2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Learning a Dynamical System Model for a Spatiotemporal Field Using a Mobile Sensing Robot

Xiaodong Lan and Mac Schwager

Abstract— **This paper considers the problem of learning dynamic spatiotemporal fields using measurements from a single sensing robot. We first introduce a widely used parametric dynamical model for the spatiotemporal field. We then propose a motion strategy that can be used by a single sensing robot to collect sensor measurements about the field. Our motion strategy is designed to collect sufficient information at repeated locations but different times along an arbitrarily chosen periodic trajectory. In conjunction with the measurements collected, we propose a new learning algorithm based on subspace identification to learn the parameters of the dynamical model. We prove that the parameters learned by our algorithm will converge to the true parameters as long as the periodic trajectory is uniformly observable and the number of measurements collected by the robot goes to infinity. The performance of our algorithm is demonstrated in numerical simulations.**

I. Introduction

In this paper we investigate the problem of learning how a stable dynamic spatiotemporal field changes over time. For example, consider using a single sensing robot to monitor the temperature in an area of the ocean, the concentration of a particular chemical over a lake, or the radioactivity in a region around a nuclear accident. In order to monitor such fields properly [1, 2], it is important that we know how the field varies both spatially and temporally. We first introduce a dynamical model for these kinds of spatiotemporal fields, similar to the model proposed in our previous work [3, 4]. However, in our previous work we assumed the dynamics of the field were known, and our task was to estimate the state of the field. In this work, we assume the dynamics themselves, as well as the state, are unknown. Specifically, we propose a motion strategy for a single sensing agent to move around an arbitrarily chosen periodic trajectory to collect sufficient measurements about the field. Then based on the measurements collected using the motion strategy, we propose an associated learning algorithm based on subspace identification to learn how the field changes in time.

Subspace identification is a well-studied subject [5, 6, 7, 8]. It has been successful in identifying linear time invariant (LTI) systems. This type of approach first uses singular value decomposition (SVD) to estimate the model dimensionality and the underlying state sequence, and then it derives parameter estimates using least squares. The main three subspace identification algorithms are canonical variate analysis (CVA) [9], multi-variable output-error state space (MOESP) [10] and numerical algorithms for subspace state space system identification (N4SID) [11]. The similarities on consistent estimation and asymptotic properties between these algorithms were discussed in [12]. A unifying theorem was proposed in [13] to state the relationships between these three algorithms. A good overview about subspace identification is given in [14, 15]. Recently, the authors in [16] proposed a subspace identification based algorithm to obtain a minimal-dimension state-space model in the canonical form, thus connecting the subspace identification method with the prediction error method(PEM)[7], which is used to estimate the classical transfer function parameters.

Our learning algorithm is different from the above algorithms in two respects. First, we design motion strategies that critically influence the data collection process itself, while the data in classical subspace ID is considered to be given. Secondly, data collected by moving agents leads to a linear time-varying system, while subspace identification algorithms typically deal with linear time-invariant (LTI) systems. The motion strategies executed by the mobile robot are along a periodic trajectory. Our use of periodic trajectory is inspired by [17, 18]. In [17], the authors use the classical LTI subspace identification to get the column space of the time-varying observability matrices and transform them into the same state basis to identify the parameters of the system. In [18], the authors introduce a factorization method to form a predictor that predicts the outputs of the system. Using this predictor, LTI subsapce identification is applied to estimate the state sequence from which the LPV system matrices can be constructed. Different from the algorithms described above, in this paper we deal with two subproblems: how to collect data and how to use the collected data to learn the dynamics of the spatiotemporal field. Specifically, we use mobile point sensors to collect noisy measurements of the field at the waypoints of a uniformly observable periodic trajectory, then we use singular value decomposition (SVD) to discover the relationship between the hidden states and the measurements by correlating the past and future measurements.

The rest of this paper is organized as follows. The problem is formulated in Section II. Our learning algorithms are described in Section III. Section IV presents the results of numerical simulations, and we discuss conclusions in Section V.

This work was supported in part by ONR grant N00014-12-1-1000. We are grateful for this support.

X. Lan is with the Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA, xlan@bu.edu.

M. Schwager is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA, schwager@stanford.edu.

978-1-5090-5994-2/$31.00 ©2017 AACC

II. Problem Formulation

In this section we formulate the problem considered in this paper.

