Learning a Linear Dynamical System Model for Spatio-temporal Fields Using a Group of Mobile Sensing Robots

Xiaodong Lan [a], Mac Schwager [b],

[a] *Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA*

[b] *Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA*

Abstract

This paper considers the problem of learning dynamic spatiotemporal fields using sensor measurements from a group of mobile sensing robots. We first introduce a general dynamical model for the spatiotemporal field. We then propose a family of motion strategies that can be used by a group of mobile robotic sensors to move along a uniformly observable periodic trajectory to collect point samples about the field. Our motion strategies are designed to collect enough information from enough locations at enough different times for the robots to learn the dynamics of the field. In conjunction with these motion strategies, we propose a new learning algorithm based on subspace identification to learn the parameters of the dynamical model. We prove that as the number of data collected by the robots goes to infinity, the parameters learned by our algorithm will converge to the true parameters. The performance of our algorithm is demonstrated in numerical simulations and verified by considering the surface water temperature in the Caribbean Sea, which is a physical spatio-temporal field.

Key words: Subspace Identification; Spatio-temporal Fields.

1 Introduction

In this paper we investigate the problem of learning how a stable dynamic spatiotemporal field changes over time. For example, consider using a group of mobile sensing robots 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 [Smith et al., 2012, Lan and Schwager, 2015], it is important that we know how the field varies both spatially and temporally. We first propose a dynamical model for these kinds of spatiotemporal fields, similar to the model proposed in our previous work [Lan and Schwager, 2013, Lan and Schwager, 2014]. 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 family of motion strategies for sensing agents to move around an arbitrarily chosen periodic trajectory to collect sufficient data about the field. Then based on the data collected using each 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 [Siddiqi et al., 2008, Van Overschee and De Moor, 1996, Ljung, 1999, Katayama, 2006]. 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) [Larimore, 1990], multi-variable output-error state space (MOESP) [Verhaegen, 1994] and numerical algorithms for subspace state space system identification (N4SID) [Van Overschee and De Moor, 1994]. A unifying theorem was proposed in [Van Overschee and De Moor, 1995] to state the similarities between these three algorithms. A good overview about subspace identification is given in [Viberg, 1995, Qin, 2006]. Our learning algorithm is different from classical subspace identification 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

Email addresses: xlan@bu.edu (Xiaodong Lan), schwager@stanford.edu (Mac Schwager).

time-invariant (LTI) systems. The motion strategies executed by the mobile robots are along a periodic trajectory. Our use of periodic trajectory is inspired by [Felici et al., 2007, Van Wingerden and Verhaegen, 2009]. In [Felici et al., 2007], 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 [Van Wingerden and Verhaegen, 2009], 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 about 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.

Our work is also related to learning Hidden Markov Models (HMMs) [Hsu et al., 2012, Siddiqi et al., 2010] and learning Predictive State Representations (PSRs) [Rosencrantz et al., 2004, Boots and Gordon, 2010, Boots et al., 2011]. In [Hsu et al., 2012] the authors use the idea of performing a Canonical Correlation Analysis (CCA) between past and future observations to uncover the information about HMMs. In [Siddiqi et al., 2010], the authors introduce a generalization of HMMs called Reduced-Rank Hidden Markov Model (RR-HMM) and apply the spectral learning algorithm in [Hsu et al., 2012] to learn RR-HMM. PSRs are generalizations of Partially Observable Markov Decision Processes (POMDPs) [Littman et al., 2002, Singh et al., 2004]. In [Rosencrantz et al., 2004], the authors propose a variant of PSRs called transformed predictive state representations (TPSRs) and provide an efficient principal-components-based algorithm for learning a TPSR. In [Boots and Gordon, 2010], the authors combine reinforcement learning with subspace identification and propose an algorithm called Predictive State Temporal Difference (PSTD) to learn TPSRs. In [Boots et al., 2011], the authors present a spectral algorithm for learning a PSR and use the learned PSR to do path planning for a vision-based mobile robot. Our algorithm is different from all these in that we explicitly take into account the motion of the sensing robots as they move and take data, rather than considering data that is given to the algorithm in a batch.

The rest of this paper is organized as follows. The problem is formulated in Section 2. Our learning algorithms are described in Section 3 and Section 4. Section 5 presents the results of numerical simulations, and we discuss conclusions in Section 6.

Table 1
Main notation used in this paper.

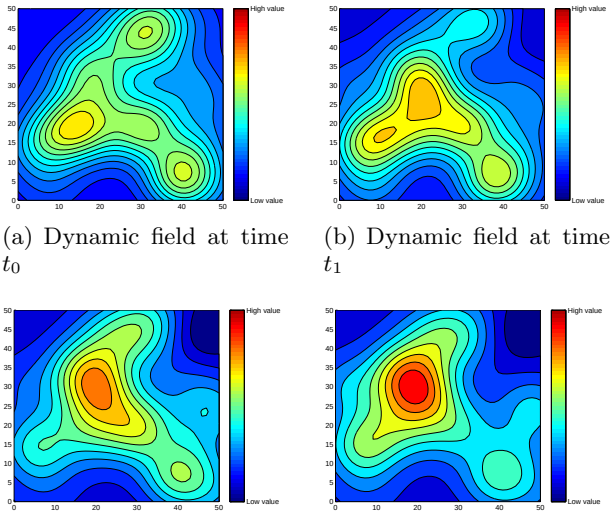
$\phi(q,t)$	Scalar field value at point q at time t.
x_t	Time-changing weights.
$C(q)$	Spatial basis functions.
N_s	Number of mobile sensors.
N_c	Number of consecutive pairs.
p_t^i	Position of the i-th mobile sensor at time t.
y_t^i	Measurement from the i-th sensor at time t.
y_t^{mult}	Composite measurements from N_s sensors.
σ	A discrete periodic trajectory consisting of a set of waypoints $\{\sigma_1, \sigma_2, \ldots, \sigma_T\}$ with period T.
λ_i	Eigenvalues of the A matrix.
r	Number of steps one sensor stays at one waypoint.
d, K_d	User-defined parameters with $d = K_d * r$.
$\mathbf{y}_{t,r\|t+d-1}$	Column vector of stacked mobile sensor outputs.
$\mathbf{y}^{\text{ref}}_{t,r\|t+d-1}$	Column vector of stacked mobile sensor outputs and reference sensor outputs.
$Y_{t,r\|t+d-1}$	Block Hankel matrix.
$X_{t,r}$	Stacked weight matrix.
$\widehat{X}_{t,r}$	Estimated stacked weight matrix.
$O_{t,r\|d}$	Extended observability matrix.
$\widehat{O}_{t,r\|d}$	Estimated extended observability matrix.

2 Problem formulation

In this section we formulate the problem considered in this paper. As discussed in the previous section, there are many examples of relevant spatio-temporal fields in the real world, including temperature fields, radioactivity intensity fields, magnetic and electric fields, chemical concentration fields, and many more. We cannot realistically measure the field everywhere at all times. Hence we must build a model of the field using a limited number of spatial and temoral samples. We can then use the constructed model to predict the field at any location at any time. In this paper we propose a family of learning algorithms and sensor motion strategies to learn the parameters of a spatio-temporal field model.

2.1 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, see Figure 1. 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*



(a) Dynamic field at time t_0

(b) Dynamic field at time t_1

(c) Dynamic field at time t_2

(d) Dynamic field at time t_3

Fig. 1. Dynamic environment is changing with time $t_0 < t_1 < t_2 < t_3$. The color map indicates the field values. Points on the same contour line have the same field value.

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 environmental fields [Cressie, 1990, Cressie, 1993]. Note that $C(q)$ can be any arbitrary set of spatial basis. Typical choices of basis functions are 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 t_0 denote the initial time and x_0 the initial state vector. 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, $rank\big([Q^{1/2}, AQ^{1/2}, A^2Q^{1/2}, \ldots, A^{n-1}Q^{1/2}]\big) = n$. In this paper, given an unknown stable scalar spatiotemporal field modeled in this way, our goal is to learn how the field changes. That is, our goal is to learn the A matrix, the spatial basis functions $C(q)$, and the noise covariance matrix Q. We will use mobile sensors to measure the field at a set of locations in a certain sequence in order to collect the required data to learn the model parameters.

