

---

# Optimal Mobile Sensor Motion Planning Under Nonholonomic Constraints for Parameter Estimation of Distributed Systems

---

Zhen Song<sup>†</sup>, YangQuan Chen<sup>†</sup>, JinSong  
Liang<sup>†</sup> and Dariusz Uciński<sup>‡</sup>

<sup>†</sup>CSOIS, Department of Electrical and Computer Engineering  
Utah State University, Logan, Utah 84322, USA  
{zhensong, yqchen, jsliang}@ieee.org

<sup>‡</sup> Institute of Control and Computation Engineering  
University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra,  
POLAND  
D.Ucinski@issi.uz.zgora.pl.

**Abstract:** This paper presents a numerical solution for a mobile sensor motion trajectory scheduling problem under nonholonomic constraints of a project named MAS-net, which stands for Mobile Actuator-Sensor network. The motivation of the MAS-net project, at the first stage, is to estimate diffusion system parameters by networked mobile sensors. Each sensor is mounted on a differentially-driven mobile robot to observe the diffusing fog. In other words, this project requires to observe a parabolic distributed parameter systems (DPS) by nonholonomic networked mobile sensors. This paper reformulates this problem in the framework of optimal control and proposes a procedure to obtain a numerical solution by using RIOTS (Schwartz et al., 1997) and Matlab PDE Toolbox. The objective function of this method is designed to minimize the effect of the sensing noise. Extensive simulation results are presented for illustration.

**Keywords:** Distributed parameter system, sensor trajectory, motion planning, RIOTS, optimal control, MAS-net, sensor networks, networked mobile robots.

**Reference** This paper should be cited as follows: Song, Z., Chen, Y.Q., Liang, J.S., and Uciński, D., 'Optimal Mobile Sensor Motion Planning Under Nonholonomic Constraints for Parameter Estimation of Distributed Systems', Int. J. Intelligent System Tech. and Applications, Vol. X, No. Y, pp 1234-5678.

**Biographical notes:** Zhen Song is currently a PhD candidate in the Department of Electronics and Computer Engineering at Utah State University in USA. He received his Master's degree from the same department and the same university in 2003. His research areas include distributed parameter systems, sensor networks, localization and navigation, intelligent control systems, sensing and perception for mobile robots.

YangQuan Chen is presently an assistant professor of Electrical and Computer Engineering Department and the Acting Director for CSOIS



(Center for Self-Organizing and Intelligent Systems) at Utah State Univ. He obtained his Ph.D. from Nanyang Tech. Univ. (NTU), Singapore in 1998. Dr Chen has 12 US patents granted and 2 US patent applications published. He published over 200 academic papers, 2 textbooks, 2 research monographs, and (co)authored over 50 industrial reports. Dr. Chen has been an Associate Editor in the Conference Editorial Board of IEEE Control Systems Society since 2002. He is a founding member of the ASME subcommittee of “Fractional Dynamics” in 2003. He is a senior member of IEEE, a member of ASME and a member of International Society for Information Fusion.

JinSong Liang is a senior engineer of Maxtor Corporation for servo product development. He received his Ph.D. from the Mechanical and Aerospace Engineering Dept. of Utah State University in 2005 and he also received a M.Sc. degree at the Electrical and Computer Engineering Department at Utah State University in 2005.

Dariusz Uciński is currently an associate professor at the University of Zielona Góra, Poland. He received Ph.D. (1992) and D.Sc. (2000) degrees in automatic control and robotics from the Technical University of Wrocław, Poland. He is the author of the research monograph *Optimal Measurement Methods for Distributed Parameter System Identification*, and a co-author of two textbooks. For sixteen years his major activities have been concentrated on measurement optimization and sensor network design for parameter estimation in distributed systems. Other areas of expertise include optimum experimental design, algorithmic optimal control, robotics and cellular automata. Since 1992 he has been the scientific secretary of the editorial board of the International Journal of Applied Mathematics and Computer Science. He is the Secretary of the IEEE Technical Committee on Distributed Parameter Systems and a member of SIAM.

---

## 1 Introduction

A wide class of processes are those whose behavior is described by partial differential equations (PDEs) owing to the inherent spatial and temporal variabilities of their states. They are commonly termed the distributed parameter systems (DPSs), which occupy now an important place in control and systems theories (Curtain and Zwart, 1995; Lasiecka and Triggiani, 2000; Zwart and Bontsema, 1997; Omatu and Seinfeld, 1989; Neittaanmäki and Tiba, 1994; Christofides, 2001). One of the basic and most important questions in DPSs is parameter estimation, which refers to the determination from observed data of unknown parameters in the system model such that the predicted response of the model is close, in some well-defined sense, to the process observations. For that purpose, the system’s behavior or response is observed with the aid of some suitable collection of discrete sensors which reside at prespecified spatial locations. However, the resulting measurements are incomplete in the sense that the entire spatial state profile is not available. Moreover, the measurements are inexact by virtue of inherent errors of measurement associated with transducing elements and also because of the measurement environment. These factors lead to the question of where to locate sensors so that the information content of the resulting outputs with respect to the distributed state and PDE model be as high as possible.



Both researchers and practitioners do not doubt that making use of sensors placed in an ‘intelligent’ manner may lead to dramatic gains in the achievable accuracy of the resulting parameter estimates, so efficient sensor location strategies are highly desirable. In turn, the complexity of the sensor location problem implies that there are very few sensor placement methods which are readily applicable to practical situations. What is more, they are not well known among researchers. This generates keen interest in the potential results, as the motivations to study the sensor location problem stem from practical engineering issues. Optimization of air quality monitoring networks is among the most interesting ones. One of the tasks of environmental protection systems is to provide expected levels of pollutant concentrations. But to produce such a forecast, a smog prediction model is necessary which is usually chosen in the form of an advection-diffusion partial-differential equation. Its calibration requires parameter estimation, e.g. the unknown spatially-varying turbulent diffusivity tensor should be identified based on the measurements from monitoring stations. Since measurement transducers are usually rather costly and their number is limited, we are faced with the problem of how to optimize their locations in order to obtain the most precise model. Other stimulating applications include, among other things, groundwater modelling, recovery of valuable minerals and hydrocarbon from underground permeable reservoirs, gathering measurement data for calibration of mathematical models used in meteorology and oceanography, automated inspection in static and active hazardous environments where trial-and-error sensor planning cannot be used (e.g. in nuclear power plants), or emerging smart material systems.

