1)}$, and for the bandlimited model, $\mathcal{X}_{\text{BL}}$,

$$z(t) = [1, \sin(\omega t), \cos(\omega t), \ldots, \sin(P\omega t), \cos(P\omega t)]^\top$$

and $\boldsymbol{M} \in \mathbb{R}^{d \times (2P+1)}$.

A difficulty compared to the static case is that spectral factorization of polynomial Gram matrices is not straightforward and becomes brittle in the presence of noise. It is thus desirable to develop trajectory estimation methods that do not require full polynomial factorization. We show that this is possible at the expense of additional anchor measurements.

A. Known Spectral Factor

We start by assuming that we have access to *some* spectral factor $\overline{\boldsymbol{X}}(t)$ such that $\boldsymbol{G}(t) = \overline{\boldsymbol{X}}(t)^\top \overline{\boldsymbol{X}}(t)$. In this case, to estimate the unknown rotation and translation, we assume that at L distinct times $\tau_1, \ldots, \tau_L$ we measure positions of points $\mathcal{I}_1, \ldots, \mathcal{I}_L$, with $\mathcal{I}_\ell$ being the index set of points whose positions are measured at τ_ℓ. We let $\boldsymbol{X}_{\mathcal{I}_\ell}$ denote the column selection of $\boldsymbol{X}(\tau_\ell)$ corresponding to indices in $\mathcal{I}_\ell$.

An estimate for $\boldsymbol{U}$ and $\boldsymbol{M}$ can be computed by solving

$$\underset{\boldsymbol{U} \in M_d(\mathbb{R}),\, \boldsymbol{M} \in \mathbb{R}^{d \times L}}{\arg\min} \sum_{\ell=1}^L \left\| \boldsymbol{X}_{\mathcal{I}_\ell} - \boldsymbol{U}\overline{\boldsymbol{X}}(\tau_\ell) - \boldsymbol{M}z(\tau_\ell)\boldsymbol{I}^\top \right\|_F^2$$

where $M_d(\mathbb{R})$ is the set of $d \times d$ orthonormal matrices and $L = P + 1$ (resp. $2P + 1$) for polynomial (resp. bandlimited) trajectories. This is a non-convex problem because $M_d(\mathbb{R})$ is a non-convex set.

The above optimization can be decoupled as in standard Procrustes analysis provided that there exists a time $\widetilde{\tau}_\ell \in \{\tau_1, \ldots, \tau_L\}$ at which we know the positions of at least $d+1$ anchors. In this case, $\boldsymbol{U}$ can be estimated at this time alone using the technique described in Section II-B. Once the rotation $\widehat{\boldsymbol{U}}$ is found, we can estimate the matrix $\boldsymbol{M}$ by solving the following convex problem:

$$\widehat{\boldsymbol{M}} = \underset{\boldsymbol{M} \in \mathbb{R}^{d \times L}}{\arg\min} \sum_{\ell=1}^L \left\| \boldsymbol{M}z(\tau_\ell) - \frac{1}{N_{\tau_\ell}}\left(\boldsymbol{X}_{\mathcal{I}_\ell}(\tau_\ell) - \widehat{\boldsymbol{U}}\overline{\boldsymbol{X}}(\tau_\ell)\right)\boldsymbol{I} \right\|_2^2$$

where $N_{\tau_\ell} \geq 1$ for $\ell \geq 2$. Finally, we note that matching d instead of $d+1$ points would leave us with a flip ambiguity, so $d+1$ is indeed the smallest number of anchors that lets us use the Procrustes analysis.

B. Unknown Spectral Factor (Practical Algorithm)

The previous section implies that $L + d$ anchor points are necessary to estimate the rotation $\boldsymbol{U}$ and translation $\boldsymbol{M}$ provided that a spectral factor $\overline{\boldsymbol{X}}(t)$ of $\boldsymbol{G}(t)$ is given. Unfortunately, algorithms for spectral factorization rely on unstable computations involving determinants and are often computationally demanding, which makes them unsuitable for our application where noise can be significant [46]. To avoid this step, we propose a method which relies on additional anchor measurements.

Algorithm 2: Spectral Factorization.

1: **procedure** SpectralFactorization ($\widehat{\boldsymbol{D}}(t)$, $\{\boldsymbol{X}_{\mathcal{I}_\ell}\}_{l=1}^L$) $\quad \triangleright \boldsymbol{X}_{\mathcal{I}_\ell}$: Anchor points at different times, $\tau_1, \ldots, \tau_L$.

2: **for** $l \in \{1, \ldots, L\}$ **do**

3: $\quad \widehat{\boldsymbol{G}}(\tau_l) \leftarrow -\frac{1}{2}\boldsymbol{J}_N \widehat{\boldsymbol{D}}(\tau_l)\boldsymbol{J}_N$

4: $\quad \overline{\boldsymbol{X}}(\tau_l) \leftarrow \widehat{\boldsymbol{G}}(\tau_l)^{1/2}$

5: $\quad$ Solve for $\widehat{\boldsymbol{U}}_{\tau_l}$ using Procrustes analysis

6: $\quad$ Estimate the translation at time τ_ℓ:

$$\widehat{\boldsymbol{x}}(\tau_l) \leftarrow \frac{1}{N_{\tau_l}}(\boldsymbol{X}_{\mathcal{I}_\ell} - \widehat{\boldsymbol{U}}\overline{\boldsymbol{X}}_{\mathcal{I}_\ell}(\tau_l))\boldsymbol{I}$$

