ISSUES IN
AUTONOMOUS MOBILE SENSOR NETWORKS

A Dissertation

by

AVINASH GOPAL DHARNE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2007

Major Subject: Mechanical Engineering

ISSUES IN
AUTONOMOUS MOBILE SENSOR NETWORKS

A Dissertation

by

AVINASH GOPAL DHARNE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Suhada Jayasuriya
Committee Members,	Alexander Parlos
	Shankar Bhattacharyya
	Jennifer Welch
Head of Department,	Dennis O'Neal

May 2007

Major Subject: Mechanical Engineering

ABSTRACT

Issues in

Autonomous Mobile Sensor Networks. (May 2007)

Avinash Gopal Dharne, B.E., SPCE, University of Mumbai;

M.S., Texas A&M University

Chair of Advisory Committee: Dr. Suhada Jayasuriya

Autonomous mobile sensor networks consist of a number of autonomous mobile robots equipped with various sensors and tasked with a common mission. This thesis considers the topology control of such an ad hoc mobile sensor network. In particular, I studied the problem of controlling the size, with respect to a distance metric, of the network for general interactive forcing among agents. Developed is a stability result, allowing one to design force laws to control the spread of the network. Many of the current results assume a known and/or fixed topology of the graph representing the communication between the nodes, i.e. the graph laplacian is assumed constant. They also assume fixed and known force-laws. Hence, the results are limited to time-invariant dynamics. The research considers stability analysis of sensor networks, unconstrained by specific forcing functions or algorithms, and communication topologies. Since the graph topologies are allowed to change as the agents move about, the system dynamics become discontinuous in nature. Filippov's calculus of differential equations with discontinuous right hand sides is used to formally characterize the multi-agent system with the above attributes. Lyapunov's Stability Theory, applied to discontinuous systems, is then used to derive bounds on the norm of the system states given bounds on its initial states and input.

The above derived stability results lend themselves to the derivation of methods for the design of algorithms or force-laws for mobile sensor networks. The efficacy of the derived results is illustrated through several examples where it is shown how they may be used for synthesizing a topology managing strategy. Examples are given of designing force-laws that limit the network in a desired area.

To My Mother and Father

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to Dr. Suhada Jayasuriya for his able guidance, advise and support as I worked on this project. I would like to thank my committee members, Dr. Alexander Parlos, Dr. Shankar Bhattacharyya and Dr. Jennifer Welch for their counsel and support. I would also like to thank my fiancée Shweta Parab for her support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

New technologies are radically reducing the size, cost, weight and power consumption of sensors and making it more and more feasible for sensor networks to be deployed in a cost-effective manner. Mobile sensor networks are sensory systems that combine sets of sensors on mobile platforms, along with spatially fixed (static) sensors, to form heterogeneous, dynamic, ad-hoc mobile sensor networks. Each node in a sensor network can be thought of as a mobile sensor platform consisting of:

1. a sensor suite to sense the environment,
2. a general purpose processor with limited computational capability,
3. some memory for storing data,
4. a low cost radio transceiver, and,
5. a battery power supply.

Sensor networks can be deployed to form a decentralized network, distributed over a spatial area, to provide real-time sensory information with the benefits of scalability, modularity and robustness [1, 2, 3, 4]. With the additional feature of mobility, such sensory networks accrue the benefits of a dynamic topology, improved robustness to environmental uncertainty and survivability to sensor failure [5]-[8]. A dynamic topology permits the sensor network to optimally position and align its sensors to improve sensing metrics such as error covariance and signal-to-noise ratio (SNR), perform distributed dynamic sensor fusion, all the while permitting fast response to dynamically changing environments and sensing

---

This dissertation follows the style of *IEEE Transactions on Automatic Control.*

assignments. Such a network can then be made autonomous giving it decision making capabilities based on the information its own sensors collect. In short, the network will operate independent of human intervention once it has been deployed. Such a network can perform estimation, detection and categorization for applications such as condition monitoring, surveillance, search and rescue, and emergency management [9]-[15].

For example, one application for mobile sensor networks is in offshore oil and gas exploration and condition monitoring. It is envisaged that a team of autonomous underwater vehicles, equipped with a sensor suite, would serve as a network, permitting them to map the seabed using ultrasonic sensors, monitor the condition of piping and oil platform infrastructure and provide real-time visual sensing of drilling and construction. Another, more dynamic and uncertain application, is emergency management. In the advent of a flood or earthquake affecting an urban area mobile sensor networks can be dispatched to provide detailed, real-time information on the environment. Yet another example of a network would be the 'Dominator' UAV network being envisaged by the USAF for surveillance, monitoring and engagement of targets on a cluttered battlefield [16]. A human in the loop provides initial directives to a group of agents regarding location, area of surveillance/mapping and data requested, and the sensor nodes then configure their operational mode depending on their required function. The mobile sensors share their sensed data within the mobile ad-hoc network, fusing and transmitting the information back to a command center or remote nodes and managing the flow of the sensed data. They react to their own network fused data and event detections, changing modes of operation, while accommodating directives from the command center.

Deployment of sensor networks ultimately depends on the ability to position sensor nodes so that each node is able to collect data from its surrounding environment in an optimal manner. There are situations where the sensor nodes will reach equilibrium with respect to their spatial distribution (topology), in which case the network nodes can essentially become stationary once such a configuration is reached (See Figure 1). It is also

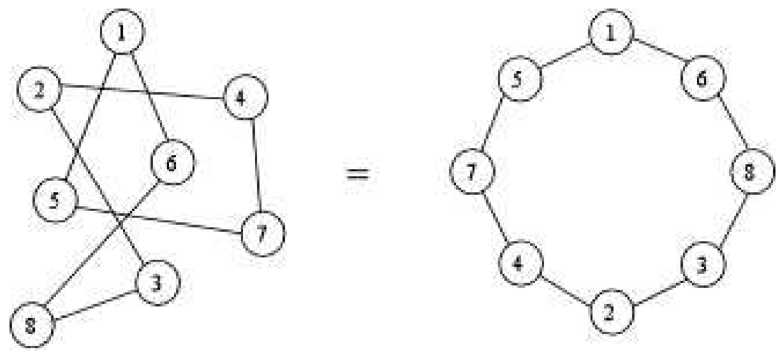

Fig. 1. Nodes move from their arbitrary initial positions to the desired topology

feasible and desirable to exploit the mobility of the agents to loiter around a final stable equilibrium state (topology) to improve the validity of sensed data gathered by its sensor suite. Mobility may also be necessitated by the sensing problem, such as in search and rescue and monitoring of a dynamic environmental event. It is important to recognize that the sensor nodes be distributed in the region of interest in an a priori uncontrolled and unorganized way to decrease the costs associated with infrastructure and to alleviate the need for careful global planning down to the final configuration the network is expected to reach.

To reach this final configuration nodes of an autonomous mobile sensor network exchange information with a small subset of nodes in the network, specifically those nodes which are within the range of its wireless communication hardware. Generally, in order to conserve power, this communication might be limited to the robots exchanging their position information only. Plus, in order to save power and money, the range of such hardware may be limited too. Thus the basic problem faced by a group of mobile nodes is how to propagate the network autonomously under communication constraints, i.e. how to use local information to decide one's motion so that a global goal is achieved. This is generally achieved by coming up with force-laws that translate the position and/or velocity information being exchanged by the robots into force commands for their actuators. [17] presents some force-laws which cause the sensor network to arrange itself in a pattern of hexagons or squares.

