From Algorithms to Architectures in Cyber-Physical Networks

Magnus Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology

Abstract

In this paper, we provide a formalism for Cyber-Physical Networks (CPN) that explicitly calls out the dynamic and the computational aspects of such networks, thus allowing for questions concerning algorithms and architectures to be asked in a systematic manner. The developments are motivated by a power-balancing example over the power grid, and a number of general, CPN-relevant questions are posed using the proposed formalism.

1 Introduction

Cyber-physical systems (CPS) are, at their core, characterized by fundamentally different models of computation. On the physical side, the Laws of Physics apply, i.e., differential equations describe the dynamics of the systems. On the cyber side, discrete models dictate the evolution of the computations. The result is a hybrid dynamic system, and, by now, a rich body of work exists for characterizing, modeling, designing, and analyzing such systems, thus providing a general model for CPS. (For a representative sample, see [2, 3, 16, 27, 29] and references therein.)

However, one aspect of CPS that has not yet received the same *systematic* treatment[1], is the fact that such systems are oftentimes interconnected, e.g., as is the case in power grids, precision agriculture infrastructure, smart building controls, and mobile sensor and communication networks, just to name a few, [1, 8, 10, 12, 14, 18, 22, 28, 30]. There certainly is a vast literature on networked systems in terms of coordinated controls, e.g., [4, 7, 21, 26], but a formalism that explicitly calls out the *cyber* and the *physical* aspects of such networks has been somewhat absent. The purpose of this paper is by no means to provide the ultimate answer to how such a formalism should be constructed. Instead, we simply highlight some key features of such networks, where physical interconnections between physical nodes have to co-exist with an overlaid computational, information-exchange network, thus creating a network (or really a network of networks) that also must be characterized by different computational models. The discussion will be motivated by power-grid examples, and we call the resulting networks *Cyber-Physical Networks*, or CPN. As such, this short paper should be understood as a small step towards characterizing CPN, as opposed to a complete treatment of the subject; such a treatment does not yet exist.

2 Power Balancing on the Grid

As a first installation of what we mean by a CPN, and what special considerations must be taken when considering such networks, consider a highly idealized problem on the power grid. As renewable resources drop in cost and approach price parity with fossil power, intermittent sources will become a larger part of total generation. Additionally, power generation will be more distributed, with residential customers more frequently having generation capacity. As this shift occurs, the lines between producer and consumer become less clear [15].

[1] There is indeed a rich literature on example classes within the networked CPS domain, but we do not yet have something akin to a general hybrid automata formalism for such systems.

In light of this shift towards distributed decision-making on the grid, one can model the power grid as a collection of connected agents using a DC power flow model. Let the *physical* grid network be given by an undirected graph $G_P = (V, E_P))$, where $V = \{1, \dots, N\}$ is the set of agents on the grid, and E_P is the set of physical tie lines. Moreover, assume for simplicity that each agent would like to receive $\rho_i,\ i = 1, \dots, N$ (watts) during an upcoming time period, with a negative ρ_i meaning a net production as opposed to a net consumption.

One problem could then be to try to decide what agent should be producing what, in order to solve

$$\min_p \sum_{i=1}^{N} \|p_i - \rho_i\|^2,$$

where $p = [p_1, \dots, p_N]^T$ and $p_i,\ i = 1, \dots, N$ is the actual net power load (or generation) of agent i. The solution to this problem is obviously to simply set $p_i = \rho_i,\ i = 1, \dots, N$, but this is clearly nonsense. What gets consumed must ultimately be produced somewhere, which means that the power flow must be balanced, i.e.,

$$\sum_{i=1}^{N} p_i = 0.$$

Note that this is a *physical* constraint in that any solution to the operational problem of "Who should be producing what?" must respect this constraint. And, even though the steady-state DC-flow constraint is purely algebraic, this constraint must be replaced by a differential (or even algebraic-differential) if the grid dynamics is taken into account.

