Sensor Motion Planning in Distributed Parameter Systems Using Turing's Measure of Conditioning

Dariusz Uciński and YangQuan Chen

Abstract— **We present a technique for planning sensor motions in a specified two-dimensional spatial domain in such a way as to make the Hessian of the parameter estimation cost well conditioned. The framework is based on the use of Turing's measure of conditioning, whose minimization yields the confidence regions for the parameters as spherical as possible. Since this does not necessarily guarantee a high information content in the measurements, an additional constraint is imposed on the D-efficiency of the solutions. Then the approach converts the problem to an optimal control one in which both the control forces of the sensors and the initial sensor positions are optimized. Numerical solutions are then obtained using the MATLAB PDE toolbox and the RIOTS_95 optimal control toolbox which handles various constraints imposed on the sensor motions.**

I. INTRODUCTION

In numerous applications, one is confronted with the task of estimating unknown parameters in mathematical models from observations of the underlying physical phenomenon being modelled. If this phenomenon is distributed in nature, i.e., its state depends not only on time, but also on spatial coordinates, then we deal with a distributed-parameter system (DPS) for which appropriate mathematical modelling most often yields partial differential equations (PDEs). The inability to take distributed measurements of process states then leads to the question of where to locate sensors so that the information content of the resulting signals be as high as possible. The sensor location problem for parameter estimation was attacked from various angles, but the results communicated by most authors are limited to the selection of stationary sensor positions (for reviews, see [1], [2]). An appealing alternative to such an approach is to apply spatially-movable sensors. This is due to the fact that sensors are not assigned to fixed positions which are optimal only on the average, but are capable of tracking points which provide at a given time moment best information about the parameters to be identified. Consequently, by actively reconfiguring a sensor system we can expect the minimal value of an adopted design criterion to be lower than the one for the stationary case. It is important to note that planning techniques developed for moving sensors can prove useful in many areas of automation. A possibility of using moving observations does arise in a variety of applications, e.g., air pollutants in the environment are often measured using data gathered by monitoring cars moving in an urban area and atmospheric variables are measured using instruments carried in an aircraft. What is more, technological advances in communication systems and the growing ease in making small, low power and inexpensive mobile systems now make it feasible to deploy a group of networked vehicles in a number of environments [3], [4]. A cooperated and scalable network of vehicles, each of them equipped with a single sensor, has the potential to substantially improve the performance of the observation systems. Applications in various fields of research are being developed and interesting ongoing projects include extensive experimentation based on testbeds. The problem to be discussed in this paper cought our attention while working on one of such experimental platforms, namely the MAS-net lab testbed being a distributed system equipped with two-wheeled differentially driven mobile robots capable of sensing the states of DPSs described by diffusion and wave equations [5], [6].

Surprisingly, few works have been reported regarding a systematic approach to mobile observer planning and the problem still waits for satisfactory solutions. Rafajłowicz [7] considers the determinant of the Fisher Information Matrix (FIM) associated with the parameters to be estimated as a measure of the identification accuracy and looks for an optimal time-dependent measure, rather than for the trajectories themselves. On the other hand, Uciński [2], [8], apart from generalizations of Rafajłowicz's results, develops some computational algorithms based on the FIM. He reduces the problem to a state-constrained optimal-control one for which solutions are obtained via the methods of successive linearizations which is capable of handling various constraints imposed on sensor motions. In turn, the work [9] was intended as an attempt to properly formulate and solve the time-optimal problem for moving sensors which observe the state of a DPS so as to estimate some of its parameters.

In the sensor location literature, various design criteria defined on the FIM are considered, but among them the determinant (which leads to the so-called D-optimum designs) is the most intensively studied of all since its use leads to minimum volumes of the confidence ellipsoids for the estimates. However, a serious drawback of the D-optimality criterion is that the volume of a confidence ellipsoid may be small because the ellipsoid is elongated. In practice, this means that there may be linear functionals of parameters which will be estimated with a very large variance, even though the measurements have been collected in D-optimal experimental

The work of D. Uciński was supported by the State Committee for Scientific Research under Grant 3 T11C 035 27.