Such networks present unique and difficult issues, such as formation control [18, 19, 20, 21, 23, 22] and co-operative robotics [24, 25, 26], which need to be resolved, for them to function efficiently and effectively. Generally communication constraints are placed on the robots. Because of these, force-laws cease to be effective when the robots move a certain distance away from each other and the graph representing the communication between nodes keeps changing as the nodes move. Therefore, in addition to the final steady state of the sensor network, we need to have information about the transients too. If the transient response of the network of nodes is not very good, then there is a chance the nodes will lose communication with each other during propagation and the network might fail. Establishing the stability properties of the sensor network would allow us to determine its transient properties, its tracking performance and its ability to reject noise or disturbances and allow us to design a force-law we implement will perform as per expectations. Section 1 gives a more complete literature survey of the work done in this field and outlines the deficiencies in the current results. It also outlines what improvements we have attempted to make in order to advance the state-of-the-art.

Another issue of importance considered here is Localization. For most of the algorithms or force-laws referred above to work, robots need to be aware of their current location in the area of operation. We have done some work in this area which allows robots to localize and keep track of their location in a given environment. The motivation and background for this is detailed in Subsection 2.

A. Literature Survey

1. Network Modeling and Analysis

As mentioned above, the problem we wish to tackle is that of the modeling and analysis of a network of mobile sensor nodes. These nodes communicate with each other and decide their motion based on the information that is exchanged between them. How a robot uses the information at hand, i.e. how it converts the position information it has into force inputs to its actuators, depends on the mission of the network. If in general, the network is limited to a specific area, such as a network of robots used for search and rescue or surveillance and monitoring, the main focus of the algorithm is to ensure an even distribution of the nodes in the area and collision avoidance. Potential functions are often used for such applications [27, 28, 29]. These are generally of the form $F = \frac{k}{r^2}$, where r is distance between two nodes and k is a constant. These force-laws provide collision avoidance and are especially useful in environments with known obstacles. Coverage Control is another issue which has received attention [30, 31]. If the network of robots is expected to display a 'flocking' behavior, then in addition to the above, there also arise problems of agreement [32, 34, 33]. [32, 33] give results on average-consensus problem, i.e. the problem of nodes converging on a single point, for the case where the graph representing node communications is strongly connected and balanced. [35] gave results on a group of mobile robots achieving consensus in their heading angles by changing their heading at every time instant to the average of the heading angles of its neighbors. [36, 37] tackled multiple goals by using algorithms that have multiple terms each taking care of collision avoidance and agreement respectively.

There are two separate components which need to be considered when trying to model a network of mobile robots. One is the robot itself which has its own dynamics. The other is the communication in the network which gives the robot the information it needs to decide its next action. Graph theory can be used to represent the communication taking place between the nodes. The way these transmitted positions are used to determine force input to a robot gives rise to different kinds of models. Murray and Fax [38] modeled the mobile network as a graph. They assumed linear robot dynamics and linear feedback law governing the mapping of robot position differences to force input. However, the method requires simultaneous stabilization of multiple LTI systems and hence may not be scalable. More importantly, it assumes that the communication topology is fixed *a priori* and does not change. [39] extended this work by proving that an appropriate decentralized linear stabilizing feedback exists iff the communication graph has a rooted directed spanning tree. Cremean and Murray [40] attempted a stability analysis of robots modeled as nonlinear systems. However, this work is of very little applicability to mobile sensor networks because it assumed, among other things, asymptotic stability of individual robots which generally are only neutrally stable. Moreau [41] has tackled the problem of agreement in networks. i.e. all robots converge to a point in space. Moreau assumed a special function which governed the robot update rule. Specifically, he assumed that the new position of the robot at time $t+1$ is strictly inside the convex hull of the locations of that robot and its neighbors (in the sense of Graph theory). He also extended the results to time dependent communication links [42], ie. a changing topology. Angeli and Bliman [43] extended his result to include delayed robot measurements. They also relaxed the convexity condition but imposed others like bijectivity and Lipschitz conditions. Vicsek [44] applied Moreau's ideas to robot heading instead of position and showed that all robots finally 'agree' on a common heading. Morse [45] extended Vicsek's work by considering time delays in the system. Mesbahi [46, 47] gave some results on the agreement problems in Random Graphs. However, Random Graph theory may not be a good tool for analysis of mobile robot networks because inter-node connections in such networks are governed by deterministic criteria like the effective operating range of communicating hardware rather than by a fixed probability. Gazi and Passino [48] gave some stability results and state-vector norm bounds on complete networks, i.e a network where every node is connected to every other node, driven by a specific forcing function. Krishnaprasad and Leonard [18, 21, 23] model the network as robots having a constant speed and actuate their headings. They have developed control laws which cause the robots to follow a predetermined path or to follow each other in circles. Results on Mesh and string stability [49, 50] consider a string or network of agents with fixed and constrained communication and spatial topology and develop results indicating that a disturbance introduced in the system is damped out or dies down as it travels down the mesh or string. Recently [51] gave some consensus results on networks with time-delay under the assumption that the graphs representing communication have spanning trees.

Much of the work detailed above assumes fixed communication topology. Also, not all of it is scalable. In many cases, specific control laws under fixed and/or specific network topologies are analyzed. Most importantly, even where time-varying network topologies are considered, not much attention is given to the transient properties of the force-laws proposed. While most of the force-laws or algorithms proposed are effective in converging to the desired final state of the network, special attention needs to be paid to intermediate states of the network as well. This is so because if the transient states of the network are too large, they may lose connectivity with the rest of the network and thus fail. Such a study will allow the design of forcing functions which achieve desired final states while ensuring that the network does not fail in the process of reaching the final states. There is a need for a comprehensive stability analysis, unconstrained by specific forcing functions and topologies, to alleviate deficiencies and facilitate practical methods to analyze and design force-laws for mobile sensor networks. In this work, to begin with, we assume simple, neutrally stable, linear models for the robot dynamics. We model the force law of each node as a sum of two parts. The linear part represents the differences between node positions and can be incorporated into the state equations using the graph Laplacian. The second part is a possibly non-linear function of the position of the nodes which are communicating with the given node. We propose to assume, quite reasonably, that this function is norm-bounded by known bounds. We remove the condition of fixed topology and allow the graph edges to change with the positions of the nodes. This leads to a system of differential equations where the right-hand side is discontinuous. A. F. Filippov [52] has developed a calculus for solving such differential equations. Shevitz and Paden [53] have used this calculus to develop a Lyapunov stability theory for such systems. We use this result to prove a converse Lyapunov theorem guaranteeing the existence of a Lyapunov function for the system. This result is then used to develop bounds on the norm of the system states when the non-linear part of the force law is similarly bounded. This work can be used to design the parameters of the force-laws so that the network states satisfy required norm-bounds.

In summary, the current work on stability analysis of mobile sensor networks is limited by the assumptions of fixed communication topology and/or fixed force-laws. Some of it is also not scalable or based on hard to justify assumptions such as asymptotically stable agents or communication between random agents. We strive to alleviate these shortcomings and use the stability theory so developed to control the transient states of the network, exercise some control on the topology of the network and design force-laws satisfying performance specifications.

2. Robot Localization Using Fuzzy Logic

As is evident from the previous sections, the operation of a mobile sensor network can be enhanced by location awareness in individual mobile sensor nodes. This information is used for tasks such as topology control, collision avoidance and development of routing protocols. Security of routing algorithms can also be enhanced if location information is available [55].

Localization can be sub-divided into the problems of global position estimation and local position tracking. Global position estimation is the process of determining the position of the node in an *a priori* known map of the environment, given that the robot knows only the map and the fact that it is on it somewhere. This information can then be used to plan and navigate complex environments reliably. Once a node has been located on a map, local position tracking is the problem of keeping track of that position over time. This information is required for local manipulation tasks. Both these subsystems are essential for the functioning of a truly autonomous robot.