7: $\quad$ Estimate point positions at time τ_ℓ:

$$\widehat{\boldsymbol{X}}(\tau_l) \leftarrow \widehat{\boldsymbol{U}}_{\tau_l}\overline{\boldsymbol{X}}(\tau_l) + \widehat{\boldsymbol{x}}(\tau_l)\boldsymbol{I}^\top$$

8: **end for**

9: Find the trajectory:

$$\Theta \leftarrow \underset{\Theta \in \mathcal{A}}{\arg\min} \sum_{\ell=1}^L \|\boldsymbol{X}_\Theta(\tau_\ell) - (\widehat{\boldsymbol{U}}_{\tau_\ell}\overline{\boldsymbol{X}}(\tau_\ell) + \widehat{\boldsymbol{x}}(\tau_l)\boldsymbol{I}^\top)\|_F^2$$

10: **return** $\boldsymbol{X}_\Theta(t)$

11: **end procedure**

Assume that at each of L distinct times we measure positions of at least $d+1$ anchors; as before, denote the anchor indices at time τ_ℓ by $\mathcal{I}_\ell$, and the corresponding positions by $\boldsymbol{X}_{\mathcal{I}_\ell}$. Now we can use Procrustes analysis at each time individually (that is, applied to constant matrices that are evaluations of time-varying matrices at these particular times) to estimate rotation and translation, $\widehat{\boldsymbol{U}}_{\tau_l}$ and $\widehat{\boldsymbol{x}}(\tau_l)$ at time τ_l. Denote by $\overline{\boldsymbol{X}}(\tau_\ell)$ any matrix such that $\overline{\boldsymbol{X}}(\tau_\ell)^\top \overline{\boldsymbol{X}}(\tau_\ell) = \boldsymbol{G}(\tau_\ell)$; since this involves only constant matrices, we can use the eigendecomposition method described in Section II-A to compute $\overline{\boldsymbol{X}}(\tau_\ell)$.

Note that in doing so, there is no guarantee that these "marginal" estimates for the rotation correspond to the unique global $\boldsymbol{U}$ we are looking for, because we do not exploit any temporal model in computing the spectral factors $\overline{\boldsymbol{X}}(\tau_\ell)$. In other words, all $\widehat{\boldsymbol{U}}_{\tau_l}$ could be distinct, and in principle they will. Nevertheless, we can use them to estimate the trajectory by solving the following problem:

$$\boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta} \in \mathcal{A}}{\arg\min} \sum_{l=1}^L \left\| \boldsymbol{X}_\boldsymbol{\Theta}(\tau_l) - \left(\widehat{\boldsymbol{U}}_{\tau_l}\overline{\boldsymbol{X}}(\tau_l) + \widehat{\boldsymbol{x}}(\tau_l)\boldsymbol{I}^\top\right) \right\|_F^2, \tag{15}$$

and using $\boldsymbol{X}_{\boldsymbol{\Theta}^*}(t)$ as our estimate of the trajectory.

The logic behind (15) is that even though the matrices $\widehat{\boldsymbol{U}}_{\tau_\ell}$ are "wrong," the product $\widehat{\boldsymbol{U}}_{\tau_l}\overline{\boldsymbol{X}}(\tau_l)$ is correct thanks to the anchors. With sufficiently many marginal estimates, there is a unique set of polynomial trajectories passing through them. The described procedure is summarized in Algorithm 2 and the complete KEDM trajectory localization algorithm with anchors in Algorithm 3.

Algorithm 3: End-to-end KEDM Algorithm.

1: **procedure** KEDM $(\{\boldsymbol{X}_{\mathcal{I}_\ell}\}, \{t_i\}, \{\widetilde{\boldsymbol{D}}_{t_i}\}, \{\boldsymbol{W}_i\})$
2:     $\widehat{\boldsymbol{D}}(t) = \texttt{SDR}(\{t_i\}, \{\widetilde{\boldsymbol{D}}_{t_i}\}, \{\boldsymbol{W}_i\})$
3:     $\boldsymbol{X}_\Theta(t) = \texttt{SpectralFactorization}(\widehat{\boldsymbol{D}}(t),$
      $\{\boldsymbol{X}_{\mathcal{I}_\ell}\})$
4:     **return** $\boldsymbol{X}_\Theta(t)$
5: **end procedure**

VI. SIMULATION RESULTS

In this section we empirically evaluate different aspects of the proposed algorithm. We first study the influence of sampling time distribution in Section VI-A as this choice affects the other experiments. In Section VI-B, we look at the maximum achievable measurement sparsity:[6] KDGP measurements are a sequence of incomplete EDMs and it is interesting to understand what proportion of missing entries we can tolerate.[7] Finally, in Section VI-C we study the effect of measurement noise on the quality of the estimated trajectories.

We end this section by applying our algorithms to a synthetic problem of satellite localization from noisy and very sparse distance measurements.

A. Distribution of Sampling Times

The measurements in Algorithm 3 are a sequence of (incomplete) snapshots of KEDM at different times, $\{\boldsymbol{W}_i \circ \mathcal{D}(\boldsymbol{X}(t_i))\}_i$. We experiment with different choices of sampling times $\{t_i\}_i$. To exclude the influence of other factors, we assume having access to all pairwise distances, and we contaminate the measurements by noise. Note that without noise, we can compute the Gramian basis simply by solving a linear system of equations so that any sampling strategy with sufficiently many samples gives the perfect estimation.

