Optimal Mobile Sensing Policy for Parameter Estimation of Distributed Parameter Systems: Finite Horizon Closed-loop Solution

Christophe Tricaud and YangQuan Chen

1 Introduction

In control engineering practice, it is very common to estimate the parameters of a system given a mathematical model. Using observations or measurements, one can parameterize the model using different techniques. Sometimes, when the system to be modeled is spatially and temporally dynamic (i.e. the states depend on both time and space), common lumped parameter input-output relationships cannot characterize the system dynamics and instead, we must use partial differential equations (PDEs) for modeling.

However, making observations or measurements of the states of a distributed parameter system is not an easy and straightforward task. One needs to consider the location of the sensors so that the gathered information best helps model parameter identification. This problem of sensor location is not new but most of the work achieved so far is limited to stationary sensors in the framework of wireless sensor networks [5, 14, ].

Mobile sensors that are now available both via mobile robots or unmanned air vehicles (UAVs), offer a much more interesting alternative to stationary sensors for distributed parameter systems characterization. Indeed, when looking for optimal

\*This work is supported in part by the NSF International Research and Education in Engineering (IREE) grant #0540179. Christophe Tricaud is supported by Utah State University President Fellowship (2006-2007).
location of a stationary sensor, one only seeks the average optimal over time and space. However, the optimal location to gather sensible data about a given distributed parameter system is not necessarily static but in most cases dynamic. It is therefore logical to expect a better estimation of the PDE based system if mobile sensors are used than only considering stationary sensors.

Areas of application of such mobile sensing techniques includes air pollutants monitoring using cars equipped with sensors on the ground and aircrafts in the air. In addition, low cost platforms for mobile sensors with wireless communications are now available and becoming cheaper and cheaper. As said, a set of such autonomous vehicles equipped with sensors can potentially improve the efficiency of the measurements. As technology evolves, it is necessary and practically meaningful to consider using mobile sensors for optimal measurement of distributed parameter systems with an objective of unknown parameter estimation.

In this paper, we consider this type of research problem first introduced in [11, ], where the optimal observations of a DPS based on diffusion equations were made by two-wheeled differentially driven mobile robots equipped with sensors.

In the field of mobile sensor trajectory planning in a distributed parameter system setting, few approaches have been developed. So far, the available solutions are not quite practically appealing. For example, Rafajówicz [9, ] investigated the problem using the determinant of the Fisher Information Matrix (FIM) associated with the parameters to be estimated. The determinant of the FIM is used as a metric evaluating the accuracy of the parameters estimation. However, the results are more of an optimal time-dependent measure than a trajectory. In [14, ] and [13, ], Uciński reformulated the problem of time-optimal path planning into a state-constrained optimal control one which allows the addition of different constraints on the dynamics of the sensor. In [15, ], Uciński and Chen tried to properly formulate and solve the time-optimal problem for moving sensors observing the state of a DPS for optimal parameter estimation. In [11, ], realistic constraints to the dynamics of the mobile sensor are considered when a differential-drive mobile robot in the framework of the MAS-net Project [3, ]. It is pinpointed in [11, ] that one of the fundamental problems in DPS-parameter estimation using mobile sensors is that the optimal paths for the DPS-parameter estimation are conditional on the very parameters’ value that have to be estimated which are unknown. Given parameters, how to optimally plan the motion trajectories of the mobile sensors has been known in the literature [14, 7, ], where the purpose of mobile sensing is to best estimate the parameters. Clearly, there is a “chick-and-egg” problem regarding the optimal mobile sensor motion planning and parameter estimation for distributed parameter systems.