Extensive research has been done on localization for wireless networks. A general survey is found in [56]. Here, only localization techniques suitable for mobile ad hoc sensor networks are discussed. The approaches taken to achieve localization in sensor networks differ in their assumptions about the network deployment and the hardware's capabilities.

Centralized location techniques involve the sensor nodes transmitting data to a central machine which computes the location of the nodes [57, 58]. However these approaches require large amounts of communication between nodes and a central computer. In addition, centralized computation of position is difficult, undesirable and incompatible with the basic philosophy of a mobile ad-hoc sensor network. Hence distributed localization methods are concentrated upon.

Two important methodologies of distributed localization are range-based and range-free localization. Range-based localization [59, 60] are hardware intensive methods which localize a robot using such techniques as time of arrival, received signal strength, time difference of arrival of two different signals (TDOA), and angle of arrival (AOA). These schemes need special or expensive hardware.

Range-free schemes for mobile robots are mainly probabilistic estimation schemes. If both the robot motion and sensor measurement models are assumed to be Gaussian, then a neat Kalman Filter [61] can be designed. These filters provide a robust and accurate localization scheme but provide the best results only when the probability distributions are Gaussian. Introduced in [62] is a grid based Markov localization approach which eliminated the restriction on the probability density function. However, this approach is computationally intensive. Hence Monte Carlo simulations have been used to get around this problem [63, 64].

In this work, we use a grid based approach in which a node's location is represented by its belief or confidence that it stands at a certain point in this grid. Various sensor measurements carried out by the node may provide it with their confidence in its location at each grid point. This confidence function would be constructed a-priori based on the known accuracy and precision of the sensor. The node will typically have multiple sensors on it which will give it their independent confidence functions of its location. The node can then combine all of these with its own confidence in its location to obtain its new confidence in its location. This combination will be done using Fuzzy Logic and heuristic rules [65] instead of strict probabilistic rules. This approach has a few advantages over the previous work. First, it is computationally less intense while being unconstrained by any specific probability distribution function. Second, by setting the heuristic rules in a proper manner, we can tackle the possibility that a minority of the multiple sensors are not providing accurate information to the robot. The approach is intuitive and easy to program. The theoretical and experimental work done here is detailed in [66].

B. Structure

This document is structured as follows. Chapter II first presents mathematical preliminaries and background information on the topics covered in this thesis, namely, Graph Theory and Differential Equations with discontinuous right-hand sides. Then it presents the work and results pertaining to the modeling and analysis of Sensor Networks. Chapter III outlines some examples where the above results are applied to design of force laws for networks. Chapter IV details the theoretical and experimental results obtained for Robot Localization. Chapter V presents the conclusions and suggests future directions of research.

CHAPTER II

MODELING AND CONTROL OF AUTONOMOUS MOBILE SENSOR NETWORKS

A. Mathematical Preliminaries

1. Graph Theory Preliminaries

A *graph* $\mathcal{G}$ consists of a set of unique vertices, denoted $\mathcal{V}$, and a set of edges, denoted $\mathcal{E}$. Each element of the set $\mathcal{E}$ connects two distinct elements of the set $\mathcal{V}$, meaning that the graph has no self-loops. We also assume that each element of $\mathcal{E}$ is unique. Elements of the set $\mathcal{E}$, and hence the graph they define, can be *directed* or *undirected*. In our work, the graphs we define are always undirected. If every possible edge between all possible pairs of the elements of $\mathcal{V}$ exists, the graph is said to be *complete*.

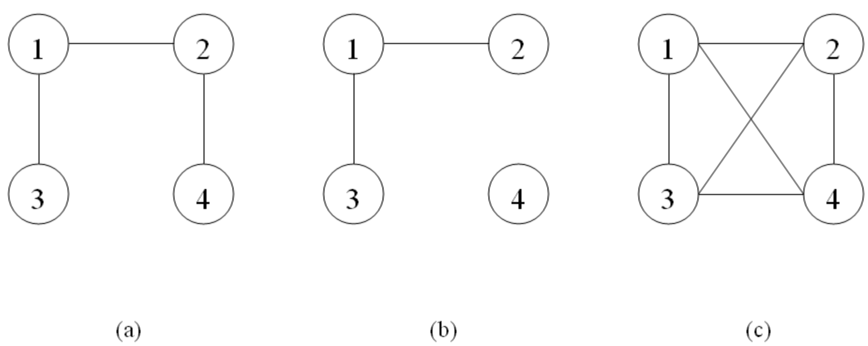

Fig. 2. (a)Connected graph (b)Disconnected graph (c)Complete graph.

A *path* on $\mathcal{G}$ of length N from v_0 to v_N is an ordered set of distinct vertices $[v_0 \ldots v_N]$ such that $(v_{i-1}, v_i) \in \mathcal{E} \;\forall\; i \in [i, N]$. A graph in which a path exists from every vertex to every other vertex is said to be *connected*. A graph in which disjoint subsets of vertices exist that cannot be joined by any path is termed *disconnected*. An *N-cycle* on $\mathcal{G}$ is a path for which $v_0 = v_N$. A graph without cycles is said to be *acyclic*. A graph with the property that the set of all cycle lengths has a common divisor $k > 1$ is said to be *k-periodic*. The relationship between graph theory and control theory makes use of matrices associated with a graph. For the purpose of defining these matrices, we assume that the vertices of $\mathcal{G}$ are enumerated, and each is denoted v_i. Given a graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$, the *Adjacency Matrix Ad* is defined as,

$$(ad)_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Since by convention, no self loops are allowed, i.e. $(ad)_{ii} = 0 \; \forall \; i$. The graph *Laplacian* is then defined as

$$L = D - Ad, \tag{2.2}$$

where D is a diagonal matrix whose i^{th} diagonal entry is the number of neighbors of v_i, i.e. the number of other vertices it is connected to. For the graph shown in Figure 2(a), the Laplacian will be

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{2.3}$$

Note that the rows of L each sum to zero. The structure of the spectrum of L is also interesting. The following can be shown to be true by observing that the rows of L necessarily sum to zero.

1. Zero is an eigenvalue of L. The associated eigenvector is $\mathbf{1}^T$.

2. If $\mathcal{G}$ is connected, the zero eigenvalue of is simple.

3. If $\mathcal{G}$ is undirected, then all eigenvalues of L are real.

2. Differential Equations with Discontinuous RHSs

In this section we review the Filippov solution concept for differential equations with discontinuous right-hand sides.

Filippov Solutions: We consider the vector differential equation

$$\dot{x} = f(x,t) \tag{2.4}$$

where $f : R^n x R \rightarrow R^n$ is discontinuous but measurable and essentially locally bounded. We first define what it means to be a solution of this equation.

Definition 1 (Filippov) *A vector function $x(\cdot)$ is called a solution of [2.4] on $[t_0,t_1]$ if $x(\cdot)$ is absolutely continuous on $[t_0,t_1]$ and for almost all $t \in [t_0,t_1]$*

$$\dot{x} \in K[f](x,t) \tag{2.5}$$

where

$$K[f](x,t) \equiv \bigcap_{\delta>0} \bigcap_{\mu N \equiv 0} \overline{co} f(B(x,\delta) - N, t) \tag{2.6}$$

and $\bigcap_{\mu N \equiv 0}$ denotes the intersection over all sets N of Lebesgue measure zero. An equivalent definition is: there exists $N_f \subset R^n$, $\mu N_f = 0$ such that for all $N \subset R^n$, $\mu N = 0$

$$K[f](x,t) \equiv \overline{co} \, \{\lim f(x_i) | x_i \rightarrow x, x_i \notin N_f \cup N\}. \tag{2.7}$$

The content of Filippovs solution is that the tangent vector to a solution, where it exists, must lie in the convex closure of the limiting values of the vector field in progressively smaller neighborhoods around the solution point (see Figure 3). It is important in the above definition that we discard sets of measure zero. This technical detail allows solutions to be defined at points even where the vector field itself is not defined, such as at the interface of two regions in a piecewise defined vector field.