D. Uciński is with the Institute of Control and Computation Engineering, University of Zielona Góra, 65-246 Zielona Góra, Poland d.ucinski@issi.uz.zgora.pl

Y. Chen is with the Department of Electrical and Computer Engineering, Utah State University, Logan, Utah 84322, USA yqchen@ece.usu.edu

conditions. In this work we outline an approach to overcome this severe impediment. It is based on the use of Turing's measure of conditioning for the FIM, whose minimization yields the confidence regions for the parameters as spherical as possible. This measure has been severely criticized for producing solutions with large volumes of the resulting confidence ellipsoids. However, we demonstrate here how to combine both the criteria so as to eliminate their individual disadvantages and maintain a low complexity regarding possible implementations of the resulting strategy in practice. Moreover, we also show how the proposed formulation can be transformed into an equivalent optimal control problem in Mayer form, which can be efficiently solved by the MATLAB toolbox RIOTS_95, a high-performance tool for solving optimal control problems [10].

II. SENSOR LOCATION PROBLEM IN QUESTION

Consider a DPS described by the partial differential equation

$$\frac{\partial y}{\partial t} = \mathcal{L}\big(y, \theta\big) \quad \text{in } \Omega \times]0, t_f[\tag{1}$$

subject to given boundary and initial conditions, where $\Omega \subset \mathbb{R}^2$ is a fixed, bounded, open set with sufficiently smooth boundary $\partial\Omega$, $y = y(x, t; \theta)$ denotes the scalar state at a spatial point $x \in \bar{\Omega} = \Omega \cup \partial\Omega$ and time instant $t \in T = [0, t_f]$, $t_f < \infty$, and $\mathcal{L}$ signifies a (possibly nonlinear) differential operator which involves first- and second-order spatial derivatives and may include terms accounting for forcing inputs which are given *a priori*. In this description, $\theta \in \mathbb{R}^m$ represents an unknown constant parameter vector which must be estimated using observations of the system.

In what follows, we consider the observations provided by N moving pointwise sensors. Let $x^j : T \longrightarrow \Omega_{\mathrm{ad}}$ be the trajectory of the j-th sensor, where $\Omega_{\mathrm{ad}} \subset \Omega$ stands for the region where measurements can be made. The observations are assumed to be of the form

$$z(t) = y_{\mathrm{m}}(t) + \varepsilon_{\mathrm{m}}(t), \quad t \in T, \tag{2}$$

where

$$y_{\mathrm{m}}(t) = \mathrm{col}[y(x^1(t), t), \dots, y(x^N(t), t)], \tag{3}$$
$$\varepsilon_{\mathrm{m}}(t) = \mathrm{col}[\varepsilon(x^1(t), t), \dots, \varepsilon(x^N(t), t)], \tag{4}$$

$z(t)$ is the N-dimensional observation vector and $\varepsilon = \varepsilon(x, t)$ is a white Gaussian noise process (a formal time derivative of a Wiener process) whose statistics are

$$\begin{cases} \mathrm{E}\big\{\varepsilon(x, t)\big\} = 0, \\ \mathrm{E}\big\{\varepsilon(x, t)\varepsilon(x', t')\big\} = \sigma^2 \delta(x - x')\delta(t - t'), \end{cases} \tag{5}$$

$\sigma > 0$ being the standard deviation of the measurement noise and δ the Dirac delta function concentrated at the origin. The assumption that we are in a position to observe directly the system state is made only for simplicity of presentation. The approach outlined in what follows can easily be generalized to indirect observation of state variables.

Sensor trajectories which guarantee the best accuracy of the least-squares estimates of θ can be found by choosing x^j, $j = 1, \dots, N$ so as to minimize some scalar measure of performance Ψ defined on the *average Fisher Information Matrix* (FIM) *Fisher Information Matrix* (FIM) [7] which is widely used in optimum experimental design theory for lumped systems [11], [12]. When the time horizon is large, the nonlinearity of the model with respect to its parameters is mild and the measurement errors are independently distributed and have small magnitudes, the inverse of the FIM constitutes a good approximation of the covariance matrix for the estimate of θ [11], [12].

For notational convenience, introduce

$$s(t) = (x^1(t), x^2(t), \dots, x^N(t)), \quad \forall\, t \in T \tag{6}$$

and set $n = \dim(s(t))$. The FIM has the following representation [2]:

$$M(s) = \sum_{j=1}^{N} \int_0^{t_f} g(x^j(t), t) g^{\mathsf{T}}(x^j(t), t)\, \mathrm{d}t, \tag{7}$$

where $g(x, t) = \nabla_\theta y(x, t; \theta)\big|_{\theta=\theta^0}$ denotes the vector of the so-called *sensitivity coefficients*, θ^0 being a prior estimate to the unknown parameter vector θ [2], [8].