A. Spatiotemporal Field Model

Consider a dynamic scalar field $\phi(q,t) : \mathcal{Q} \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}$, where $\mathcal{Q} \subset \mathbb{R}^2$ is the domain of interest in the environment, and $q \in \mathcal{Q}$ is an arbitrary point in the domain. We model the spatiotemporal field as a linear combination of *static* spatial basis functions $C(q) : \mathcal{Q} \mapsto \mathbb{R}^{1 \times n}$, where $C(q) = [c_1(q) \cdots c_n(q)]$, with *time-changing* weights $x_t : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^n$, where $x_t = [x_t^1 \cdots x_t^n]^T$, so that

$$\phi(q,t) = C(q)x_t. \tag{1}$$

This is a common way to represent spatiotemporal fields [19, 20]. Note that $C(q)$ can be any arbitrary set of spatial basis. In this paper, we choose Gaussian radial basis functions where the ith element of $C(q)$ is given by $c_i(q) = Ke^{-\|q-q_i\|^2/2\sigma_c^2}$, where q_i is the center position of the basis function. Other choices include Fourier basis and orthogonal polynomial basis. To model the time changing weights x_t, we let the weights be driven by a linear discrete-time stochastic system of the form

$$x_{t+1} = Ax_t + w_t, \tag{2}$$

where A is a $n \times n$ matrix and $w_t \in \mathbb{R}^n$ is Gaussian white noise process with distribution $w_t \sim N(0,Q)$, and Q is the covariance matrix which is positive definite and symmetric. Let x_0 the initial state vector at time $t = 0$. It is assumed that x_0 is Gaussian distributed, independent of w_t. We also assume that A is stable, that is, all its eigenvalues are strictly inside the unit circle of the complex plane. The pair $(A, Q^{1/2})$ is assumed controllable, that is, $rank\big([Q^{1/2}, AQ^{1/2}, A^2Q^{1/2}, \ldots, A^{n-1}Q^{1/2}]\big) = n$. Note that using this model and Gaussian spatial basis functions, the unknown parameters are the scaling coefficient K, basis function centers q_i, deviation parameter σ_c, noise covariance matrix Q and the eigenvalues of A matrix. Here we will consider the eigenvalues of the A matrix. The reason is that the eigenvalues capture the underlying dynamics of such a dynamic system. As long as we get the eigenvalues, we can construct different A matrices and different state sequences to represent the same dynamics, and these A matrices are similar to each other. For different A matrices and state sequences, we will get different K, q_i and σ_c parameters, but the predicted field values will be the same.

B. Mobile Sensor Model

In order to learn the parameters listed above, we need to collect data. We will use a single *point* sensor to take measurements about the field. Here *point* means the sensor can only measure the field value at its current position. In order to take measurments at other positions, it has to move to other positions and take measurements there. Assume the sensor's position is at p_t, the sensor model is given by

$$y_t = \phi\big(p_t, t\big) + v_t = C\big(p_t\big)x_t + v_t \tag{3}$$

where $y_t \in \mathbb{R}$ is the noisy measured field value, $v_t \in \mathbb{R}$ is a Gaussian white noise process, independent of x_t, and with Gaussian distribution $v_t \sim N(0,R)$.

C. Learning Accuracy Metric

As discussed above, our goal is to use a single mobile sensor to collect measurements about the field and learn the parameters of the model introduced in (1) and (2). In order to measure the performance of a learning algorithm, we need to define a metric. An intuitive metric would be calculating the sum of errors of all the parameters to their true parameters. As discussed in part II-A, the scaling coefficient K, basis function centers q_i and deviation parameter σ_c depend on the A matrix, and it is the eigenvalues of the A matrix that capture the decay rates and natural frequencies of the field dynamics [21]. Therefore, an error metric based on the differences between the learned eigenvalues and the true eigenvalues would be legitimate.

$$\text{Error} = \sum_{i=1}^{n} \{|\text{real}\left(\widehat{\lambda}_i\right) - \text{real}\left(\lambda_i\right)| + |\text{imag}\left(\widehat{\lambda}_i\right) - \text{imag}\left(\lambda_i\right)|\}, \tag{4}$$

where $\widehat{\lambda}_i$ and λ_i are the learned eigenvalues and the true eigenvalues, respectively, and $\text{real}\left(\lambda_i\right)$ and $\text{imag}\left(\lambda_i\right)$ are the real parts and the imaginary parts of eigenvalue λ_i, respectively.

III. Learning Algorithm

In this section we propose an algorithm based on subspace identification [6, 7, 8] to learn the eigenvalues of the A matrix. We first introduce the motion strategies that the sensor will use to collect measurements.

A. Motion Strategy

The mobile sensor will move along a periodic path $\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{T-1})$ with period T. At each time t, a mobile sensor located at $p_t = \sigma_k$ can either move forward one step, $p_{t+1} = \sigma_{k+1}$ (where we define $\sigma_{T+1} := \sigma_1$), or it can stay in place, $p_{t+1} = \sigma_k$. Here we assume that the distance between two consecutive waypoints is bounded by a maximum distance and we are able to control the sensor to move from one waypoint to another. A motion strategy is a periodic sequence of move and stay actions for a mobile sensor along σ. Before we move on, we first give a definition that will be used to explain what trajectory to use for the mobile sensor to collect data. This definition is widely used for linear time-varying systems [22].

