2005 IEEE/RSJ International Conference on Intelligent Robots and Systems

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

Zhen Song†, *Student Member, IEEE*, YangQuan Chen†, *Senior Member, IEEE*,
Jinsong Liang†, *Student Member, IEEE* and Dariusz Uciński‡, *Member, IEEE*,

†CSOIS, Department of Electrical and Computer Engineering
Utah State University, Logan, Utah 84322, USA
{zhensong, yqchen, jsliang}@ieee.org
‡ 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 nonholomonic 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 differential-drive 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 [1] and Matlab PDE Toolbox [2]. The objective function of this method is designed to minimize the effect of the sensing noise. Extensive simulation results are presented for illustration.**

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

I. Introduction

The Motivation of MAS-net Project. The application background of this paper is a project called mobile actuator-sensor networks (MAS-net) [3]–[8]. This project is proposed to synergize 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.

From the theoretical aspect, the project is featured with real-time parameter estimation and state estimation of a class of distributed parameter systems (DPSs) by a swarm of mobile sensors with nonholonomic constraints and limited communication capability. In addition, mobile actuators (e.g., a mobile robot equipped with a chemical neutralizer dispenser) with the same nonholonomic constraints will be added to control the DPS (basically, to reduce the concentration) with the help of the mobile sensors.

System Identification for DPS. While observing a DPS, it is most often impossible to measure the system states over an entire spatial domain, and therefore the problem of where to locate the measurement sensors becomes very important. It is not trivial to estimate the parameters or the states of the DPS by a limited number of sensors. The study of the DPS identification problem started about 40 years ago, but the number of papers that discussed the sensor-motion-scheduling problem is still limited. Many of them discussed the optimal-placement problem for static sensors. Some researchers discussed state estimation [9]–[11], some focused on parameter identification [12]–[14], some addressed both simultaneously [15].

The model-based adaptive measurement and control problem of the MAS-net project is formulated in [4], [5]. To implement this distributed control system, the parameter estimation for the DPS is required, and the choice of best experimental conditions for that purpose is referred to as an "optimum experimental design" problem as discussed in [16]–[20].

The recent publications [21], [22] are closely related to the MAS-net estimation problem. In [21], [22], the dynamic-sensor-motion scheduling problem is studied intensively with many practical considerations such as robust design, collision avoidance, etc.

Motivation for Mobile Sensors. For distributed parameter system identification, using a group of movable sensors for measurement has obvious advantages [21]. However, as also pointed out in [21], [22], there exists a fundamental problem for the DPS-parameter estimation: the optimal sensor locations for the DPS-parameter estimation depend on those unknown parameters to be identified. In [21], this "chicken-and-egg" problem is described as "*if you tell me the parameters of a DPS, I promise to design an experiment to measure them optimally.*"

One of the solutions to this problem is to design robust sensor trajectory/placement scheduling schemes which are not sensitive to the unknown parameters, such as [23] and Chapter 6 of [21].

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 [22]. 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

Corresponding author: Prof. YangQuan Chen, Center for Self-Organizing and Intelligent Systems, Dept. of Electrical and Computer Engineering, 4160 Old Main Hill, Utah State University, Logan, UT 84322-4160. **T**: (435)7970148, **F**: (435)7973054, **W**: www.csois.usu.edu

problem of how to "close" the loop of this system.

In this paper, in the vein of [21], [22], 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 nonholomonic constraints as in MASmotes [6], the two wheeled differentially-driven mobile robots, in our MAS-net project [3]–[8].

II. Problem Formulation of the Sensor-Motion Scheduling for Diffusion Systems

A. The Dynamic Model of Differential-Drive Robots

MASmote [6] 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 as follows:

- m: the weight of the robot.
- I: the inertia of the robot along the z axis. Note that I is a scalar. In the literature, $\mathcal{M}$ can be a 3 by 3 inertia matrix of the robot. The I in (1) is the entry at the 3rd row and 3rd column of $\mathcal{M}$, i.e. $\mathcal{M}_{(3,3)}$.
- l: the length of the robot's axis.
- r: wheel radius. The left and right wheels have the same radius.
- α: the yaw angle as shown in Fig. 1.
- (x, y): the coordinate of the center of the axis.
- τ_l,τ_r: the torque applied on the left and right wheel, respectively.