If we let r_{ij} be the net power flow from an agent j, adjacent to agent i, i.e.,

$$p_i = \sum_{(i,j)\in E_P} r_{ij},$$

then we can let q_i denote potentials (acting like phasor angles in the network) in the sense that

$$r_{ij} = q_j - q_i,$$

which in turn produces an unconstrained minimization problem that is equivalent to the previous, constrained problem

$$\min_q \sum_{i=1}^{N} \Big\| \sum_{(i,j)\in E_P} (q_j - q_i) - \rho_i \Big\|^2,$$

where $q = [q_1, \dots, q_N]^T$.

By letting L_P be the graph Laplacian associated with G_P (e.g., [21]) and setting $\rho = [\rho_1, \dots, \rho_N]^T$, the ensemble version of this problem becomes

$$\min_q \|L_P q - \rho\|^2.$$

The set of global minimizers can be obtained by setting the derivative of the cost equal to zero,

$$2L_P^2 q - 2L_P\rho = 0.$$

Note that these minimizers are not unique since, if $q^\star$ is a global minimizer, then so is $q^\star + \alpha\mathbf{1}$, for any $\alpha \in \mathbb{R}$, where $\mathbf{1} = [1, \dots, 1]^T$. However, the corresponding, optimal power level

$$p_i^\star = \sum_{(i,j)\in E_P} (q_i^\star - q_j^\star)$$

is indeed unique. But, finding the roots to the derivative is a centralized operation since it involves a direct matrix inversion. An alternative would be to update q using a descent search. In other words, if $q(k)$ is the q-value at iteration k, one update law could be

$$q(k+1) = q(k) - \gamma(k)(L_P^2 q(k) - L_P\rho).$$

If the step-size $\gamma(k)$ is chosen correctly and the network is connected, this will indeed converge to a global minimizer.

The point behind this entire computation – originally performed in more general setting in [25] – is the fact that the square of the Laplacian shows up in the update equation. To see what this means, consider the element-wise (node-level) version of the gradient descent update law

$$q_i(k+1) = q_i(k) - \gamma(k)\left[\sum_{(i,j)\in E_P}\left(\sum_{(i,\ell)\in E_P}(q_i(k)-q_\ell(k)) - \sum_{(j,s)\in E_P}(q_j(k)-q_s(k))\right) - \sum_{(i,j)\in E_P}(\rho_i-\rho_j)\right].$$

From here it is clear that in order for agent i to update its q-value, not only is access to its neighbors' information needed, but also its neighbors' neighbors'. In other words, the L_P^2-term means that the information required to perform this computation cannot be obtained through the topology encoded in G_P but through the square-graph G_P^2. What is at play here is a coupling of physics (through power balancing) and computational requirements. One way of phrasing this is that the super-imposed cyber network G_C must, at a minimum, be a subgraph of the square of the physical graph, i.e., $G_C \subseteq G_P^2$, if a gradient descent algorithm applied to the (least-square power-deviation problem) is to be employed. This, by no means, implies that the square of the physical graph is fundamentally the smallest information graph required to solve this problem, but it does indicate that the algorithmic choices made at the cyber-level have architectural implications. (Finding the actual, minimal information requirement is a highly interesting question that, as of yet, does not have a general solution.) And, in subsequent sections, we will leverage this example when producing a CPN formalism that supports architectural questions in an explicit manner.

3 Cyber-Physical Networks

A CPN is comprised of (at least) two interacting networks, G_P and G_C, where $G_P = V_P \times E_P$, with V_P being the set of physical nodes, and $E_P \subseteq V_P \times V_P$ encodes the existence of physical couplings between the nodes.[2] This is illustrated in Figure 1, where the physical network G_P is the lower network (darker vertices), i.e., in that example

$$\begin{aligned} V_P &= \{1_P, 2_P, 3_P, 4_P, 5_P\} \\ E_P &= \{(1_P,2_P),(2_P,1_P),(1_P,3_P),(2_P,3_P),(3_P,4_P),(4_P,2_P),(4_P,5_P),(5_P,4_P)\}. \end{aligned}$$

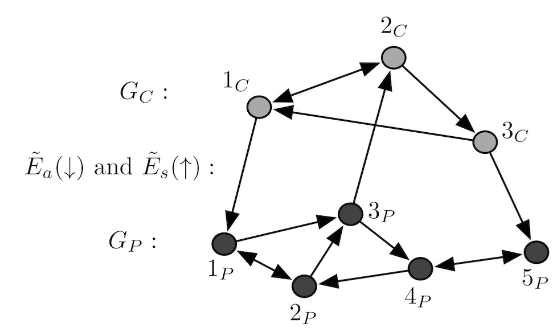

Figure 1: An example CPN comprised of constituent G_P, G_C, $\tilde{E}_a$, and $\tilde{E}_s$ components. The up-arrow in $\tilde{E}_s$ and the down-arrow in $\tilde{E}_a$ show the direction in which actuation (from cyber to physics) and sensing (from physics to cyber) act.