The sensor placement problem was attacked from various angles, but the results communicated by most authors are limited to the selection of stationary sensor positions (for reviews, see (Kubrusly and Malebranche, 1985; Uciński, 2005*a*, 2000*aa*, 2000*b,a*; Uciński and Korbicz, 1999; Uciński, 2001), apart from generalizations of Rafajłowicz’s results, develops some computational algorithms based on the FIM. The problem is then reduced to a state-constrained optimal-control one for which solutions are obtained via gradient techniques capable of handling various constraints imposed on sensor motions. In turn, the work (Uciński and Chen, 2005) 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. Note that the idea of moving observations has also been applied in the context of state estimation (Khapalov, 1992; Nakano and Sagara, 1981, 1988; Carotenuto et al., 1987), but those results can hardly be exploited in the framework considered here as those authors make extensive use of some specific features of the addressed problem (e.g., the linear dependence of the current state on the initial state for linear systems).

It should be emphasized that 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 (Ögren et al., 2004; Chong and Kumar, 2003; Sinopoli et al., 2003; Cassandras and Li, 2005; Martínez and Bullo, 2006). 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 caught 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 equations (Moore et al., 2004*b*; Chen et al., 2004*b*).

This project is proposed to combine the latest sensor network technologies with mobile robotics for an application-oriented high-level task: characterization, estimation and control of an undesired diffusion process by networked movable or mobile actuators and sensors. One potential solution is to estimate the parameters in a “closed-loop” or ‘on-line”, or a “recursive” approach, as mentioned in the last chapter of (Patan, 2004). This idea can be explained as follows. With arbitrary initial values of the unknown parameters, the system starts to drive sensors in an “optimal” trajectory with respect to those parameters. Sensor data are then collected while the sensors are moving. Based on the collected data, parameter estimates are improved and the moving sensor trajectories are then updated accordingly. Then, the sensors are driven to follow the newly updated trajectories based on the parameters estimated. Through this “closed-loop” iteration or the recursive on-line adaptation, the estimated parameters converge to the true values of the DPS. This so-called “online” mode was listed as one of the important future research efforts.

From the control system perspective, the trajectory scheduling procedure can be called “control for sensing,” and the parameter updating procedure is “sensing for control.” When these two parts are connected with an “online” or “recursive” strategy, the whole system is a closed-loop controlled system. Control theory can then be applied to improve the performances. Currently, it is still an open problem of how to “close” the loop of this system.

In this paper, in the vein of (Ucinski, 2005*b*; Patan, 2004), we focus on the “control for sensing” part, that is, given an estimate of the DPS parameters, how to drive the mobile sensors optimally in the sense that the effect of the sensor noise can be minimized. We present a numerical solution for a mobile sensor motion trajectory scheduling problem under nonholonomic constraints as in MASmotes (Wang et al., 2004), the two wheeled differentially-driven mobile robots, in our MAS-net project (Moore and Chen, 2004; Moore et al., 2004*a*; Chen et al., 2004*a*; Wang et al., 2004; Chen, 2005; Arora, 2005).

The rest of this paper is organized as follows. The formulation of the MAS-net estimation problem is described in Sec. 2, in which the dynamic model for differentially-driven mobile robots is presented in Sec. 2.1 and the objective function for the optimal sensor motion scheduling is described in Sec. 2.3. Section 2.4 reformulates the problem in the framework of optimal control. In Sec. 3, a numerical solution procedure for this problem is presented. A Matlab optimal control toolbox RIOTS is briefly described in Sec. 3.1 and Sec. 3.2 describes a method to incorporate the Matlab PDE Toolbox and the RIOTS, cf. Sec. 3.1. Some illustra-



$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 2b & 0 & 0 \\ 0 & 2b & 0 \\ 0 & 0 & bl^2/2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} r \cos(\alpha) & r \cos(\alpha) \\ r \sin(\alpha) & r \sin(\alpha) \\ -rl/2 & rl/2 \end{bmatrix} \begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix} \quad (1)$$

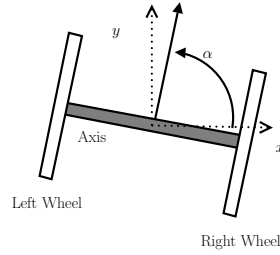
tive simulation results are presented in Sec. 4 with remarks on the obtained results. Section 5 concludes this paper. For readers' convenience, Appendix 5 lists all the notations used in this paper.

## 2 Problem Formulation of the Sensor-Motion Scheduling for Diffusion Systems

In this section, the model of our diffusion system and the model of our differentially-driven robots are presented in Sec. 2.1 and Sec. 2.2, respectively. After introducing a class of objective functions in Sec. 2.3, the MAS-net estimation problem is reformulated in the framework of optimal control, and ready to be solved by RIOTS.

### 2.1 The Dynamic Model of Differentially-Driven Robots

MASmote (Wang et al., 2004) is a differentially-driven ground mobile robot as illustrated in Fig. 1. Its dynamic model can be described by (1), where the symbols are defined and listed in the Appendix.



**Figure 1** A differentially-driven mobile robot

In (1), the mobile robot is represented in a form of a second order system. For convenience, the corresponding state space form can be easily derived by introducing  $\mathbf{x}$ , the extended system state vector defined as  $\mathbf{x} := [x \ y \ \alpha \ \dot{x} \ \dot{y} \ \dot{\alpha}]^T$ , and  $\tau$  is defined as  $\tau = [\tau_l, \tau_r]^T$ .

To have a compact notation, let us define matrices  $A_1$  and  $B_1$  as

$$A_1 := \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2b/m & 0 & 0 \\ 0 & 0 & 0 & 0 & -2b/m & 0 \\ 0 & 0 & 0 & 0 & 0 & -bl^2/(2I) \end{bmatrix},$$

and

$$B_1 := \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ r \cos(\alpha)/m & r \cos(\alpha)/m \\ r \sin(\alpha)/m & r \sin(\alpha)/m \\ -rl/(2I) & rl/(2I) \end{bmatrix}.$$

Thus, the robot dynamics can be written as

$$\dot{\mathbf{x}} = A_1 \mathbf{x} + B_1 \tau. \quad (2)$$

Note that  $B_1$  depends on  $\mathbf{x}$ .