Definition 1 (Uniformly Observable): The tuple $(A, C(\cdot), \sigma)$ is uniformly observable if $\exists \delta \in \mathbb{Z}^+$ and positive constants β_1 and β_2 such that $0 \prec \beta_1 I \preceq (O_t^{\delta+1})^T O_t^{\delta+1} \preceq \beta_2 I, \forall t$, where $O_t^{\delta+1} = \big[C(\sigma_t);\ C(\sigma_{t+1})A;\ \ldots;\ C(\sigma_{t+\delta})A^{\delta}\big]$, and we call the path σ a uniformly observable path.

Here square bracket notation with semicolon as the delimiter $[a; b]$ indicates that the first row is row vector a and the second row is row vector b.

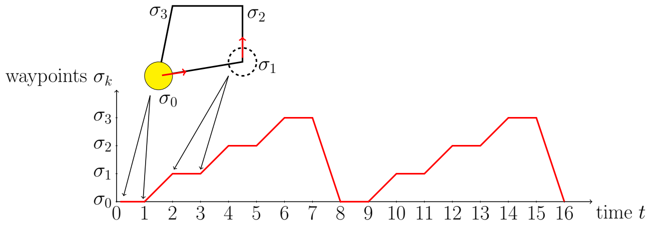


Fig. 1. An example trajectory along a periodic path with period 4 in the time frame. The sensor stays at one waypoint for $r = 2$ steps.

In order to use data from the mobile sensor to learn the eigenvalues of A, the data must contain sufficient information about x_t. Requiring that $(A, C(\cdot), \sigma)$ be uniformly observable guarantees that there is sufficient information content in the data along the sensor's path. The requirement to use a uniformly observable path to collect data also means that the classical subspace identification algorithms do not apply in our case. Classical subspace identification requires C to be a constant matrix, however in our case it changes in time according to $C(\sigma_k)$. This time varying nature is inherent in our problem, since our sensors must move in order to collect spatially distributed information about the field, but moving itself makes the $C(\sigma_k)$ matrix change in time.

Next we will explain how we control the mobile sensor to move along the periodic path σ and take measurements of the field at the waypoints of the trajectory. To simplify notation, we let $C_k = C(\sigma_k)$. Let the mobile sensor start at waypoint σ_0 at time 0 and stay at point σ_0 for r steps, where $r \geq 2$. Denote the corresponding sensor outputs by $(y_0, y_1, \ldots, y_{r-1})$. Then let the mobile sensor move to the next waypoint σ_1 and stay there for r steps with outputs $(y_r, y_{r+1}, \ldots, y_{2r-1})$. The mobile sensor will repeat this *move and stay for r steps* process along the periodic trajectory. Note that $r = 1$ (the mobile sensor moves at every time step) is prohibited. Moving in this way, the mobile sensor will come back to its original waypoint every Tr steps. That is, the trajectory's period is Tr. Hence, $C_k = C_{k+Tr}$. An example trajectory is given in Figure 1.

Next we write the measurements collected by the mobile sensor in a block matrix form. First, define a stacked output column vector as

$$\mathbf{y}_{t,r|t+d-1} := [y_t; y_{t+1}; \ldots y_{t+r-1}; | y_{t+r}; y_{t+r+1}; \ldots y_{t+2r-1}; | \ldots | y_{t+(K_d-1)r}; \ldots y_{t+K_d r-1}] \quad (5)$$

where $d = K_d * r, K_d \in \mathbb{Z}^+$. Both d and K_d are user-defined parameters. An example stacked output column vector when $r = 2$ and $K_d = 3$ is given as follows.

$$\mathbf{y}_{0,2|5} := [y_0; y_1; y_2; y_3; y_4; y_5]$$

Next we define an output block matrix, which is usually called the output block Hankel matrix.

$$Y_{t,r|t+d-1} := [\mathbf{y}_{t,r|t+d-1} \quad \mathbf{y}_{t+Tr,r|t+Tr+d-1} \quad \cdots \mathbf{y}_{t+(N-1)Tr,r|t+(N-1)Tr+d-1}] \quad (6)$$

where N is the number of columns in $Y_{t,r|t+d-1}$ and $N \gg n$, $Y_{t,r|t+d-1} \in \mathbb{R}^{d \times N}$. Note that for two adjacent elements in any one row of $Y_{t,r|t+d-1}$, they correspond to the field measurements taken at the same waypoint, and the elapsed time between these two measurements is Tr steps. Next we define a stacked coefficient matrix as follows.

$$X_{t,r} = \begin{bmatrix} x_t & x_{t+Tr} & \ldots & x_{t+(N-1)Tr} \end{bmatrix}, \quad X_{t,r} \in \mathbb{R}^{n \times N} \quad (7)$$