[2]In general, the graphs are directed, i.e., the edge-set is made up of ordered vertex-pairs.

The cyber-part of the network, $G_C = V_C \times E_C$, encodes the information flow among computational nodes, i.e., the edges in this graph denote communication channels between cyber agents – as opposed to dynamical coupling terms. In Figure 1, G_C is the upper graph (lighter vertices), with

$$\begin{array}{l} V_C = \{1_C, 2_C, 3_C\} \\ E_C = \{(1_C, 2_C), (2_C, 1_C), (2_C, 3_C), (3_C, 1_C)\}. \end{array}$$

The way these two networks come together to form a CPN, G_{CP}, is through the coupling between cyber-nodes and physical nodes. And, there are two distinctly different ways in which these two types of nodes can interact, namely through sensing and actuation. As such, we define two more edge sets, $\tilde{E}_a \subseteq V_C \times V_P$ and $\tilde{E}_s \subseteq V_P \times V_C$, where the subscripts denote *sensing* and *actuation*, respectively. The interpretation is that cyber-node i can influence (directly) physical node j if and only if $(i, j) \in \tilde{E}_a$, while it can sense physical node j if and only if $(j, i) \in \tilde{E}_s$. In Figure 1, these two edge sets are

$$\tilde{E}_a = \{(1_C, 1_P), (3_C, 5_P)\}, \quad \tilde{E}_s = \{(3_P, 2_C)\}.$$

The resulting CPN is obtained through the union of these constituent components, i.e.,[3]

$$G_{CP} = (V_P \cup V_C, E_P \cup E_C \cup \tilde{E}_a \cup \tilde{E}_s).$$

Returning to the power-balancing example, in this case

$$\begin{array}{l} V_P \simeq V_C \\ E_C = E_P \cup \{(i, j) \in V_P \times V_P \mid \exists k \in V_P \text{ s.t. } (i, k) \in E_P \text{ and } (k, j) \in E_P\} \\ \tilde{E}_a = \tilde{E}_s = \{(i, i) \mid i \in V_P\}. \end{array}$$

Now, associate a state $x_i,\ i = 1, \ldots, N_P$, $(|V_P| = N_P)$, with each physical node[4] and use $x_P = [x_1, \ldots, x_{N_P}]^T$ to denote the aggregate. Moreover, let $u_j,\ j = 1, \ldots, N_C$, $(|V_C| = N_C)$, be a decision variable/control signal associated with the cyber nodes, the physical constraints can be written on the form

$$\dot{x} = F(x, u), \quad G(x, u) = 0.$$

But, the differential coupling constraints must respect the sparsity pattern of the underlying network, since they encode pairwise dynamic couplings, and we denote this physical sparsity pattern by

$$F \in \text{sparse}_P(G_{CP}),$$

which means that the phsyical nodes can only "affect" each other directly if they form an edge in E_P, while the decision variables can only "affect" the physical node states if they form an edge in $\tilde{E}_a$. For example, if we assume that F is differentiable in both arguments, then the sparsity pattern can be encoded – following the construction in [9] – as

$$\begin{array}{l} F \in \text{sparse}_P(G_{CP}) \;\Leftrightarrow \\ \left((i, j) \notin E_P, i \neq j \Rightarrow \left[\frac{\partial F}{\partial x_i}\right]_j = 0,\ \forall (x, u) \right) \text{ and } \left((i, j) \notin \tilde{E}_a \Rightarrow \left[\frac{\partial F}{\partial u_i}\right]_j = 0,\ \forall (x, u) \right). \end{array}$$

Examples of such couplings are the Kuramoto coupled oscillator models [11, 17, 20] or the Bergen-Hill power exchange model [6], just to name a few.