Let the true, noiseless distances be $d_{ij}(t) = \|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|$, and noisy measurements given as

$$\widetilde{d}_{ij}(t) = d_{ij}(t) + n_{ij}(t) \tag{16}$$

where $n_{ij}(t) \sim \mathcal{N}(0, \sigma^2)$ is iid measurement noise. The corresponding KEDMs are $\boldsymbol{D}(t) = [d_{ij}^2(t)]_{ij}$ and $\widetilde{\boldsymbol{D}}(t) = [\widetilde{d}_{ij}^2(t)]_{ij}$.

To compare the different sampling protocols, we average the reconstruction error over many trajectory and noise realizations. The reconstruction error is defined as

$$e_D(t) = \frac{\|\boldsymbol{D}(t) - \widehat{\boldsymbol{D}}(t)\|_F}{\|\boldsymbol{D}(t)\|_F}.$$

where $\widehat{\boldsymbol{D}}(t) = \texttt{SDR}((\{t_i\}, \{\widetilde{\boldsymbol{D}}_{t_i}\}, \{\boldsymbol{W}_i\})_{i=1}^T)$ is the KEDM estimated by Algorithm 3. The goal is to determine which sampling pattern minimizes $e_D(t)$ for all t in the interval of interest $[T_1, T_2]$. In Figure 5, we show the average errors for the following sampling patterns:

---

[6]We use the term "sparsity" to refer to sparse or subsampled measured data, as is common in the inverse problems theory.

[7]In all experiments we sample the positive semidefinite constraint at random times. We have found empirically that this choice does not matter much, unlike the choice of measurement times. The exact details can be found in the reproducible code at https://github.com/swing-research/kedm/
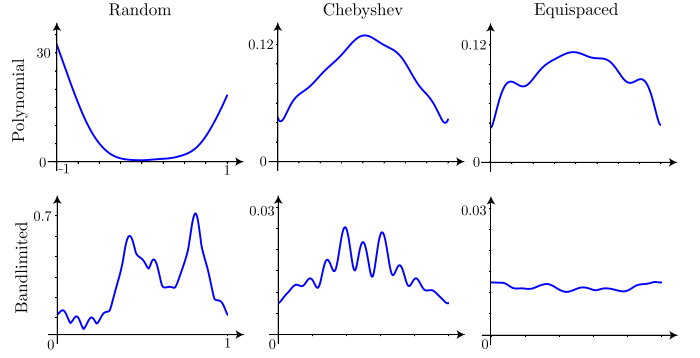

Fig. 5. Relative reconstruction error $e_D(t)$ averaged over $M = 200$ realizations. The number of points is $N = 10$, ambient dimension $d = 2$, trajectory degree $P = 3$ and noise variance $\sigma^2 = 1$ for both models. The trajectory parameters, $\boldsymbol{A}_p$, are drawn iid Gaussian—real valued for polynomial and complex for bandlimited with complex exponential basis. The sampled interval of interest is $[-1, 1]$ for the polynomial and $[0, 1]$ for the bandlimited model.

- random: $t_i \sim \text{Unif}([T_1, T_2])$,
- Chebyshev: $t_i = \frac{1}{2}(T_1 + T_2) + \frac{1}{2}(T_2 - T_1)\cos(\frac{2i-1}{2T}\pi)$,
- equispaced: $t_i = T_1 + (T_2 - T_1)\frac{i}{T}$,

where $i = 1, \ldots, T$. We can see that random sampling performs poorly for both the polynomial and the bandlimited model. Chebyshev and equispaced nodes give a similar relative error, with equispaced nodes performing slightly better for the bandlimited model. Studying individual realizations shows that the worst-case error for Chebyshev and equispaced sampling is on the same order as the average error, but it is much worse for random sampling: large reconstruction errors occur when two consecutive measurement times are far apart. In the following experiments, we use equispaced measurement times.

All experiments were run on a laptop with a 2.9 GHz Core i5 processor and 16 GB of memory, using the cvxpy package [47], [48]. The interior point methods used by solvers in cvxpy tend to become slow as the number of points and the polynomial degree grow (e.g., for $N \geq 20$, $P \geq 5$), and should be replaced by faster, tailor-made optimizers. In this proof-of-concept paper, we used cvxpy for convenience; a study of faster solvers is left to future work.

B. Measurement Sparsity

Trajectory estimation from distances is a nonlinear sampling problem, with trajectory models allowing us to trade spatial for temporal samples. Here we empirically study the maximum sparsity level for spatial measurements. Given a sequence of measurement masks $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_T \in \{0,1\}^{N \times N}$, the sparsity level, $0 \leq S \leq 1$, is defined as the ratio of average to total number of pairwise distances:

$$S = \frac{1}{\binom{N}{2}} \frac{1}{T} \sum_{i=1}^{T} \# \text{ of missing measurements at time } t_i.$$

We can expect the maximum sparsity level to vary with factors such as the trajectory model, temporal sampling pattern, measurement masks, and noise. To evaluate it, we fix parameters the trajectory class, degree, number of points, and ambient dimension. We declare a localization experiment successful if