Optimal sensor trajectories can be found by choosing s so as to maximize some scalar function Ψ of the information matrix. The introduction of the design criterion permits to cast the sensor location problem as an optimization problem, and the criterion itself can be treated as a measure of the information content of the observations. Several choices exist for such a function [11], [12] and the most popular one is the D-optimality criterion

$$\Psi[M] = \log\det(M). \tag{8}$$

Its use yields the minimal volume of the confidence ellipsoid for the estimates.

Unfortunately, application of the D-optimality criterion may lead to the FIMs which correspond to very elongated confidence ellipsoids and, therefore, to situations where the uncertainty of the inference based on the resulting parameter estimates will still be high. In this context, a desirable additional characteristic of the solutions is that the confidence ellipsoids be as spherical as possible. As a measure of sphericality, we propose here the following criterion, called *Turing's measure of conditioning* [13]:

$$\Upsilon[M] = \frac{1}{m}\sqrt{\mathrm{trace}(M)\,\mathrm{trace}(M^{-1})}. \tag{9}$$

It is an easy exercice to check that its minimum value is unity, which is achievable only for spherical confidence regions. As was shown in [2, Sec. 2.5], the second Gâteaux derivative at a global minimum $\widehat{\theta}$ of the least-squares criterion (**??**) is approximately equal, up to a constant multiplier, to the corresponding FIM. Thus, minimization of $\Upsilon(M)$ with respect to s will simultaneously make an approximation H to the Hessian of the estimation cost well conditioned in the sense that it will yield minimization with respect to s of the Frobenius condition number defined as [11]

$$\mathfrak{J}(s) = \sqrt{\mathrm{trace}[H(s)]\,\mathrm{trace}[H^{-1}(s)]}. \tag{10}$$

The direct use of Turing's measure of conditioning has been criticized [2, p. 94] since, in spite of its clear rationale, the approach only guarantees that the condition number is close to unity and no more than that. This means that we might have a low value of $\mathfrak{J}$ and, at the same time, little information about the parameters.

Ideally, measurements should yield confidence ellipsoids which are as spherical as possible and simultaneously guarantee an acceptable level of the information content of the collected observations. To reach this compromise, we propose here an approach which relies on the notion of the D-efficiency which quantifies the D-suboptimality of given trajectories. In much the same way as in the classical optimum experimental design [11], we define it here as follows:

$$E_D(s) = \left\{ \frac{\det(M(s))}{\det(M(s^\star))} \right\}^{1/m}, \tag{11}$$

where $s^\star$ stands for the D-optimal trajectories which can be determined beforehand. Setting a reasonable positive threshold $\eta < 1$, we can impose the constraint

$$E_D(s) \geq \eta \tag{12}$$

which will yield a D-suboptimal yet reasonable solution. It follows immediately that (12) is equivalent to the constraint

$$\Psi[M(s)] \leq C, \tag{13}$$

where $C = \Psi[M(s^\star)] - m\log(\eta)$.

Accordingly, in the sequel, we shall be concerned with minimization of the criterion

$$\Phi[M(s)] = \operatorname{trace}(M(s)) \operatorname{trace}(M^{-1}(s)), \tag{14}$$

which is equivalent to minimizing (9), subject to (13).

III. LIMITATIONS ON SENSOR MOVEMENTS

A. Dynamics

We assume that the sensors are conveyed by vehicles whose motions are described by

$$\dot{s}(t) = f(s(t), u(t)) \quad \text{a.e. on } T, \quad s(0) = s_0 \tag{15}$$

where a given function $f : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$ is required to be continuously differentiable, $s_0 \in \mathbb{R}^n$ defines an initial sensor configuration, and $u : T \to \mathbb{R}^r$ is a measurable control function which satisfies

$$u_l \leq u(t) \leq u_u \quad \text{a.e. on } T \tag{16}$$

for some constant vectors u_l and u_u.

Given any initial sensor configuration s_0 and any control function, there is a unique absolutely continuous function $s : T \to \mathbb{R}^n$ which satisfies (15) a.e. on T. In what follows, we will call it the state trajectory corresponding to s_0 and u, and make the following notational convention: if s appears without mention in a formula, it is always understood that a control u and initial condition s_0 have been specified and s is the trajectory corresponding to u and s_0 through (15).

