Proceedings of the IEEE
International Conference on Mechatronics & Automation
Niagara Falls, Canada • July 2005

`Diff/Wave-MAS2D`: A Simulation Platform for Measurement and Actuation Scheduling in Distributed Parameter Systems with Mobile Actuators and Sensors

Jinsong Liang, *Student Member, IEEE* and YangQuan Chen, *Senior Member, IEEE*

Center for Self-Organizing and Intelligent Systems (CSOIS)
Dept. of Electrical and Computer Engineering
4160 Old Main Hill, Utah State University, Logan, UT 84322-4160, USA

***Abstract*— This paper presents a simulation platform called `Diff/Wave-MAS2D` we have developed for measurement scheduling and distributed controls in distributed parameter systems with networked moving sensors and moving actuators. The background information for this simulation platform is introduced and the main features of the software packages are discussed. Two illustrative simulation examples (diffusion control and wave equation control) are demonstrated to show some of the capabilities and unique features of this simulation platform.**

***Index Terms*— Sensor networks, actuator networks, diffusion process, wave equation, simulation.**

I. BACKGROUND

Sensor networks are drawing increased attention from research communities, industry sectors, and government agencies. As stated in [1], sensor networks will “have significant impact on a broad range of applications relating to national security, health care, the environment, energy, food safety, and manufacturing. The convergence of the Internet, communications, and information technologies with techniques for miniaturization has placed sensor technology at the threshold of a period of major growth.” Recent surveys on sensor networks [2], [3], [4], [5], [6] also indicate the importance of sensor networks research. Many on-going efforts are focused on various specific issues in sensor networks such as sensor structures [7], [8], [9], communication [3], [10], data processing and sensor fusion methods [2], [4], [11], sensor deployment and localization [2], [4], [12], calibration [13], [14], etc.

However, from dynamic systems control point of view, the sensor networks should be part of a complete closed-loop system, especially a closed-loop distributed control system, with a specific mission defined. Although a few systems consisting sensor networks have been proposed [15], [16], [17] to carry out specific missions, neither of the missions chooses closed-loop distributed control as the final goal. So far, there is no such real-time, closed-loop distributed feedback control system involving networked actuators and sensors [18], [19]. However, there are applications that require the use of mobile actuators and sensor networks, which we denote MAS-Net. These include the following motivating examples.

- Application Scenario 1 (**land**): In this case, each networked sensor is mounted on a ground mobile robot. The mission is to determine the safe radiation boundary of the radiation field from possibly multiple, or even moving, radiation sources. Each robot is actuated according to spatial and temporal sensed information (radiation gradient, spatial position, etc.) from more than one actuated or mobile sensors.
- Application Scenario 2 (**water**): This case is similar to Application Scenario 1 if the toxic diffusion source is a one-time pouring and the diffusion is in steady state. However, the boundary may be dynamically evolving as the toxic source keeps polluting the pollutant into the reservoir. The actuated or mobile sensors here are autonomous boats mounted with toxic chemical concentration sensors. The boats are commanded according to the spatial-temporal sensed information from more than one sensor. Furthermore, assume that some of the boats (not all of the boats) are equipped with relevant neutralizing chemicals to make the water detoxified. By proper design of the distributed sensing and actuation/control strategies, it is possible to control the zone or shape of the toxic region to match the given desirable zone/shape. Now, we have a complex distributed feedback control system that is more challenging than the networked actuators and sensors themselves.
- Application Scenario 3 (**air**): This scenario is similar to the above water case, but it is more complicated since 3D space must be explored. Here, the actuated or mobile sensors are unmanned aerial vehicles (UAVs) equipped with concentration detectors and anti-contamination chemical agent(s) distributors. For this case, a more detailed description can be found in [20].