This paper, for the first time, solved this problem by the proposed optimal interlaced mobile sensor motion planning and parameter estimation. The problem formulation is given in detail with a numerical solution for generating and refining the mobile sensor motion trajectories for parameter estimation of the distributed parameter system. The basic idea is to use the finite horizon control type of scheme. First, the optimal trajectories are computed in a finite time horizon based on the assumed parameter values. For the following time horizon, the parameters of the distributed parameter system are estimated using the measured data in the previous time horizon, and the optimal trajectories are updated accordingly based on these estimated parameters obtained. Simulations are offered to illustrate the advantages
of the proposed interlaced method over the non-interlaced techniques. We call the proposed scheme “on-line” or “real-time” which offers practical solutions to optimal measurement and estimation of a distributed parameter system when mobile sensors are used. It should be mentioned that this “on-line” problem has been recognized in the last chapter of [7, ] as an “extremely important” research effort.

2 Optimal Measurement Problem Definition

Let us consider the scalar (possibly non-linear) DPS defined by

$$\frac{\partial y}{\partial t} = F(x_1, x_2, t, y, \frac{\partial y}{\partial x_1}, \frac{\partial y}{\partial x_2}, \frac{\partial^2 y}{\partial x_1^2}, \frac{\partial^2 y}{\partial x_2^2}),$$

with initial conditions and boundary conditions of the following form

$$y(x_1, x_2, 0) = y_0(x_1, x_2)$$

$$E(x_1, x_2, t, y, \theta) = 0,$$

where $\Omega \in \mathbb{R}^2$ is a fixed, bounded, open set with sufficiently smooth boundary $\partial \Omega$. The points in $\Omega$ are denoted as $x = (x_1, x_2)$. $F$ and $E$ are some known functions; $y_0$ is a given initial state; $t$ is the time and $T$ is a fixed known time interval. $y = y(x_1, x_2, t)$ represents the state variable with values in $\mathbb{R}$.

From now on, we will assume that $u$ depends on the vector $\theta \in \mathbb{R}^m$ of unknown parameters to be estimated from measurements made by $N$ moving sensors over the observation horizon $T$. We will denote $x^j : T \rightarrow \Omega_{ad}$ the moving trajectory or path of the $j$-th sensor, where $\Omega_{ad} \subset \Omega$ is the area where measurements are possible.

The observations are assumed to be of the form

$$z(t) = y_m(t) + \epsilon_m(t), t \in T,$$

where

$$y_m(t) = \text{col}[y(x^1(t), t), ..., y(x^N(t), t)],$$

$$\epsilon_m(t) = \text{col}[\epsilon(x^1(t), t), ..., \epsilon(x^N(t), t)],$$

$z(t)$ is the vector of measurements of dimension $N$ and $\epsilon = \epsilon(x, t)$ is a zero-mean white Gaussian noise.

With such a setup, the problem of parameter identification is as follows. Given the model (1) and the measurements $z$ along the trajectories $(x^j)$, $j = 1, \cdots, N$, obtain an optimal estimation $\hat{\theta} \in \Theta_{ad}$ ($\Theta_{ad}$ being the set of admissible parameters) that minimizes the weighted least squares criterion given in [2, 6, ], and reformulated in section 3:

$$J(\theta) = \frac{1}{2} \int_0^T \|z(t) - \hat{y}(x, t; \theta)\|^2 dt$$

where $\hat{y}(x, t; \theta)$ denotes the solution to (1) corresponding to a given parameter $\theta$, $\| \cdot \|$ is the Euclidean norm.
The parameter estimate \( \hat{\theta} \) is influenced by the sensors trajectories \( \mathbf{x}^j \) and our goal is to obtain the best estimates of the system parameters. It is hence logical to decide on the trajectory based on a quantitative measure related to the expected accuracy of the parameter estimates to be obtained from the data collected. The Fisher Information Matrix (FIM) \([9, 12, \]) is proved to be suitable criterion when looking for best measurements in distributed parameter systems. Its inverse constitutes an approximation of the covariance matrix for the estimate of \( \theta \) \([16, 4, 1, \]). Let us write

\[
s(t) = (\mathbf{x}^1(t), ..., \mathbf{x}^N(t)), \forall t \in T
\]

and let \( n = \dim(s(t)) \). Under such conditions, the FIM can be written as \([8, \]

\[
M(s) = \sum_{j=1}^{N} \int_0^T g(\mathbf{x}^j(t), t) g^T(\mathbf{x}^j(t), t) dt
\]

where \( g(\mathbf{x}, t) = \nabla_{\theta} y(x_1, x_2; \theta)|_{\theta = \theta_0} \) is the vector made of the sensitivity coefficients, \( \theta_0 \) being a previous estimate of the unknown parameter vector \( \theta \) \([14, 13, \].