B. Pathwise State Constraints

In reality, some restrictions on the motions are inevitably induced. First of all, all sensors should stay within the admissible region Ω_{ad} where measurements are allowed. We assume that it is a compact set defined as follows:

$$\Omega_{\mathrm{ad}} = \big\{ x \in \bar{\Omega} : b_i(x) \leq 0,\, i = 1, \dots, I \big\} \tag{17}$$

where b_i's are given continuously differentiable functions. Accordingly, the conditions

$$\alpha_{ij}(s(t)) = b_i(x^j(t)) \leq 0, \quad \forall\, t \in T \tag{18}$$

must be fulfilled, where $1 \leq i \leq I$ and $1 \leq j \leq N$.

Moreover, we can restrict the admissible distances between the sensors by imposing the constraints

$$\beta_{ij}(s(t)) = R^2 - \|x^i(t) - x^j(t)\|^2 \leq 0, \quad \forall\, t \in T \tag{19}$$

where $1 \leq i < j \leq N$ and R stands for a minimum allowable distance which guarantees that the measurements taken by the sensors can be considered as independent.

To shorten notation, after relabelling, we rewrite constraints (18) and (19) in the form

$$\gamma_\ell(s(t)) \leq 0, \quad \forall\, t \in T \tag{20}$$

where γ_ℓ, $\ell = 1, \dots, IN$ tally with (18), whereas γ_ℓ, $\ell = IN + 1, \dots, [I + (N-1)/2]N$ coincide with (19). In the sequel, $\bar{\nu}$ stands for the set if indices $\big\{1, \dots, \nu\big\}$, $\nu = [I + (N-1)/2]N$.

IV. OPTIMAL CONTROL FORMULATION

The goal in the optimal measurement problem is to determine the forces (controls) applied to each vehicle conveying a sensor, which minimize the design criterion $\Phi[\,\cdot\,]$ defined on the FIMs of the form (7), which are determined unequivocally by the corresponding trajectories, subject to the limitation (13) on the possible loss of D-optimality, constraints (16) on the magnitude of the controls and induced state constraints (20). In order to increase the degree of optimality, in our approach we will regard s_0 as a control parameter vector to be chosen in addition to the control function u. Clearly, the correctness of such a formulation necessitates some additional restrictions on the smoothness of sensitivity coefficients g. In what follows, we thus assume the continuity of g and $\partial g/\partial x$.

The above formulation can be interpreted as the following optimization problem: Find the pair (s_0, u) which minimizes

$$J(s_0, u) = \Phi[M(s)] \tag{21}$$

over the set of feasible pairs

$$\mathcal{P} = \big\{ (s_0, u) \mid u : T \to \mathbb{R}^r \text{ is measurable}, \\ u_l \leq u(t) \leq u_u \ \text{ a.e. on } T,\ s_0 \in \Omega_{\mathrm{ad}}^N \big\}, \tag{22}$$

subject to the constraint (13) and the pathwise state inequality constraints (20).

Evidently, its high non-linearity excludes any possibility of finding closed-form formulae for its solution. Accordingly, we must resort to numerical techniques. A number of possibilities exist in this respect [14], but before exploiting them, observe that in spite of its apparently non-classical form, the resulting optimal-control problem can be easily cast as a classical Mayer problem where the performance index is defined only via terminal values of state variables.

V. EQUIVALENT CANONICAL PROBLEM

The aim of this section is to convert our problem into a canonical optimal control one with an endpoint cost, an endpoint inequality constraint, and inequality-constrained trajectories [14]. Such a transcription will make it possible to employ existing software packages for numerically solving dynamic optimization and optimal control problems.

For notational convenience, define the function $\mathrm{svec} : \mathbb{S}^m \to \mathbb{R}^{m(m+1)/2}$, where $\mathbb{S}^m$ denotes the subspace of all symmetric matrices in $\mathbb{R}^{m\times m}$, that takes the lower triangular part (the elements only on the main diagonal and below) of a symmetric matrix A and stacks them into a vector a:

$$\begin{aligned} a &= \mathrm{svec}(A) \\ &= \mathrm{col}[A_{11}, A_{21}, \dots, A_{m1}, A_{22}, A_{32}, \dots, A_{m2}, \dots, A_{mm}]. \end{aligned} \tag{23}$$

Similarly, let $A = \mathrm{Smat}(a)$ be the symmetric matrix such that $\mathrm{svec}(\mathrm{Smat}(a)) = a$ for any $a \in \mathbb{R}^{m(m+1)/2}$.