Motivated by these examples, in [21], we introduced two high-level tasks or missions, i.e., the diffusion boundary determination and zone control via mobile actuator-sensor networks. As a means of exploring the algorithmic, com-

---

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`, **E**: `yangqquan.chen@usu.edu`

putational, and practical issues of the problems, we have developed a 2-D MAS-Net hardware test-bed introduced in [22]. It is well-known that software simulation plays an equally important (in some situations, more important) role as hands-on, real hardware based experiments. There are lots of readily available mathematical tools for solving PDE problems, such as MATLAB PDE Toolbox [23], FEMLAB [24], Nastran [25], ANSYS [26]. However, if we study these tools carefully, we find that neither is suitable for simulating the problems mentioned above due to the following reasons:

- Neither is designed with movable sensors in mind. Although this can be worked around with careful post-process, it makes code writing for on-line sensor scheduling hard and error-prone.
- Neither is designed with movable actuators in mind. This shortcoming is vital, since it makes simulation of closed-loop control of PDEs very hard, if not impossible.

In view of the above problems, we developed our own software simulation platform, `Diff-MAS2D` and `Wave-MAS2D`, for simulation of measurement scheduling and distributed controls in distributed parameter systems equipped with moving sensors and moving actuators.

The paper is organized as follows. In Sec. II, the main features of the `Diff-MAS2D` and `Wave-MAS2D` are introduced. Two simulation examples are shown in Sec III to demonstrate the capabilities and the unique features of `Diff-MAS2D` and `Wave-MAS2D`. Since `Diff-MAS2D` and `Wave-MAS2D` are still in beta stage, in Sec.IV, some future work is introduced. Finally, Sec. V concludes the paper.

II. DIFF/WAVE-MAS2D SIMULATION PLATFORM

`Diff-MAS2D` and `Wave-MAS2D` are able to simulate closed-loop distributed control of the diffusion process and the wave equation, respectively, using movable sensors and movable actuators. Specifically, `Diff-MAS2D` is used to solve the following PDE:

$$\frac{\partial u}{\partial t} = k\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f(\tilde{u}, x, y, t), \tag{1}$$

where $u = u(x, y, t)$ is the physical variable to be controlled; $0 \leq x \leq 1$ and $0 \leq y \leq 1$ is the spatial domain; $t \geq 0$ is the time domain; k is a positive real constant related to system parameters; $f(\tilde{u}, x, y, t)$ is a combination of control and disturbances

$$f(\tilde{u}, x, y, t) = f_c(\tilde{u}(x, y, t), x, y, t) + f_d(x, y, t),$$

where $\tilde{u}(x, y, t)$ is the measured data of $u(x, y, t)$ from the movable sensors; $f_c(\tilde{u}(x, y, t), x, y, t)$ is the control applied by the movable actuators; $f_d(x, y, t)$ is the disturbance. The exact format of $f_c(\tilde{u}(x, y, t), x, y, t)$ depends on the closed-loop control law designed by the user.

`Diff-MAS2D` and `Wave-MAS2D` are designed as complete simulation environments within Matlab for the end user. Internally, $u(x, y, t)$ is discretized in spatial domain manually using finite difference method; while the discretization and integration in time domain are left to Matlab. There are two main advantages with the different treatments of spatial domain and time domain. First, the coding is greatly simplified, since $u(x, y, t)$ is only discretized manually in spatial domain. Second, since the discretization and integration in time domain are handled by Matlab itself, for the end user, the large amount of Matlab functions can be utilized in the path planning of the sensors/actuators and in the control law design.

For the end user, arbitrary combination of the following two types of boundary conditions can be used as boundary conditions for each boundary ($x = 0$, $x = 1$, $y = 0$, and $y = 1$).

- Dirichlet boundary condition
$$u = C \tag{2}$$
where C is a real constant.
- Neumann boundary condition
$$\frac{\partial u}{\partial n} = C_1 + C_2 u \tag{3}$$
where C_1 and C_2 are two real constants; n is the outward direction normal to the boundary.