We also define a matrix consisting of $(A, C(\cdot))$ in (8). This is usually called the extended observability matrix.

$$O_{t,r|d} := [C_t; \ldots C_{t+r-1}A^{r-1}; | C_{t+r}A^r; \ldots | C_{t+(K_d-1)r}A^{(K_d-1)r}; \ldots C_{t+K_d r-1}A^{K_d r-1}] \quad (8)$$

where $O_{t,r|d} \in \mathbb{R}^{d \times n}$. Note that here $C_t = C_{t+1} = \ldots = C_{t+r-1} = C_k$ if the sensor stays at waypoint σ_k at time t. After defining these block matrices, and according to (2) and (3), we can write the relationship between $Y_{t,r|t+d-1}$, $X_{t,r}$ and $O_{t,r|d}$ as

$$Y_{t,r|t+d-1} = O_{t,r|d} X_{t,r} + \mathit{Noise}, \quad (9)$$

where the matrix *Noise* term comes from the process noise w_t and measurement noise v_t. Note that $O_{t,r|d}$ contains information about the A matrix. By equation (9), we can separate $O_{t,r|d}$ from the sensor outputs if $\text{rank}(X_{t,r}) = n$. That is, there should be at least n independent columns in $Y_{t,r|t+d-1}$ and in this case $O_{t,r|d}$ will have the same column space as $Y_{t,r|t+d-1}$. Then we can perform a singular value decomposition (SVD) for $Y_{t,r|t+d-1}$ to get the column space of $O_{t,r|d}$.

B. Learning Algorithm

By equation (9), we can get the column space of $O_{t,r|d}$ from the sensor outputs. We can also get the row space of $X_{t,r}$ from the sensor outputs if $\text{rank}(O_{t,r|d}) = n$. It can be proved that as long as the data are collected along a uniformly observable periodic trajectory, the rank of $O_{t,r|d}$ will be equal to n. Next we prove a lemma which indicates that we can reconstruct the coefficients x_t from the sensor output data. This lemma will be important for us to prove the main theorem of this paper which states how to use the collected data to learn the eigenvalues of A. Also, the reconstructed x_t will be helpful when we learn the parameters of the spatial basis functions $C(q)$. Here we want to point out that the reconstructed x_t from the sensor outputs by Lemma 1 is the same as the estimated state given by Kalman filter [23]. Hence Lemma 1 can be seen as a different form of the Kalman filter. First we define some notation required to state the lemma. The operator $E[\cdot]$ is the expected value operator.

Definition 2: $\widehat{x}_t := E[x_t]$, $\Sigma^\infty := \lim_{t\to\infty} E[x_t x_t^T]$, $\Sigma_t := E[(x_t - \widehat{x}_t)(x_t - \widehat{x}_t)^T]$, $G_t := E[x_{t+1} y_t^T]$, $\Lambda_i^t := E[y_{t+i} y_t^T]$, $\Lambda_0^t := E[y_t y_t^T]$, $\Lambda_{-i}^t := E[y_{t-i} y_t^T]$, where $i \in \mathbb{Z}^+$.

If A is stable, then it is easy to verify that:

$$\Sigma^{\infty} = A\Sigma^{\infty}A^T + Q, \quad G_t = A\Sigma^{\infty}C_t^T + S,$$
$$\begin{cases} \Lambda_i^t = C_{t+i}A^{i-1}G_t \\ \Lambda_0^t = C_t\Sigma^{\infty}C_t^T + R \\ \Lambda_{-i}^t = G_{t-i}^T(A^{i-1})^T C_t^T \end{cases} \tag{10}$$

Definition 3: $\Delta_i^t := (A^{i-1}G_t \quad A^{i-2}G_{t+1} \quad \dots \quad AG_{t+i-2} \quad G_{t+i-1})$,

$$L_i^t = \begin{pmatrix} \Lambda_0^t & \Lambda_{-1}^{t+1} & \Lambda_{-2}^{t+2} & \dots & \Lambda_{1-i}^{t+i-1} \\ \Lambda_1^t & \Lambda_0^{t+1} & \Lambda_{-1}^{t+2} & \dots & \Lambda_{2-i}^{t+i-1} \\ \dots & \dots & \dots & \dots & \dots \\ \Lambda_{i-1}^t & \Lambda_{i-2}^{t+1} & \Lambda_{i-3}^{t+2} & \dots & \Lambda_0^{t+i-1} \end{pmatrix} \tag{11}$$

$$N_i^t = \begin{pmatrix} \Lambda_i^t & \Lambda_{i-1}^{t+1} & \Lambda_{i-2}^{t+2} & \dots & \Lambda_1^{t+i-1} \\ \Lambda_{i+1}^t & \Lambda_i^{t+1} & \Lambda_{i-1}^{t+2} & \dots & \Lambda_2^{t+i-1} \\ \dots & \dots & \dots & \dots & \dots \\ \Lambda_{2i-1}^t & \Lambda_{2i-2}^{t+1} & \Lambda_{2i-3}^{t+2} & \dots & \Lambda_i^{t+i-1} \end{pmatrix} \tag{12}$$