Consider the matrix-valued function

$$\Pi(s(t), t) = \sum_{j=1}^{N} g(x^j(t), t) g^{\mathsf{T}}(x^j(t), t). \tag{24}$$

Setting $r : T \to \mathbb{R}^{m(m+1)/2}$ as the solution of the differential equations

$$\dot{r}(t) = \mathrm{svec}(\Pi(s(t), t)), \quad r(0) = 0, \tag{25}$$

we have

$$M(s) = \mathrm{Smat}(r(t_f)), \tag{26}$$

i.e., minimization of $\Phi[M(s)]$ thus reduces to minimization of a function of the terminal value of the solution to (25).

Introducing the augmented state vector

$$q(t) = \begin{bmatrix} s(t) \\ r(t) \end{bmatrix}, \tag{27}$$

we obtain

$$q_0 = q(0) = \begin{bmatrix} s_0 \\ 0 \end{bmatrix}. \tag{28}$$

Then the equivalent canonical optimal control problem consists in finding a pair $(q_0, u) \in \bar{\mathcal{P}}$ which minimizes the performance index

$$\bar{J}(q_0, u) = \phi(q(t_f)) \tag{29}$$

subject to

$$\begin{cases} \dot{q}(t) = \varphi(q(t), u(t), t), \\ q(0) = q_0, \\ \psi(q(t_f)) \le C, \\ \bar{\gamma}_\ell(q(t)) \le 0, \quad \forall\, t \in T, \quad \ell \in \bar{\nu}, \end{cases} \tag{30}$$

where

$$\begin{aligned} \bar{\mathcal{P}} = \big\{ (q_0, u) \mid u : T \to \mathbb{R}^r \text{ is measurable}, \\ u_l \le u(t) \le u_u \text{ a.e. on } T,\ s_0 \in \Omega_{\mathrm{ad}}^N \big\}, \end{aligned} \tag{31}$$

and

$$\varphi(q, u, t) = \begin{bmatrix} f(s(t), u(t)) \\ \mathrm{svec}(\Pi(s(t), t)) \end{bmatrix}, \tag{32}$$

$$\psi(q(t)) = \Psi[\mathrm{Smat}(r(t))], \tag{33}$$

$$\bar{\gamma}_\ell(q(t)) = \gamma_\ell(s(t)). \tag{34}$$

The above problem in canonical form can be solved using one of the existing packages for numerically solving dynamic optimization problems, such as RIOTS_95 [10], DIRCOL [15] or MISER [16]. In our implementation, we employed the first of them, i.e., RIOTS_95, which is designed as a MATLAB toolbox written mostly in C and runs under Windows 98/2000/XP and Linux. It provides an interactive environment for solving a very broad class of optimal control problems. The implemented numerical methods are supported by the theory outlined in [14], which uses the approach of consistent approximations.

VI. ILLUSTRATIVE EXAMPLE

Having developed the method to design sensor motions minimizing Turing's measure of conditioning subject to a guaranteed D-efficiency of the experiment, we go straight to a demonstrative example. Consider the two-dimensional diffusion equation

$$\frac{\partial y}{\partial t} = \nabla \cdot (\mu \nabla y) + F \tag{35}$$

for $x \in \Omega = (0,1)^2$ and $t \in [0,1]$, subject to homogeneous initial and Dirichlet boundary conditions, where $F(x,t) = 20 \exp(-50(x_1 - t)^2)$. The assumed form of the diffusion coefficient is

$$\mu(x) = \theta_1 + \theta_2 x_1 + \theta_3 x_2. \tag{36}$$

We note that the forcing term in (35) imitates the action of a line source whose support is constantly oriented along the x_2-axis and moves with constant speed from the left to the right boundary of Ω. Given three moving sensors whose dynamics is defined by the simple model

$$\dot{s}(t) = u(t), \quad s(t_0) = s_0,$$

along with the constraints

$$|u_i(t)| \le 0.7, \quad \forall\, t \in T, \quad i = 1, \dots, 6$$

imposed on the controls, we are interested in designing their trajectories so as to obtain estimates of θ_1, θ_2 and θ_3 for which the confidence ellipsoid would be as spherical as possible and have a reasonably small volume.