The main features of `Diff-MAS2D` also include:

- Any number of sensors and actuators can be used.
- Sensors and actuators can be collocated (bound together) or non-collocated (separated).
- Disturbances can be movable and time-varying.
- Movement of sensors and actuators can be open-loop (designed by the user as functions of time only) or closed-loop (designed by the user as functions of t, $\tilde{u}(x, y, t)$, sensor position/velocity, and actuator position/velocity).
- Arbitrary control algorithms can be applied in $f_c(\tilde{u}(x, y, t), x, y, t)$.
- `Diff-MAS2D` is designed with easy-to-use in mind. The user needs only consider the high-level problems, such as the control law design and on-line sensor/actuator scheduling, without knowing the internal implementation details of `Diff-MAS2D`.
- `Diff-MAS2D` is designed as a platform to simulate distributed control systems, *i.e.*, the movement of each sensor/actuator can be designed individually based on the information (position and velocity) of other sensors/actuators and the measurement of the sensors.

`Wave-MAS2D` has the same features as `Diff-MAS2D` except that `Wave-MAS2D` solves the following wave equation

$$\frac{\partial^2 u}{\partial t^2} = k\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f(\tilde{u}, x, y, t), \tag{4}$$

The flow chart of `Diff/Wave-MAS2D` is shown in Fig. 1.

III. TWO SIMULATION EXAMPLES

To show the capabilities and the unique features of `Diff-MAS2D` and `Wave-MAS2D`, we will simulate the following problems.

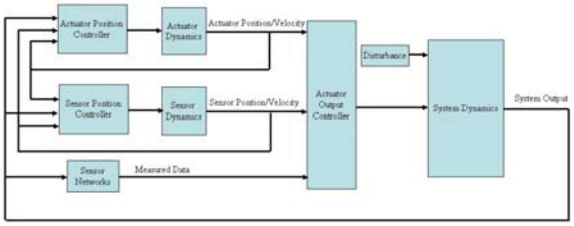

Fig. 1. Simulation flow chart of `Diff/Wave-MAS2D`

A. Example 1: A Simple Pollution Tracking and Control Problem

In this simulation, we will control the diffusion process (1). The initial condition is $u(x, y, 0) = 0$ and the boundary condition for all four boundaries is $\partial u/\partial n = 0$.

Assume there is a point disturbance $f_d = 5$, which can be viewed as a moving pollution source with a constant point concentration, with the following sinusoidal movement in x direction only

$$x = 0.3 \sin(0.628t) + 0.5, \quad y = 0.25.$$

The objective is to control $u(x, y, t)$ to be as close to zero as possible using a small number of actuators and sensors. To achieve this goal, the best strategy is probably to let the actuators track and catch the pollution source and apply the control f_c. By doing this, most part of the pollution can be eliminated as soon as the pollution is emitted, before it spreads and pollutes the whole area.

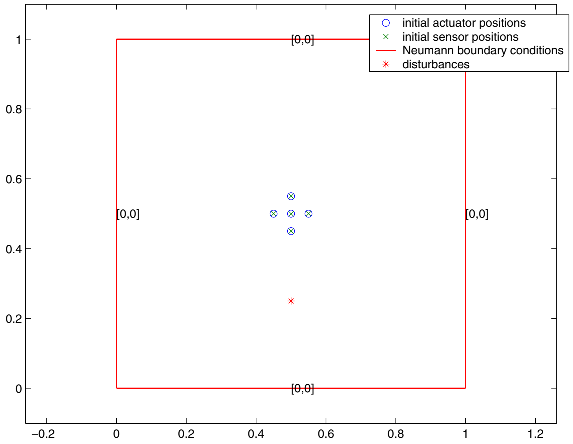

Fig. 2. Initial actuator/sensor layout for Example-1

To track the pollution source, the actuators/sensors are designed to move in the direction of $\nabla\tilde{u}(x, y, t)$, the gradient of $u(x, y, t)$ around the position of sensors/actuators. To calculate the gradient, we use five sensors and five actuators and choose the collocated scheme. The sensors/actuators are formed in a cross shape, shown in Fig. 2, the initial layout ($t = 0$) of the simulation. The gradient is approximately calculated by (denoting the three sensors in horizontal position as s1, s2, s3, and the other two sensors in vertical position as s4 and s5)