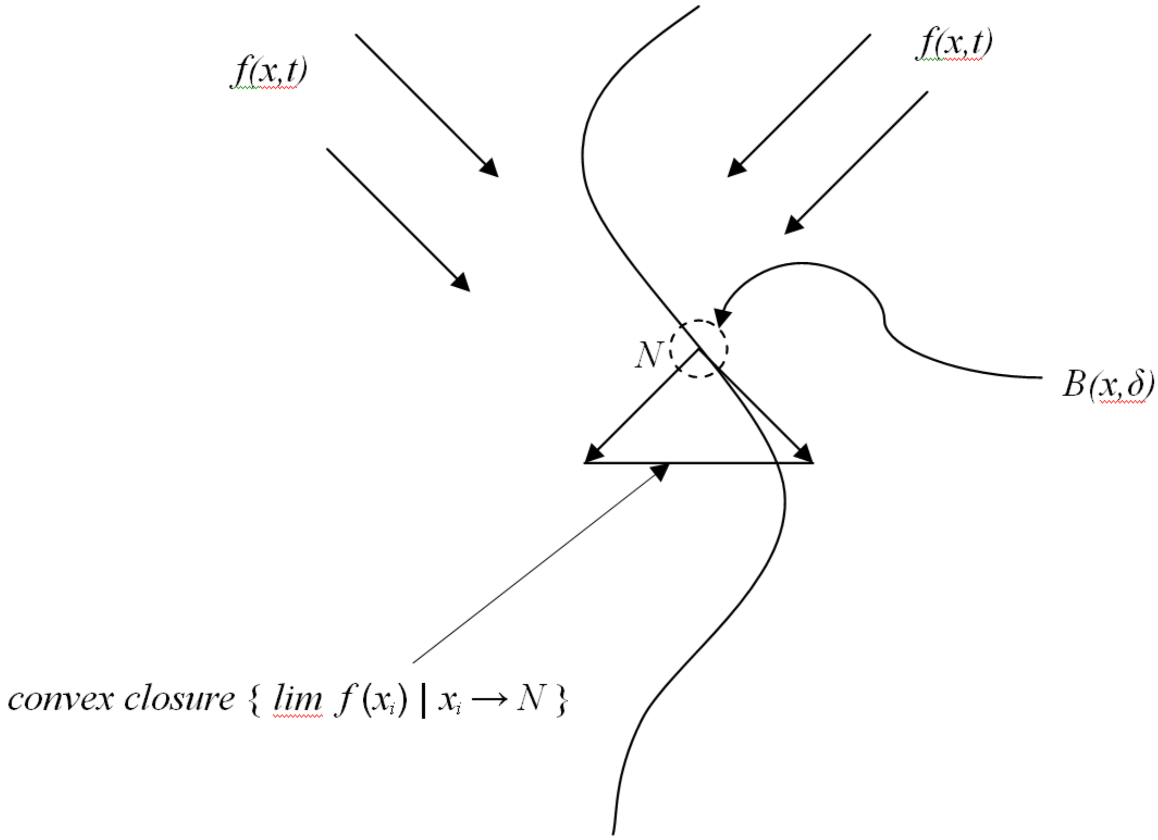

Fig. 3. Solutions of differential equations with discontinuous RHS

B. Main Results

We start our analysis by considering a network of mobile robots each having standard second order linear dynamics with viscous friction. The robots have wireless communication devices which can function to a distance of say r units. All robots weigh 1 unit and have identical dynamics and communication devices. We assume 2-dimensional $(x-y)$

space. Then for robot i,

$$\dot{\mathbf{X}}_i = \begin{bmatrix} \dot{x}_i \\ \ddot{x}_i \\ \dot{y}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -c & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -c \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \\ y_i \\ \dot{y}_i \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_{ix} \\ f_{iy} \end{bmatrix} = A\mathbf{X}_i + B\mathbf{U}_i. \tag{2.8}$$

Here c is the viscous friction constant and f_{ix} and f_{iy} is the force the robot exerts in the x and y direction respectively to propel itself. The robot "outputs", i.e. transmits its current position x_i and y_i. So,

$$\mathbf{Y}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_i = C\mathbf{X}_i. \tag{2.9}$$

Let $n_n =$ number of nodes in the network, and $n_y =$ number of outputs per node $= 2$. We are going to assume a certain structure for the forces f_{ix} and f_{iy}. In particular,

$$f_{ix} = \psi_{ix}(\mathbf{X}) - g \sum_{j \in \mathcal{N}_i} (x_i - x_j) = \psi_{ix}(\mathbf{X}) - g\delta_{xi} \tag{2.10}$$

and

$$f_{iy} = \psi_{iy}(\mathbf{X}) - g \sum_{j \in \mathcal{N}_i} (y_i - y_j) = \psi_{iy}(\mathbf{X}) - g\delta_{yi}. \tag{2.11}$$

In other words,

$$\mathbf{U}_i = \psi_i(\mathbf{X}) - g \sum_{j \in \mathcal{N}_i} (\mathbf{Y}_i - \mathbf{Y}_j) = \psi_i(\mathbf{X}) - g\delta_i. \tag{2.12}$$

Here $\mathbf{X}$ is the stacked state vector of the entire system, i.e. $\mathbf{X} = [\mathbf{X}_1{}^T \dots \mathbf{X}_{n_n}{}^T]^T$ and $\psi_i : \mathbb{R}^{4n_n} \to \mathbb{R}^2$ are possibly non-linear functions of $\mathbf{X}$. The summation is over all the j "neighbors" of node i, that is to say all the nodes in the network at a distance less than r units from node i. This structure of a force law is not very unusual. For example, consider a network where nodes behave as if they are attached by springs of stiffness g and unstretched length r with their neighbors. The force being computed by each node then is,

$$\begin{aligned} \mathbf{U}_i &= \sum_{j\in\mathcal{N}_i} \frac{g(r-d_j)}{d_{ij}}(\mathbf{Y}_i - \mathbf{Y}_j) && (2.13) \\ &= gr\sum_{j\in\mathcal{N}_i} \frac{1}{d_{ij}}(\mathbf{Y}_i - \mathbf{Y}_j) - g\sum_{j\in\mathcal{N}_i}(\mathbf{Y}_i - \mathbf{Y}_j) && (2.14) \\ &= \psi_i(\mathbf{X}) - g\sum_{j\in\mathcal{N}_i}(\mathbf{Y}_i - \mathbf{Y}_j), && (2.15) \end{aligned}$$

where d_{ij} = distance between nodes i and j = $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

Using Kronecker algebra notation and setting $\mathbf{U} = [\mathbf{U}_1{}^T \ldots \mathbf{U}_{n_n}{}^T]^T$ we get,

$$\dot{\mathbf{X}} = (I_{n_n} \otimes A)\mathbf{X} + (I_{n_n} \otimes B)\mathbf{U}. \quad (2.16)$$

But

$$\mathbf{U} = \Psi - g(L \otimes I_{n_y})(I_{n_n} \otimes C)\mathbf{X} \quad (2.17)$$

where, $\Psi = [\psi_1{}^T \ldots \psi_{n_n}{}^T]^T$ and L is the graph Laplacian defined above. Substituting equation (2.17) in (2.16), we get,

$$\dot{\mathbf{X}} = [(I_{n_n} \otimes A) - g(I_{n_n} \otimes B)(L \otimes I_{n_y})(I_{n_n} \otimes C)]\mathbf{X} + (I_{n_n} \otimes B)\Psi. \quad (2.18)$$