If the dynamics were linear, this condition would correspond to

$$\dot{x} = Ax + Bu,$$

[3]If one explicitly wants to call out the different types of edges and vertices in G_{CP}, one can add a labeling function to the definition of the CPN.

[4]For the sake of notational simplicity, we assume these states to be scalar, without loss of generality.

with $x \in \mathbb{R}^{N_P}$, $u \in \mathbb{R}^{N_C}$, and the sparsity pattern meaning that $(i,j) \notin E_P \Rightarrow A_{ji} = 0$ as long as $i \neq j$, and $(i,j) \notin \tilde{E}_a \Rightarrow B_{ji} = 0$. Following the CPN in Figure 1, this means that

$$\dot{x} = \begin{bmatrix} A_{11} & A_{12} & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & A_{24} & 0 \\ A_{31} & A_{32} & A_{33} & 0 & 0 \\ 0 & 0 & A_{43} & A_{44} & A_{45} \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix} x + \begin{bmatrix} B_{11} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & B_{35} \end{bmatrix} u.$$

The algebraic constraints do not have to exhibit the same sparsity patterns since they are typically used to encode global constraints, such as limited resources, e.g., [23], or, as in the case with power balancing, a net constraint on the power produced and consumed in the network, i.e., if $x_i = q_i$ in the previous discussion,

$$\sum_{i=1}^{N_P} \sum_{(i,j)\in E_P} (x_i - x_j) = 0.$$

The key feature of the power-balancing example is that the algorithmic issues associated with the decision-making process must be taken into account as well. And, as the process operates in discrete time, say, in the case of synchronous updates, the decision process would update every Δt steps[5],

$$u((k+1)\Delta t) = f(x(k\Delta t), u(k\Delta)),$$

with the cyber sparsity pattern

$$f \in \text{sparse}_C(G_{CP})$$

that encodes the fact that j only has access to i's information if $(i,j) \in E_C$, i.e., i can talk to j over the cyber-network. Moreover, cyber-node j can only sense physical node j's state (or output) value if $(i,j) \in \tilde{E}_s$[6]. If we, once more, assume that f is differentiable, then this condition can be stated as

$$f \in \text{sparse}_C(G_{CP}) \;\Leftrightarrow$$
$$\left((i,j) \notin E_C, i \neq j \Rightarrow \left[\frac{\partial f}{\partial u_i}\right]_j = 0,\ \forall (x,u)\right) \text{ and } \left((i,j) \notin \tilde{E}_s \Rightarrow \left[\frac{\partial f}{\partial x_i}\right]_j = 0,\ \forall (x,u)\right).$$

In the case of linear systems with outputs, the last condition simply states that the C-matrix must have zeroes in locations where the corresponding cyber node can not sense the physical node. Using the example in Figure 1, the measurement equation becomes

$$y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{32} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} x.$$

Putting all of this together gives a hybrid system that can be written on impulsive form[7] as

$$\text{CPN model:} \begin{cases} \text{dynamics:} & \begin{cases} \dot{x}(t) = F(x(t), u(t)) \\ \dot{\tilde{x}}(t) = 0 \\ \dot{u}(t) = 0 \end{cases} \\ & \\ \text{computation:} & \begin{cases} x(k\Delta t^+) = x(k\Delta t^-) \\ \tilde{x}(k\Delta t^+) = x(k\Delta t^-) \\ u(k\Delta t^+) = f(\tilde{x}(k\Delta t^-), u(k\Delta t^-)), \end{cases} \end{cases}$$

where $\tilde{x}(t)$ contains the latest, sampled state value.

[5]It is possible to construct a similar, asynchronous model, but the formalism becomes significantly more involved.

[6]It should be noted that even though the formulation of f is as a memory-less function, one can augment this to include observer-like structures. However, it is not, in general, clear what part of the estimated state (physics) should be associated with what cyber node unless they are identical or each cyber node has its own estimate of the full state.

[7]This formulation boils down to a zero-order-hold implementation of a digital controller but more elaborate hold mechanisms can be employed as well.