By choosing \( s \) so as to minimize some scalar function \( \Psi \) of the information matrix one can find optimal sensor trajectories. There are different possibilities for such a function \([16, 4, 1, \] but the most commonly used is the D-optimality criterion

\[
\Psi(M) = - \log \det(M)
\]

which minimizes the volume of the confidence ellipsoid for the estimates. In our case, it will represent a measure of the information content of the observations for DPS parameter estimation.

3 Parameter Estimation

The purpose of the parameter estimation is to determine the value of the parameters \( \hat{\theta} \in \Theta_{ad} \) from a set of available measurements. Those measurements should be close to the output of the model in order, for the parameters, to be well estimated. The parameter estimation problem is generally transformed into a static parametric optimization problem minimizing a weighted least squares criterion \( J(\theta) \) as given below.

\[
J(\theta) = \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{K} \int_0^T \| z^j(t_k) \|
\]

\[
- \mathcal{H}(\hat{y}(\mathbf{x}^j(t_k), t_k; \theta), \mathbf{x}^j(t_k), t_k) dt
\]

where \( \hat{y}(\mathbf{x}, t; \theta) \) represents the solution of (1) for a given \( \theta \) and \( \mathcal{H}(\cdot) \) is the function mapping its argument in \( \mathbb{R}^r \) such that

\[
z^j(t_k) = \mathcal{H}(\hat{y}(\mathbf{x}^j(t_k), t_k; \theta), \mathbf{x}^j(t_k), t_k) + \epsilon(\mathbf{x}^j(t_k), t_k).
\]
4 A DPS and Its Mobile Sensors

To get ready for simulation demonstration, let us start with a generic DPS model describing a diffusion process with unknown parameters. Then, we define the mobile sensors used for taking measurements of this system. Our ultimate goal is to best identify the unknown DPS parameters using these mobile sensors.

4.1 A Diffusion Process Model

The model used for a specific diffusion process is the same as in [11, ] except that the parameters are assumed unknown. This allows us to compare the results between different estimation techniques.

\[
\frac{\partial y(x_1, x_2, t)}{\partial t} = \frac{\partial}{\partial x_1}(\kappa(x_1, x_2) \frac{\partial y(x_1, x_2, t)}{\partial x_1}) \\
+ \frac{\partial}{\partial x_2}(\kappa(x_1, x_2) \frac{\partial y(x_1, x_2, t)}{\partial x_2}) \\
+ 20 \exp(-50(x_1 - t)^2),
\]

\( (x_1, x_2) \in \Omega = (0, 1) \times (0, 1), t \in T, \)

\[ y(x_1, x_2, 0) = 0 \]

\[ y(x_1, x_2, t) = 0 \]

\[ T = \{ t | t \in (0, 1) \} \]

\[ \kappa = \theta_1 + \theta_2 x_1 + \theta_3 x_2 \]

\[ \theta_1 = 0.1, \theta_2 = 0.6, \theta_3 = 0.8 \]

where \( y(x_1, x_2, t) \) is the concentration of the considered diffusing substance, \( \kappa(x_1, x_2) \) is the diffusion coefficient for the spatial coordinate \((x_1, x_2)\); \( t \) is the time and \( \theta_1, \theta_2, \theta_3 \) are the unknown values of the parameters to be estimated. The assigned values for \( \theta_1, \theta_2, \theta_3 \) are just for simulation comparison purpose.