To solve the multi-robot-motion-scheduling problems in Sec. 4, we need to write the dynamics of three robots as a single dynamic system. Denote the states of each robot in (2) as  $\mathbf{x}^{(1)}$ ,  $\mathbf{x}^{(2)}$ , and  $\mathbf{x}^{(3)}$ , respectively. After defining

$$\mathbf{x}_3 := \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \mathbf{x}^{(3)} \end{bmatrix}, \quad A_3 = \begin{bmatrix} A_1^{(1)} & 0 & 0 \\ 0 & A_1^{(2)} & 0 \\ 0 & 0 & A_1^{(3)} \end{bmatrix},$$

$$B_3 = \begin{bmatrix} B_1^{(1)} & 0 & 0 \\ 0 & B_1^{(2)} & 0 \\ 0 & 0 & B_1^{(3)} \end{bmatrix}, \quad \text{and } \tau_3 = \begin{bmatrix} \tau^{(1)} \\ \tau^{(2)} \\ \tau^{(3)} \end{bmatrix},$$

where  $A_1^{(j)}, B_1^{(j)}$  are for the  $j$ -th robot, the dynamics of all three robots can be written compactly as follows:

$$\dot{\mathbf{x}}_3 = A_3 \mathbf{x}_3 + B_3 \tau_3. \quad (3)$$

## 2.2 The Model of the Diffusion Process

For comparison purposes, here we use the same diffusion system model as in Example 4.1 in (Ucinski, 2005b). We rewrite it using our notation in the following form:

$$\begin{aligned} \frac{\partial u(x, y, t)}{\partial t} &= \frac{\partial}{\partial x} \left( \kappa(x, y) \frac{\partial u(x, y, t)}{\partial x} \right) \\ &+ \frac{\partial}{\partial y} \left( \kappa(x, y) \frac{\partial u(x, y, t)}{\partial y} \right) \\ &+ 20 \exp(-50(x-t)^2), \\ &(x, y) \in \Omega = (0, 1) \times (0, 1), t \in T, \\ u(x, y, 0) &= 0, (x, y) \in \Omega, \\ u(x, y, t) &= 0, (x, y, t) \in \partial\Omega \times T, \\ T &:= \{t | t \in (0, 1)\}, \\ \kappa(x, y) &= c_1 + c_2 x + c_3 y, \\ &c_1 = 0.1, c_2 = -0.05, c_3 = 0.2, \end{aligned}$$

where  $u(x, y, t)$  is the concentration,  $(x, y)$  is the spatial coordinate,  $c_1, c_2, c_3$  are the nominal parameters, and  $t$  is the time.

### 2.3 The Objective Function for Sensor-Motion Scheduling

In this paper, the aim of the optimization is to minimize the sensor noise effect. For the  $i$ -th mobile sensor, its observation is assumed as follows:

$$z^{(i)}(t) = u(\mathbf{x}_{\ell oc}^{(i)}(t), t) + \epsilon(\mathbf{x}_{\ell oc}^{(i)}(t), t), \quad (4)$$

where  $\mathbf{x}_{\ell oc}^{(i)}$  signifies the two-element vector formed of the first two components of  $\mathbf{x}^{(i)}$  and  $\epsilon$  is the white noise with statistics

$$E\{\epsilon(x, y, t)\} = 0,$$

$$E\{\epsilon(x, y, t)\epsilon(x', y', t')\} = \sigma^2 \delta(x - x') \delta(y - y') \delta(t - t').$$

The positions are in the domain of the diffusion process, i.e.  $(x, y) \in \Omega$  and  $(x', y') \in \Omega$ . Note that here the prime does not mean a derivative or a transpose. The  $\delta$  is Dirac's delta function, and  $\sigma$  is a positive constant.

The objective function is chosen to be the so-called D-optimum design criterion defined on the Fisher Information Matrix (FIM) (Ucinski, 2005b). Up to a constant multiplier, the FIM constitutes the inverse of the covariance matrix for the least-squares estimator defined as the minimizer of the following "fit-to-data" criterion (Ucinski, 2005b):

$$J_1(c) = \frac{1}{2} \int_T \|z(t) - \hat{u}(\mathbf{x}_{\ell oc}, t; c)\|^2 dt. \quad (5)$$

The notation  $\hat{\cdot}$  in (5) indicates the predicted value. For  $N$  robots,  $J_1(c)$  becomes

$$J_N(c) = \sum_{j=1}^N \frac{1}{2} \int_T \|z^{(j)}(t) - \hat{u}^{(j)}(\mathbf{x}_{\ell oc}, t; c)\|^2 dt. \quad (6)$$

Then, the FIM of  $N$  robots is defined as follows:

$$M = \sum_{j=1}^N \int_0^{t_f} \left( \frac{\partial u(\mathbf{x}_{\ell oc}^{(j)}(t), t)}{\partial c} \right)^T \left( \frac{\partial u(\mathbf{x}_{\ell oc}^{(j)}(t), t)}{\partial c} \right) dt. \quad (7)$$

Note that  $\mathbf{x}^{(j)}$  is the state vector of the  $j$ -th robot. The readers should not confuse  $\mathbf{x}$  with the spatial variable  $x$  which is a scalar. Here  $c$  is the parameter vector in the DPS to be identified, and the partial derivatives are evaluated at  $c = c^0$ , a preliminary estimate of  $c$ .

Note that the FIM  $M$  is a matrix. Thus, there are many metrics that can be defined on it. The D-optimality criterion used in this paper is defined as

$$\Psi(M) = -\ln \det(M). \quad (8)$$

Other optimization criteria are described and compared in (Ucinski, 2005b).

The objective function for the MAS-net estimation problem is to minimize  $J_2(\mathbf{x}) = \Psi(M)$ . Our goal here is to find the optimal control function  $\tau \in L_\infty^{2N}[t_0, t_f]$  for  $N$  two wheel differentially-driven mobile sensors together with the initial states  $\mathbf{x}(t_0) = \xi \in \mathbb{R}^K$  where  $K = 6N$  and  $t \in [t_0, t_f] = [0, 1]$ , such that  $\Psi(M)$  is minimized.