2.2 Mobile Sensor Model

Assume we have $N_s \geq 1$ mobile sensors at positions $p_t^i \in \mathcal{Q}, i = 1, 2, \ldots, N_s$. Each mobile sensor can only measure the field value at a single point at each time step, with some additive sensor noise

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

where p_t^i is the i-th sensor's position, $y_t^i \in \mathbb{R}$ is the i-th sensor output, $v_t^i \in \mathbb{R}^{l\times 1}$ is a Gaussian white noise process, independent of x_t, and with Gaussian distribution $v_t^i \sim N(0, R^i)$. We allow v_t^i to be correlated with w_t, and denote the covariance matrix between w_t and v_t^i by S^i, however, we do not know R^i nor S^i. We also assume that the sensors are given a single periodic path $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_T, \sigma_1) \in \mathcal{Q}^{T+1}$ with period T. At each time t, a mobile sensor located at σ_k can either move forward one step, $p_{t+1}^i = \sigma_{k+1}$ (where we define $\sigma_{T+1} := \sigma_1$), or it can stay in place, $p_{t+1}^i = \sigma_k$. A motion strategy is a periodic sequence of move and stay actions for each mobile sensor p_t^i along σ.

2.3 Learning Accuracy Metric

As discussed above, our goal is to use mobile sensors to collect measurements about the field at the waypoints of one trajectory and learn how the field changes. Note that it is the eigenvalues of the A matrix that capture the decay rates and natural frequencies of the field dynamics [Kailath, 1980]. In order to measure how close a learned system matrix $\hat{A}$ is to a true system matrix A, we define an error metric based on the differences between the eigenvalues of the two matrices

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

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. Next, we state the learning problem formally.

Problem 1 *Given an unknown spatiotemporal scalar field $\phi(q,t)$ modeled by* (1) *and* (2)*, use N_s mobile sensors* (3) *to collect field measurements to learn the A matrix and the spatial basis functions $C(q)$, while minimizing the learning error defined in* (4)*.*

Remark 1 *Note that Problem 1 consists of two subproblems: how to collect data using mobile sensors and how to use the collected data to learn the parameters of the model. This is different from classical subspace identification and machine learning problems, in which the data*

are usually given and the goal is to use these data to extract features or learn parameters. In our case, we care about both of the two subproblems since how to collect data will influence how we learn the parameters. If the data are not collected properly, we cannot use these data to learn the parameters.

3 Learning the A Matrix

In this section we propose an algorithm based on subspace identification [Van Overschee and De Moor, 1996, Ljung, 1999, Katayama, 2006] to learn the eigenvalues of the A matrix. We first consider the case that we have only one mobile sensor, $N_s = 1$, see Figure 2(a).

3.1 One mobile sensor

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.

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.*

This definition is based on a similar notion for linear time-varying systems in [Jazwinski, 2007]. Note that here we use MATLAB notation, where $[a; b]$ indicates that the first row is a and the second row is b. In the following sections, without specific explanation, the square bracket $[\cdot]$ means MATLAB notation.

In order to use data from the mobile sensor to learn 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 $C(p_t)$. 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 $C(p_t)$ matrix change in time.

3.1.1 Motion Strategy

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 point $p_t = \sigma_k$ at time t and stay at point σ_k for r steps, where $r \geq 2$. Denote the corresponding sensor outputs by $(y_t, y_{t+1}, \ldots, y_{t+r-1})$, here since we only have one sensor, we omit the superscript i of y_t^i, p_t^i, v_t^i and R^i. Then let the mobile sensor move to the next waypoint σ_{k+1} and stay there for r steps with outputs $(y_{t+r}, y_{t+r+1}, \ldots, y_{t+2r-1})$. 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. This case will be considered in the next subsection. Also, the mobile sensor will come back to its original waypoint every $Tr + 1$ steps. Hence, $p_t = p_{t+Tr+1}$ and $C_t = C_{t+Tr+1}$. Next we write the measurements collected by the mobile sensor in a block matrix form. First, define a stacked output column vector as

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

where $d = K_d * r, K_d \in \mathbb{Z}^+$. Both d and K_d are user-defined parameters. Next we define an output block matrix as in (6), which is usually called the output block Hankel matrix.

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

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.

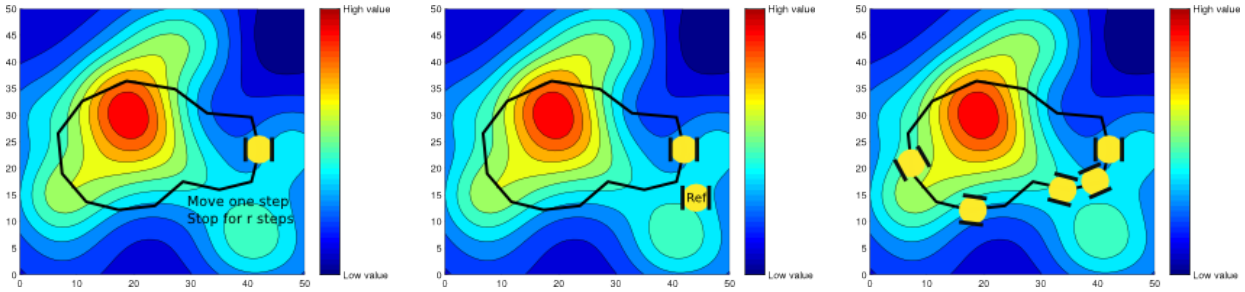
$$X_{t,r} = \begin{bmatrix} x_t & x_{t+Tr} & \ldots & x_{t+(N-1)Tr} \end{bmatrix}, \quad X_{t,r} \in \mathbb{R}^{n \times N} \quad (7)$$

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

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

where $C_t = C(p_t)$ and $O_{t,r|d} \in \mathbb{R}^{d \times n}$. Note that here $C_t = C_{t+1} = \ldots = C_{t+r-1} = C(p_i)$, since the sensor stays at waypoint p_i for r steps. This is still true for other C matrices when the mobile sensor stays at one waypoint for r steps. That is, when the sensor stays at one waypoint, it will keep taking measurements about the field at that point. Hence, 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

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



(a) One mobile sensor executing moving and stopping motion.

(b) One mobile sensor with a synchronized static reference sensor.

(c) Multiple synchronized mobile sensors with a fixed network structure.

Fig. 2. Using different number of mobile sensors to move along periodic trajectories to collect field data.

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 the noise is small and $\text{rank}(X_{t,r}) = n$. 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 $O_{t,r|d}$. We will prove in the later sections that we can indeed get the column space of $O_{t,r|d}$ from the sensor output data. Therefore, in order to write equation (9) and separate information about the A matrix from the sensor outputs, the elements in each row of $Y_{t,r|t+d-1}$ should correspond to the measurements taken at the same waypoint. That is, the mobile sensor has to come back to the same position to take measurements after some time. This explains why we use periodic trajectories to collect data.

3.1.2 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 turns out 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 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 [Kalman, 1960]. 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

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

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 can be found in Appendix A.1.

Remark 2 *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 $y_0, y_1, \ldots, y_{t-1}$. Note that Kalman filter only uses partial information of the outputs. It can be verified that the following is true: $\widehat{x}_{i+t} = \Delta_i^t (L_i^t)^{-1} \left[y_t;\ y_{t+1};\ \ldots; y_{t+i-1} \right]$. 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 A matrix. In this theorem, we assume that we are given a uniformly observable periodic path such that we can control the mobile sensor to move along this trajectory. Exactly how to construct such a path will be the subject of future research. Also, for the field dynamics, the pair $(A, Q^{1/2})$ is assumed controllable.

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 a uniformly observable periodic path 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:

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

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

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

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

The proof of this theorem can be found in Appendix A.2.

Remark 3 *The learned matrices $\widehat{O}_{t,r|d}$ and $\widehat{X}_{t,r}^{svd}$ 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 $\widehat{X}_{t,r}^{svd} = \Gamma \widehat{X}_{t,r}$, 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:

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

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

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 $\widehat{A} = \bar{\widehat{O}}_{t,r|d}^\dagger \underline{\widehat{O}}_{t,r|d}$, where $\bar{\widehat{O}}_{t,r|d}$ and $\underline{\widehat{O}}_{t,r|d}$ 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.