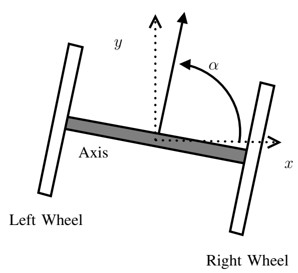

Fig. 1. A differentially-driven mobile robot

In (1), the mobile robot is represented in a form of a 2^{nd} 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 τ is defined as $\tau = [\tau_l, \tau_r]^T$. To have a compact notation, let us define matrix 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

$$\dot{\mathbf{x}} = A_1\mathbf{x} + B_1\tau. \tag{2}$$

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

To solve the multi-robot-motion-scheduling problems in Sec. IV, 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

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

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

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:

$$\dot{\mathbf{x}}_3 = A_3\mathbf{x}_3 + B_3\tau_3. \tag{3}$$

B. The Model of the Diffusion Process

For comparison purposes, here we use the same diffusion system model as in Example 4.1 in [21]. We rewrite it using our notation in the following form:

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

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.

C. 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:

$$z^{(i)}(t) = u(\mathbf{x}^{(i)}(t), t) + \epsilon(\mathbf{x}^{(i)}(t), t), \tag{4}$$

where ϵ is 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 δ is Dirac's delta function, and σ 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) [21]. Up to a constant multiplier, the FIM constitutes the

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

inverse of the covariance matrix for the least-squares estimator defined as the minimizer of the following "fit-to-data" criterion [21]:

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

The notation ˆ in (5) indicates the predicted value. For N robots, $J_1(c)$ becomes

$$J_1(c) = \sum_{j=1}^{N} \frac{1}{2} \int_T \|z^{(j)}(t) - \hat{u}^{(j)}(\mathbf{x}, t; c)\|^2 \mathrm{d}t. \quad (6)$$

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

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

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 [21].

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 $J_2(\mathbf{x})$ is minimized.

D. Problem Reformulation in the Optimal Control Framework

According to the general optimal control problem formulation in RIOTS [1], 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

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

subject to the following conditions and constraints:

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

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. $g(\mathbf{x}^{(i)})$ is called the sensitivity function, where

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

Then, the FIM in (7) is

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

Define the Mayer states as

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

where

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

Denote χ_{dl} the stack vector which stacks all the entries on the diagonal and below the diagonal of χ 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 χ_{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:

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

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

III. Finding a Numerical Solution of the Optimal Mobile Sensor Motion Scheduling Problem

A. 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 [1], [24]. The most important function in this optimal control toolbox is `riots` explained in detail on page 73 of [1].

```
[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. IV, 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, $\chi_{\mathbf{d1}}$ 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 τ.
- `t`: time.
- `Umin`, `Umax`: bounds for τ.

The definitions of other parameters are described in [1].

B. 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:

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

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. II-B.

IV. Illustrative Simulations

A. Differential Drive vs. Omni-Directional Drive

In [21], the robot model is a simple kinematic model:

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

where $\omega(t)$ is the angular speed vector, and r is the radii of the wheels. Obviously, (15) is an approximation. In this paper, we refer a robot that subjects to the kinematic in (15) 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 four 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° and subcase(2b) of -15°.

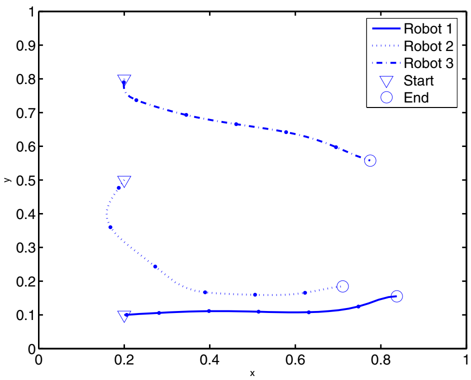

Fig. 2. The optimal sensor trajectories of omni-directionally-driven robots (case 1)

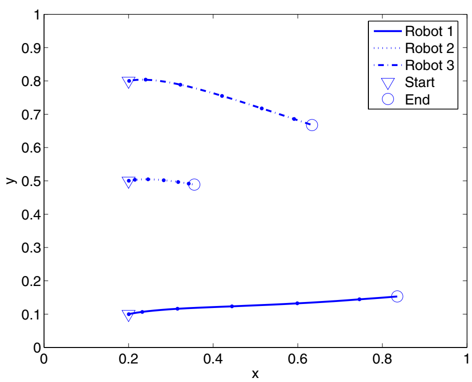

Fig. 3. The optimal sensor trajectories of differentially-driven robots: 15° initial yaw angle (case 2a)

- 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.

According to the above definitions, Fig. 2 shows the results for case 1; Fig. 3 for case 2(a); Fig. 4 for case 2(b); Fig. 5 for case 3; and Fig. 6 for case 4. 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. 3 and Fig. 4 is especially interesting.

B. Comparison of Robots with Different Capabilities

Here we consider two more cases to see .

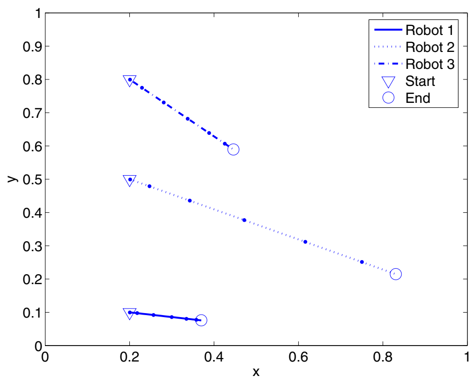

Fig. 4. The optimal sensor trajectories of differentially-driven robots: -15° initial yaw angle (case 2b)

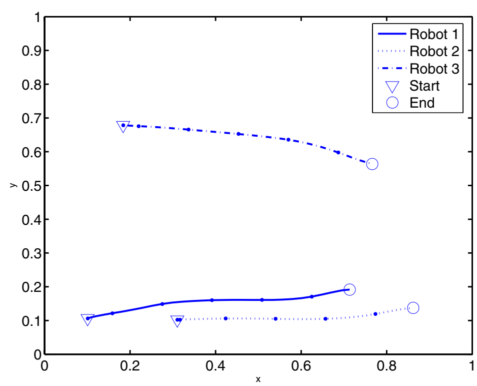

Fig. 5. The optimal sensor trajectories of omni-directionally-driven robots (case 1) using optimal initial conditions (case 3)

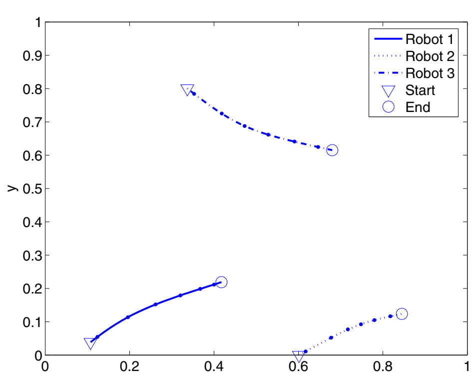

Fig. 6. The optimal sensor trajectories of differentially-driven robots using optimal initial conditions (case 4)

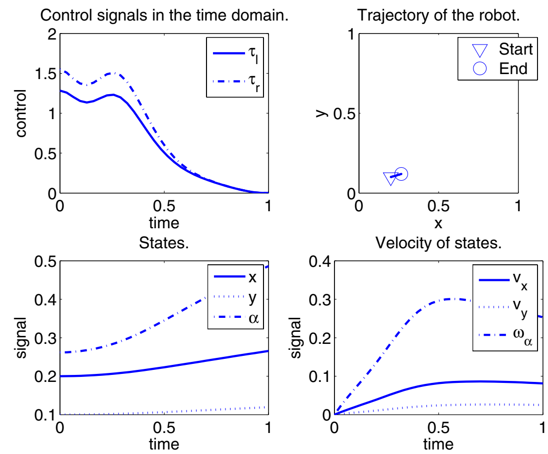

Fig. 7. The optimal trajectory of weak differential-drive robots.

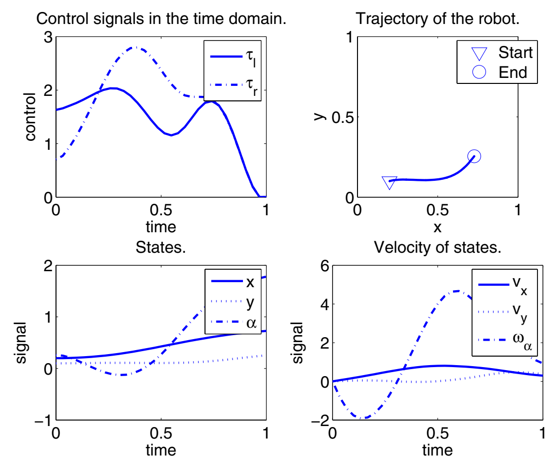

Fig. 8. The optimal trajectory of strong differential-drive robots: initial yaw angle is 15°.

- case 5: using a single “weak” robot, whose weight is 0.5 and the range of its torque for each wheel is ± 10.