#### 2.4 Problem Reformulation in the Optimal Control Framework

According to the general optimal control problem formulation in RIOTS (Schwartz et al., 1997), our optimal mobile sensor motion scheduling problem can be formulated as follows:

$$\min_{(\tau, \xi) \in L_{\infty}^{2N}[t_0, t_f] \times \mathbb{R}^K} J(\tau, \xi) \quad (9)$$

where

$$J(\tau, \xi) = g_0(\xi, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} l_o(t, \mathbf{x}, \tau) dt$$

subject to the following conditions and constraints:

$$\begin{aligned} \dot{\mathbf{x}} &= h(t, \mathbf{x}, \tau), \\ \mathbf{x}(t_0) &= \xi, \quad t \in [t_0, t_f], \\ \tau_{\min}^{(j)}(t) &\leq \tau^{(j)}(t) \leq \tau_{\max}^{(j)}(t), \quad j = 1, \dots, N, t \in [t_0, t_f], \\ \xi_{\min}^{(j)}(t) &\leq \xi^{(j)}(t) \leq \xi_{\max}^{(j)}(t), \quad j = 1, \dots, K, t \in [t_0, t_f], \\ l_{ti}(t, \mathbf{x}(t), \tau(t)) &\leq 0, \quad t \in [t_0, t_f], \\ g_{ei}(\xi, \mathbf{x}(t_f)) &\leq 0, \quad g_{ee}(\xi, \mathbf{x}(t_f)) = 0. \end{aligned}$$

For our optimal motion scheduling problem,  $\dot{\mathbf{x}} = h(t, \mathbf{x}, \tau) = A_1 \mathbf{x} + B_1 \tau$  for the single robot case and for three robot cases  $\dot{\mathbf{x}}_3 = h(t, \mathbf{x}_3, \tau_3) = A_3 \mathbf{x}_3 + B_3 \tau_3$ . Here, we define  $l_0(\xi, \mathbf{x}(t_f)) = 0$  and  $g_0(\xi, \mathbf{x}(t_f)) = \Psi(M)$  to simplify the numerical computation. This technique is called solving an “equivalent Mayer problem.” To understand the equivalent Mayer problem, let us start from the definition of some new notation.  $g(x, y, t)$  is called the sensitivity function, where

$$g(x, y, t) := \left( \frac{\partial u(x, y, t)}{\partial c} \right)^T.$$

Then, the FIM in (7) is

$$M = \sum_{j=1}^N \int_{t_0}^{t_f} g(\mathbf{x}_{\ell oc}^{(j)}(t), t) g^T(\mathbf{x}_{\ell oc}^{(j)}(t), t) dt. \quad (10)$$

Define the Mayer states as

$$\chi_{(i,j)}(t) := \int_{t_0}^t \varpi_{(i,j)}(\tau) d\tau. \quad (11)$$

where

$$\varpi_{(i,j)}(t) := \sum_{l=1}^N g_{(i)}(\mathbf{x}_{\ell oc}^{(l)}(t), t) g_{(j)}(\mathbf{x}_{\ell oc}^{(l)}(t), t). \quad (12)$$

Denote by  $\chi_{\text{dl}}$  the stack vector which stacks all the entries on the diagonal and below the diagonal of  $\chi$  to a vector. Then, the extended Mayer state vector  $\tilde{\mathbf{x}}$  can be expressed as

$$\tilde{\mathbf{x}} := \begin{bmatrix} \mathbf{x} \\ \chi_{\text{dl}} \end{bmatrix}.$$

Comparing (11) and (10), one can easily observe the key point of this equivalent Mayer problem. That is,  $\chi(t_f) = M$  and  $\chi_{\text{dl}}$  contains all the information of  $M$  since  $M$  is symmetric. After replacing the extended state vector  $\mathbf{x}$  with the extended Mayer vector  $\tilde{\mathbf{x}}$ , we can get  $M$  without explicit integration.

Thus, when considering the equivalent Mayer problem, the models used for RIOTS are as follows:

$$(13) \quad \dot{\tilde{\mathbf{x}}} = \begin{bmatrix} A_1 \mathbf{x} + B_1 \tau \\ \varpi_{\text{dl}} \end{bmatrix},$$

$$(14) \quad \dot{\tilde{\mathbf{x}}}_3 = \begin{bmatrix} A_3 \mathbf{x} + B_3 \tau \\ \varpi_{\text{dl}} \end{bmatrix}.$$

### 3 Finding A Numerical Solution of the Optimal Mobile Sensor Motion Scheduling Problem

#### 3.1 A Brief Introduction to RIOTS

RIOTS stands for “recursive integration optimal trajectory solver.” It is a Matlab toolbox designed to solve a very broad class of optimal control problems as defined in (9). When executing under Matlab script mode, the following configuration files need to be provided: `sys_l.m`, `sys_h.m`, `sys_g.m`, `sys_init.m`, `sys_acti.m`. They are the  $l_o$ ,  $h$ ,  $g_o$  functions in (9) and two initial conditions, respectively. Detailed instructions on how to prepare these files and many sample problems can be found in (Schwartz et al., 1997; ?). The most important function in this optimal control toolbox is `riots` explained in detail on page 73 of (Schwartz et al., 1997).

```
[u,x,f,g,lambda2] = riots([x0,{fixed},{x0min,x0max}]),u0,t,
Umin,Umax,params,[miter,{var},{fd}],ialg,
{[eps,epsneq,objrep,bigbnd]}, {scaling},{disp},{lambda1}).
```

The parameters useful for understanding our numerical experiments here are as the follows:

- `x0`: initial values of  $\tilde{\mathbf{x}}$ .
- `fixed`: a vector to specify which entries in `x0` are fixed and which entries are not. Later in Sec. 4, results for two configurations are presented by changing `fixed` which are cases of “fixed initial states” and “unfixed initial states”, respectively. For the first case, the robots’ initial conditions,  $\mathbf{x}$ , are fixed. For the second case,  $\chi_{\text{dl}}$  is fixed so that the robots start from the optimal starting positions.
- `x0min`, `x0man`: bounds of the initial conditions.
- `u0`: initial values of the control functions  $\tau$ .



- $t$ : time.
- $U_{\min}, U_{\max}$ : bounds for  $\tau$ .

The definitions of other parameters are described in (Schwartz et al., 1997).

### 3.2 Using Matlab PDE Toolbox Together with RIOTS

The sensitivity function is generated before the function call of `riots` by Matlab PDE Toolbox. The procedure of solving the sensitivity function amounts to finding the solutions of the followings equations:

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot (\kappa \nabla u) + 20 \exp(-50(x_1 - t)^2), \\ \frac{\partial g_{(1)}}{\partial t} = \nabla \cdot \nabla u + \nabla \cdot (\kappa \nabla g_{(1)}), \\ \frac{\partial g_{(2)}}{\partial t} = \nabla \cdot (x \nabla u) + \nabla \cdot (\kappa \nabla g_{(2)}), \\ \frac{\partial g_{(3)}}{\partial t} = \nabla \cdot (y \nabla u) + \nabla \cdot (\kappa \nabla g_{(3)}) \end{cases}$$

where  $\nabla = (\partial/\partial x, \partial/\partial y)$ . Note that there are three  $g$  functions since there are 3 parameters  $c_1, c_2, c_3$  in Sec. 2.2.