4.2 Mobile Sensor Model

Sensor Dynamics

We assume that the sensors are mounted on vehicles whose dynamics are described by the following equation

\[ s(t) = f(s(t), u(t)) \quad \text{a.e. on} \quad T, \quad s(0) = s_0 \quad (11) \]

where the function \( f : \mathbb{R}^n \mathbb{R}^r \rightarrow \mathbb{R}^n \) is continuously differentiable; \( s_0 \text{in}(\mathbb{R})^n \) represents the initial position of the sensors, and \( u : T \rightarrow (\mathbb{R})^r \) is a measurable control function satisfying the following inequality

\[ u_l \leq u(t) \leq u_u \quad \text{a.e. on} \quad T \quad (12) \]

for some constant vectors \( u_l \) and \( u_u \). We assume that all the vehicles have to stay within an admissible region \( \Omega_{ad} \) (a given compact set) where measurements are possible. \( \Omega_{ad} \) can be conveniently defined as
\[ \Omega_{ad} = \{ x \in \Omega : b_i(x) = 0, i = 1, \cdots, I \} \quad (13) \]
where the \( b_i \) are known continuously differentiable functions. That is, the following constraints have to be satisfied:
\[ h_{ij}(s(t)) = b_i(x^j(t)) \leq 0, \forall t \in T \quad (14) \]
where \( 1 \leq i \leq I \) and \( 1 \leq j \leq N \). For simpler notation, we reformulate the conditions described in (14) in the following way
\[ \gamma_l(s(t)) \leq 0, \forall t \in T, \quad (15) \]
where \( \gamma_l, l = 1, \cdots, \nu \) tally with (14), \( \nu = I \times N \).

It is possible to consider additional constraints on the path of each vehicle such as specific dynamics, collision avoidance and any other constraints. For example, we can restrict the minimum distance between the vehicles. Such constraint can be achieved by forcing the following condition
\[ \beta_{ij}(s(t)) = R^2 - \| x^i(t) - x^j(t) \|^2 \leq 0, \forall t \in T \quad (16) \]
where \( 1 \leq i < j \leq N \) and \( R \) is the minimum distance ensuring that the measurements taken by the sensors can be considered as uncorrelated, or ensuring that the vehicles will not collide each other during the experiment.

**Sensor model**

Measurements from the \( j^{th} \) sensor are modeled as follow:
\[ z^j(t_k) = u(x^j, t_k) + \epsilon(x^j, t_k) \quad (17) \]
where \( \epsilon \) is the measurement noise considered with zero-mean, Gaussian, uncorrelated in time and space. Its statistics are therefore
\[ E[\epsilon(x^j, t)] = 0, \quad (18) \]
\[ E[\epsilon(x^j, t)\epsilon(x^i, \tau)] = \delta_{ij}\delta(t - \tau)\sigma \quad (19) \]
where \( x^j \in \Omega, x^i \in \Omega, \delta_{ij} \) and \( \delta(\cdot) \) stand for Kronecker’s and Dirac’s delta symbols, respectively.

### 5 Interlaced Optimal Trajectory Planning

#### 5.1 Optimal Trajectory Planning

In order to solve the problem described in Sec. 2, we need to reformulate the problem in the optimal control framework. The solver used for this optimal control problem is called RIOTS \([10, \cdots]\). RIOTS stands for “recursive integration optimal trajectory solver.” It is a Matlab toolbox programmed to solve a very broad class of optimal control problems. According to \([10, \cdots]\), our optimal trajectory planning problem can be solved using the RIOTS toolbox if rephrased as follows
\[ \min_{(u, \xi) \in L^2N_{\infty}(t_0, t_f) \times \mathbb{R}^K} J(u, \xi) \quad (20) \]
where
\[ J(u, \xi) = g_0(\xi, x(t_f)) + \int_{t_0}^{t_f} l_0(x, t, u) dt \] (21)
and is subject to the following conditions and constraints
\[ \dot{x} = h(x, t), \]
\[ x(t_0) = \xi, t \in [t_0, t_f], \]
\[ u_{min}^{(j)}(t) \leq u^{(j)}(t) \leq u_{max}^{(j)}(t), j = 1, ..., N, t \in [t_0, t_f], \]
\[ \xi_{min}^{(j)}(t) \leq \xi^{(j)}(t) \leq \xi_{max}^{(j)}(t), j = 1, ..., K, t \in [t_0, t_f], \]
\[ l_t(x(t), t, \tau(t)) \leq 0, t \in [t_0, t_f], \]
\[ g_{ei}(\xi, x(t_f)) \leq 0, g_{ee}(\xi, x(t_f)) = 0. \]