While numerically integrating the system of ODEs (25), the values of the vector of the sensitivities $g = \mathrm{col}[g_1, g_2, g_3]$ along sensor trajectories are indispensable. Assuming the

TABLE I
INFLUENCE OF THE GUARANTEED D-EFFICIENCY.

η	$\Phi(M)$	$\lambda_1(M)$	$\lambda_2(M)$	$\lambda_3(M)$
1.0	117.4386	7.9647	11.5393	529.0066
0.8	51.8741	7.9444	11.6147	220.0239
0.6	41.0240	6.8170	9.0278	139.169
0.4	34.4599	5.0071	6.2348	81.2906
0.2	29.3238	2.6983	3.2887	35.7465
0.0	9.2098	0.2494e–8	0.3114e–8	0.3610e–8

nominal values $\theta_1^0 = 0.1$, $\theta_2^0 = -0.05$ and $\theta_3^0 = 0.2$, they are defined as solutions to the following system of PDEs [2]:

$$\begin{cases} \dfrac{\partial y}{\partial t} = \nabla \cdot (\mu \nabla y) \ + F, \\ \dfrac{\partial g_1}{\partial t} = \nabla \cdot \nabla y \qquad + \nabla \cdot (\mu \nabla g_1), \\ \dfrac{\partial g_2}{\partial t} = \nabla \cdot (x_1 \nabla y) + \nabla \cdot (\mu \nabla g_2), \\ \dfrac{\partial g_3}{\partial t} = \nabla \cdot (x_2 \nabla y) + \nabla \cdot (\mu \nabla g_3), \end{cases} \tag{37}$$

in which the first equation constitutes the original state equation and the second, third and fourth equations result from its differentiation with respect to θ_1, θ_2 and θ_3, respectively. The initial and Dirichlet boundary conditions for all the four equations are homogeneous.

We numerically solved (37) using some routines of the MATLAB PDE toolbox and stored g_1, g_2 and g_3 interpolated at the nodes of a rectangular grid in a four-dimensional array (we applied uniform partitions using 21 grid points per each spatial dimension and 31 points in time), cf. Appendix I in [2] for details. Despite the impossibility of employing the graphical user interface of the toolbox (it is tailored to single PDEs, and not to systems of PDEs), we could still solve (37) using command-line functions. The GUI was applied here only to conveniently define the spatial domain Ω (which is a unit square) and then to export the resulting decomposed geometry matrix to MATLAB's workspace. Since values of g may have been required at points which were not necessarily nodes of that grid, the relevant interpolation was thus performed using cubic splines in space (to this end, MATLAB's procedure `interp2` was used) and linear splines in time. Since, additionally, the derivatives of g with respect to spatial variables and time were going to be required, these derivatives were approximated numerically using the central-difference formula.

The next step consisted in using RIOTS_95 to determine sensor trajectories minimizing Turing's measure of conditioning for the FIM. For that purpose, we retained the partition of the time interval $[0, 1]$ used to solve (37) and considered the components of the control u in the class of cubic splines on the same grid. The system dynamics was integrated using the classical fourth-order Runge-Kutta method.