## 4 Illustrative Simulations

### 4.1 Differential Drive vs. Omni-Directional Drive

In (Ucinski, 2005b), the robot model is a simple kinematic model:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix}, \quad \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad (15)$$

where  $u_x$  and  $u_y$  are horizontal and vertical control components, respectively. This form is equivalent to

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = r\omega(t), \quad \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad (16)$$

where  $\omega(t)$  is the angular speed vector, and  $r$  is the radii of the wheels. In this paper, we refer a robot that subjects to the kinematic in (16) a proximal ‘‘omni-directionally-driven robot’’ since the velocity can be set arbitrarily. When the robot is differentially driven, we are interested to see the difference in the optimal sensor motion scheduling. The following five cases are compared first:

- case 1: Omni-directionally-driven robots starting from a fixed given initial state vector.



- case 2: Differentially-driven robots with a fixed given initial state vector. Moreover, we consider two subcases: subcase (2a) has an initial yaw angle of  $15^\circ$  and subcase (2b) of  $-15^\circ$ .
- case 3: Omni-directionally-driven robots without a fixed given initial state vector. We assume that the optimal static sensor location problem is solved first. Use this obtained optimal position as the initial states and seek the optimal sensor motion trajectories.
- case 4: The same as in case 3 but using differentially-driven mobile robots.
- case 5: Differentially-driven robots start from the left boundary. The initial positions and yaw angles are optimized to minimize the cost function, as far as the positions are still on the boundary.

According to the above definitions, Fig. 2 and Fig. 3 show the optimized trajectories for case 1 and the associated cost function  $J$ ; Fig. 4 and Fig. 5 for case 2(a); Fig. 6 and Fig. 7 for case 2(b); Figs. 8, 9 for case 3; Figs. 10,11 for case 4; and Figs. 12,13 for case 5. From these figures, we have the following observations:

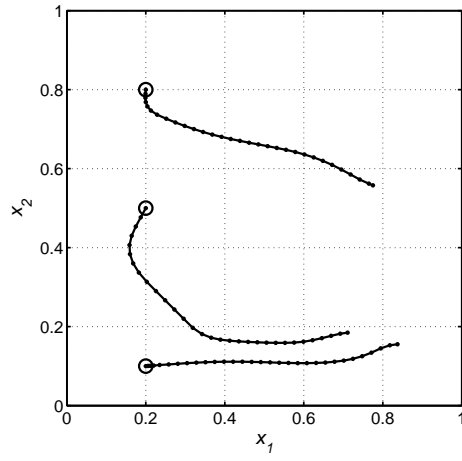
- Differentially-driven robots are less likely to change the orientation. The optimal mobile sensor trajectories in cases 2 and 4 have smaller curvatures compared with that in cases 1 and 3.
- No matter what robot dynamics is, the robots tend to move along the same trend. This can be observed by comparing cases 1, 2(a), 2(b) and cases 3, 4.
- For multi-robot cases, the final positions of the robots tend to be evenly distributed. Comparison on Fig. 4 and Fig. 6 is especially interesting. Due to the difference of the initial orientation, the final positions of the robots are significantly different. However, the trend is robust to the initial orientation.
- For cases with different configurations, the range of the cost function,  $J$ , is about the same.

#### 4.2 Comparison of Robots with Different Capabilities

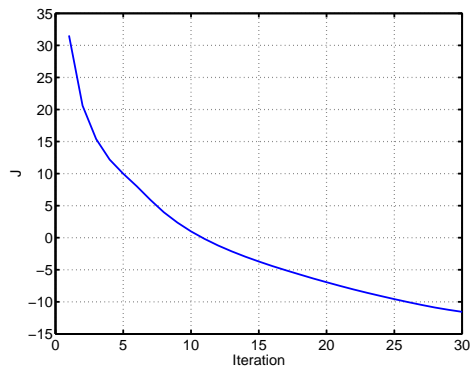
From the robot design prospect, it is important to compare the robots with different configurations, such as different motor power. Here we consider two more cases.

- case 6: using a single “weak” robot, whose weight is 0.5 and the range of its torque for each wheel is  $\pm 10$ . See Fig. 14.
- case 7: using a single “strong” robot, whose weight is 0.05 and the range of its torque for each wheel is  $\pm 100$ . See Fig. 15.

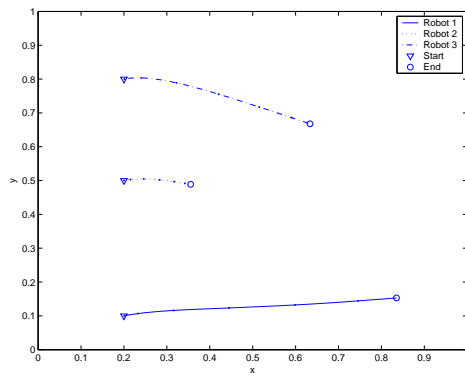
With the same fixed initial states, and the same  $T$ , the robot in case 6 moves shorter than in case 7 as seen from Fig. 14 and Fig. 15. This matches our intuition that it is desirable for the sensors to measure the DPS states at more spatial locations whenever possible. In other words, it is better to increase the power of the robots.



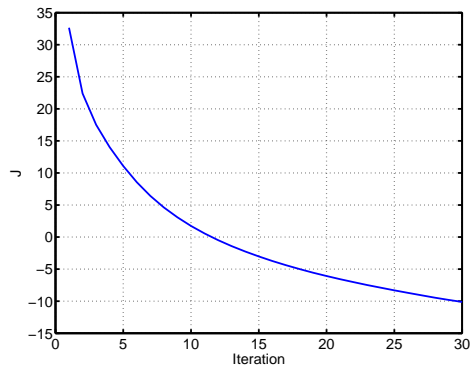
**Figure 2** The optimal sensor trajectories of omni-directionally-driven robots (case 1).



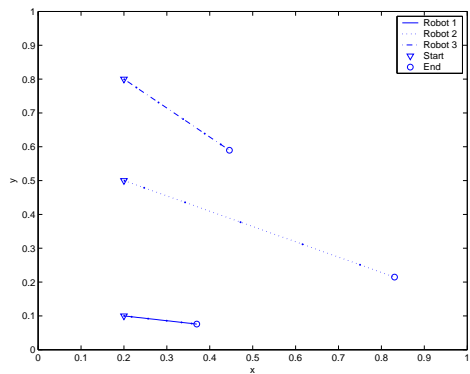
**Figure 3** The cost function of omni-directionally-driven robots (case 1).