Using the property, $(P \otimes Q)(R \otimes S) = PR \otimes QS$ if the dimensions are appropriate and due to the fact that the number of inputs to each node equals the number of outputs, equation (2.18) can be simplified to

$$\dot{\mathbf{X}} = [(I_{n_n} \otimes A) - g(L \otimes BC)]\mathbf{X} + (I_{n_n} \otimes B)\Psi. \quad (2.19)$$

Now, consider the first part of equation 2.19,

$$\dot{\mathbf{X}} = [(I_{n_n} \otimes A) - g(L \otimes BC)]\mathbf{X} = f(\mathbf{X}). \quad (2.20)$$

Now consider the center of gravity (C.G) of the network of nodes. Since the dynamics in the y-direction are analogous to the ones in the x-direction, we only write out the x-dynamics of the system below. Then,

$$x_{cg} = \frac{1}{n_n}\sum_{i=1}^{n_n} x_i \tag{2.21}$$

$$\therefore \quad \dot{x}_{cg} = \frac{1}{n_n}\sum_{i=1}^{n_n} \dot{x}_i \tag{2.22}$$

$$\therefore \quad \ddot{x}_{cg} = \frac{1}{n_n}\sum_{i=1}^{n_n} \ddot{x}_i$$

$$\therefore \quad \ddot{x}_{cg} = \frac{1}{n_n}\sum_{i=1}^{n_n} (-c\dot{x}_i - g\delta x_i) \tag{2.23}$$

$$i.e. \quad \ddot{x}_{cg} = -c\dot{x}_{cg} - g\frac{1}{n_n}\sum_{i=1}^{n_n} \delta x_i. \tag{2.24}$$

The second term in equation (2.24) is zero. This is because the term $\sum_{i=1}^{n_n} \delta x_i$ is equivalent to $\mathbf{1}^T L\mathbf{x}$ and $\mathbf{1}$ is the eigenvector of the symmetric matrix L corresponding to the zero eigenvalue. Thus

$$\ddot{x}_{cg} = -c\dot{x}_{cg}. \tag{2.25}$$

Thus if the initial condition of the system is such that $\sum_{i=1}^{n_n} \dot{x}_i = 0$ and $\sum_{i=1}^{n_n} \dot{y}_i = 0$, then the C.G of the system is stationary. Equation (2.25) also shows that that the dynamics of the C.G are not subject to inter-node forces and the C.G of the system, if it is subject to a non zero initial velocity condition, will eventually come to rest.

Now if we define $\mathbf{Z} = [\ldots \quad z_{xi} \quad \dot{z}_{xi} \quad z_{yi} \quad \dot{z}_{yi} \quad \ldots]^T$ where

$$z_{xi} = x_i - x_{cg}, \quad \dot{z}_{xi} = \dot{x}_i - \dot{x}_{cg}, \quad z_{yi} = y_i - y_{cg}, \quad \dot{z}_{yi} = \dot{y}_i - \dot{y}_{cg}, \tag{2.26}$$

then we can substitute equation (2.25) into (2.20) to get,

$$\dot{\mathbf{Z}} = (I_{n_n} \otimes A)\mathbf{Z} - g(L \otimes BC)\mathbf{X}. \tag{2.27}$$

Now $\mathbf{Z} = \mathbf{X} - [\dots \;\; x_{cg} \;\; \dot{x}_{cg} \;\; y_{cg} \;\; \dot{y}_{cg} \;\; \dots]^T$. Hence,

$$(L \otimes BC)\mathbf{Z} = (L \otimes BC)\mathbf{X} - (L \otimes BC)[\dots \;\; x_{cg} \;\; \dot{x}_{cg} \;\; y_{cg} \;\; \dot{y}_{cg} \;\; \dots]^T.$$

The second term above is zero for the B and C matrices above and for any L because the $\mathbf{1}^T$ vector is an eigenvalue of L corresponding to the zero eigenvalue and the L matrix is being effectively multiplied by the vectors $[x_{cg} \dots x_{cg}]$ and $[y_{cg} \dots y_{cg}]$. Hence $(L \otimes BC)\mathbf{Z} = (L \otimes BC)\mathbf{X}$. So we can rewrite equation (2.27) as

$$\dot{\mathbf{Z}} = [(I_{n_n} \otimes A) - g(L \otimes BC)]\,\mathbf{Z} = f(\mathbf{Z}). \tag{2.28}$$

This equation looks like a linear dynamical equation. But the graph Laplacian is not constrained to be constant. It varies as a discontinuous function of the node positions and thus the right hand side of this equation is discontinuous. The set of points at which it changes value are ones where one or more pairs of nodes in the network are at a distance r from each other. Between such points, there is a continuum of points where the Laplacian is constant.

Comment 1 *Note that since there are a finite number of nodes, there are a finite number of possible graphs, connected or disconnected, and thus there is a finite number of points at which a change of Laplacian takes place. However in this work we assume a connected graph.*

Let Ω be the set of zero measure of points at which the L matrix switches value and Ω_l, $l = 1, 2, \dots, q$ be the continuous sets of non-zero measure in which the graph is constant with the Laplacian L_l. Then the absolutely continuous function $\mathbf{X}(t)$ is a Filippov solution [52] of equation (2.28) if it satisfies the differential inclusion,

$$\dot{\mathbf{Z}} \in K[f(\mathbf{Z})](\mathbf{Z}) \tag{2.29}$$

When the current state of the network $\mathbf{Z}(t) \in \Omega_l$ then $K[f(\mathbf{Z})](\mathbf{Z}) = f(\mathbf{Z})$. At the points of discontinuity, the derivative $\dot{\mathbf{Z}}$ lies in the convex hull of the points representing the limits of the value of $\dot{\mathbf{Z}}$ as $\mathbf{Z}$ approaches the point of discontinuity from various directions. So

$$\begin{aligned} K[f(\mathbf{Z})](\mathbf{Z}) &= \sum_{l=1}^{q} \alpha_l.(I_{n_n} \otimes A - gL_l \otimes BC)\mathbf{Z}, \qquad \alpha_l \geq 0, \sum_l \alpha_l = 1, \\ &= (I_{n_n} \otimes A)\mathbf{Z} - g\sum_{l=1}^{q} \alpha_l.(L_l \otimes BC)\mathbf{Z}, \\ &= \left[\dot{z}_{xi}, \, (-c\dot{z}_{xi} - g\sum_{l=1}^{q} \alpha_l \delta z_{xil}), \, \dot{z}_{yi}, \, (-c\dot{z}_{yi} - g\sum_{l=1}^{q} \alpha_l \delta z_{yil}) \right]^T, \end{aligned} \tag{2.30}$$

$i = 1, 2, \ldots, n_n$. At points $\mathbf{Z} \in \Omega_l$, $\alpha_l = 1$ and all other α_i's$= 0$. δz_{xil} and δz_{yil} are δz_{xi} and δz_{yi} corresponding to the graph represented by the Laplacian L_l.

We now state the Lyapunov Stability Theorem for systems with D.Es having discontinuous RHSs. The proof is identical to the one in [54] for the continuous case except for some relations holding 'almost everywhere' instead of everywhere. The theorem uses some results from [53].

Theorem 1 (Lyapunov) *Let $\dot{x} = f(x)$ be essentially locally bounded and $0 \in K[f(x)](0)$ in a domain $D \subset \mathbb{R}^n$ containing $x = 0$. Let $x(\cdot)$ be a Filippov solution of the above system. Let $V(x) : \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz, regular function. Then,*

1. *$V(x)$ is absolutely continuous, $(d/dt)V(x)$ exists almost everywhere and*

$$\frac{d}{dt}V(x) \ \in^{a.e} \ \dot{\tilde{V}}(x) \tag{2.31}$$

where

$$\dot{\tilde{V}}(x) \ := \bigcap_{\xi \in \partial V(x)} \xi^T K[f](x) \tag{2.32}$$

and $\partial V(x)$ is the generalized gradient of $V(x)$[1].