4 Conclusions: From Algorithms to Architectures?

The model in the previous section attempts to capture ways in which a physical network interacts with a cyber-network through actuators and sensors. The dynamic coupling constraints as well as the physical interaction network are typically given *a priori* since the Laws of Physics are what they are, and the design task is to construct effective ways of controlling and coordinating such networks, i.e., design the cyber part. And, the "standard" question pursued in the literature is one of designing algorithms with provable performance properties. If we, abstractly, call such a property $\mathcal{P}$, we would have:

Problem 1: Given (G_{CP}, F), find f such that $\mathcal{P}$ is satisfied.

As an example, the many variations to the consensus problem fall squarely under this formulation (see [26] and the references therein), with the added restriction that $G_P = G_C$, i.e., no distinction is made between physical and cyber agents since, for example, in multi-robot networks, they are indeed the same. A variation and extension to this question that has been proposed involves the issue of what actuators and sensors are really needed to solve the problem, i.e., the way G_C interacts with G_P is part of the design process as well:

Problem 2: Given (G_C, G_P, F), find $(\tilde{E}_a, \tilde{E}_s, f)$ such that $\mathcal{P}$ is satisfied.

The literature on leader-selection mechanisms in multi-agent networks, as well as the sensor placement problem in spatially distributed systems are manifestations of this problem, e.g., [5, 13, 19, 24]. But, one can take this question in a slightly different direction and ask, as was done in the power-balancing example, what flow of information is required in G_C in order to make a given algorithm f effective:

Problem 3: Given (G_P, F, f), find $(G_C, \tilde{E}_a, \tilde{E}_s)$ such that $\mathcal{P}$ is satisfied.

Or, why stop there? The full-fledged problem simply starts with the physics and super-imposes a cyber-structure as well as an algorithm on top of it:

Problem 4: Given (G_P, F), find $(G_C, \tilde{E}_a, \tilde{E}_s, f)$ such that $\mathcal{P}$ is satisfied.

These problems have both an algorithmic (f) and an architectural (G_{CP}) aspect to them, with the architectural aspects becoming more pronounced as one goes down the list of problems. What this means is really that we have only begun to scratch the surface of CPN, and significant work remains to be done in order to fully harness their expected utilities. And, in this paper, we propose a possible formalism for describing such networks in a systematic manner. In fact, this paper should by no means be understood as providing a whole host of answers to questions related to CPN. Instead, it should be thought of as providing an organizing principle under which a number of different problem classes can be characterized, modeled, and (hopefully) successfully addressed.

References

[1] I. F. Akyildiz, E. Cayirci, W. Su and Y. Sankarasubramaniam. A Survey on Sensor Networks, *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, Aug. 2002.

[2] R. Alur and D.L. Dill. A Theory of Timed Automata, *Theoretical Computer Science*, Vol. 126, pp. 183-235, 1994.

[3] P.J Antsaklis (Ed.). Special Issue on Hybrid Systems: Theory and Applications, *Proceedings of the IEEE*, Vol. 88, No.7, pp.879-887, Jul 2000.

[4] H. Bai, M. Arcak, and J. Wen. *Cooperative Control Design: A Systematic, Passivity-Based Approach*, Springer, Berlin, 2011.

[5] B. Bamieh, M. R. Jovanovic, P. Mitra, and S. Patterson. Coherence in Large-Scale Networks: Dimension Dependent Limitations of Local Feedback. *IEEE Transactions on Automat. Control*, Vol. 57, No. 9, pp. 2235-2249, Sept. 2012.

[6] A. R. Bergen and D. J. Hill, A Structure Preserving Model for Power System Stability Analysis, *IEEE Transactions on Power Apparatus and Systems*, Vol. 100, No. 1, pp. 25-35, 1981.

[7] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*, Princeton University Press, Princeton, New Jersey, U.S.A., 2009.

[8] C. Y. Chong and S. P. Kumar. Sensor Networks: Evolution, Opportunities and Challenges, *Proceedings of the IEEE*, Vol. 91, No. 8, pp. 1247 - 1256, Aug. 2003.

[9] J.P. de la Croix and M. Egerstedt. A Control Lyapunov Function Approach to Human-Swarm Interactions, *American Control Conference*, Chicago, IL, June 2015.