$$\nabla\tilde{u}(x, y, t) \approx [\frac{\tilde{u}_{s3} - \tilde{u}_{s1}}{x_{s3} - x_{s1}}, \frac{\tilde{u}_{s4} - \tilde{u}_{s5}}{x_{s4} - x_{s5}}].$$

A simple proportional control law is chosen for the actuators to eliminate the pollution. Denote the five actuators as a1 through a5. The control applied by each actuator is formulated as

$$f_{ai} = -45\tilde{u}_{si}, \quad i = 1, \cdots, 5. \tag{5}$$

The evolution of $u(x, y, t)$ can be viewed through an animation generated by the post-process program. We will use a few pictures to show snapshots of the control results.

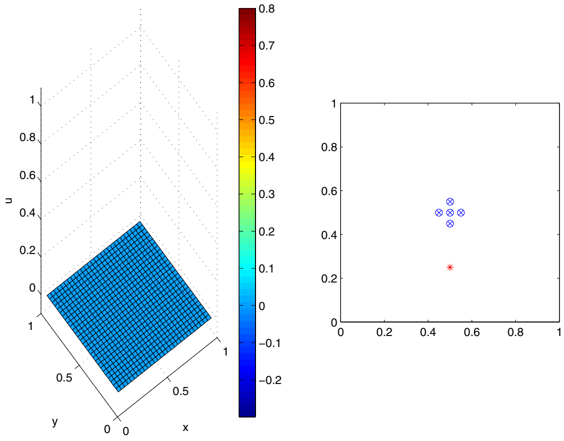

Fig. 3. $u(x, y, t)$, sensor/actuators/disturbance positions at $t = 0$

Figure 3 shows $u(x, y, 0)$ and the positions of the sensors, actuators, and the disturbance at $t = 0$. The left part of the figure is a 3D plot of $u(x, y, 0)$ and the right part of the figure shows the current positions of the sensors, actuators, and the disturbance.

Figure 4 shows $u(x, y, 0.8)$ and the positions of the sensors, actuators, and the disturbance at $t = 0.8$. We can observe a peak in the 3D plot, caused by the pollution source. Since the pollution has not spread to the position of the sensors/actuators, the sensors and the actuators are at almost the same position as in Fig. 3.

Figure 5 shows $u(x, y, 7.0)$ and the positions of the sensors, actuators, and the disturbance at $t = 7.0$. We can see that the sensors/actuators have started to track the pollution source, although not catching it yet. From the 3D plot, the environment can be seen as having been polluted (the color changed from blue to green).

Figure 6 shows $u(x, y, 7.9)$ and the positions of the sensors, actuators, and the disturbance at $t = 7.9$. We can see that the sensors/actuators have caught the pollution source. The peaks in Fig. 4 and Fig. 5 disappeared, showing the effect of the control law (5). After $t = 7.9$, the pollution source is caught till the end of simulation, as shown in Fig. 7.