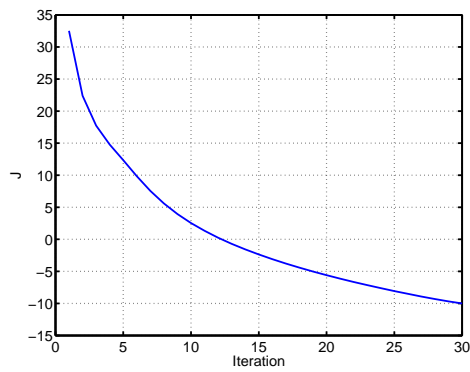
**Figure 4** The optimal sensor trajectories of differentially-driven robots:  $15^\circ$  initial yaw angle (case 2a)



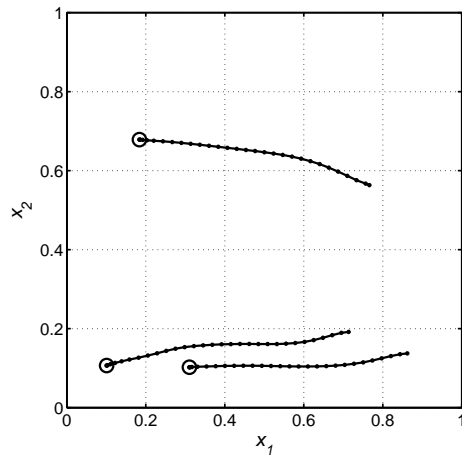
**Figure 5** The cost function of differentially-driven robots:  $15^\circ$  initial yaw angle (case 2a).



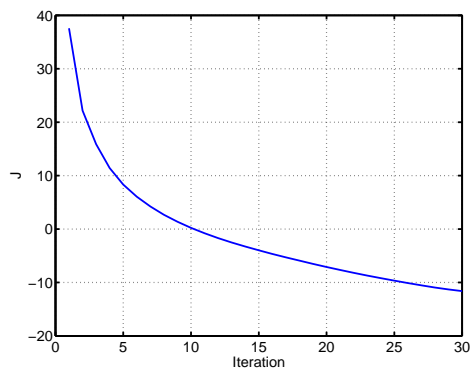
**Figure 6** The optimal sensor trajectories of differentially-driven robots:  $-15^\circ$  initial yaw angle (case 2b)



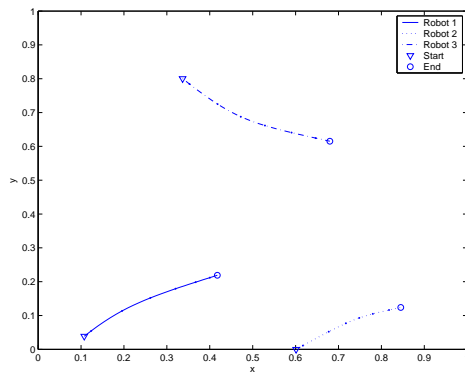
**Figure 7** The cost function of differentially-driven robots:  $-15^\circ$  initial yaw angle (case 2b).



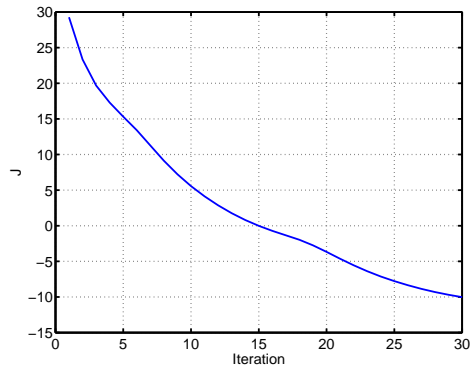
**Figure 8** The optimal sensor trajectories of omni-directionally-driven robots using optimal initial conditions (case 3).



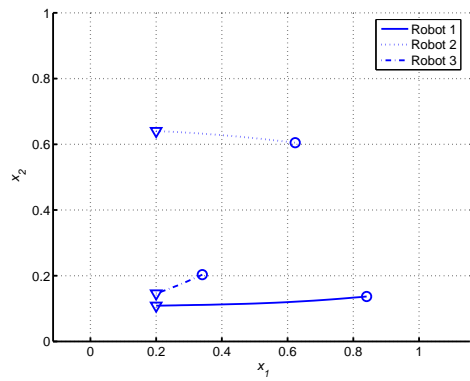
**Figure 9** The cost function of omni-directionally-driven robots using optimal initial conditions (case 3).



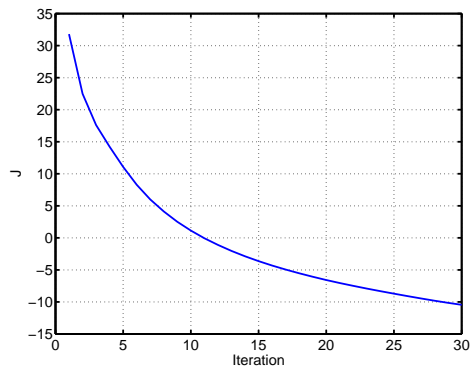
**Figure 10** The optimal sensor trajectories of differentially-driven robots using optimal initial conditions (case 4).



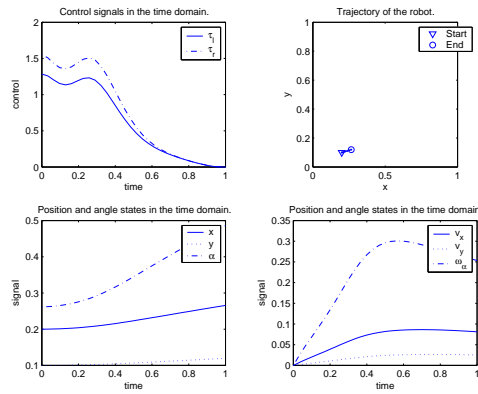
**Figure 11** The cost function of differentially-driven robots using optimal initial conditions (case 4).



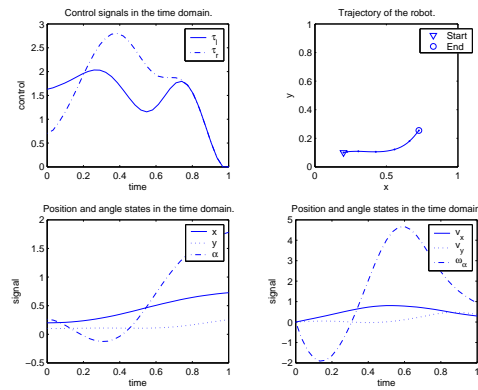
**Figure 12** The optimal sensor trajectories of differentially-driven robots starting from the boundary (case 5).