[10] K. Deng, P. Barooah, and P. G. Mehta. Mean-Field Control For Energy Efficient Buildings. In *Proceedings of American Control Conference*, pp. 30443049, Montreal, Canada, 2012.

[11] F. Dorfler and F. Bullo. Synchronization and Transient Stability in Power Networks and Non-Uniform Kuramoto Oscillators. In *American Control Conference*, pp. 930937, 2010.

[12] G.N. Ericsson. Cyber Security and Power System Communication-Essential Parts of a Smart Grid Infrastructure, *IEEE Transactions on Power Delivery*, Vol. 25, No. 3, pp. 1501–1507, 2010.

[13] M. Fardad, F. Lin, and M. Jovanovic. Algorithms for Leader Selection in Large Dynamical Networks: Noise-Free Leaders. *IEEE Conference on Decision and Control and European Control Conference*, pp. 7188 7193, 2011.

[14] A. Foucquier, S. Robert, F. Suard, L. Stephan, and A. Jay. State of the Art in Building Modelling and Energy Performances Prediction: A Review. *Renewable and Sustainable Energy Reviews*, Vol. 23, pp. 272288, 2013.

[15] S. Grijalva and M.U. Tariq. Prosumer-Based Smart Grid Architecture Enables a Flat, Sustainable Electricity Industry. *Innovative Smart Grid Technologies (ISGT)*, pp. 1-6, Jan. 2011.

[16] T. A. Henzinger. The Theory of Hybrid Automata, *Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS)*, IEEE Computer Society Press, pp. 278-292, 1996.

[17] A. Jadbabaie, N. Motee, and M. Barahona. On the Stability of the Kuramoto Model of Coupled Nonlinear Oscillators. *American Control Conference*, pp. 4296-4301, 2004.

[18] H. Jaleel, A. Rahmani, and M. Egerstedt. Probabilistic Life Time Maximization of Sensor Networks. *IEEE Transactions on Automatic Control*, Vol. 58, No. 2, pp. 534-539, Feb. 2013.

[19] H. Kawashima and M. Egerstedt. Manipulability of Leader-Follower Networks Under a Rigid-Link Approximation. *Automatica*, Vol. 50, No. 3, pp. 695-796, March 2014.

[20] Y. Kuramoto. *Chemical Oscillations, Waves, and Turbulence*. Springer- Verlag, Berlin and New York, 1984.

[21] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, Princeton, New Jersey, U.S.A., 2010.

[22] M. Nazari, Z. Costello, J. Feizollahi, S. Grijalva, and M. Egerstedt. Distributed Frequency Control of Prosumer-Based Electric Energy Systems. *IEEE Transactions on Power Systems*, No. 99, pp. 1-9, March 2014.

[23] A. Nedic and A. Ozdaglar. On the Rate of Convergence of Distributed Subgradient Methods for Multiagent Optimization. *IEEE Conference on Decision and Control*, pp. 4711-4716, 2007.

[24] S. Patterson and B. Bamieh. Leader Selection for Optimal Network Coherence, *IEEE Conference on Decision and Control*, pp. 2692 2697, 2010.

[25] T. Ramachandran, Z. Costello, P. Kingston, M. Egerstedt, and S. Grijalva. Distributed Power Allocation in Prosumer Networks. *IFAC Workshop on Estimation and Control of Networked Systems*, Santa Barbara, CA, Sept. 2012.

[26] W. Ren and R. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*, Springer, Berlin, 2008.

[27] A.J. van der Schaft and J.M. Schumacher. An Introduction to Hybrid Dynamical Systems, *Lecture Notes in Control and Information Sciences*, Vol. 251, Springer-Verlag, London, 2000.

[28] A. Srinivasan. *Handbook of Precision Agriculture: Principles and Applications*, Food Products Press, New York, 2006.

[29] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*, Springer, 2009.

[30] Y. R. Tsai. Sensing Coverage for Randomly Distributed Wireless Sensor Networks in Shadowed Environments, *IEEE Transactions on Vehicular Technology*, Vol. 57, No. 1, pp. 556 - 564, Jan. 2008.