[1] *This part of the theorem is proved and the terms defined in [53].*

If in addition, the function $V(x)$ satisfies

$$V(0) = 0 \quad \textit{and} \quad V(x) > 0 \quad \textit{in}\, D - \{0\} \tag{2.33}$$

then

2. $\dot{\hat{V}}(x) \leq 0$ *in D implies $x = 0$ is a stable equilibrium point.*

3. $\dot{\hat{V}}(x) < 0$ *in $D - \{0\}$ implies x=0 is an asymptotically stable equilibrium point.*

Proof: See Khalil [54]. ■

Corollary 1 *The dynamic system repersented by equation (2.28) is stable.*

Proof: We choose the following Lyapunov function

$$V(\mathbf{Z}) = \frac{1}{2}\left(g\mathbf{z_x}^T L\mathbf{z_x} + g\mathbf{z_y}^T L\mathbf{z_y} + \dot{\mathbf{z}}_\mathbf{x}^T\dot{\mathbf{z}}_\mathbf{x} + \dot{\mathbf{z}}_\mathbf{y}^T\dot{\mathbf{z}}_\mathbf{y}\right), \tag{2.34}$$

where $\mathbf{z_x}$, $\dot{\mathbf{z}}_\mathbf{x}$, $\mathbf{z_y}$, $\dot{\mathbf{z}}_\mathbf{y}$ are the vectors in $\mathbb{R}^{n_n}$ representing the positions and velocities of the nodes with respect to the Center of Gravity. This function is defined on the domain $D = \mathbb{R}^{4n_n}$. Mathematically speaking, the matrix L representing a connected graph, has a single zero eigen-value whose eigen-space is spanned by the vector $\mathbf{1}$. Hence the Lyapunov function in equation (2.34) can have value zero at the non-zero state vector $\mathbf{Z} = [\mathbf{z_x} = \mathbf{a}^T \quad \mathbf{z_y} = \mathbf{b}^T \quad \dot{\mathbf{z}}_x = \mathbf{0}^T \quad \dot{\mathbf{z}}_y = \mathbf{0}^T]^T$. However, this value of the state vector, physically means that all the nodes are converged at one single point while the C.G of the system is at a point (a,b) units away from them. This is clearly impossible unless both a and b are zero. Hence the state-space for this problem, naturally excludes the eigen-space corresponding to the zero eigen-vector of the connected L matrix. Hence the Lyapunov Function chosen in equation (2.34) is positive definite. Also note that $V(\mathbf{Z})$ is a discontinuous function of $\mathbf{Z}$ because of the discontinuity of L. Hence we have to define the generalized gradient of

$V(\mathbf{Z})$ as

$$\partial V = \left[\ldots g\sum_{l=1}^{q}\alpha_l\delta z_{xil},\quad \dot{z}_{xi},\quad g\sum_{l=1}^{q}\alpha_l\delta z_{yil},\quad \dot{z}_{yi},\ldots\right]^T \qquad i=1,2,\ldots,n_n. \tag{2.35}$$

At points $\mathbf{Z}\in\Omega_l$, $\alpha_l=1$ and all other α_i's$=0$. Then, from equations (2.30), (2.32) and (2.35),

$$\begin{aligned}\dot{\tilde{V}}(\mathbf{Z}) &= \bigcap_{\xi\in\partial V(\mathbf{Z})}\xi^T K[f](\mathbf{Z})\\ &= \bigcap\sum_{i=1}^{n_n}\left[g\dot{z}_{xi}\sum_{l=1}^{q}\alpha_l\delta z_{xil} - c\dot{z}_{xi}^2 - g\dot{z}_{xi}\sum_{l=1}^{q}\beta_l\delta z_{xil} + g\dot{z}_{yi}\sum_{l=1}^{q}\alpha_l\delta z_{yil} - c\dot{z}_{yi}^2 - g\dot{z}_{yi}\sum_{l=1}^{q}\beta_l\delta z_{yil}\right]\end{aligned} \tag{2.36}$$

Now in the regions $\mathbf{Z}\in\Omega_l$, $\alpha_l=\beta_l=1$, all other α's and β's equal zero and the summation in equation (2.36) simplifies to $\sum_{i=1}^{n_n} -c\dot{x}_i{}^2 - c\dot{y}_i{}^2 \;\leq\; 0 \;\forall\; \mathbf{X}$. When $\mathbf{X}\in\Omega$, then the α's and β's can vary independently within their constraints. When the corresponding α's and β's match, the summation in equation (2.36) simplifies as above but at other times its value is indefinite. However, $\dot{\tilde{V}}(\mathbf{Z})$ is an intersection of all these sets and since we have some sets which are strictly non-positive, the intersection of all these sets is strictly non-positive. So

$$\dot{\tilde{V}}(\mathbf{Z}) \;\leq\; 0. \tag{2.37}$$

Hence using Theorem 1, the system is stable. ■

We now attempt to use the discontinuous version of the LaSalle's theorem from [53] to get a better idea of the equilibrium states of the system.

Theorem 2 (LaSalle) *Let Γ be a compact set such that every Filippov solution to the autonomous system $\dot{x}=f(x),\;\; x(0)=x(t_0)$ starting in Γ is unique and remains in Γ for all $t\geq t_0$. Let $V:\Gamma\rightarrow\mathbb{R}$ be a time independent regular function such that $v\leq 0$ for all $v\in\dot{\tilde{V}}$. Define $S=\{x\in\Gamma\mid 0\in\dot{\tilde{V}}\}$. Then every trajectory in Γ converges to the largest invariant set M, in the closure of S.*

Proof: See [53]. ■

Corollary 2 *The dynamic system represented by equation (2.28) is asymptotically stable about its Center of Gravity, i.e. around* $\mathbf{Z} = \mathbf{0}$.

Proof: Consider the function V defined in equation (2.34). This function is defined on a large enough ball Γ of $\mathbb{R}^{4n_n}$. This can be done because as stated above the final states of the system are bounded. From equation (2.37), we know that the set $S = \{\mathbf{Z} \mid \dot{\mathbf{z}}_x = \mathbf{0},\ \dot{\mathbf{z}}_y = \mathbf{0}\}$. The largest invariant set in S is $M = \{\mathbf{Z} \mid \mathbf{z_x} = \mathbf{0},\ \mathbf{z_y} = \mathbf{0},\ \dot{\mathbf{z}}_\mathbf{x} = \mathbf{0},\ \dot{\mathbf{z}}_\mathbf{y} = \mathbf{0}\}$. This is true because of our assumption of connectedness of the graph which implies that L has only a single zero eigen-value with corresponding eigenvector $\mathbf{1}$ and because as argued above, a and b have to be zero. Hence applying Theorem 2, we can conclude that all the nodes will converge to a single point in the 2-dimensional space, which will also be its center of gravity. The C.G has been shown to be always stable and bounded above. Hence, the system is asymptotically stable around its C.G. ■

Corollary 3 *For the dynamical system (2.28), there exists a positive constant a independent of* t_0 *and a class* $\mathcal{KL}$ *function* $\beta(.,.)$*, such that*

$$\|\mathbf{Z}(t)\| \leq \beta(\|\mathbf{Z}(t_0)\|, t - t_0), \quad \forall\ t \geq t_0 \geq 0, \ \forall\ \|\mathbf{Z}(t_0)\| < a. \tag{2.38}$$