TABLE I
MAXIMAL SPARSITY FOR THE POLYNOMIAL MODEL AND $d = 2$

$P \backslash N$	5	6	7	8	9	10	11	12	13	14	15
$P = 1$	0.1	0.2	0.28	0.39	0.44	0.46	0.52	0.56	0.57	0.60	0.62
$P = 2$	0.1	0.2	0.33	0.35	0.41	0.46	0.51	0.54	0.57	0.60	0.62
$P = 3$	0.1	0.2	0.28	0.35	0.41	0.46	0.51	0.54	0.57	0.59	0.62
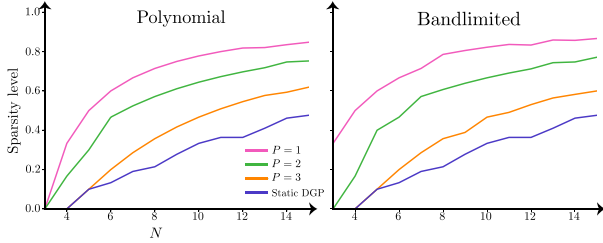
Fig. 6. The estimated sparsity level $\widehat{S}$ for polynomial degrees P and numbers of points N. The success threshold δ is set to 0.99 and the target fraction of successful reconstructions q to 0.9.

the relative trajectory mismatch,

$$e_X = \int_{\mathcal{T}} \|\boldsymbol{X}(t) - \widehat{\boldsymbol{X}}(t)\|_F / \|\boldsymbol{X}(t)\|_F \, \mathrm{d}t,$$

which we approximate by discretizing $\mathcal{T}$, is below some pre-scribe threshold δ. We are interested in numerically evaluating the probability that the localization succeeds (within tolerance δ) if on average over sampling times, m pairwise distances are missing. Denote this probability by $p(\delta, m)$. We would like to find conditions on m such that $p(\delta, m)$ is large. In particular, for $0 \le q < 1$, let $\overline{m}(\delta, q)$ be the largest m such that $p(\delta, m) \ge q$.

We run M localization trials for different realizations of random trajectories, and denote the number of succesful trials by M_1. For a given average number of missing pairwise distances m, the probability of correct localization is estimated as $\widehat{p}_M(\delta, m) = \frac{M_1}{M}$. The estimate of $\overline{m}(\delta, q)$ is then simply

$$\widehat{\overline{m}}(\delta, q) := \max \{ m \ : \ \widehat{p}_M(\delta, m) \ge q \}.$$

To compute $\widehat{\overline{m}}(\delta, q)$, we increase the number of missing mea-surements per sampling time, m, and count the number of δ-accurate estimates to compute $\widehat{\overline{m}}(\delta, q)$ and the corresponding $\widehat{S}(\delta, q) = \widehat{\overline{m}}(\delta, q) / \binom{N}{2}$.

In the first experiment, we fix the number of sampling times, T, and vary the number of points N and polynomial (or ban-dlimited) degree P. Specifically, in Figure 6 we choose $T = 7$ for polynomial and $T = 13$ for bandlimited models.

As expected, we observe that for a fixed N, as P grows (and consequently the number of parameters) the allowable sparsity level decreases, meaning that more complicated trajectories require more spatial samples. This is due to fact that ratio of number of measurements, which is fixed in this case, to number of parameters decreases. Importantly, compared to the static DGP, we see that KEDMs and the proposed semidefinite relaxation allow us to measure fewer distances at any given time, and compensate for this by sampling at multiple times.
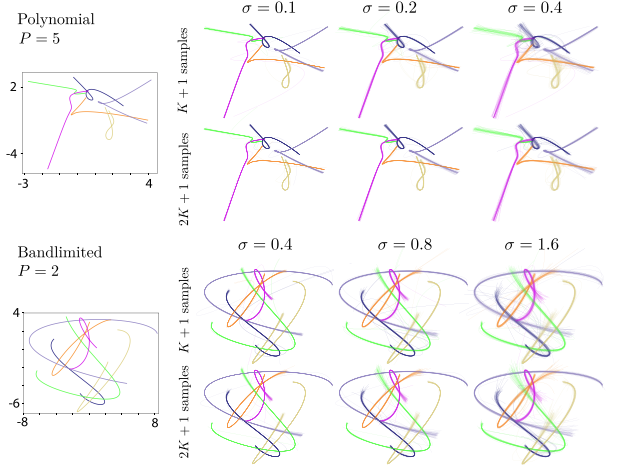

Fig. 7. Estimated trajectories, $\widehat{\boldsymbol{X}}(t)$, for $N = 6$ points in $\mathbb{R}^2$ at different levels of measurement noise and number of temporal measurements. The time interval of interest is $t \in [-1, 1]$ for polynomial and $t \in [0, 1]$ for bandlimited trajectories.

In the second experiment we attempt to better characterize the observed spatio–temporal sampling tradeoff. To this end, we fix the parameters so that the ratio of the number of measurements to the number of the degrees of freedom is constant. That is, we keep the number of sampling times proportional to the number of basis Gramians, $T = K + 1$ for the polynomial and $T = 2K + 1$ for the bandlimited model.

As Tables I and II show, with this scaling the sparsity level is approximately constant as the polynomial degree P grows. In other words, even though the trajectories become more and more complicated, we can keep the number of spatial measurements fixed as long as we adjust the number of temporal sampling instants. The empirical observation that the required number of measurements scales linearly with the number of the degrees of freedom suggests that the proposed algorithms require an order-optimal number of samples.