Lemma 1: Define $P_t := \Sigma^{\infty} - \Sigma_t$, and given $\widehat{x}_0 = 0$ and $P_0 = 0$, then the Kalman filter state estimate $\widehat{x}_t$ defined by the following recursive formulas:

$$\begin{aligned} \widehat{x}_t &= A\widehat{x}_{t-1} + K_{t-1}(y_{t-1} - C_{t-1}\widehat{x}_{t-1}), \\ K_{t-1} &= (G_{t-1} - AP_{t-1}C_{t-1}^T)(\Lambda_0^{t-1} - C_{t-1}P_{t-1}C_{t-1}^T)^{-1} \\ P_t &= AP_{t-1}A^T + (G_{t-1} - AP_{t-1}C_{t-1}^T) \\ &\quad \times (\Lambda_0^{t-1} - C_{t-1}P_{t-1}C_{t-1}^T)^{-1}(G_{t-1} - AP_{t-1}C_{t-1}^T)^T, \end{aligned}$$

can be written as

$$\widehat{x}_t = \Delta_t^0 (L_t^0)^{-1} \left[y_0; y_1; \dots; y_{t-1}\right] \tag{13}$$

and P_t can be written as

$$P_t = \Delta_t^0 (L_t^0)^{-1} (\Delta_t^0)^T \tag{14}$$

The proof of this lemma is omitted here and can be found online at http://people.bu.edu/xlan/files/subIDproof.pdf. Other proofs in this paper are also omitted and can also be found in the same place.

Remark 1: In Lemma 1, (13) indicates that the Kalman filter state estimate $\widehat{x}_t$ can be expressed as a linear combination of the past outputs $y_0, y_1, \dots, y_{t-1}$. Note that Kalman filter only uses partial information of the outputs. It can be verified that the following is true: $\widehat{x}_{i+t} = \Delta_i^t (L_i^t)^{-1} \left[y_t; \quad y_{t+1}; \quad \dots; y_{t+i-1}\right]$. Hence, the estimate for the block coefficient matrix $X_{t,r}$ defined in (7) can be written as

$$\widehat{X}_{t,r} = \Delta_d^t (L_d^t)^{-1} Y_{t,r|t+d-1}. \tag{15}$$

Based on this lemma, next we prove a theorem which states how we can use the data collected along a uniformly observable periodic path σ to learn the eigenvalues of the A matrix.

Define an operator on two matrices $\Pi_1 \in \mathbb{R}^{i\times k}$ and $\Pi_2 \in \mathbb{R}^{j\times k}$ by Π_1/Π_2, which denotes the orthogonal projection of the row space of Π_1 onto the row space of Π_2. Hence, $\Pi_1/\Pi_2 = \Pi_1\Pi_2^T(\Pi_2\Pi_2^T)^{\dagger}\Pi_2$, where $\cdot^{\dagger}$ denotes the Moore-Penrose pseudo-inverse.

Theorem 1: Given an unknown stable scalar spatiotemporal field modeled by (1) and (2), stack the measurements collected by one mobile sensor along an arbitrarily chosen uniformly observable periodic trajectory in a block Hankel matrix as (6), define $H_t^d := Y_{t+d,r|t+2d-1}/Y_{t,r|t+d-1}$, then as the number of measurements $N \to \infty$, we have

$$H_t^d = O_{t,r|d}\widehat{X}_{t,r} \tag{16}$$

If we do singular value decomposition for H_t^d:

$$H_t^d = (U_1 \quad U_2)\begin{pmatrix} S_1 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} = U_1 S_1 V_1^T$$

then the number of basis functions in the spatiotemporal fields is equal to the number of non-zero singular values. Also,

$$\widehat{O}_{t,r|d} = U_1 S_1^{1/2}, \quad \widehat{X}_{t,r}^{\text{svd}} = S_1^{1/2} V_1^T, \tag{17}$$

where $\widehat{O}_{t,r|d}$ and $\widehat{X}_{t,r}^{\text{svd}}$ are the learned extended observability matrix and the learned matrix for the estimate of the block coefficient matrix, respectively.

Remark 2: The learned matrices $\widehat{O}_{t,r|d}$ and $\widehat{X}_{t,r}^{\text{svd}}$ are not exactly the same as the original matrices $O_{t,r|d}$ and $\widehat{X}_{t,r}$, respectively. They are the same up to a similarity transformation, that is, $\widehat{O}_{t,r|d}\Gamma = O_{t,r|d}$ and $\widehat{X}_{t,r}^{\text{svd}} = \Gamma\widehat{X}_{t,r}$, where Γ is a non-singular similarity transformation matrix.