3.2 One mobile sensor, one static sensor

When the mobile senor moves at each step ($r = 1$), we can not use the same method as above. In this case, the extended observability matrix becomes $O_{t,1|d} = \left[C_t; C_{t+1} A; \ldots; C_{t+K_d-1} A^{K_d-1} \right]$ and $C_t =$

$C_{t+T}, d = K_d r = K_d$. Hence we can not use (18) and (19) to learn A. However, if we have two extended observability matrices $O_{t,1|d}$ and $O_{t+1,1|d}$, then the following equation holds: $\bar{O}_{t+1,1|d}A = \underline{O}_{t,1|d}$. Therefore we can use Theorem 1 to learn $\widehat{O}_{t+1,1|d}$ and $\widehat{O}_{t,1|d}$ first, then use $A = \bar{\widehat{O}}^{\dagger}_{t+1,1|d}\underline{\widehat{O}}_{t,1|d}$ to learn A. The problem with this approach is that the learned $\widehat{O}_{t+1,1|d}$ and $\widehat{O}_{t,1|d}$ may not stay in the same state space. In order to transform them into the same state space, we propose to use a *static* reference sensor to collect measurements, see Figure 2(b). We assume that the reference sensor has the same sensing model as the mobile sensor (3). We also assume that it is synchronized in time with the mobile sensor, that is, they take measurements at the same instant. Denote the reference sensor's position by p^{ref}, its corresponding C matrix by $C^{\text{ref}} = C(p^{\text{ref}})$, its output by y_t^{ref}. Then we reconstruct the output block Hankel matrix $Y^{\text{ref}}_{t,1|t+d-1}$ by replacing $\mathbf{y}_{t,1|t+d-1}$ in (6) with $\mathbf{y}^{\text{ref}}_{t,1|t+d-1} = \left[y_t^{\text{ref}}; y_t; y_{t+1}^{\text{ref}}; y_{t+1}; \ldots; y^{\text{ref}}_{t+K_d-1}; y_{t+K_d-1}\right]$, then the corresponding extended observability matrix is $O^{\text{ref}}_{t,1|d} = \left[C^{\text{ref}}; C_t; C^{\text{ref}}A; C_{t+1}A; \ldots; C^{\text{ref}}A^{d-1}; C_{t+K_d-1}A^{Kd-1}\right]$. Note that $O^{\text{ref}}_{t,1|d}$ consists of $O_{t,1|d}$ and O_t^{ref} which are defined as follow.

$$O_{t,1|d} = \left[C_t; C_{t+1}A; \ldots; C_{t+K_d-1}A^{K_d-1}\right]$$
$$O_t^{\text{ref}} = \left[C^{\text{ref}}; C^{\text{ref}}A; \ldots; C^{\text{ref}}A^{d-1}\right].$$

By Theorem 1, we can learn $\widehat{O}^{\text{ref}}_{t,1|d}$ by using output data $(y_t^{\text{ref}}, y_t, y_{t+1}^{\text{ref}}, y_{t+1}, \ldots)$ and learn $\widehat{O}^{\text{ref}}_{t+1,1|d}$ by using output data $(y_{t+1}^{\text{ref}}, y_{t+1}, y_{t+2}^{\text{ref}}, y_{t+2}, \ldots)$. Since the learned extended observability matrices are not the same as the original extended observability matrices (they are equivalent to the original matrices up to a similarity transformation), we denote this relationship by the following equations:

$$\widehat{O}^{\text{ref}}_{t,1|d}\Gamma_t = O^{\text{ref}}_{t,1|d}, \quad \widehat{O}^{\text{ref}}_{t+1,1|d}\Gamma_{t+1} = O^{\text{ref}}_{t+1,1|d}, \qquad (21)$$

where Γ_t and Γ_{t+1} are non-singular similarity transformation matrices. Then we can learn the A matrix from $\widehat{O}^{\text{ref}}_{t,1|d}$ and $\widehat{O}^{\text{ref}}_{t+1,1|d}$ by the following theorem.

Theorem 2 *Given two learned extended observability matrices $\widehat{O}^{ref}_{t,1|d}$ and $\widehat{O}^{ref}_{t+1,1|d}$, and their relationships with the original extended observability matrix given by* (21)*, then the basis transformation between $\widehat{O}^{ref}_{t,1|d}$ and $\widehat{O}^{ref}_{t+1,1|d}$ is given by $\Gamma = \Gamma_{t+1}\Gamma_t^{-1} = (\widehat{O}^{ref}_{t+1})^{\dagger}\widehat{O}^{ref}_t$, and the learned $\widehat{A}$ is given by $\widehat{A} = \left(\overline{\widehat{O}_{t+1,1|d}\Gamma}\right)^{\dagger}\underline{\widehat{O}}_{t,1|d}$ as long as $K_d \geq n+1$, where $\overline{\widehat{O}_{t+1,1|d}\Gamma}$ is the upper $K_d - 1$ blocks of $\widehat{O}_{t+1,1|d}\Gamma$, $\underline{\widehat{O}}_{t,1|d}$ is the lower $K_d - 1$ blocks of $\widehat{O}_{t,1|d}$.*

The proof of this theorem can be found in Appendix A.3.

3.3 Multiple mobile sensors

Here we extend the strategy for using one mobile sensor to collect data and learn the A matrix to the case of multiple sensors. When using multiple mobile sensors to collect data, we assume that they start at different waypoints on the periodic trajectory. We also assume that the relative spacing between them is fixed, that is, they will move at the same time and stop at the same time, and they are synchronized in time, see Figure 2(c). Finally, we assume that $N_s < T$, so that there is not more than one mobile sensor per waypoint on the path σ. If $N_s \geq T$, this means we can put at least one sensor on each waypoint of the periodic trajectory and there is no need for the sensors to move. Denote each sensor's output by $(y_t^1, y_t^2, \ldots, y_t^{N_s})$ and use them to construct a composite output by $y_t^{\text{mult}} = [y_t^1; y_t^2; \ldots; y_t^{N_s}]$. We reconstruct the output block Hankel matrix $Y^{\text{mult}}_{t,r|t+d-1}$ by replacing $\mathbf{y}_{t,r|t+d-1}$ in (6) with

$$\mathbf{y}^{\text{mult}}_{t,r|t+d-1} := [y_t^{\text{mult}}; y_{t+1}^{\text{mult}}; \ldots y^{\text{mult}}_{t+r-1}; \big| y^{\text{mult}}_{t+r}; y^{\text{mult}}_{t+r+1}; \ldots$$
$$y^{\text{mult}}_{t+2r-1}; \big| \ldots \big| y^{\text{mult}}_{t+(K_d-1)r}; \ldots y^{\text{mult}}_{t+K_d r-1}] \qquad (22)$$

then the corresponding extended observability matrix is

$$O^{\text{mult}}_{t,r|d} := [C_t^{\text{mult}}; \ldots C^{\text{mult}}_{t+r-1}A^{r-1}; \big| C^{\text{mult}}_{t+r}A^r; \ldots \big|$$
$$C^{\text{mult}}_{t+(K_d-1)r}A^{(K_d-1)r}; \ldots C^{\text{mult}}_{t+K_d r-1}A^{K_d r-1}] \qquad (23)$$

where $C_t^{\text{mult}} = \left[C_t^1; C_t^2; \ldots; C_t^{N_s}\right]$ is the composite C matrix consisting of the C matrices corresponding to all the multiple sensors. Note that we still have $C_t^{\text{mult}} = C_{t+1}^{\text{mult}} = \ldots = C^{\text{mult}}_{t+r-1}$ if these sensors stay at their waypoints for r steps. We can still use Theorem 1 to learn $\widehat{O}^{\text{mult}}_{t,r|d}$. The next step is to use the learned $\widehat{O}^{\text{mult}}_{t,r|d}$ to learn A, which is somewhat different than for the single sensor case. Before we describe how to use $\widehat{O}^{\text{mult}}_{t,r|d}$ to learn A, we first give a definition which will be used in the following context.