C. Noisy Measurements

We again quantify the influence of noise by the relative trajectory mismatch. We fix a trajectory, shown in Figure 7, and a set of distance sampling times $\{t_k\}_{k=0}^{K}$, and generate many realizations of noisy measurement sequences $\widetilde{\mathcal{D}}_{t_0}, \ldots, \widetilde{\mathcal{D}}_{t_K}$ with the same noise variance σ^2. The i.i.d. noise is added to the non-squared distances. The empirical trajectory mismatch is an average of relative trajectory mismatches over realizations, $\frac{1}{M} \sum_m e_X^{(m)}$.

TABLE II
MAXIMAL SPARSITY FOR THE BANDLIMITED MODEL AND $d = 2$

$P\backslash N$	5	6	7	8	9	10	11	12	13	14	15
$P = 1$	0.1	0.26	0.33	0.39	0.44	0.48	0.51	0.56	0.57	0.60	0.63
$P = 2$	0.1	0.2	0.28	0.35	0.41	0.44	0.49	0.53	0.56	0.60	0.63
$P = 3$	0.1	0.2	0.28	0.35	0.38	0.46	0.49	0.53	0.56	0.58	0.60
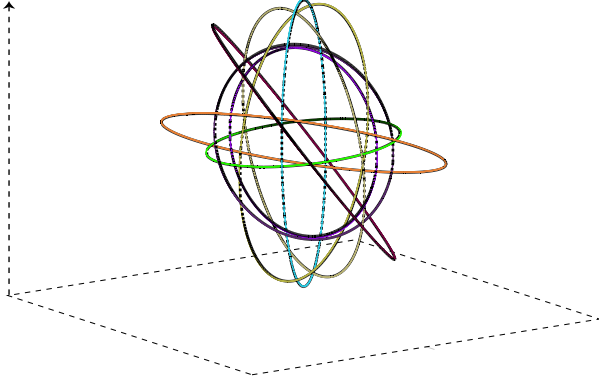
Fig. 8. Reconstructing the trajectories of 8 orbiting satellites. Colored and dashed lines represent actual and estimated trajectories. All satellites have the same angular frequency with $P = 1$. The measurement matrices are missing about $9/10$ measurements, and noise level is set to $\sigma = 0.05$. The average reconstruction error is $\frac{1}{M} \sum_i e_D(t_i) = 0.01$.
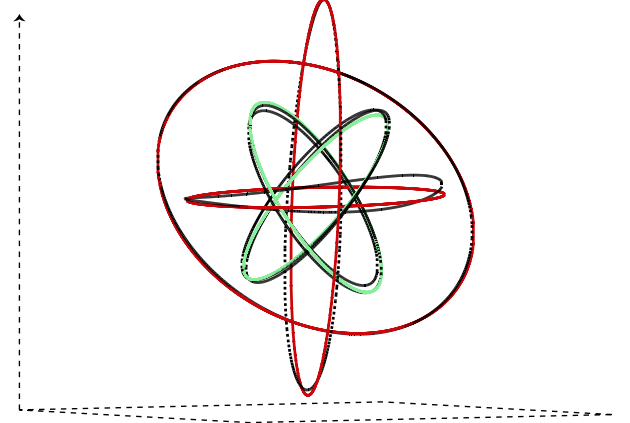

Fig. 9. Reconstructing the trajectories of 5 orbiting satellites with angular frequencies of ω_0 and $2\omega_0$. The measurement matrices are 80% sparse, and average reconstruction error is $\frac{1}{M} \sum_i e_D(t_i) = 0.03$.

In Figure 7, we show many estimated trajectories $\widehat{\boldsymbol{X}}(t)$. As expected, the mismatch increases with measurement noise σ^2 and decreases with the number of measurements. In all cases, the estimated trajectories concentrate around the true ones.

D. A Stylized Application: Satellite Positioning

In this section we apply KEDMs in a stylized satellite positioning scenario where measurements are both very sparse and noisy. We consider a set of satellites moving with constant angular velocity, with angular frequency being an integer multiple of the fundamental frequency ω_0. This is a limitation of the current bandlimited model which we intend to address in future work. Such trajectories have the form

$$\boldsymbol{x}(t) = \boldsymbol{R} \begin{pmatrix} a\cos(\omega t) \\ b\sin(\omega t) \\ 0 \end{pmatrix},$$

where $\boldsymbol{R}$ is a 3×3 rotation matrix.

The set of all satellite trajectories,

$$\boldsymbol{X}(t) = [\boldsymbol{x}_1(t, p_1), \ldots, \boldsymbol{x}_N(t, p_N)]$$

follows the bandlimited trajectory model. Concretely,

$$\boldsymbol{x}(t, p) = \boldsymbol{a}_1 \cos(p\omega_0 t) + \boldsymbol{a}_2 \sin(p\omega_0 t)$$

is the trajectory of a satellite whose angular frequency is p times the fundamental frequency ω_0 and $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R}^3$. The ensemble trajectory, $\boldsymbol{X}(t)$, is a bandlimited trajectory of degree $P = \max_n p_n$.