By theorem 1, we can learn the extended observability matrix $\widehat{O}_{t,r|d}$ from the measurements collected along a uniformly observable periodic trajectory. Next we will talk about how to use the learned extended observability matrix $\widehat{O}_{t,r|d}$ to learn the A matrix.

Consider the extended observability matrix defined in (8), we can extract two submatrices from it:

$$\begin{aligned} \bar{O}_{t,r|d} &:= [C_t; \dots; C_{t+r-2}A^{r-2}; | C_{t+r}A^r; \dots | \\ &\qquad C_{t+(K_d-1)r}A^{(K_d-1)r}; \dots; C_{t+K_d r-2}A^{K_d r-2}] \end{aligned} \tag{18}$$

$$\begin{aligned} \underline{O}_{t,r|d} &:= [C_{t+1}A; \dots; C_{t+r-1}A^{r-1}; | C_{t+r+1}A^{r+1}; \dots | \\ &\qquad C_{t+(K_d-1)r+1}A^{(K_d-1)r+1}; \dots; C_{t+K_d r-1}A^{K_d r-1}]. \end{aligned} \tag{19}$$

Note that (18) is the submatrix which extracts the upper $r-1$ blocks in each block separated by the vertical line while (19) is the submatrix which extracts the lower $r-1$ blocks. For each block separated by the vertical line, the C matrices are equal since the sensor stays at the same waypoint to take measurements. So the relationship between the two submatrices is as follows.

$$\bar{O}_{t,r|d}A = \underline{O}_{t,r|d} \Rightarrow A = \bar{O}_{t,r|d}^{\dagger}\underline{O}_{t,r|d} \tag{20}$$

Equation (20) always holds as long as $\bar{O}_{t,r|d}$ is full column rank. In order to guarantee that $\bar{O}_{t,r|d}$ is full column rank, we just need to make sure that $(r-1)K_d \geq n$. Therefore, we can learn A from $\widehat{O}_{t,r|d}$ by using $\widehat{A} = \bar{\widehat{O}}_{t,r|d}^{\dagger}\widehat{\underline{O}}_{t,r|d}$, where $\bar{\widehat{O}}_{t,r|d}$ and $\widehat{\underline{O}}_{t,r|d}$ are the corresponding submatrices of $\widehat{O}_{t,r|d}$. Note that the learned $\widehat{A}$ may not be the same as the original A matrix. They are equivalent up to a similarity transformation.

C. Learning Spatial Basis Functions

In this section we describe how to learn the parameters of the spatial basis functions $C(q)$. By Theorem 1, when we learn the extended observability matrix, we also get an estimate of the coefficient matrix which is given by $\widehat{X}_{t,r}^{\text{svd}}$, see equation (17). Note that $\widehat{X}_{t,r}^{\text{svd}} = [\hat{x}_t, \hat{x}_{t+Tr}, \hat{x}_{t+2Tr}, \dots, \hat{x}_{t+(N-1)Tr}]$. That is, it consists of the estimated state sequence in every Tr steps. Based on equation (3), we can learn the spatial basis function parameters using least square curve fitting. For example, if the basis functions are Gaussians, $c_i(p_t) = Ke^{-\|p_t - q_i\|^2/2\sigma_c^2}$, then we can learn K, q_i and σ_c by solving the nonlinear program

$$(q_1^*, \dots, q_n^*, K^*, \sigma_c^*) = \underset{q_1,\dots,q_n,K,\sigma_c}{\arg\min} \sum_{j=1}^{N} \|y_{t+(j-1)Tr} - C_t \hat{x}_{t+(j-1)Tr}\|^2. \quad (21)$$

In the case of Gaussian basis functions, the parameters appear nonlinearly, and the objective function is nonconvex, hence we must use a nonlinear optimization algorithm. Other choices of spatial bases may lead to objective functions that are convex in the unknown parameters, allowing for the use of more efficient optimization algorithms. This is an interesting topic for future work.

IV. Numerical Simulations

In this section, we consider the performance of our algorithm by applying the proposed algorithm to a simulated spatiotemporal field with known parameters. We simulate a spatiotemporal field in a 50×50 unit square region and we use Gaussian radial basis functions with $c_i(q) = Ke^{\frac{-\|q-q_i\|}{2\sigma_c^2}}$. We arbitrarily choose a uniformly observable periodic trajectory to collect data. One naive choice would be a TSP tour visiting each basis center in a predefined order. These TSP tours will be guaranteed uniformly observable. The search of optimal uniformly observable trajectory to collect data remains an open question. We then apply the proposed motion strategy to control the sensing robot to move along the periodic path and take measurements at the waypoints of the path. These measurements will serve as the only inputs to our algorithm. Based on the collected measurements, we first apply the proposed algorithm to learn the eigenvalues of the A matrix, then we use the nonlinear curve fitting solver *lsqcurvefit* with multiple starting points in MATLAB to solve (21), thus learning the corresponding parameters for the Gaussian basis functions. We run 100 Monte Carlo experiments and each experiment has different noise sequence. The simulation information can be accessed at http://people.bu.edu/xlan/files/simInfo.zip. The learned eigenvalues are shown in Figure 2. This figure shows the learned eigenvalues across all the runs for a first order system and a sixth order system. The distribution of the learned eigenvalue errors is shown in Table I. This table shows that as more data is used, the learning error tends to be smaller, especially for high dimensional systems.