Definition 4 (Consecutive Pair) *Two mobile sensors are a consecutive pair if they are always positioned at consecutive waypoints, so that when one is at waypoint σ_k, the other is either at σ_{k+1} or σ_{k-1}.*

Sensors in a consecutive pair will collect measurements at the same waypoint when the pair moves forward for one step. That is, assume sensor one and sensor two in a consecutive pair collect measurements at waypoints σ_k and σ_{k+1}. If they move forward for one time step, then they will collect measurements at waypoints σ_{k+1} and σ_{k+2}, respectively. In this case, both of them collect measurements at waypoint σ_{k+1} in two consecutive time steps. Denote one consecutive pair by $k \twoheadrightarrow j$, which means that sensor k and sensor j are a consecutive pair

and sensor k will move from waypoint i to waypoint $i+1$ if sensor j moves from waypoint $i+1$ to waypoint $i+2$. Denote the number of consecutive pairs in a fixed sensor network by N_c. Note that $0 \leq N_c \leq N_s - 1 (N_s < T)$ and N_c is determined by the initial positions of these sensors. Denote all the consecutive pairs by $k_1 \twoheadrightarrow j_1, k_2 \twoheadrightarrow j_2, \ldots, k_{N_c} \twoheadrightarrow j_{N_c}$. Similar to (18) and (19), we can extract two submatrices from $O^{\text{mult}}_{t,r|d}$. We modify each block separated by $|\cdot|$ in (18) and (19) in the following way.

$$\bar{O}^{\text{mult}}_{t,r|d} := [C^{\text{mult}}_t; \ldots; C^{\text{mult}}_{t+r-2}A^{r-2}; C^{j_1}_{t+r-1}A^{r-1}; \ldots; C^{j_{N_c}}_{t+r-1}A^{r-1}; | C^{\text{mult}}_{t+r}A^r; \ldots] \quad (24)$$

$$\underline{O}^{\text{mult}}_{t,r|d} := [C^{\text{mult}}_{t+1}A; \ldots; C^{\text{mult}}_{t+r-1}A^{r-1}; C^{k_1}_{t+r}A^r; \ldots; C^{k_{N_c}}_{t+r}A^r; | C^{\text{mult}}_{t+r+1}A^{r+1}; \ldots] \quad (25)$$

Note that $C^{j_i}_{t+r-1} = C^{k_i}_{t+r}$ for all the $i = 1, 2, \ldots, N_c$, and $C^{j_i}_{t+r-1}$ is the C matrix corresponding to sensor j_i in C^{mult}_{t+r-1} and $C^{k_i}_{t+r}$ is the C matrix corresponding to sensor k_i in C^{mult}_{t+r}. So $\bar{O}^{\text{mult}}_{t,r|d}A = \underline{O}^{\text{mult}}_{t,r|d}$. Compared with (18) and (19), in these two equations we use the measurements collected at the same waypoints by the consecutive pairs. Hence, when using multiple sensors to collect data, we can use the collected data to construct $Y^{\text{mult}}_{t,r|t+d-1}$ first, then use Theorem 1 to learn $\widehat{O}^{\text{mult}}_{t,r|d}$ and extract $\bar{\widehat{O}}^{\text{mult}}_{t,r|d}$ and $\underline{\widehat{O}}^{\text{mult}}_{t,r|d}$, then we can learn A by $\widehat{A} = (\bar{\widehat{O}}^{\text{mult}}_{t,r|d})^{\dagger}\underline{\widehat{O}}^{\text{mult}}_{t,r|d}$.

3.4 Conditions on parameters

When we use Theorem 1 and $\widehat{A} = (\bar{\widehat{O}}^{\text{mult}}_{t,r|d})^{\dagger}\underline{\widehat{O}}^{\text{mult}}_{t,r|d}$ to learn A, we need to make sure that $\bar{\widehat{O}}^{\text{mult}}_{t,r|d}$ is full column rank. Since we collect data along a uniformly observable trajectory, in order to guarantee $\bar{\widehat{O}}^{\text{mult}}_{t,r|d}$ is full column rank, we just need to make sure that the number of rows in $\bar{\widehat{O}}^{\text{mult}}_{t,r|d}$ is greater than or equal to n. Next we give a theorem which states what conditions the parameters N_s, N_c, r and K_d have to satisfy in order to make sure $\bar{\widehat{O}}^{\text{mult}}_{t,r|d}$ is full column rank. This theorem also applies to the single mobile sensor case.

Theorem 3 *When using N_s $(1 \leq N_s < T)$ mobile sensors to collect data along a uniformly observable periodic trajectory, in order to use theorem 1 to learn the A matrix, the parameters N_s, N_c, r and K_d have to satisfy the following conditions. If $N_s = 1$ and $r = 1$, use a synchoronized static reference sensor and choose $K_d \geq n+1$. Otherwise, the parameters must be chosen to satisfy*

$$N_s(r-1)K_d + N_c(K_d - 1) \geq n \quad (26)$$

The proof of this theorem can be found in Appendix A.4.

Remark 4 *In theorem 3, when $N_s > 1$ and $r = 1$, this means that all sensors move at every time step along the periodic trajectory to take measurements. In this case, N_c can not be zero. That is, in order to learn A, there must be consecutive pairs if all the multiple sensors keep moving, since we can only use the overlap measurements taken by these consecutive pairs to learn the A matrix.*

4 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 $\widehat{X}^{\text{svd}}_{t,r}$, see equation (17). Note that $\widehat{X}^{\text{svd}}_{t,r} = [\hat{x}_t, \hat{x}_{t+T}, \hat{x}_{t+2T}, \ldots, \hat{x}_{t+(N-1)T}]$. That is, it consists of the estimated state sequence in every T 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

$$(q_1^*, \ldots, q_n^*, K^*, \sigma_c^*) = \underset{q_1,\ldots,q_n,K,\sigma_c}{\arg\min} \sum_{j=1}^{N}\sum_{i=1}^{N_s} \|y^i_{t+(j-1)T} - C(p^i_{t+(j-1)T})\hat{x}_{t+(j-1)T}\|^2. \quad (27)$$

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.

5 Numerical Simulations

In this section, we consider the performance of our algorithms by simulating a spatio-temporal field and applying the proposed algorithms to learn a dynamic model for the field. Then we use the data from the US National Oceanographic and Atmospheric Administration (NOAA) Operational Model Archive and Distribution System (NOMADS) [NOAA,] to learn a sea surface temperature model for the Caribbean Sea.

5.1 Simulated Spatio-temporal Field

When we simulate a spatio-temporal field, we choose the environment as a 50×50 unit square region (with arbitrary spatial units). We assume that the basis functions are Gaussian radial basis functions with $c_i(q) =$

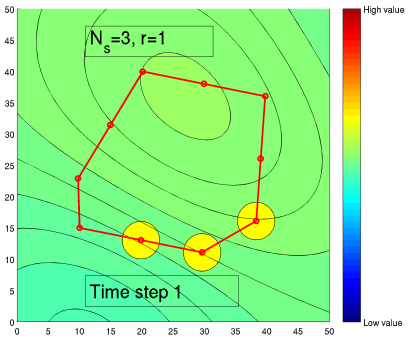
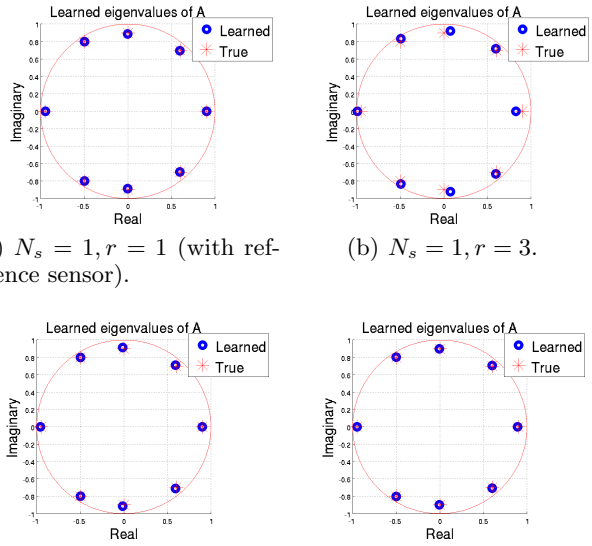


Fig. 3. Periodic trajectory used to collect measurements of the simulated spatio-temporal field.