We apply Algorithm 3 in two experiments. In Figure 8, we show trajectories of $N = 8$ satellites with the same orbiting frequency ω_0. Since the ellipses are of different sizes, the inner

points can also be interpreted as vehicles on the earth. We measure 3 noisy pairwise distances, out of 28 available, per sampling time instant. This could model, for instance, occlusions by the earth and other adversarial effects. We compensate for undersampling in space by oversampling in time, taking samples at $T = 30$ different times. Similarly, in Figure 9 we show $N = 5$ satellites with angular frequencies ω_0 and $2\omega_0$, that is, with $P = 2$; we measure only 2 pairwise distances per sampling time instant (these are extremely sparse measurements with which static localization is hopeless), at $T = 30$ sampling times. As figures show, in both experiments, we successfully reconstruct trajectories of the satellites.

VII. Conclusion

In this paper, we extended the algebraic tools for localization from distances to the case when points are moving. We defined kinetic Euclidean distance matrices for polynomial and bandlimited trajectories, and we derived algorithms based on semidefinite programming to solve the associated trajectory localization problem. The chosen trajectory models are expressive and can approximate continuous trajectories commonly used in localization and tracking.

The key step in our method is to represent the time-varying Gram matrices as time-varying linear combinations of certain constant matrices. This allowed us to rewrite the localization problem as a semidefinite program. Same as in the static case, the actual localization involves an additional spectral factorization step. However, for polynomial matrices, this is much harder than a simple SVD, and especially from noisy data like those that we get. We circumvent the related difficulties by deriving a spectral factorization method that directly uses anchor measurements.

We demonstrated through numerical experiments that the proposed algorithms can indeed reconstruct model trajectories from sparse and noisy measurements, and that they can explore the tradeoff between the number of distances measured at any given time, and the number of sampling times.

A. Future Work

Both the polynomial and the bandlimited trajectories are special cases of a general class of subspace trajectories. Conceptually, one should be able to derive the localization theory for general subspace trajectories (for example, one could mix bandlimited and polynomial trajectories). Doing this cleanly is not trivial and is part of ongoing work. Instead of adopting deterministing trajectory models, we could think of stochastic models. The KEDM would then become a random object and ideas of statistical inference could be used to extract point position information. Stochastic models should play particularly well with SLAM.

In the presented numerical experiments we empirically explored the tradeoff between the spatial and the temporal measurements. An interesting and important line of work is to characterize this tradeoff analytically, as a function of the algorithm used for localization.

Finally, an interesting thing happens once we depart from static EDMs: instead of only measuring inter-point distances, we can measure relative (vector or scalar) velocities, and the same goes for anchors. It is of considerable practical interest to derive tractable optimization procedures that take into account these general kinetic measurements.

APPENDIX A
SPECTRAL FACTORIZATION OF THE GRAMIAN

Let q stand for t for the polynomial model, or $e^{j\omega t}$ for the bandlimited model. Similarly, let $\mathcal{P} = \{0, \dots, P\}$ for polynomial or $\mathcal{P} = \{-P, \dots, P\}$ for bandlimited, and $\mathcal{P} + \mathcal{P} \overset{\text{def}}{=} \{p_1 + p_2 : p_1, p_2 \in \mathcal{P}\}$.

Lemma 1: Let $\boldsymbol{G}(q) = \sum_{p \in \mathcal{P}+\mathcal{P}} \boldsymbol{B}_p q^p$ (with $\boldsymbol{B}_p \in \mathbb{C}^{N \times N}$) be rank-$d$ and positive semidefinite. Then there exists a unique (up to a $d \times d$ left unitary factor) $d \times N$ matrix $\boldsymbol{X}(q) = \sum_{p \in \mathcal{P}} \boldsymbol{A}_k q^p$ such that $\boldsymbol{G}(q) = \boldsymbol{X}(q)^H \boldsymbol{X}(q)$.

The statement has been proved for Laurent matrix polynomials in [44]. For $q = t$ it is equivalent to spectral factorization of polynomial matrices on the real line. Ephremidze [43] proved the full rank version; an entirely parallel construction to those in [43], [44] implies that it holds of rank-deficient matrices.

ACKNOWLEDGMENT

REFERENCES

[1] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM Rev.*, vol. 56, no. 1, pp. 3–69, 2014.

[2] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, Oct. 2006.

[3] N. Krislock and H. Wolkowicz, "Explicit sensor network localization using semidefinite representations and facial reductions," *SIAM J Optim.*, vol. 20, no. 5, pp. 2679–2708, Jan. 2010.

[4] S. Bai and H. Qi, "Tackling the flip ambiguity in wireless sensornetwork localization and beyond," *Digit. Signal Process.*, vol. 55, pp. 85–97, Jul. 2016.

[5] I. Dokmanić, J. Ranieri, and M. Vetterli, "Relax and unfold: Microphone localization with Euclidean distance matrices," in *Proc. 23rd Eur. Signal Process. Conf.,* 2015, pp. 265–269.

[6] I. Dokmanić and M. Vetterli, "Room helps: Acoustic localization with finite elements," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 2617–2620.

[7] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proc. Nat. Acad. Sci.*, vol. 110, no. 30, pp. 12 186–12 191, Jun. 2013.

[8] R. Parhizkar, I. Dokmanić, and M. Vetterli, "Single-channel indoor microphone localization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 1434–1438.

[9] A. Singer, "A remark on global positioning from local distances," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 28, pp. 9507–9511, 2008.

[10] L. Liberti and C. Lavor, "Open research areas in distance geometry," in *Open Problems in Optimization and Data Analysis*. Berlin, Germany: Springer, 2018, pp. 183–223.

[11] B. Hendrickson, "The molecule problem: Exploiting structure in global optimization," *SIAM J Optim.*, vol. 5, no. 4, pp. 835–857, 1995.