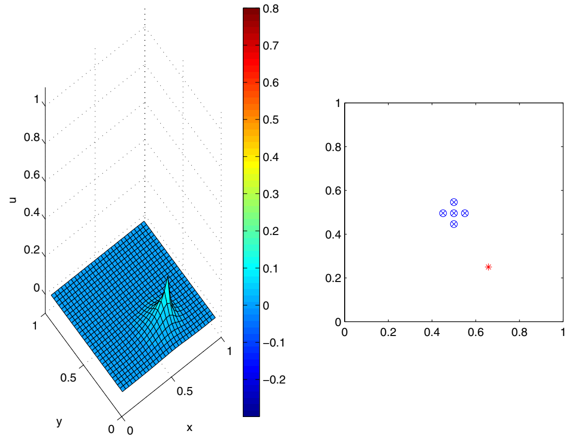

Fig. 4. $u(x, y, 0.8)$, sensor/actuators/disturbance positions at $t = 0.8$

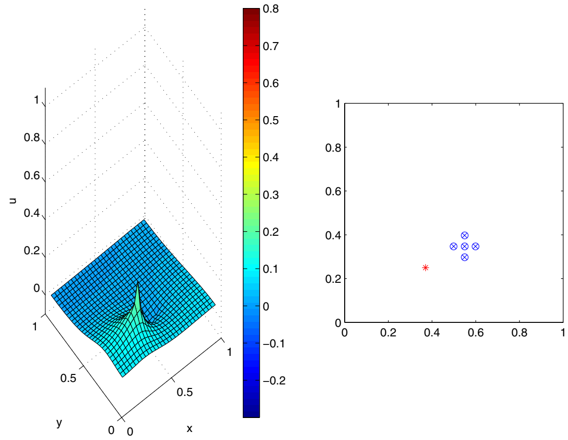

Fig. 5. $u(x, y, 7.0)$, sensor/actuators/disturbance positions at $t = 7.0$

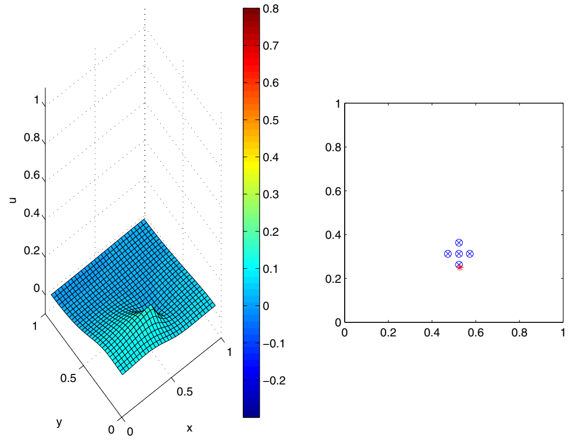

Fig. 6. $u(x, y, 7.9)$, sensor/actuators/disturbance positions at $t = 7.9$

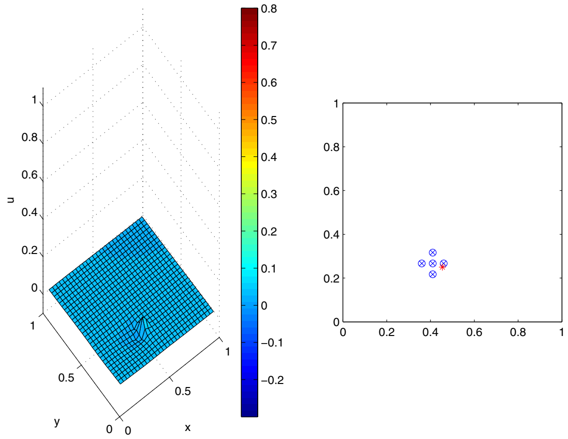

Fig. 7. $u(x, y, 19.7)$, sensor/actuators/disturbance positions at $t = 19.7$

Finally, to compare the effects of different path planning and control algorithms, a possible benchmark is to compare the time profile of total pollution over the whole area:

$$u_I(t) = \int_0^1\int_0^1 u(x,y,t)\mathrm{d}x\mathrm{d}y \tag{6}$$

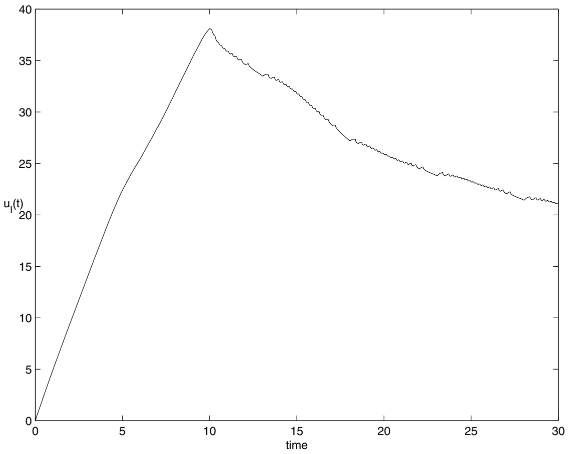

Fig. 8. Integration of $u(x, y, t)$ over x and y

$u_I(t)$ is shown in Fig. 8. Before the pollution source is captured by the actuators at $t \approx 10$, the total pollution is always increasing. After the pollution source is captured, not only the instant pollution emission is eliminated, due to the diffusion process, the total pollution level is also dropping. Note that $u_I(t)$ can be used as the performance index for optimal distributed controller designs.

The above simulation example shows some of the capabilities and features of `Diff-MAS2D`. To the best of the authors' knowledge, `Diff-MAS2D` is the only available software package capable of simulating control of the diffusion process using movable sensors and actuators. With `Diff-MAS2D`, we might be able to give answers to some hard questions, such as

- Given a specification, what are the minimal number of sensors and the minimal number of actuators required?
- What are the advantages of disadvantages of the collocated scheme and the non-collocated scheme?
- How to perform optimal control design and performance evaluation?
- What are the effects of various network induced uncertainties?
- Any better control laws than (5)?
- Any better schemes to track the disturbances?
- ······

Using `Diff/Wave-MAS2D`, some interesting research results were reported in [27], [28], [29] on various aspects of distributed feedback control of a 2D diffusion process using mobile actuator sensor networks involving central Voronoi tessellations techniques.

B. *Example 2: A Simple Vibration Suppression Control Problem*

In this simulation, we assume there is a square membrane governed by (4) with fixed boundaries ($\partial u = 0$). To meet the high accuracy requirement on the displacement $u(x, y, t)$, we will use 81 smart sensors and 81 smart actuators, fixed or embedded built-in on the surface of the membrane, to suppress some unavoidable vibrations. The layout of the sensors/actuators is shown in Fig. 9. Note that, unlike in the Example-1, here all sensors and actuators are spatially fixed but densely distributed. However, we can schedule only small portion of the sensors and actuators. The locations of the selected sensors and actuators could also change over time in a way specified by the designed distributed sensing and actuation policies.

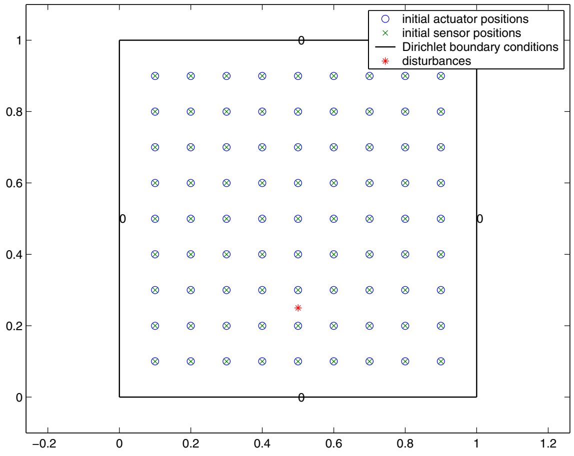

Fig. 9. Sensors/actuators layout for Example-2

Assume the initial condition is $u(x, y, 0) = 0$. The objective is to control $u(x, y, t)$ as close to zero as possible. For demonstration purpose, we choose the following simple proportional control law for the actuators, although obviously a poor control law for a second order system without damping

$$f_{ai} = -30\tilde{u}_{si}, \quad i = 1, \cdots, 81. \tag{7}$$

Assume the system is subject to an impulse disturbance at $t = 0$. With the actuators deactivated , the displacement of the membrane is shown in Fig. 10. If the actuators are activated using the control law (7), the displacement of the membrane is shown in Fig. 11. Compare Fig. 10 and Fig. 11, we can see that the controlled system behaves much better, even if a poor controller (7) is used.