$Ke^{\frac{-\|q-q_i\|}{2\sigma_c^2}}$ and the basis centers q_i are distributed in a uniform grid over the environment. We choose a uniformly observable periodic trajectory to collect data, see Figure 3. The motion strategy for the mobile sensors is as discussed in the previous sections. Note that all the data are collected along the same periodic path. Our goal is to use these noisy data to learn how the field in the whole environment changes. Note that for the same spatiotemporal field, the data collected by different periodic trajectories are different.

We first apply the proposed algorithms to the data collected to learn the A matrix, then we use the nonlinear curve fitting solver *lsqcurvefit* with multiple starting points in MATLAB to solve (27), thereby learning the corresponding parameters for the Gaussian basis functions. Note that the learned A matrix may not equal the original A matrix. They are the same up to a similarity transformation. However the eigenvalues of A are invariant under similarity transformations. So we evaluate the learning accuracy by equation (4). Figure 4 shows the eigenvalues of the A matrix learned by using a single sensor and a reference sensor (a), a single sensor with wait time $r = 3$ (b), three sensors with wait time $r = 1$ (c), and three sensors with wait time $r = 3$ (d). For all experiments $N = 10000$ and $d = 15$. The figure indicates that the learned eigenvalues are close to the true eigenvalues.

Figure 5 shows how the eigenvalue error defined in (4) changes with an increasing number of data points N for the same four simulation scenarios. This figure shows that the error decreases as N increases. Note that the figure is not meant to compare the four motion and learning strategies since the data used for each strategy is different. The sensors are in different locations at different times for each strategy, and different strategies will collect different amounts of data in the same time window. Hence there is not a simple or fair way to compare the strategies. Intuitively using more sensors to collect data will improve learning performance since during the same time window more data is collected. This intuition is reflected in the figure.

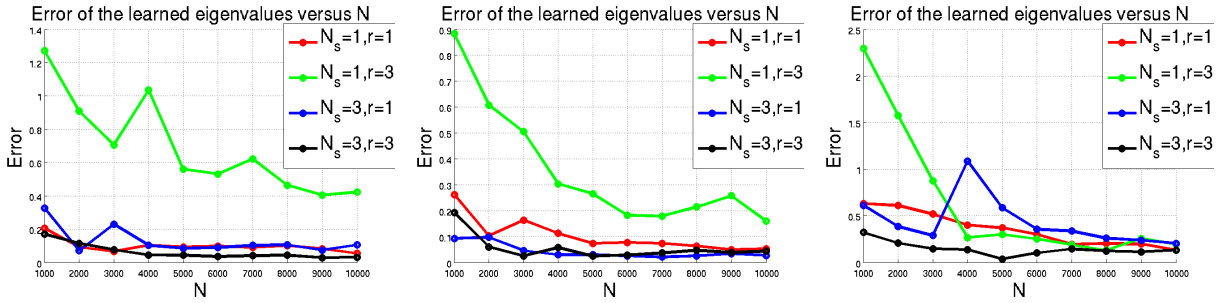


(a) $N_s = 1, r = 1$ (with reference sensor). (b) $N_s = 1, r = 3$.

(c) $N_s = 3, r = 1$. (d) $N_s = 3, r = 3$.

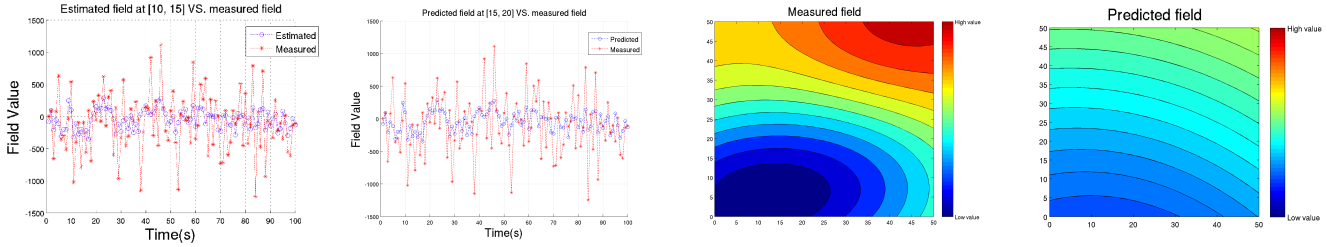
Fig. 4. Learned eigenvalues for the A matrix of a spatiotemporal field represented by 8 basis functions with $N = 10000$ and $d = 15$. The learned eigenvalues (blue circles) lie nearly on top of the true ones (red crosses).

Next we use the procedure proposed in Section 4 to learn the spatial basis functions. In this simulated spatiotemporal field, the basis functions are assumed to be Gaussian basis functions. Because the learned coefficient matrix $\widehat{X}^{\text{svd}}_{t,r}$ is not the same as the original coefficient matrix $\widehat{X}_{t,r}$ (see Equation (17)), the learned Gaussian basis function parameters may not equal the original parameters that we used to generate the spatio-temporal field. In order to verify that we learned these parameters correctly, we calculate the estimated field values based on the learned parameters and compare the estimated field values with the measured field values at two locations in the field, see Figure 6, Figure 7, Figure 8 and Figure 9. The first location is a waypoint on the sensing trajectory at location $(10, 15)$, hence the we call the learned field value "estimated." The second point is at a location where no sensor measurement was ever collected $(15, 20)$, so we call the learned field value "predicted." Note that the estimated/predicted field is based on the learned basis functions and the learned state (coefficient) sequence, which is Kalman filtered and the effect of the noise is decreased, while the measured field is corrupted by process and measurement noise. Therefore, the estimated/predicted field is a Kalman-filtered version of the measured field, but using the learned model A, $C(q)$. This can explain why the estimated and predicted fields do not match the measured fields, but they have the same changing tendency. We also calculate the predicted fields of every location in the environment at one time instant and plot them as a colormap. In the colormap, red stands for high values while blue stands for low values. We compare the predicted fields with the measured fields in the last two subplots of Figure 6, Fig-



(a) $d = 15$ and $n = 8$. (b) $d = 15$ and $n = 6$. (c) $d = 15$ and $n = 4$.

Fig. 5. Error decreases as N increases.



(a) Estimated field versus measured field at $(10, 15)$. (b) Predicted field versus measured field at $(15, 20)$. (c) Measured field colormap in the whole environment. (d) Predicted field colormap in the whole environment.

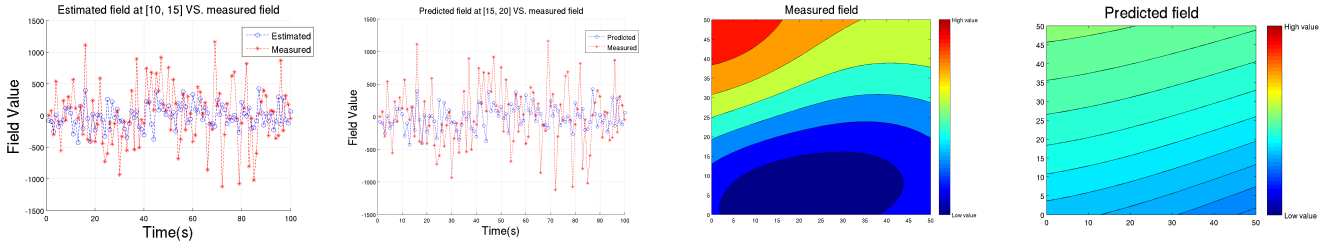
Fig. 6. Parameters learned by using one reference sensor and one mobile sensor. The first two plots show the estimated and predicted field values at two locations based on the learned model parameters A, $C(q)$, compared with the measured field values at those locations. The estimated/predicted field represents a Kalman-filtered version of the measured field, but filtered using the learned dynamical model. The last two plots show the colormap of the whole field at one time instant for the measured and learned fields.

ure 7, Figure 8 and Figure 9. As discussed above, the estimated and predicted fields are not exactly the same with the measured fields, but they show close field distributions in the whole environment.