Next we use the procedure proposed in Section III-C to learn the spatial basis functions. In this simulated spatiotemporal field, the basis functions are Gaussian radial basis functions. Because the learned coefficient matrix $\widehat{X}_{t,r}^{\text{svd}}$ is not the same as the original coefficient matrix $\widehat{X}_{t,r}$ (see Equation (17)), the learned Gaussian basis function parameters will not equal the original parameters that we used to generate the spatiotemporal field. In order to verify that we learned these parameters correctly, we calculate the predicted field values based on the learned parameters and compare the predicted field values with the measured field values. We calculate the mean error between the predicted field values and the measured field values at 100 random locations. We also run 100 Monte Carlo simulations and calculate the average mean error. The result is shown in Table II. From this table, we can see that as more data is used to learn the model, the mean error between the predicted fields and the measured fields will be smaller. An example of the predicted fields compared with the measured fields are shown in Figure 3.

V. Conclusions and Future Work

In this paper we proposed an algorithm to collect noisy measurements of a spatiotemporal field using a single mobile sensor, and use the collected data to learn a linear state space model for the dynamics of the field. We first introduce a model for the field. Then we control a mobile sensor to move along a uniformly observable periodic trajectory to collect measurements of the field at the waypoints of the trajectory. By correlating the future measurements and the past measurements, we discover the relationship between the hidden states and the measurements and learn the dynamics of the field. We also use nonlinear optimization to learn the parameters of the spatial basis functions for the field. We verify the performance of our algorithm by considering a simulated spatiotemporal field, for which we know the ground truth. In the future, we plan to extend this work to use multiple sensing robots to collect measurements and learn the field. Finding the optimal periodic uniformly observable trajectory to collect data is another interesting future research direction.

References

[1] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *Robotics, IEEE Transactions on*, vol. 28, no. 2, pp. 410–426, 2012.

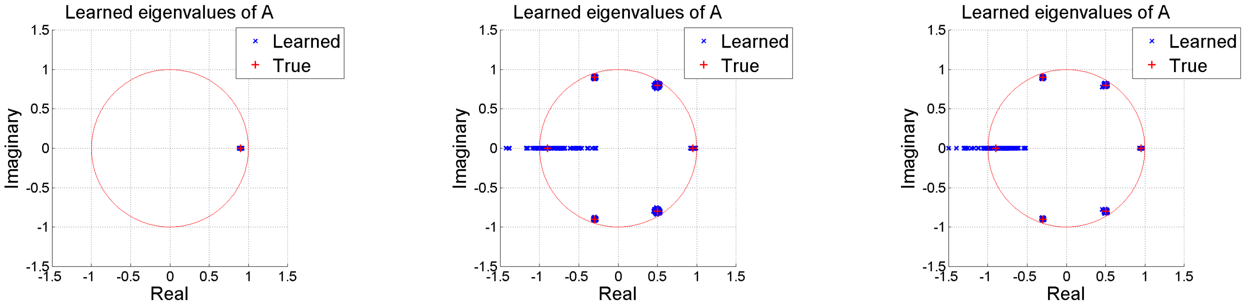
[2] X. Lan and M. Schwager, "Rapidly-exploring random cycles: Persistent estimation of spatio-temporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, 2016.

[3] X. Lan and M. Schwager, "Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 2415–2420, IEEE, 2013.

[4] X. Lan and M. Schwager, "A variational approach to trajectory planning for persistent monitoring of spatiotemporal fields," in *2014 American Control Conference (ACC)*, 2014.

[5] S. M. Siddiqi, B. Boots, and G. J. Gordon, "A constraint generation approach to learning stable linear dynamical systems," in *Proceedings of Advances in Neural Processing Information Systems (NIPS 2007)*, vol. 20, pp. 1329–1336, 2008.

[6] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: theory, implementation, applications*. Kluwer academic publishers, 1996.



(a) Learned eigenvalue for a one dimensional system, $N = 1000$, $r = 3$, and $d = 15$.

(b) Learned eigenvalues for a sixth dimensional system, $N = 5000$, $r = 3$, and $d = 15$.