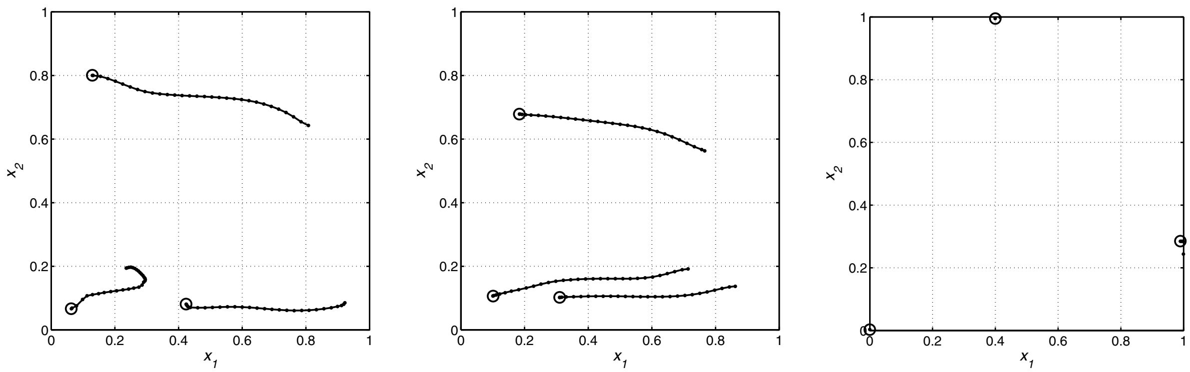
All computations were performed using a low-cost PC (Pentium 4, 2.40 GHz, 512 MB RAM) running Windows 2000 and Matlab 7 (R14). In order to avoid getting stuck in a local minimum, they were repeated several times from different initial solutions. Each run took from three to six minutes. The optimal trajectories obtained for the guaranteed D-efficiency set at the level of $\eta = 0.8$ are shown in Fig. 1(a). The corresponding sensor controls are displayed in Fig. 2. For comparison, Fig. 1(b) shows the D-optimal trajectories, which corresponds to $\eta = 1$, and Fig. 1(c) contains the graphical display of the trajectories obtained when there was no constraint on the guaranteed D-efficiency (i.e., for $\eta = 0$). The latter confirms again that minimization of Turing's measure of conditioning with no reference to other criteria, such as D-optimality, may yield very poor results. We obtained $\det(M) = 2.8 \times 10^{-26}$, which means that the corresponding FIM is practically singular. This is owing to the sensor trajectories reducing to single points located close to the boundary $\partial\Omega$ which does not provide any information about the parameters (we had imposed zero Dirichlet boundary conditions for (35)). The numerical results for different selections of the guaranteed D-efficiences η are displayed in Table 1 (the $\lambda_i(M)$'s denote the eigenvalues of the FIM). As had been expected, the constraints (13) proved active in the optimal solutions. Note that in the D-optimal solution ($\eta = 1$) one eigenvalue is approximately fifty times larger than the others, which means that the expected confidence ellipsoid for the estimates (as assessed by M^{-1}) will be extremely flat. Significant gains in the values of Turing's measure, and thereby improved shapes of the confidence ellipsoid, at the cost of light losses in the degree of D-optimality after application of the proposed approach should then be stressed.

VII. CONCLUSIONS

The paper has dealt, from formulation to numerical evaluation of a special example, with the design of moving sensor trajectories which, on one hand, make the Hessian of the parameter estimation cost well conditioned and, on the other hand, still guarantee a reasonably small volume of the confidence ellipsoid for the estimates. So far, this sensor location problem has been customarily addressed with no attention paid to the form of this ellipsoid, which often resulted in its harmfully elongated form. Of significant importance is the fact that the proposed formulation can be transcribed into an equivalent optimal control problem in Mayer form, and that it can be then efficiently solved by the MATLAB toolbox RIOTS_95, a high-performance tool for solving optimal control problems, combined with the Partial Differential Equation Toolbox.

REFERENCES

[1] C. S. Kubrusly and H. Malebranche, "Sensors and controllers location in distributed systems — A survey," *Automatica*, vol. 21, no. 2, pp. 117–128, 1985.
[2] D. Uciński, *Optimal Measurement Methods for Distributed-Parameter System Identification*. Boca Raton, FL: CRC Press, 2005.



(a) (b) (c)

Fig. 1. Optimal sensor trajectories: Turing's measure of conditioning with the guaranteed D-efficiency set as 0.8 (a), genuine D-optimality criterion (b), and Turing's measure of conditioning without limitations on the D-efficiency. The initial sensor positions are marked with open circles, and the sensors positions at the consecutive points of the time grid are denoted by discs.