5.2 NOAA NOMADS Ocean Temperature Model

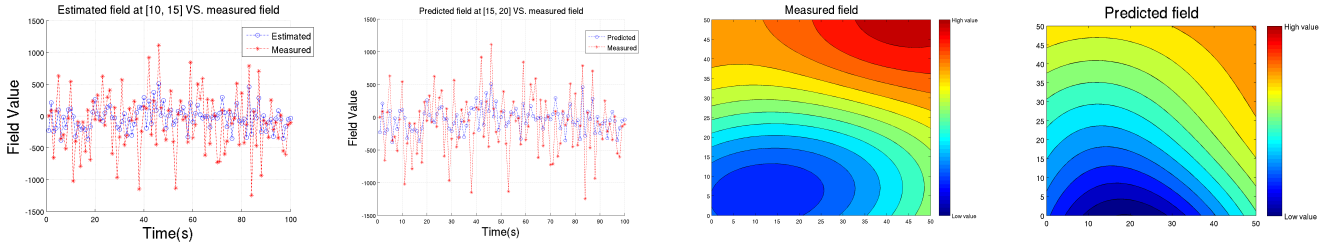
Next we apply the proposed motion and learning algorithms to learn a temperature model for the surface sea water temperature of the Caribbean Sea. We first choose a trajectory for the mobile sensors to collect data, see Figure 10. We get this trajectory by using the path planning algorithm that we proposed previously in [Lan and Schwager, 2015]. We simulate the data collecting process by extracting data along this trajectory using the NOAA Operational Model Archive and Distribution System (NOMADS) [NOAA,]. From NOMADS, we can get the sea water temperature at any location in the sea. For the temperature at one location, the NOMADS can give a measurement every 3 hours. In the data collecting process, we choose the time step to be 6 hours, that is, it takes 6 hours for a mobile sensing boat to travel from one waypoint to its next waypoint and take measurements. We use 5 mobile sensors to collect data. These sensors execute a motion strategy in which they stop at each waypoint for 3 steps. We have found that the other motion strategies we describe in this paper achieve similar results. We also assume that the basis functions are Gaussian, so based on the collected data we will use the proposed learning algorithms to learn A, K, $q_1, \ldots, q_n$ and σ_c.

Since we are considering a real spatio-temporal field, we do not have a ground truth for the parameters of the field. For example, we do not know what are the true eigenvalues of the A matrix for this temperature field. The only way for us to evaluate the performance of our algorithms on the collected data is to compare the estimated and predicted temperatures with the temperature from NOMADS, see Figure 11. The left plot shows the estimated field value, which is a Kalman filtered version of the measured field using the learned model parameters for the filter, at a location where data was taken. The center-left plot shows the predicted temperature at a location where data was never taken (off the sampling trajectory). Both temperatures track the measured field, however there seems to be a small offset for the predicted field value. The right two plots show the whole measured and predicted field at a single time instant. The two colormaps are close, but not the same. One reason is the presence of model uncertainties. That is, the ocean temperature may not fit well with our linear model. A nonlinear model may be needed if more



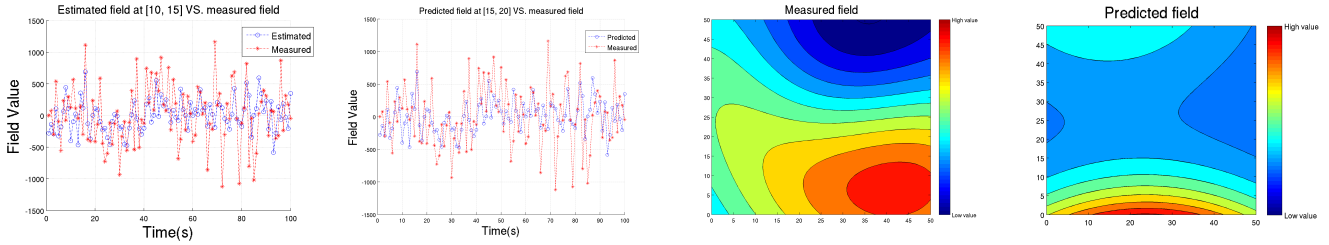
(a) Estimated field versus measured field at $(10, 15)$. (b) Predicted field versus measured field at $(15, 20)$. (c) Measured field colormap in the whole environment. (d) Predicted field colormap in the whole environment.

Fig. 7. Parameters learned by using one mobile sensor executing $N_s = 1$ and $r = 3$ motion strategy. The first two plots show the estimated and predicted field values at two locations based on the learned model parameters A, $C(q)$, compared with the measured field values at those locations. The estimated/predicted field represents a Kalman-filtered version of the measured field, but filtered using the learned dynamical model. The last two plots show the colormap of the whole field at one time instant for the measured and learned fields.



(a) Estimated field versus measured field at $(10, 15)$. (b) Predicted field versus measured field at $(15, 20)$. (c) Measured field colormap in the whole environment. (d) Predicted field colormap in the whole environment.

Fig. 8. Parameters learned by using 3 mobile sensors executing $N_s = 3$ and $r = 1$ motion strategy. The first two plots show the estimated and predicted field values at two locations based on the learned model parameters A, $C(q)$, compared with the measured field values at those locations. The estimated/predicted field represents a Kalman-filtered version of the measured field, but filtered using the learned dynamical model. The last two plots show the colormap of the whole field at one time instant for the measured and learned fields.



(a) Estimated field versus measured field at $(10, 15)$. (b) Predicted field versus measured field at $(15, 20)$. (c) Measured field colormap in the whole environment. (d) Predicted field colormap in the whole environment.

Fig. 9. Parameters learned by using 3 mobile sensors executing $N_s = 3$ and $r = 3$ motion strategy. The first two plots show the estimated and predicted field values at two locations based on the learned model parameters A, $C(q)$, compared with the measured field values at those locations. The estimated/predicted field represents a Kalman-filtered version of the measured field, but filtered using the learned dynamical model. The last two plots show the colormap of the whole field at one time instant for the measured and learned fields.

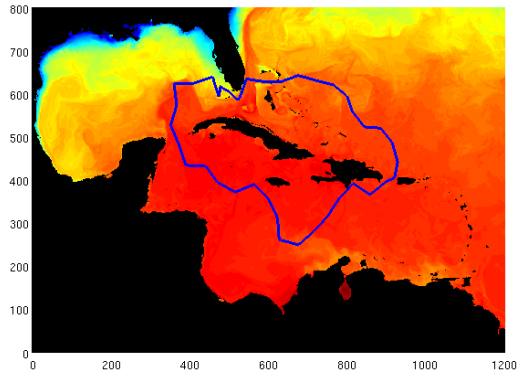


Fig. 10. Blue curve is the trajectory used to collect measurement data of the surface sea water temperature in the Caribbean Sea. One unit length in the plot is equal to 2.304 miles in geography.

accurate predictions are desired.

6 Conclusions and Future Work

In this paper we proposed several algorithms to collect noisy measurements of a spatiotemporal field using mobile sensors, and use the collected data to learn a linear state space model for the dynamics of the field. We first propose a model for the field. Then we control mobile sensors 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 algorithms by considering a simulated spatio-temporal field, for which we know the ground truth. Then we test the algorithms on a physical sptatio-temporal field, which is the surface sea water temperature in the Caribbean Sea as recorded by the NOAA NOMADS database. The predicted field values based on the learned parameters from our algorithm are close to the measured values of the field. In the future, we plan to investigate how to construct an optimal uniformly observable periodic trajectory to collect data to minimize the learning error, subject to motion and fuel constraints. This is usually referred as *informative path planning* in the robotics community. We are also interested in finding parsimonious spatial basis functions to represent the spatial component of environmental spatio-temporal fields.

Acknowledgements

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

References

[Boots and Gordon, 2010] Boots, B. and Gordon, G. J. (2010). Predictive state temporal difference learning. In *NIPS*, pages 271–279.

[Boots et al., 2011] Boots, B., Siddiqi, S. M., and Gordon, G. J. (2011). Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966.

[Cressie, 1990] Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, 22(3):239–252.

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

[Felici et al., 2007] Felici, F., Van Wingerden, J.-W., and Verhaegen, M. (2007). Subspace identification of mimo lpv systems using a periodic scheduling sequence. *Automatica*, 43(10):1684–1697.