In the case of our optimal trajectory planning problem, \( \dot{x} = h(t, x, u) = Ax + Bu. \) Instead of defining \( l_0(\xi, x(t_f)) = \Psi(M) \), we choose to define \( g_0(\xi, x(t_f)) = \int_{t_0}^{t_f} \Psi(M) dt \), in order to lower the amount of calculations.

The methodology used further is known as the equivalent Mayer problem. We define the Mayer states as
\[ \chi_{i,j}(t) := \sum_{l=1}^{N} \int_{t_0}^{t} g_{i}(x^l, t)g_{j}^T(x^l, t) dt \] (22)
where \( g_{i}(x^l, t) \) is the sensitivity function of the \( l \)-th sensor regarding the \( i \)-th parameter.

According to (7), \( \chi_{i,j}(t) \) is the entry of the FIM corresponding to the \( i \)-th row and \( j \)-th column.

Denote \( \chi_{dl} \) the vector where all the entries on the diagonal and below the diagonal of \( \chi \) are stacked. The extended Mayer state vector \( \tilde{x} \) is then defined as
\[ \tilde{x} = \begin{bmatrix} x \\ \chi_{dl} \end{bmatrix} \] (23)
Since \( M \) is symmetric, \( \chi_{dl} \) contains all the information of \( M \). This reveals the interest of introducing the equivalent Mayer problem in our framework. When we exchange the extended state vector \( x \) with the extended Mayer vector \( \tilde{x} \), we can get \( M \) without having to compute any integration. Thus, when considering the equivalent Mayer problem, the model used for RIOTS is as follows:
\[ \dot{\tilde{x}} = \dot{x} + u. \] (24)

The sensitivity function is obtained “offline” or beforehand using the Matlab PDE Toolbox, prior to the function call of RIOTS by Matlab. The computation of the sensitivity function requires solutions of the followings equations:
\[ \frac{\partial y}{\partial t} = \nabla \cdot (\kappa \nabla y) + 20e^{50(x_1 - t)^2} \]
\[ \frac{\partial \theta_1}{\partial t} = \nabla \cdot \nabla y + \nabla \cdot (\kappa \nabla \theta_1) \]
\[
\frac{\partial \theta_2}{\partial t} = \nabla \cdot (x_1 \nabla y) + \nabla \cdot (\kappa \nabla \theta_2)
\]
\[
\frac{\partial \theta_3}{\partial t} = \nabla \cdot (x_2 \nabla y) + \nabla \cdot (\kappa \nabla \theta_3)
\]

where \( \nabla = \partial / \partial x_1 + \partial / \partial x_2 \). Note that there are three \( \theta \) equations since there are 3 parameters \( \theta_1, \theta_2, \theta_3 \).

### 5.2 Measurements and Parameters Estimation

Once the optimal trajectories have been computed, the measurements are done as described in Sec. 4.2. However, the observations are completed until the end of the finite horizon for which the trajectory was computed. Instead, after a fraction of the horizon, the data gathered so far are used to refine the estimation of the parameters values.