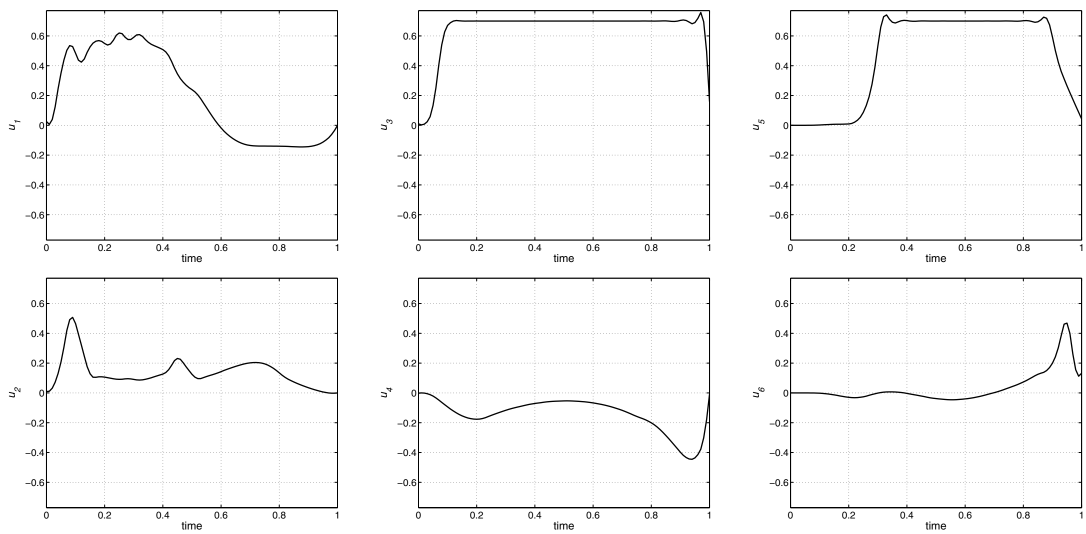


Fig. 2. Controls minimizing Turing's measure of conditioning with the guaranteed D-efficiency set as 0.7.

[3] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[4] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4–5, pp. 436–463, 2005.

[5] K. L. Moore, Y. Chen, and Z. Song, "Diffusion-based path planning in mobile actuator-sensor networks (MAS-net): Some preliminary results," in *Proc. SPIE Defense and Security Symposium on Intelligent Computing: Theory and Applications II (OR53)*, Orlando, FL, USA, Apr. 12–16, 2004, published on CD-ROM.

[6] Y. Chen, K. L. Moore, and Z. Song, "Diffusion boundary determination and zone control via mobile actuator-sensor networks (MAS-net) – Challenges and opportunities," in *Proc. SPIE Defense and Security Symposium on Intelligent Computing: Theory and Applications II (OR53)*, Orlando, FL, USA, Apr. 12–16, 2004, published on CD-ROM.

[7] E. Rafajłowicz, "Optimum choice of moving sensor trajectories for distributed parameter system identification," *International Journal of Control*, vol. 43, no. 5, pp. 1441–1451, 1986.

[8] D. Uciński, "Optimal sensor location for parameter estimation of distributed processes," *International Journal of Control*, vol. 73, no. 13, pp. 1235–1248, 2000.

[9] D. Uciński and Y. Chen, "Time-optimal path planning of moving sensors for parameter estimation of distributed systems," in *Proc. 44th IEEE Conference on Decision and Control, and the European Control Conference 2005,* Seville, Spain, 2005, published on CD-ROM.

[10] A. L. Schwartz, E. Polak, and Y. Chen, *A Matlab Toolbox for Solving Optimal Control Problems. Version 1.0 for Windows*, May 1997. [Online]. Available: http://www.schwartz-home.com/˜adam/RIOTS/

[11] É. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*, ser. Communications and Control Engineering. Berlin: Springer-Verlag, 1997.

[12] V. V. Fedorov and P. Hackl, *Model-Oriented Design of Experiments*, ser. Lecture Notes in Statistics. New York: Springer-Verlag, 1997.

[13] É. Walter and L. Pronzato, "Qualitative and quantitative experiment design for phenomenological models — A survey," *Automatica*, vol. 26, no. 2, pp. 195–213, 1990.

[14] E. Polak, *Optimization. Algorithms and Consistent Approximations*, ser. Applied Mathematical Sciences. New York: Springer-Verlag, 1997.

[15] O. von Stryk, *User's Guide for DIRCOL, a Direct Collocation Method for the Numerical Solution of Optimal Control Problems. Version 2.1*, Fachgebiet Simulation und Systemoptimierung, Technische Universität Darmstadt, Nov. 1999. [Online]. Available: http://www.sim.informatik.tu-darmstadt.de/index/leftnav.html.en

[16] L. S. Jennings, M. E. Fisher, K. L. Teo, and C. J. Goh, *MISER 3: Optimal Control Software, Version 2.0. Theory and User Manual*, Department of Mathematics, University of Western Australia, Nedlands, 2002. [Online]. Available: http://www.cado.uwa.edu.au/miser/