[12] B. A. Hendrickson, "The molecule problem: Determining conformation from pairwise distances," Cornell University, Ithaca, NY, USA, Tech. Rep. 90-1159, 1990.

[13] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 12–30, Nov. 2015.

[14] L. J. Guibas, "Kinetic data structures–a state of the art report," in *Robotics: The Algorithmic Perspective*. Boston, MA, USA: A. K. Peters, 1998, Art. no. 191210.

[15] A. Cornejo and R. Nagpal, "Distributed range-based relative localization of robot swarms," in *Algorithmic Foundations of Robotics XI*. Berlin, Germany: Springer, 2015, pp. 91–107.

[16] J. Matthaei, T. Krüger, S. Nowak, and U. Bestmann, "Swarm exploration of unknown areas on mars using slam," in *Proc. Int. Micro Air Vehicle Conf. Flight Competition*, 2013.

[17] J. S. Jaffe *et al.*, "A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics," *Nature Commun.*, vol. 8, p. 14189, 2017, Art. no. 14189.

[18] Q. Wu, A. Tinka, K. Weekly, J. Beard, and A. M. Bayen, "Variational Lagrangian data assimilation in open channel networks," *Water Resour. Res.*, vol. 51, no. 4, pp. 1916–1938, Apr. 2015.

[19] J. Morales, P. Roysdon, Z. M. Kassas, "Signals of opportunity aided inertial navigation," in *Proc. ION GNSS Conf.*, Portland, OR, USA, pp. 1492–1501, 2016.

[20] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[21] M. Kreković, I. Dokmanić, and M. Vetterli, "Echoslam: Simultaneous localization and mapping with acoustic echoes," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 11–15.

[22] A. Hernandez Ruiz, L. Porzi, S. Rota Bulò, and F. Moreno-Noguer, "3D CNNs on distance matrices for human action recognition," in *Proc. ACM Multimedia Conf.*, 2017, pp. 1087–1095.

[23] A. Mucherino and D. S. Gonçalves, "An approach to dynamical distance geometry," in *Proc. Int. Conf. Geometric Sci. Inf.*, Springer, 2017, pp. 821–829.

[24] A. Mucherino, J. Omer, L. Hoyet, P. R. Giordano, and F. Multon, "An application-based characterization of dynamical distance geometry problems," *Optim. Lett.*, pp. 1–15, 2018, doi: 10.1007/s11590-018-1302-6.

[25] "Protein folding—Wikipedia, the free encyclopedia," 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Protein_folding&oldid=862790 016, Accessed: Oct. 19, 2018.

[26] K. Menger, "Untersuchungen über allgemeine Metrik," *Math. Ann.*, vol. 100, no. 1, pp. 75–163, Dec. 1928.

[27] I. J. Schoenberg, "Remarks to maurice frechet's article 'Sur la definition axiomatique D'Une classe D'Espace distances vectoriellement applicable Sur L'Espace De Hilbert'," *Ann. Math.*, vol. 36, pp. 724–732, 1935.

[28] L. M. Blumenthal, *Theory and Applications of Distance Geometry*. Oxford, U.K.: Clarendon Press, 1953.

[29] M. Browne, "The Young-Householder algorithm and the least squares multidimensional scaling of squared distances," *J. Classification*, vol. 4, no. 2, pp. 175–190, 1987.

[30] J. C. Gower, "Euclidean distance geometry," *Math. Sci.*, vol. 7, no. 1, pp. 1–14, 1982.

[31] J. C. Gower, "Properties of Euclidean and non-Euclidean distance matrices," *Linear Algebra Appl.*, vol. 67, pp. 81–97, 1985.

[32] J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*. Palo Alto, CA, USA: Meboo Publishing, 2011.

[33] N. Krislock and H. Wolkowicz, "Euclidean distance matrices and applications," in *Handbook on Semidefinite, Conic and Polynomial Optimization*. Boston, MA, USA: Springer, Jan. 2012, pp. 879–914.

[34] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Math. Program.*, vol. 109, no. 2/3, pp. 367–384, Mar. 2007.

[35] P. Tabaghi and I. Dokmanić, "Real polynomial gram matrices without real spectral factors," 2019, *arXiv:1903.04085*.

[36] L. Blumenthal and B. Gillam, "Distribution of points in n-space," *Amer. Math. Monthly*, vol. 50, no. 3, pp. 181–185, 1943.

[37] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I–convex underestimating problems," *Math. Program.*, vol. 10, no. 1, pp. 147–175, 1976.

[38] E. M. Smith and C. C. Pantelides, "A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex minlps," *Comput. Chem. Eng.*, vol. 23, no. 4/5, pp. 457–478, 1999.

[39] P. H. Schönemann, "A solution of the orthogonal procrustes problem with applications to orthogonal and oblique rotation," Ph.D. dissertation, Dept. Psych., Graduate College Univ., Illinois, 1964.

[40] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 842–849.

[41] C.-C. Wang, C. Thorpe, and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," in *Proc. IEEE Intell. Vehicles Symp.*, 2003, pp. 416–421.

[42] R. F. Bass and K. Gröchenig, "Random sampling of multivariate trigonometric polynomials," *SIAM J. Math. Anal.*, vol. 36, no. 3, pp. 773–795, Jan. 2005.