[Hsu et al., 2012] Hsu, D., Kakade, S. M., and Zhang, T. (2012). A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480.

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

[Kailath, 1980] Kailath, T. (1980). *Linear systems*, volume 1. Prentice-Hall Englewood Cliffs, NJ.

[Kalman, 1960] Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(Series D):35–45.

[Katayama, 2006] Katayama, T. (2006). *Subspace methods for system identification*. Springer.

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

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

[Lan and Schwager, 2015] Lan, X. and Schwager, M. (2015). Rapidly-exploring random cycles: Persistent estimation of spatio-temporal fields with multiple sensing robots. *IEEE Transactions on Robotics*. **Submitted** [pdf].

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

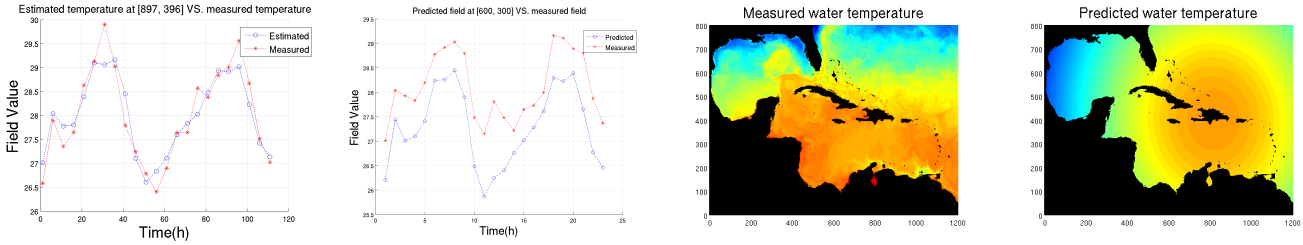
[Littman et al., 2002] Littman, M. L., Sutton, R. S., and Singh, S. (2002). Predictive representations of state. *Advances in neural information processing systems*, 2:1555–1562.

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

[NOAA,] NOAA. Noaa operational model archive and distribution system (oceannomads). `http://ecowatch.ncddc.noaa.gov/thredds/catalog/amseas/catalog.html`.

[Qin, 2006] Qin, S. J. (2006). An overview of subspace identification. *Computers & chemical engineering*, 30(10):1502–1513.

[Rosencrantz et al., 2004] Rosencrantz, M., Gordon, G., and Thrun, S. (2004). Learning low dimensional predictive representations. In *Proceedings of the 21st international conference on Machine learning*, page 88. ACM.



(a) Estimated ocean temperature at $(897, 396)$. (b) Predicted ocean temperature at $(600, 300)$. (c) Measured ocean temperature colormap. (d) Predicted ocean temperature colormap.

Fig. 11. Parameters learned by using 5 mobile sensors executing $N_s = 5$ and $r = 3$ motion strategy along the trajectory shown in Figure 10 in the Caribbean Sea. The first two plots show the estimated and predicted field values based on the learned model parameters A, $C(q)$, compared with the NOAA NOMADS temperature data. The last two plots show the colormap of the whole temperature field at one time instant for the NOMADS data and the learned model.

[Siddiqi et al., 2008] Siddiqi, S. M., Boots, B., and Gordon, G. J. (2008). A constraint generation approach to learning stable linear dynamical systems. Technical report, CMU-ML-08-101.

[Siddiqi et al., 2010] Siddiqi, S. M., Boots, B., and Gordon, G. J. (2010). Reduced-rank hidden markov models. In *International Conference on Artificial Intelligence and Statistics*, pages 741–748.

[Singh et al., 2004] Singh, S., James, M. R., and Rudary, M. R. (2004). Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 512–519. AUAI Press.

[Smith et al., 2012] Smith, S. L., Schwager, M., and Rus, D. (2012). Persistent robotic tasks: Monitoring and sweeping in changing environments. *Robotics, IEEE Transactions on*, 28(2):410–426.

[Van Overschee and De Moor, 1994] Van Overschee, P. and De Moor, B. (1994). N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

[Van Overschee and De Moor, 1995] Van Overschee, P. and De Moor, B. (1995). A unifying theorem for three subspace system identification algorithms. *Automatica*, 31(12):1853–1864.

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

[Van Wingerden and Verhaegen, 2009] Van Wingerden, J.-W. and Verhaegen, M. (2009). Subspace identification of bilinear and lpv systems for open-and closed-loop data. *Automatica*, 45(2):372–381.

[Verhaegen, 1994] Verhaegen, M. (1994). Identification of the deterministic part of mimo state space models given in innovations form from input-output data. *Automatica*, 30(1):61–74.

[Viberg, 1995] Viberg, M. (1995). Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851.

A Proofs

A.1 Proof of Lemma 1

Proof 1 *The proof for this lemma follows the proof of theorem* 6 *in [Van Overschee and De Moor, 1996]. Here we modify the proof to account for the changing C_t matrix. The proof is by induction. First, we prove that it holds for $t = 1$. Then we assume that it holds for $t = k$, we prove it holds for $t = k + 1$.*

When $t = 1$, then

$$\begin{aligned}
&\widehat{x}_1 = A\widehat{x}_0 + K_0(y_0 - C_0\widehat{x}_0) = K_0 y_0 \\
&K_0 = (G_0 - AP_0C_0^T)(\Lambda_0^0 - C_0P_0C_0^T)^{-1} = G_0(\Lambda_0^0)^{-1} \\
&\quad = \Delta_1^0(L_1^0)^{-1} \\
&\quad \Rightarrow \widehat{x}_1 = \Delta_1^0(L_1^0)^{-1}y_0
\end{aligned}$$

When $t = 1$, we can also calculate P_1:

$$\begin{aligned}
P_1 &= AP_0A^T + (G_0 - AP_0C_0^T)(\Lambda_0^0 - C_0P_0C_0^T)^{-1} \\
&\quad \cdot (G_0 - AP_0C_0^T)^T \\
&= G_0(\Lambda_0^0)^{-1}G_0^T \\
&= \Delta_1^0(L_1^0)^{-1}(\Delta_1^0)^T
\end{aligned}$$

Hence, (13) *and* (14) *hold for $t = 1$. Next we prove that if* (13) *and* (14) *hold for $t = k$, then they hold for $t = k + 1$. In this process, we will use the following formula*

for matrix inversion [Kailath, 1980].

$$
\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1}
$$

$$
= \left[\begin{array}{c|c} A_{11}^{-1} + A_{11}^{-1}A_{12} & -A_{11}^{-1}A_{12} \\ \cdot(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} & \cdot(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \\ \cdot A_{21}A_{11}^{-1} & \\ \hline -(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} & (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \\ \cdot A_{21}A_{11}^{-1} & \end{array}\right]
$$

Let's calculate P_{k+1} first:

$$
\begin{aligned}
P_{k+1} &= \Delta_{k+1}^0 (L_{k+1}^0)^{-1} (\Delta_{k+1}^0)^T \\
&= (A\Delta_k^0 \quad G_k) \begin{pmatrix} L_k^0 & (\Delta_k^0)^T C_k^T \\ C_k \Delta_k^0 & \Lambda_0^k \end{pmatrix}^{-1} \\
&\cdot \begin{pmatrix} (\Delta_k^0)^T A^T \\ G_k^T \end{pmatrix} \\
&= (A\Delta_k^0 \quad G_k) \\
&\cdot \left(\begin{array}{c|c} (L_k^0)^{-1} + (L_k^0)^{-1}(\Delta_k^0)^T & -(L_k^0)^{-1} \\ \cdot C_k^T \Delta^{-1} C_k \Delta_k^0 (L_k^0)^{-1} & \cdot(\Delta_k^0)^T C_k^T \Delta^{-1} \\ \hline -\Delta^{-1} C_k \Delta_k^0 (L_k^0)^{-1} & \Delta^{-1} \end{array}\right) \\
&\cdot \begin{pmatrix} (\Delta_k^0)^T A^T \\ G_k^T \end{pmatrix}
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta &= \Lambda_0^k - C_k \Delta_k^0 (L_k^0)^{-1} (\Delta_k^0)^T C_k^T \\
&= \Lambda_0^k - C_k P_k C_k^T \\
&= A\Delta_k^0 (L_k^0)^{-1} (\Delta_k^0)^T A^T + (G_k - A\Delta_k^0 (L_k^0)^{-1} \\
&\cdot (\Delta_k^0)^T C_k^T) \Delta^{-1} (G_k - A\Delta_k^0 (L_k^0)^{-1} (\Delta_k^0)^T C_k^T)^T \\
&= AP_kA^T + (G_k - AP_kC_k^T)(\Lambda_0^k - C_kP_kC_k^T)^{-1} \\
&\cdot (G_k - AP_kC_k^T)^T
\end{aligned}
$$