- case 6: using a single “strong” robot, whose weight is 0.05 and the range of its torque for each wheel is ± 100.

With the same fixed initial states, and the same T, the robot in case 5 moves shorter than in case 6 as seen from Fig. 7 and Fig. 8. This matches our intuition that it is desirable for the sensors to measure the DPS states at more spatial locations whenever possible.

C. On the Effect of the Initial Orientation

In additional to case 2(a) and case 2(b), the effects of different initial yaw angle is studies 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 3: three robots with 15° initial yaw angle.
- Figure 4: three robots with -15° initial yaw angle.
- Figure 8: one robots with 15° initial yaw angle.
- Figure 9: one robots with -15° initial yaw angle.

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 8 and 9 support the above statement - with different initial yaw angles, the two robots starting at the same position have different trajectory, 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.

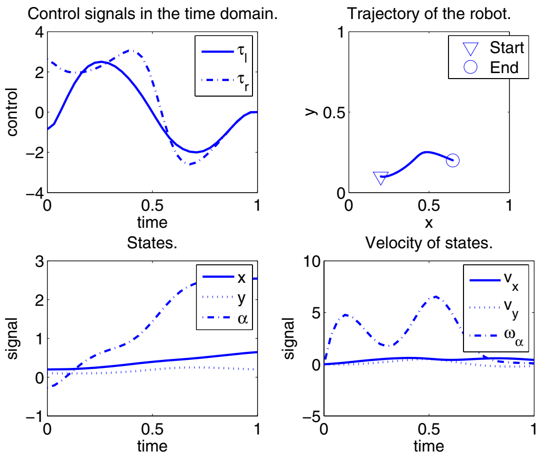

Fig. 9. The optimal trajectory of strong differential-drive robots: initial yaw angle -15°.

V. 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 differential-drive 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

YangQuan Chen would like to express his gratitude to Dr. M. Patan for sending a hardcopy of his Ph.D dissertation [22] and to Professor Kevin L. Moore for many stimulating discussions on this line of research over the past several years.

References

[1] A. Schwartz, E. Polak, and Y. Q. Chen, "RIOTS - A Matlab Toolbox for Solving Optimal Control Problems," May 1997. [Online]. Available: http://www.schwartz-home.com/˜adam/RIOTS/
[2] The MathWorks Inc., "Matlab Partial Differential Equation Toolbox," http://www.mathworks.com/products/pde/.
[3] K. L. Moore and Y. Q. Chen, "Model-based approach to characterization of diffusion processes via distributed control of actuated sensor networks," in *Proc. of The 1st IFAC Symposium on Telematics Applications in Automation and Robotics*. Helsinki University of Technology Espoo, Finland: IFAC, June 2004.
[4] K. L. Moore, Y. Q. Chen, and Z. Song, "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*. Orlando, FL., USA: SPIE, April 2004.
[5] Y. Q. 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. of SPIE Conf. on Intelligent Computing: Theory and Applications II, part of SPIE's Defense and Security*. Orlando, FL., USA: SPIE, April 2004.
[6] Z. Wang, Z. Song, P.-Y. Chen, A. Arora, D. Stormont, and Y. Chen, "MASmote - a mobility node for MAS-net (mobile actuator sensor networks)," in *Proc. of the First IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO04)*, Shenyang, China, Aug. 2004.
[7] P.-Y. Chen, "Pattern formation in mobile wireless sensor networks," Master's thesis, Utah State University, Logan Utah, USA, 2005.
[8] A. Arora, "Sensing a diffusion process using distributed robots," Master's thesis, Utah State University, Logan Utah, USA, 2005.