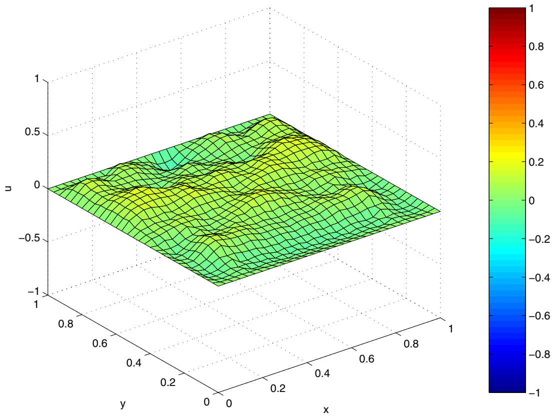

Fig. 10. $u(x, y, 80)$, actuators deactivated

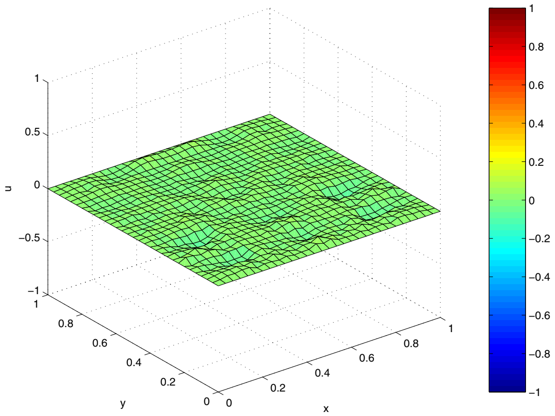

Fig. 11. $u(x, y, 80)$, actuators activated

IV. FUTURE WORK

`Diff-MAS2D` and `Wave-MAS2D` are still in beta stage. There are lots work to be done in the future. Following is a possible to-do list.

- Solve more general PDEs than (1) and (4). Specifically, k will be distributed and time-varying, *i.e.*, $k = k(x, y, t)$; first order terms such as $\partial u/\partial x$ will be added.
- Add more complex boundary conditions than (2) and (3).
- Add obstacles within the domain $0 \leq x \leq 1,\ 0 \leq y \leq 1$ for sensors and actuators to avoid. (Partly done in [29])
- Handle more complex boundaries than simple square boundaries.
- Add network induced uncertainties. (In simple scenarios, this task can be done straight forwardly)

The beta release of `Diff-MAS2D` and `Wave-MAS2D` can be found from `http://www.csois.usu.edu/ people/yqchen/tomasnet`.

V. CONCLUDING REMARKS

In this paper, we have introduced `Diff-MAS2D` and `Wave-MAS2D`, the only available simulation platform for measurement scheduling and controls in distributed parameter systems with moving sensors and moving actuators. Some of the capabilities and unique features are demonstrated in the simulation examples. The usefulness and powerfulness of `Diff-MAS2D` and `Wave-MAS2D` will be further proved in the future research.

REFERENCES

[1] NSF, "National Science Foundation. sensors and sensor networks. NSF program solicitation #NSF 03-512." `http://www.nsf.gov/pubs/2003/nsf03512/nsf03512.htm`, 2003.

[2] H. Qi, S. S. Iyengarb, and K. Chakrabarty, "Distributed sensor networks – a review of recent research," *Journal of the Franklin Institute*, vol. 338, 2001.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[4] S. Kumar, F. Zhao, and D. Shepherd, "Collaborative signal and information processing in microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 13–14, March 2002.

[5] H. Gharavi and S. P. Kumar, "Special issue on sensor networks and applications," *IEEE Proceedings*, vol. 91, no. 8, Aug. 2003.