[43] L. Ephremidze, "An elementary proof of the polynomial matrix spectral factorization theorem," *Proc. Roy. Soc. Edinburgh Section A, Math.*, vol. 144, no. 4, pp. 747–751, 2014.

[44] L. Ephremidze, I. Spitkovsky, and E. Lagvilava, "Rank-deficient spectral factorization and wavelets completion problem," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 13, no. 3, 2015, Art. no. 1550013.

[45] N. Krislock and H. Wolkowicz, "Explicit sensor network localization using semidefinite representations and facial reductions," *SIAM J. Optim.*, vol. 20, no. 5, pp. 2679–2708, 2010.

[46] L. Ephremidze, F. Saied, and I. M. Spitkovsky, "On the algorithmization of Janashia-Lagvilava matrix spectral factorization method," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 728–737, Feb. 2018.

[47] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.

[48] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *J. Control Decis.*, vol. 5, no. 1, pp. 42–60, 2018.

Puoya Tabaghi was born in Hamedan, Iran in 1993. He received the B.S. degrees in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2014, and the M.S. degree in electrical engineering from Colorado State University, Fort Collins, CO, USA, in 2016. He is currently working toward the Ph.D. degree in electrical engineering with Coordinated Science Lab, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

His areas of research interest are distance geometry, theory of machine learning, detection theory and inference.

Ivan Dokmanić received the Diploma in electrical engineering from the University of Zagreb, Zagreb, Croatia, in 2007 and the doctorate degree in computer and communication science from Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2015. Since 2019 he has been an Associate Professor with the Department of Mathematics and Computer Science, University of Basel, Basel, Switzerland, and since 2016 an Assistant Professor of ECE with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA (on leave). He has held visiting positions with EPFL (2017, 2018) and ETHZ (2019). From 2015 to 2016, he worked as a Postdoc with the Institut Langevin and Ecole Normale Supérieure in Paris. During summer 2013, he was with Microsoft Research in Redmond, Washington. Before that he was a Teaching Assistant with the University of Zagreb, a Codec Developer for MainConcept AG, and a Digital Audio Effects Designer for Little Endian Ltd. His research interests lie between data science, physics and signal processing. For his work on room shape reconstruction using sound, he was the recipient of the Best Student Paper Award at ICASSP 2011; in 2014 he received a Google Ph.D. Fellowship. He is a laureate of the EPFL Outstanding Doctoral Thesis Award and the Google Faculty Research Award. In 2019 the European Research Council (ERC) awarded him a Starting Grant.

Martin Vetterli (S'86–M'86–SM'90–F'95) received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Zurich, Switzerland, in 1981, the M.S. degree from Stanford University, Stanford, CA, USA, in 1982, and the Doctorातés Sciences degree from EPF Lausanne (EPFL), Lausanne, Switzerland, in 1986.

He was a Research Assistant with Stanford and EPFL, and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University in New York, where he was last an Associate Professor of electrical engineering and Co-Director of the Image and Advanced Television Laboratory. In 1993, he joined the University of California at Berkeley, where he was a Professor with the Department of Electrical Engineering and Computer Sciences until 1997, and has held an Adjunct Professor position until June 2010.

Since 1995, he has been a Professor of Communication Systems with EPF Lausanne, Switzerland, where he chaired the Communications Systems Division (1996/97), and heads Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research on mobile information and communication systems. He was also a Vice-President with EPFL from October 2004 to February 2011 in charge, among others, of international affairs and computing services. He has held visiting positions at ETHZ (1990) and Stanford (1998). From March 2011 to 2012, he was the Dean of the School of Computer and Communication Sciences, EPFL. From 2013 to 2016, he led the Swiss National Science Foundation and since 2017 he has been a President of EPFL. He has authored more than 190 journal papers on a variety of topics in signal/image processing and communications and holds about 50 patents and patent applications. His research interests include sampling, wavelets, multirate signal processing, computational complexity, signal processing for communications, digital image/video processing, joint source/channel coding, signal processing for sensor networks and inverse problems like acoustic tomography.

He is a Fellow of ACM, a Fellow of EURASIP, and a member of SIAM and NAE. He is on the editorial boards of *Applied and Computational Harmonic Analysis*, the *Journal of Fourier Analysis and Application,* and the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING and has been elected Foreign Member of the NAE in 2015.

He was the recipient of the Best Paper Award of EURASIP in 1984, the Research Prize of the Brown Bovery Corporation (Switzerland) in 1986, the IEEE Signal Processing Society's Senior Paper Awards in 1991, in 1996 and in 2006 (for papers with D. LeGall, K. Ramchandran, and Marziliano and Blu, respectively). He won the Swiss National Latsis Prize in 1996, the SPIE Presidential Award in 1999, the IEEE Signal Processing Technical Achievement Award in 2001, the IEEE Signal Processing Society Award in 2010 for fundamental contributions to signal processing theory, technology and education, and the IEEE Jack S. Kilby Signal Processing Medal in 2017. He is an ISI highly cited researcher in engineering. He was a member of the Swiss Council on Science and Technology from 2000 to 2003.

He was a Plenary Speaker at various conferences (e.g. IEEE ICIP, ICASSP, ISIT) and is the co-author of three books with J. Kovacevic, "*Wavelets and Subband Coding*," 1995, with P. Prandoni "*Signal Processing for Communications*," 2008 and with J. Kovacevic and V.K. Goyal, "*Foundations of Signal Processing*," 2015.