[9] C. G. Kubrusly and H. MaleBranche, "Sensors and controllers location in distributed systems," *Automatica*, vol. 21, no. 2, pp. 17–128, 1985.
[10] A. E. Jaï, "Distributed systems analysis via sensors and actuators," *Sensors and Actuators*, vol. A, no. 29, pp. 1–11, 1991.
[11] J. Korbicz and D. Uciński, "Sensors allocation for state and parameter estimation of of distributed systems." in *IUTAM symposium*, W. Gutkowsi and J. Bauer, Eds. Zakopane, Poland: Springer-Verlag, 31 Aug.-3 Sep. 1994, pp. 178–189.
[12] E. Rafajlowicz, "Design of experiments for parameter identification of the static distributged system," *Int. J. of Systems Science*, vol. 4, no. 4, pp. 349–361, 1978.
[13] Z. H. Quereshi, T. S. Ng, and G. C. Goodwin, "Optimal experimental design for identification of distributed parameter systems," *International Journal of Control*, vol. 31, no. 1, pp. 21–29, 1980.
[14] E. Rafajlowicz, "Optimal experiment design for identification of linear distributed-parameter systems: frequency domain approach," *IEEE Transactions on Automatic Control*, vol. 28, no. 7, pp. 806–808, July 1983.
[15] H. Malebranche, "Simultaneous state and parameter estimation and location of sensors for distributed systems," *International Journal of Systems Science*, no. 8, 1988.
[16] R. K. Mehra, "Optimization of measurement schedules and sensor designs for linear dynamic systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 55–64, 1976.
[17] W. G. Müller, *Collecting spatial data. Optimum design of experiments for random fields*, 2nd ed. Heidelberg: Physica-Verlag, 2001, vol. Contributions to Statistics.
[18] E. Rafajlowicz, "Design of experiments for eigenvalue identification in distributed-parameter systems," *International Journal of Control*, vol. 34, no. 6, pp. 1079–1094, 1981.
[19] ——, "Design of experimetns for parameter identification of the static distributed systems," *System Science*, vol. 4, no. 4, pp. 349–361, 1978.
[20] A. El Jai and A. J. Pritchard, *Sensors and Actuators in Distributed Systems Analysis*, ser. Ellis Horwood Series in Applied Mathematics. Chichester, West Sussex: Ellis Horwood: Ellis Horwood, John Wiley, 1988.
[21] D. Uciński, *Optimal measurement methods for distributed parameter system identification*. Boca Raton, Florida: CRC Press, 2005.
[22] M. Patan, "Optimal observation strategies for parameter estimation of distributed systems," PhD thesis, University of Zielona Gòra, Zielona Gòra, 2004.
[23] D. Uciński, "A technique of robust sensor allocation for parmeter estimation in distributed systems," in *5th European Control Conference*, P. M. Frank, Ed., Karlsruhe, Germany, 31 Aug.-3 Sep. 1999, published on CD-ROM.
[24] Y. Q. Chen and A. L. Schwartz, *Recent Developments in Optimization and Optimal Control in Chemical Engineering*, ch. RIOTS_95 - a MATLAB Toolbox for Solving General Optimal Control Problems And Its Applications to Chemical Processes.