Furthermore, this bound is of the form

$$\|\mathbf{Z}(t)\| \leq k\|\mathbf{Z}(t_0)\| e^{-\gamma(t-t_0)}, \quad \forall\ t \geq t_0 \geq 0, \ \forall\ \|\mathbf{Z}(t_0)\| < a. \tag{2.39}$$

Proof: The dynamical system (2.28) is asymptotically stable as shown above. Also, this stability is independent of t_0. Hence it is also uniformly asymptotically stable. In addition, the dynamics of the system are piecewise continuous in t and locally Lipschitz in $\mathbf{Z}$. Hence, using the definition of uniform asymptotic stability from [54], page 136, equation (2.38) holds. Furthermore the dynamics of the system are piecewise constant and linear. The states of the system are thus continuous, though not differentiable, exponential functions. Thus the equation (2.39) holds for some k and some γ. ■

We now focus on the second part of the forcing function which could be a non-linear, discontinuous function of the system states. This is considered to be a perturbation applied to the asymptotically stable system being considered till now. In this work, the input to the system is assumed to be force. Thus this perturbation vector is of the form

$$\Psi(t, \mathbf{Z}) = [\ldots \;\; 0 \;\; \psi_{xi} \;\; 0 \;\; \psi_{yi} \;\; \ldots].$$

The following two theorems state two sufficient conditions that ensure the stability of the perturbed system.

Theorem 3 *If the perturbation function* Ψ *satisfies the conditions*

$$\begin{aligned} |\psi_{xi}| &< c|\dot{z}_{xi}| \;\; \textit{and} \\ |\psi_{yi}| &< c|\dot{z}_{yi}| \;\; \forall\, i = 1, \ldots, n_n, \;\; \forall\, t > t_0 > 0, \end{aligned} \tag{2.40}$$

the system remains stable. Here c is the viscous friction constant of the nodes.

Proof: This proof derives all of its terminology from Corollary 1. The dynamic system is now changed by the addition of the perturbation term to the right hand side. We choose the Lyapunov function chosen in Corollary 1. In the regions $\mathbf{Z} \in \Omega_l$, $\alpha_l = \beta_l = 1$, all other α's and β's equal zero and the summation in equation (2.36) simplifies to

$$\sum_{i=1}^{n_n} -c\dot{z_{xi}}^2 - c\dot{z_{yi}}^2 + \dot{z_{xi}}\psi_{xi} + \dot{z_{yi}}\psi_{yi}. \tag{2.41}$$

Clearly, if the conditions of equation (2.40) are met, then this quantity is non-positive and as argued in Corollary 1, the system is stable. ■

Theorem 4 *If the perturbation function* Ψ *applies a force opposing the velocity of the nodes, the system will retain stability irrespective of the magnitude of the force.*

Proof: As seen from equation (2.41), if the condition of the theorem is met the quantity will be non-positive. Thus as argued in Corollary 1, the system is stable. ■

In order to get better results as to the effects of the perturbation function Ψ on the states of the system (2.28), we need to have a Lyapunov Function which has better properties than the one we have chosen. We now prove the existence of such a Lyapunov Function. Since we are in finite dimensional space, without loss of generality, we use the 2-norm in the following. We start with certain preliminaries regarding the unique nature of our system.

Let $\phi(\tau,t,\mathbf{Z})$ be the solution of the system (2.28) starting from $(t,\mathbf{Z})$. Our system dynamics are piecewise linear and constant functions of $\mathbf{Z}$ in t and locally Lipschitz in $\mathbf{Z}$. Each of these linear constant functions is defined by the Graph Laplacian L_i representing the connections between the nodes in the time interval t_i to t_{i+1}. Note that for finite number of nodes, there are a finite number of known graph laplacians. The function ϕ, as represented in Figure 4, is a continuous though not differentiable function of time. It is of the form,

$$\phi(\tau,t,\mathbf{Z}) = \begin{cases} \phi_0(\tau,t,\mathbf{Z}) & t \leq \tau < t_1 \\ \phi_1(\tau,t_1,\mathbf{Z}_1) & t_1 \leq \tau < t_2 \text{ and } \mathbf{Z}_1 = \mathbf{Z}(t_1) \\ \phi_2(\tau,t_2,\mathbf{Z}_2) & t_2 \leq \tau < t_3 \text{ and } \mathbf{Z}_2 = \mathbf{Z}(t_2) \\ \vdots \end{cases} \tag{2.42}$$

The $\phi_i's$ are the solutions of asymptotically stable LTI systems defined by equation (2.28) with corresponding $L_i's$, i.e. they are the solutions of asymptotically stable LTI systems

$$\dot{\mathbf{Z}} = F_i\mathbf{Z} \quad where \quad F_i = [(I_{n_n} \otimes A) - g(L_i \otimes BC)]\,. \tag{2.43}$$

Since there are a finite number of known $L_i's$, there exists a positive number H such that

$$|||F_i||| \leq H \qquad \forall \;\; i \tag{2.44}$$

The structure of the $\phi_i's$ is of the form

$$\phi_i(\tau, t_i, \mathbf{Z}_i) = \mathbf{Z}_i e^{F_i(\tau - t_i)}. \tag{2.45}$$

Thus

$$\begin{aligned} \frac{\partial \phi_i}{\partial \mathbf{Z}_i} &= e^{F_i(\tau - t_i)} \\ \therefore \left\| \frac{\partial \phi_i}{\partial \mathbf{Z}_i} \right\| &\leq e^{H(\tau - t_i)}. \end{aligned} \tag{2.46}$$

Also,

$$\|\phi(\tau, t, \mathbf{Z})\| \geq \|\mathbf{Z}\| \, e^{-H(\tau - t)} \tag{2.47}$$

With these preliminaries, we now state the following theorem.

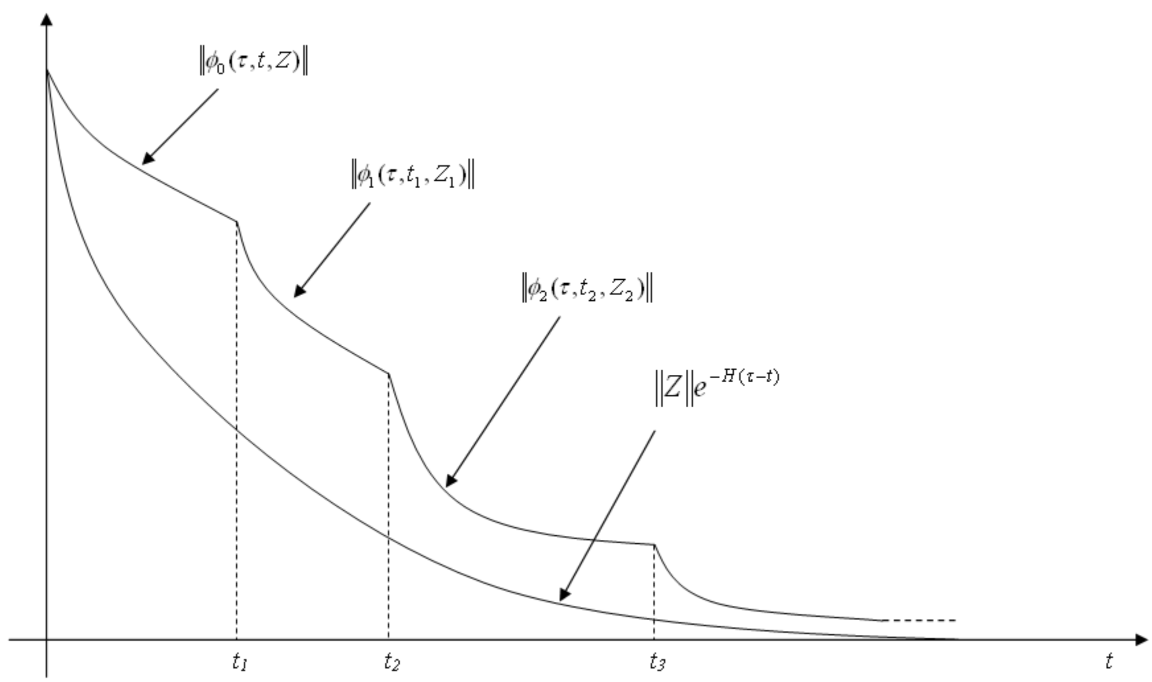

Fig. 4. $\|\phi\|$ as a function of time.