(c) Learned eigenvalues for a sixth dimensional system, $N = 10000$, $r = 3$, and $d = 15$.

Fig. 2. Learned eigenvalues over 100 Monte Carlo runs for simulated spatiotemporal fields with known ground truth.

TABLE I

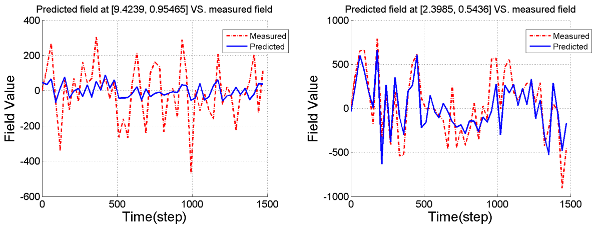
DISTRIBUTION OF LEARNING ERROR OVER 100 RUNS.

Simulation Parameters	Error < 0.1	Error < 0.2	Error $>= 0.2$
$n = 1$, $N = 100$, $r = 3$, $d = 15$	100%	100%	0%
$n = 1$, $N = 500$, $r = 3$, $d = 15$	100%	100%	0%
$n = 1$, $N = 1000$, $r = 3$, $d = 15$	100%	100%	0%
$n = 6$, $N = 1000$, $r = 3$, $d = 15$	8%	15%	85%
$n = 6$, $N = 5000$, $r = 3$, $d = 15$	26%	61%	39%
$n = 6$, $N = 10000$, $r = 3$, $d = 15$	43%	71%	29%

TABLE II

AVERAGE MEAN ERROR BETWEEN PREDICTED FIELDS AND MEASURED FIELDS AT 100 RANDOM LOCATIONS OVER 100 RUNS.

Simulation Parameters	Mean Error
$n = 1$, $N = 100$, $r = 3$, $d = 15$	-1.4245
$n = 1$, $N = 500$, $r = 3$, $d = 15$	-0.5023
$n = 1$, $N = 1000$, $r = 3$, $d = 15$	0.1266
$n = 6$, $N = 1000$, $r = 3$, $d = 15$	1.2750
$n = 6$, $N = 5000$, $r = 3$, $d = 15$	-0.6340
$n = 6$, $N = 10000$, $r = 3$, $d = 15$	-0.0805



(a) Example of predicted field versus measured field for $n = 1$.

(b) Example of predicted field versus measured field for $n = 6$.

Fig. 3. Predicted field values based on the learned model parameters are compared with the measured field values at random locations. The predicted field represents a Kalman-filtered version of the measured field.

[7] L. Ljung, *System identification: theory for the user*. Prentice Hall PTR, 1999.

[8] T. Katayama, *Subspace methods for system identification*. Springer Science & Business Media, 2006.

[9] W. E. Larimore, "Canonical variate analysis in identification, filtering, and adaptive control," in *The 29th IEEE International Conference on Decision and Control*, pp. 596–604, IEEE, 1990.

[10] M. Verhaegen, "Identification of the deterministic part of mimo state space models given in innovations form from input-output data," *Automatica*, vol. 30, no. 1, pp. 61–74, 1994.

[11] P. Van Overschee and B. De Moor, "N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, no. 1, pp. 75–93, 1994.

[12] M. Jansson and B. Wahlberg, "On consistency of subspace methods for system identification," *Automatica*, vol. 34, no. 12, pp. 1507–1519, 1998.

[13] P. Van Overschee and B. De Moor, "A unifying theorem for three subspace system identification algorithms," *Automatica*, vol. 31, no. 12, pp. 1853–1864, 1995.

[14] M. Viberg, "Subspace-based methods for the identification of linear time-invariant systems," *Automatica*, vol. 31, no. 12, pp. 1835–1851, 1995.

[15] S. J. Qin, "An overview of subspace identification," *Computers & chemical engineering*, vol. 30, no. 10, pp. 1502–1513, 2006.

[16] G. Mercère and L. Bako, "Parameterization and identification of multivariable state-space systems: A canonical approach," *Automatica*, vol. 47, no. 8, pp. 1547–1555, 2011.

[17] F. Felici, J.-W. Van Wingerden, and M. Verhaegen, "Subspace identification of mimo lpv systems using a periodic scheduling sequence," *Automatica*, vol. 43, no. 10, pp. 1684–1697, 2007.

[18] J.-W. Van Wingerden and M. Verhaegen, "Subspace identification of bilinear and lpv systems for open-and closed-loop data," *Automatica*, vol. 45, no. 2, pp. 372–381, 2009.

[19] N. Cressie, "The origins of kriging," *Mathematical Geology*, vol. 22, no. 3, pp. 239–252, 1990.

[20] N. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1993.

[21] T. Kailath, *Linear systems*, vol. 1. Prentice-Hall Englewood Cliffs, NJ, 1980.

[22] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Dover Publications, 2007.

[23] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.