**Figure 13** The cost function of differentially-driven robots starting from the boundary (case 5).



**Figure 14** The optimal trajectory of “weak” differentially-driven robots: initial yaw angle is  $15^\circ$  (case 6).



**Figure 15** The optimal trajectory of “strong” differentially-driven robots: initial yaw angle is  $15^\circ$  (case 7).

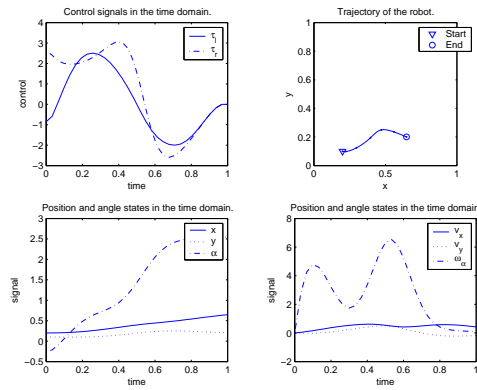
### 4.3 On the Effect of the Initial Orientation

In addition to case 2(a) and case 2(b), the effects of different initial yaw angle is studied in this section. The robots associated with each figure in this subsection have the same mechanic configurations and the same initial conditions.

Let us compare the following figures:

- Figure 4: three robots with  $15^\circ$  initial yaw angle (case 2a)
- Figure 6: three robots with  $-15^\circ$  initial yaw angle (case 2b).
- Figure 15: one robot with  $15^\circ$  initial yaw angle (case 7).
- Figure 16: one robot with  $-15^\circ$  initial yaw angle. This is the case 8.

The initial yaw angle affects the curvature of the optimal trajectory, but does not change the trend of the optimal trajectory. This indicates that the initial yaw angle matters, but not critical. Figures 15 and 16 support the above statement - with different initial yaw angles, the two robots starting at the same position have different trajectories, but their final positions are close. For multi-robot cases, the *formation* pattern of the robots tends to be similar. The optimal sensor formation along the optimal sensor trajectories is an interesting future research topic.



**Figure 16** The optimal trajectory of differentially-driven robots: initial yaw angle  $-15^\circ$  (case 8).

## 5 Conclusion

This paper presents a numerical procedure for optimal sensor-motion scheduling of diffusion systems. Given a DPS with nominal parameters, differentially-driven mobile robots move along their optimal trajectories such that the sensor noise effect on the estimation of system parameters is minimized. This optimal measurement problem is an important module for a potential closed-loop DPS parameter identification algorithm. This paper reformulates a differentially-driven robot's dynamic model in the framework of optimal control. By the combined use of two existing Matlab toolboxes for optimal control (RIOTS) and partial differential equations (Matlab PDE Toolbox), the optimal sensor-motion scheduling problem can be numerically solved successfully. Some simulation results are presented with some interesting comparative observations.

## Acknowledgment

The authors would like to express their gratitude to Dr. M. Patan for sending a hardcopy of his Ph.D dissertation (Patan, 2004) and to Professor Kevin L. Moore for many stimulating discussions on this line of research over the past several years.

## Appendix: List of Notations

- $b$ : the viscous damping coefficient;
- $m$ : weight.
- $I$ : robot's inertia with respect to the  $z$  axis (a scalar).
- $\mathcal{M}$ : inertia matrix.
- $l$ : robot's axis length, except in  $\sum_{l=1}^N$ , where  $l$  is an integer.
- $\alpha$ : yaw angle.
- $r$ : wheel radius.
- $(x, y)$ : position in 2D.
- $\theta$ : the state vector for a mechanic system.
- $M(\theta), C(\theta, \dot{\theta}), N(\theta, \dot{\theta})$ : coefficient matrices for a generic mechanic systems.
- $g$ : sensitivity function (a vector).  $g_{(i)}$  is the  $i$ th entry.
- $\tau_l, \tau_r$ : the torque (control signal) on the left and right wheel.
- $\tau$ : generic torque, or the torques for one differentially-driven robot, i.e.  $\begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix}$ .
- $\tau_3$ : the control signal for three robots.
- $\mathbf{x}$ : the extended state vector for one robot.
- $\mathbf{x}_3$ : the extended state vector for three robots.
- $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}_3$ : the extended Mayer state vector, for one and three robots respectively.
- $\cdot^{(i)}$ : a variable associated with the  $i$ th robot (sensor).
- $A_1, B_1$ : state space matrices for one differentially-driven robot.
- $A_3, B_3$ : state space matrices for three differentially-driven robots.
- $\hat{\cdot}$ : prediction.
- $[\ ]^T$ : transpose.
- $t_0$ : initial time.

- $t_f$ : final time.
- $L_\infty^m[t_0, t_f]$ : the space of essentially bounded, measurable functions, from  $[t_0, t_f]$  into  $\mathbb{R}^m$
- $N$ : number of robots (sensors).
- $\mathbb{R}$ : real number set.
- $T$ :  $\{t|t \in (t1, t2)\}$ , a period of time.
- $\Omega$ : domain.
- $\partial\Omega$ : boundary of the domain.
- $u(x, y, t)$ : concentration at the position  $(x, y)$  and time  $t$ .
- $\omega(t)$ : angular velocity of the left and right wheels.
- $\|\cdot\|_Q^2$ : weighted square.
- $\tau_{min}, \tau_{max}$ : the range of control signal. Note that they may be vectors. Each entry of  $\tau_{min}$  is the lower bound of the associated  $\tau$ .
- $\mathbf{x}_{min}, \mathbf{x}_{max}$ : refer to the above. It is the boundary of state variables.
- $\leq$  for vectors: element-wise comparison.
- $g_o, l_o$ : objective functions.
- $l_{ti}$ : trajectory constraint.
- $g_{ei}$ : end point inequality constraint.
- $g_{ee}$ : end point equality constraint.
- $\xi$ : initial value of  $\mathbf{x}$ .
- $\chi$ : Mayer states (a matrix).
- $\chi_{dl}$ : Mayer vector (lower diagonal entries of  $\chi$ ).
- $\cdot^{(i)}, \cdot^{(i,j)}$ : these subscripts are indices of a vector or a matrix.

**References**

Arora, Anisha (2005), sensing a diffusion process using distributed robots, Master’s thesis, Utah State University, Logan Utah, USA.

Carotenuto, L., P. Muraca and G. Raiconi (1987), ‘Optimal location of a moving sensor for the estimation of a distributed-parameter process’, *International Journal of Control* **46**(5), 1671–1688.