[6] C.-Y. Chong and S. Pumar, "Sensor networks: Evolution, opportunities, and challenges," *IEEE Proceedings*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[7] T. Abdelzaher, J. Stankovic, S. Son, B. Blum, T. He, A. Wood, and C. Lu, "A communication architecture and programming abstractions for real-time embedded sensor networks," in *Proceedings of The 23rd International Conference on Distributed Computing Systems Workshops*, 2003, pp. 220–225.

[8] H. Yang and B. Sikdar, "A protocol for tracking mobile targets using sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 71–81.

[9] E. Callaway, *Wireless Sensor Networks: Architectures and Protocols*. Boca Raton, FL, USA: CRC Press, 2003.

[10] H. Gharavi and K. Ban, "Multihop sensor network design for wide-band communications," *IEEE Proceedings*, vol. 91, no. 8, pp. 1235–1249, Aug. 2003.

[11] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative signal and information processing: An information-directed approach," *IEEE Proceedings*, vol. 91, no. 8, pp. 1199–1209, Aug. 2003.

[12] H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yao, "Target classification and localization in habitat monitoring," in *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing. (ICASSP '03)*, 2003, pp. IV_844–IV_847.

[13] K. Whitehouse and D. Culler, "Calibration as parameter estimation in sensor networks," in *Proceedings of the 2002 ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA., September 2002.

[14] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, "A collaborative approach to in-place sensor calibration," in *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03), also in volume 2634 of Lecture Notes in Computer Science*. Springer-Verlag, 2003, pp. 301–316.

[15] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. (ICASSP '01)*, May 2001, pp. 2033–2036.

[16] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," in *Proc. of the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, San Jose, Costa Rica, April 2001.

[17] G. S. Sukhatme, D. Estrin, D. Caron, M. Mataric, and A. Requicha, "Proposed approach for combining distributed sensing, robotic sampling, and offline analysis for in situ marine monitoring," in *In Proceedings of the Advanced Environmental and Chemical Sensing Technology- SPIE 2000, Vol. 4205*. Boston, USA: SPIE, November 2000.

[18] M. Haenggi, "Mobile sensor-actuator networks: opportunities and challenges," in *Proceedings of the 7th IEEE International Workshop on Cellular Neural Networks and Their Applications, 2002. (CNNA 2002)*, July 2002, pp. 283 –290.

[19] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry, "Distributed control applications within sensor networks," *IEEE Proceedings*, vol. 91, no. 8, pp. 1235–1249, Aug. 2003.

[20] 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.

[21] Y. Chen, K. L. Moore, and Z. Song, "Diffusion boundary determination and zone control via mobile actuator-sensor networks (masnet): Challenges and opportunities," in *SPIE Proceedings of Intelligent Computing: Theory and Applications II, part of SPIE's Defense and Security*, Orlando, FL, Apr. 2004.

[22] 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.

[23] "MATLAB," http://www.mathworks.com/.

[24] "Femlab," http://www.comsol.com/.

[25] "Nastran," http://www.nenastran.com.

[26] "ANSYS," http://www.ansys.com.

[27] Y. Chen, Z. Wang, and J. Liang, "Optimal dynamic actuator location in distributed feedback control of a diffusion process," in *Submitted to IEEE Int. Conf. on Decision and Control (CDC/ECC'05)*, 2005.

[28] ——, "Automatic dynamic flocking in mobile actuator sensor networks by central voronoi tessellations," in *Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation (ICMA05)*. Niagara Falls, Ontario, Canada: IEEE, July 29 – August 1 2005.

[29] ——, "Actuation scheduling in mobile actuator networks for spatial-temporal feedback control of a diffusion process with dynamic obstacle avoidance," in *Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation (ICMA05)*. Niagara Falls, Ontario, Canada: IEEE, July 29 – August 1 2005.