In order to determine refined values of the parameters, we use the Matlab command “\texttt{lsqnonlin}”, a routine for solving non-linear least squares problems and especially for our case, the least squares fitting problems. “\texttt{lsqnonlin}” allows the user to incorporate own function to compute. In our problem, the input of the function is a set of parameters as well as the measurements and the output is the error between the measurement and the simulated value of the measurement for the set of parameters.

\[
\min_\theta = \frac{1}{2} \sum_{i=1}^{N} f_i(\theta)^2 \quad (25)
\]

with

\[
f_i(\theta) = z^i(t_0, ..., t_k)
- \mathcal{H}(y(x^i(t_0, ..., t_k), t_0, ..., t_k; \theta), x^i(t_0, ..., t_k), t_0, ..., t_k)
\quad (26)
\]

Prior to the experiment, we determine the value of the state \( y(x, t, \theta) \) for a set of parameter value \( \theta \in \Theta_{\text{ad}} \) in an offline manner. We assume that the state variations between two values of a parameter are linear enough to allow interpolation. Using this database obtained “offline” allows faster computation of the function to be called by the optimization algorithm.

### 5.3 Summary of The Interlaced Scheme

Let us summarize the interlaced strategy step by step

1. Given a set of parameters \( \hat{\theta} \) for the DPS (its initial value being given prior to the first iteration), we design an optimal experiment, i.e., optimal trajectories for the mobile sensors to follow.

2. The sensors takes measurements along their individually assigned trajectories. Measurements are simulated taking the real value of the state along the trajectory and adding zero-mean white noise.
3. Measurement data are used to refine the estimate of the parameters using an optimization routine such as “lsqnonlin”. The optimization routine computes the parameters such that the difference between the measurements and the simulated values of the state along the trajectory is minimized. Go back to Step-1.

The above algorithm is illustrated in Fig. 1.

![Figure 1. The interlaced scheme illustrated](image)

### 6 Illustrative Simulations

In this section, we focus our attention on the performance of the methodology. The experiment is ran for different noise statistics and for each case results are given in the form of sensor trajectories and parameter estimates. For case 1, $\sigma = 0.0001$, for case 2, $\sigma = 0.001$, and for case 3, $\sigma = 0.01$. In all cases, we consider 3 mobile sensors. The control of the mobile sensors $u$ is limited between $-0.7$ and $0.7$. All three sensors have fixed initial positions ($x_1(0) = (0, 0.1)$, $x_2(0) = (0, 0.5)$ and $x_3(0) = (0.1, 0.9)$). The results for the previously defined case are respectively given in Fig. 2 for Case 1, in Fig. 3 for Case 2 and in Fig. 4 for Case 3. For each figure, subfigure (a) gives the sensor trajectories, the evolution of the estimates is shown in (b) and the measurements are given in (c).

![Figure 2. Closed-loop D-Optimum experiment for $\sigma = 0.0001$](image)

From these figures, we have the following observations:

- In all the cases, the sensors have similar trajectories as they try to follow the excitation wave along the $x_1$ axis $20 \exp(-50(x_1 - t)^2)$.

- For low noise amplitude (Cases 1 and 2), the experiment is long enough to obtain good estimates of the parameters. In case 3, the experiment is not long enough to obtain convergence.
In all cases, we can clearly observe that the trajectories of the mobile sensors change as the estimated values of the parameters are getting closer to the real values.

7 Conclusion and Future Work

In this paper, we introduced a numerical procedure for optimal sensor-motion scheduling of diffusion systems for parameter estimation. With the knowledge of the PDE governing a given DPS, mobile sensors find an initial trajectory to follow and refine the trajectory as their measurements allows to find a better estimate of the system’s parameters. Using the Matlab PDE toolbox for the system’s simulations, RIOTS Matlab toolbox for solving the optimal path-planning problem and Matlab Optimization toolbox for the estimation of the system’s parameters, we were able to solve this parameter identification problem in an interlaced manner successfully. Simulation results are presented to show both the advantages of the strategy and the convergence of the estimation.

Such an online estimation methodology sets the base for exciting future research. Indeed, we can now investigate problems related to communication between nodes such as time varying topology, communication range, information loss and other well-known problems in the scope of multi-agent systems.
Bibliography