Cassandras, Christos G. and Wei Li (2005), ‘Sensor networks and cooperative control’, *European Journal of Control* **11**(4–5), 436–463.



- Chen, Peng-Yu (2005), Pattern formation in mobile wireless sensor networks, Master's thesis, Utah State University, Logan Utah, USA.
- Chen, Yang Quan, Kevin L. Moore and Zhen Song (2004a), Diffusion boundary determination and zone control via mobile actuator-sensor networks (MAS-net) – challenges and opportunities, in 'Proc. of SPIE Conf. on Intelligent Computing: Theory and Applications II, part of SPIE's Defense and Security', SPIE, Orlando, FL., USA.
- Chen, YangQuan, Kevin L. Moore and Zhen Song (2004b), 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. Published on CD-ROM.
- Chong, Chee-Yee and Srikanta P. Kumar (2003), 'Sensor networks: Evolution, opportunities, and challenges', *Proceedings of the IEEE* **91**(8), 1247–1256.
- Christofides, Panagiotis D. (2001), *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*, Systems & Control: Foundations & Applications, Birkhäuser, Boston.
- Curtain, Ruth F. and Hans Zwart (1995), *An Introduction to Infinite-Dimensional Linear Systems Theory*, Texts in Applied Mathematics, Springer-Verlag, New York.
- Khapalov, Alexander Yu. (1992), 'Optimal measurement trajectories for distributed parameter systems', *Systems & Control Letters* **18**(6), 467–477.
- Kubrusly, C. S. and H. Malebranche (1985), 'Sensors and controllers location in distributed systems — A survey', *Automatica* **21**(2), 117–128.
- Lasiecka, Irena and Roberto Triggiani (2000), *Control Theory for Partial Differential Equations: Continuous and Approximation Theories*, Vol. I and II of *Encyclopedia of Mathematics and Its Applications*, Cambridge University Press, Cambridge.
- Martínez, Sonia and Francesco Bullo (2006), 'Optimal sensor placement and motion coordination for target tracking', *Automatica* **42**, 661–668.
- Moore, Kevin L. and Yang Quan Chen (2004), model-based approach to characterization of diffusion processes via distributed control of actuated sensor networks, in 'The 1st IFAC Symposium on Telematics Applications in Automation and Robotics', IFAC, Helsinki University of Technology Espoo, Finland.
- Moore, Kevin L., Yang Quan Chen and Zhen Song (2004a), Diffusion based path planning in mobile actuator-sensor networks (MAS-net) - some preliminary results, in 'Proc. of SPIE Conf. on Intelligent Computing: Theory and Applications II, part of SPIE's Defense and Security', SPIE, Orlando, FL., USA.
- Moore, Kevin L., YangQuan Chen and Zhen Song (2004b), 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. Published on CD-ROM.



- Nakano, Kazushi and Setsuo Sagara (1981), ‘Optimal measurement problem for a stochastic distributed parameter system with movable sensors’, *International Journal of Systems Science* **12**(12), 1429–1445.
- Nakano, Kazushi and Setsuo Sagara (1988), ‘Optimal scanning measurement problem for a stochastic distributed-parameter system’, *International Journal of Systems Science* **19**(7), 1069–1083.
- Neittaanmäki, P. and D. Tiba (1994), *Optimal Control of Nonlinear Parabolic Systems. Theory, Algorithms, and Applications*, Monographs and Textbooks in Pure and Applied Mathematics, Marcel Dekker, Inc., New York.
- Ögren, Petter, Edward Fiorelli and Naomi Ehrich Leonard (2004), ‘Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment’, *IEEE Transactions on Automatic Control* **49**(8), 1292–1302.
- Omatu, Sigeru and John H. Seinfeld (1989), *Distributed Parameter Systems: Theory and Applications*, Oxford Mathematical Monographs, Oxford University Press, New York.
- Patan, M. (2004), Optimal observation strategies for parameter estimation of distributed systems, PhD thesis, University of Zielona Góra, Zielona Góra.
- Rafałłowicz, Ewaryst (1986), ‘Optimum choice of moving sensor trajectories for distributed parameter system identification’, *International Journal of Control* **43**(5), 1441–1451.
- Schwartz, A., E. Polak and Y.Chen (1997), ‘Riots a matlab toolbox for solving optimal control problems’.  
**URL:** <http://www.schwartz-home.com/adam/RIOTS/>
- Sinopoli, Bruno, Courtney Sharp, Luca Schenato, Shawn Schaffert and S. Shankar Sastry (2003), ‘Distributed control applications within sensor networks’, *Proceedings of the IEEE* **91**(8), 1235–1246.
- Uciński, Dariusz (2000a), ‘Optimal sensor location for parameter estimation of distributed processes’, *International Journal of Control* **73**(13), 1235–1248.
- Uciński, Dariusz (2000b), ‘Optimization of sensors’ allocation strategies for parameter estimation in distributed systems’, *Systems Analysis — Modelling — Simulation* **37**, 243–260.
- Uciński, Dariusz (2001), Sensor motion planning with design criteria in output space, in A. C. Atkinson, P. Hackl and W. Müller, eds, ‘*mODa 6*, Proc. 6th Int. Workshop on Model-Oriented Data Analysis, Puchberg/Schneeberg, Austria, 2001’, Physica-Verlag, Heidelberg, pp. 235–242.
- Uciński, Dariusz (2005a), *Optimal Measurement Methods for Distributed-Parameter System Identification*, CRC Press, Boca Raton, FL.
- Ucinski, Dariusz (2005b), *Optimal measurement methods for distributed parameter system identification*, CRC press, Boca Raton, Florida.

- Uciński, Dariusz and Józef Korbicz (1999), On robust design of sensor trajectories for parameter estimation of distributed systems, *in* 'Proc. 14th IFAC World Congress, Beijing, China, 5–9 July 1999', Vol. H: Modeling, Identification, Signal Processing I, pp. 73–78.
- Uciński, Dariusz and YangQuan Chen (2005), 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'. Published on CD-ROM.
- Wang, Zhongmin, Zhen Song, Peng-Yu Chen, Anisha Arora, Dan Stormont and YangQuan Chen (2004), MASmote – a mobility node for MAS-net (mobile actuator sensor networks), *in* 'IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO)', Shenyang, China.
- Zwart, Hans and Jan Bontsema (1997), An application-driven guide through infinite-dimensional systems theory, *in* G.Bastin and M.Gevers, eds, 'European Control Conference 1997: Plenaries and Mini-Courses', CIACO, Ottignies/Louvain-la-Neuve, pp. 289–328.