This proves (14). *Next we will prove* (13).

$$
\begin{aligned}
\widehat{x}_{k+1} &= \Delta_{k+1}^0 (L_{k+1}^0)^{-1} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix} \\
&= (A\Delta_k^0 \quad G_k) \begin{pmatrix} L_k^0 & (\Delta_k^0)^T C_k^T \\ C_k \Delta_k^0 & \Lambda_0^k \end{pmatrix}^{-1} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix} \\
&= (A\Delta_k^0 \quad G_k) \cdot \\
&\left(\begin{array}{c|c} (L_k^0)^{-1} + (L_k^0)^{-1}(\Delta_k^0)^T C_k^T \Delta^{-1} & -(L_k^0)^{-1} \\ \cdot C_k \Delta_k^0 (L_k^0)^{-1} & \cdot(\Delta_k^0)^T C_k^T \Delta^{-1} \\ \hline -\Delta^{-1} C_k \Delta_k^0 (L_k^0)^{-1} & \Delta^{-1} \end{array}\right) \\
&\cdot \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix} \\
&= (A - (G_k - A\Delta_k^0 (L_k^0)^{-1} (\Delta_k^0)^T C_k^T) \Delta^{-1} C_k^T) \\
&\cdot \Delta_k^0 (L_k^0)^{-1} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{k-1} \end{pmatrix} \\
&+ (G_k - A\Delta_k^0 (L_k^0)^{-1} (\Delta_k^0)^T C_k^T) \Delta^{-1} y_k \\
&= A\widehat{x}_k + (G_k - AP_kC_k^T)(\Lambda_0^k - C_kP_kC_k^T)^{-1} \\
&\cdot (y_k - C_k \widehat{x}_k) \\
&= A\widehat{x}_k + K_k(y_k - C_k\widehat{x}_k)
\end{aligned}
$$

This proves equation (13).

A.2 Proof of Theorem 1

Proof 2 *From the definition of H_t^d and the ergodicity of y_t, as $N \to \infty$,*

$$
\begin{aligned}
H_t^d &= E[Y_{t+d,r|t+2d-1} Y_{t,r|t+d-1}^T] \\
&\cdot E[Y_{t,r|t+d-1} Y_{t,r|t+d-1}^T]^\dagger Y_{t,r|t+d-1} \\
&= N_d^t (L_d^t)^{-1} Y_{t,r|t+d-1}
\end{aligned}
$$

It is easy to verify that

$$
N_d^t = O_{t,r|d} \Delta_d^t
$$

Hence, we have

$$H_t^d = O_{t,r|d}\Delta_d^t (L_d^t)^{-1} Y_{t,r|t+d-1}$$

By equation (15)*, we can write*

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

Since $O_{t,r|d}$ is full column rank, H_t^d has the same row space with $\widehat{X}_{t,r}$. The pair $(A, Q^{1/2})$ being controllable can guarantee that $\widehat{X}_{t,r}$ is full row rank, so H_t^d has the same column space with $O_{t,r|d}$. So we can do singular value decomposition for H_t^d and find its row space and column space, respectively. Then we can get $\widehat{O}_{t,r|d}$ and $\widehat{X}_{t,r}^{svd}$ up to a similarity transformation.

A.3 Proof of Theorem 2

Proof 3 *By* (21)*, we have*

$$\widehat{O}_{t,1|d}^{ref}\Gamma_t = O_{t,1|d}^{ref}, \quad \widehat{O}_{t+1,1|d}^{ref}\Gamma_{t+1} = O_{t+1,1|d}^{ref}$$

So we have the following equation:

$$\widehat{O}_t^{ref}\Gamma_t = O_t^{ref}, \quad \widehat{O}_{t+1}^{ref}\Gamma_{t+1} = O_{t+1}^{ref} = O_t^{ref}$$

By the above equation, we have

$$\widehat{O}_t^{ref}\Gamma_t = \widehat{O}_{t+1}^{ref}\Gamma_{t+1} \Rightarrow \Gamma_{t+1}\Gamma_t^{-1} = (\widehat{O}_{t+1}^{ref})^{\dagger}\widehat{O}_t^{ref}$$

That is, the transformation matrix between $\widehat{O}_{t+1}^{ref}$ and $\widehat{O}_t^{ref}$ is given by $\Gamma = \Gamma_{t+1}\Gamma_t^{-1}$. Hence, $\widehat{O}_{t+1,1|d}\Gamma$ and $\widehat{O}_{t,1|d}$ will stay in the same basis. So we have

$$\widehat{A} = \left(\overline{\widehat{O}_{t+1,1|d}\Gamma}\right)^{\dagger}\underline{\widehat{O}}_{t,1|d}$$

A.4 Proof of Theorem 3

Proof 4 *The proof is straight forward. To prove this theorem, we use a concrete example. Let's consider the case of using three sensors to collect measurements in Figure 2(c). In this case, $N_c = 2$, $N_s = 3$. Let $r = 3$. So all the three sensors will execute the motion that they move forward for one step and stay for $r = 3$ steps and move again. Without loss of generality, let's assume that they start at time step $t = 1$ and sensor* 1 *starts at waypoint* 1*, sensor* 2 *starts at waypoint* 2*, and sensor* 3 *starts at waypoint* 3*. When $K_d = 2$, the corresponding extended observability matrix is given by*

$$O_{1,3|6}^{mult} = \begin{pmatrix} C_1^{mult} \\ C_2^{mult}A \\ C_3^{mult}A^2 \\ C_4^{mult}A^3 \\ C_5^{mult}A^4 \\ C_6^{mult}A^5 \end{pmatrix} \quad \bar{O}_{1,3|6}^{mult} = \begin{pmatrix} C_1^{mult} \\ C_2^{mult}A \\ C_3^2A^2 \\ C_3^3A^2 \\ C_4^{mult}A^3 \\ C_5^{mult}A^4 \end{pmatrix}$$

$$\underline{O}_{1,3|6}^{mult} = \begin{pmatrix} C_2^{mult}A \\ C_3^{mult}A^2 \\ C_4^1A^3 \\ C_4^2A^3 \\ C_5^{mult}A^4 \\ C_6^{mult}A^5 \end{pmatrix}$$

where $C_t^{mult} = \left[C_t^1; C_t^2; C_t^3\right]$ for all $t = 1, 2, \ldots, 6$. And $C_1^{mult} = C_2^{mult} = C_3^{mult}$, $C_4^{mult} = C_5^{mult} = C_6^{mult}$. We also have $C_3^2 = C_4^1$ and $C_3^3 = C_4^2$ since sensor 1 and sensor 2 are consecutive pairs $1 \twoheadrightarrow 2$ and sensor 2 and sensor 3 are also consecutive pairs $2 \twoheadrightarrow 3$. Hence, the number of rows in $\bar{O}_{1,3|6}^{mult}$ is 14, which is equal to $N_s(r-1)K_d + N_c(K_d - 1)$ when $N_s = 3, r = 3, K_d = 2, N_c = 2$. The $N_s(r-1)K_d$ rows come from $\left[C_1^{mult}; C_2^{mult}A; C_4^{mult}A^3; C_5^{mult}A^4\right]$ and the $N_c(K_d - 1)$ rows come from $\left[C_3^2A^2; C_3^3A^2\right]$. This proof can be generalized to any N_s, N_c, r and K_d except that $N_s = 1$ and $r = 1$ at the same time. In this case, we only have one sensor and it keeps moving along the periodic trajectory and $N_c = 0$. We need to use a static reference sensor to learn the A matrix and $K_d \geq n + 1$.