Theorem 5 *Let a be as defined in Corollary 3. Let $\mathcal{B}_a$ be the ball of radius a. There exists a function $V : [0,\infty)\times\mathcal{B}_a \to \mathbb{R}$ that satisfies the conditions,*

$$\alpha_1(\|\mathbf{Z}\|) \ \leq \ V(t,\mathbf{Z}) \ \leq \ \alpha_2(\|\mathbf{Z}\|), \tag{2.48}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{Z}} f(\mathbf{Z}) \ \leq \ -\alpha_3(\|\mathbf{Z}\|) \quad \textit{and} \tag{2.49}$$

$$\left\|\frac{\partial V}{\partial \mathbf{Z}}\right\| \ \leq \ \alpha_4(\|\mathbf{Z}\|) \tag{2.50}$$

where the α_i's are class $\mathcal{K}$ functions defined on $[0,a]$.

Proof: Consider a function $G(x)$, $x \geq 0$, such that $G' \geq 0$. Define

$$V(t,\mathbf{Z}) = \int_t^\infty G(\|\phi(\tau,t,\mathbf{Z})\|)\, d\tau \tag{2.51}$$

$$\therefore V(t,\mathbf{Z}) = \int_t^{t_1} G(\|\phi_0(\tau,t,\mathbf{Z})\|)\, d\tau + \int_{t_1}^{t_2} G(\|\phi_1(\tau,t_1,\mathbf{Z}_1)\|)\, d\tau + \int_{t_2}^{t_3} G(\|\phi_2(\tau,t_2,\mathbf{Z}_2)\|)\, d\tau + \dots \tag{2.52}$$

To talk about the existence of a suitable G, consider the derivative of V with respect to $\mathbf{Z}$.

$$\frac{\partial V}{\partial \mathbf{Z}} = \int_t^{t_1} G'(\|\phi_0\|_2)\frac{\phi_0^T}{\|\phi_0\|_2}\frac{\partial \phi_0}{\partial \mathbf{Z}}\, d\tau + \sum_i \int_{t_i}^{t_{i+1}} G'(\|\phi_i\|_2)\frac{\phi_i^T}{\|\phi_i\|_2}\frac{\partial \phi_i}{\partial \mathbf{Z}}\, d\tau \tag{2.53}$$

But as discussed above, ϕ_i's are functions of $\mathbf{Z}_i$. Thus $\frac{\partial \phi_i}{\partial \mathbf{Z}} = 0 \ \forall \ i \geq 1$. Hence,

$$\frac{\partial V}{\partial \mathbf{Z}} = \int_t^{t_1} G'(\|\phi_0\|_2)\frac{\phi_0^T}{\|\phi_0\|_2}\frac{\partial \phi_0}{\partial \mathbf{Z}}\, d\tau. \tag{2.54}$$

Therefore, from equation (2.46),

$$\begin{aligned}
\left\|\frac{\partial V}{\partial \mathbf{Z}}\right\|_2 &\leq \int_t^{t_1} G'(\|\phi_0\|_2) e^{H(\tau-t)}\, d\tau \\
&\leq \int_t^{\infty} G'(\|\phi_0\|_2) e^{H(\tau-t)}\, d\tau \quad \because\ G' \geq 0 \\
&\leq \int_t^{\infty} G'(\beta(\|\mathbf{Z}\|_2, \tau - t)) e^{H(\tau-t)}\, d\tau \\
&\leq \int_0^{\infty} G'(\beta(\|\mathbf{Z}\|_2, s)) e^{Hs}\, ds. \qquad (2.55)
\end{aligned}$$

At this point we state Massera's Lemma.

Lemma 1 (Massera) *Let* $g : [0,\infty) \rightarrow \mathbb{R}$ *be a positive, continuous, strictly decreasing function with* $g(t) \rightarrow 0$ *as* $t \rightarrow \infty$*. Let* $h : [0,\infty] \rightarrow \mathbb{R}$ *be a positive, continuous, nondecreasing function. Then, there exists a function* $G(t)$ *such that*

1. $G(t)$ *and its derivative* $G'(t)$ *are class* $\mathcal{K}$ *functions defined for all* $t \geq 0$.
2. *For any continuous function* $u(t)$ *which satisfies* $0 \leq u(t) \leq g(t)$ *for all* $t \geq 0$*, there exist positive constants* k_1 *and* k_2*, independent of u, such that*

$$\int_0^{\infty} G(u(t))\, dt \leq k_1 \quad \text{and} \quad \int_0^{\infty} G'(u(t))h(t)\, dt \leq k_2 \qquad (2.56)$$

Proof: See [54], Appendix A.6. ∎ From above Lemma, considering $\beta = g$ and $e^{Hs} = h$, we know that a function G with the required property exists and the integral in equation (2.55) exists. The integral is a class $\mathcal{K}$ function of $\|\mathbf{Z}\|_2$ since G' is a class $\mathcal{K}$ function and β is also a class $\mathcal{K}$ function in $\|\mathbf{Z}\|_2$. Thus V satisfies equation (2.50).

Now that the existence of V is proven, consider

$$\begin{aligned} V(t,\mathbf{Z}) &= \int_t^\infty G(\|\phi(\tau,t,\mathbf{Z})\|_2)\,d\tau \\ &\le \int_t^\infty G(\beta(\|\mathbf{Z}\|_2,\tau-t))\,d\tau \;=\; \int_0^\infty G(\beta(\|\mathbf{Z}\|_2,s))\,ds \;\stackrel{\text{def}}{=}\; \alpha_2(\|\mathbf{Z}\|_2) \end{aligned} \tag{2.57}$$

The existence of the integral is assured by Lemma 1. The function $\alpha_2(\cdot)$ is a class $\mathcal{K}$ function because, G is a class $\mathcal{K}$ function and β is a class $\mathcal{KL}$ function in $\|\mathbf{Z}\|_2$.

From equation (2.47),

$$\begin{aligned} V(t,\mathbf{Z}) &\ge \int_t^\infty G(\|\mathbf{Z}\|_2\,e^{-H(\tau-t)})\,d\tau \;=\; \int_0^\infty G(\|\mathbf{Z}\|_2\,e^{-Hs})\,ds \\ &\ge \int_0^{\ln(2)/H} G(\frac{1}{2}\,\|\mathbf{Z}\|_2)\,ds \;=\; \frac{\ln(2)}{H}G(\frac{1}{2}\,\|\mathbf{Z}\|_2) \;\stackrel{\text{def}}{=}\; \alpha_1(\|\mathbf{Z}\|_2) \end{aligned} \tag{2.58}$$

From equations (2.57) and (2.58), V satisfies equation (2.48).

To prove the final part,

$$\begin{aligned} \frac{\partial V}{\partial t}+\frac{\partial V}{\partial \mathbf{Z}}f(\mathbf{Z}) &= -G(\|\phi_0(t,t,\mathbf{Z})\|_2) \;+\int_t^{t_1} G'(\|\phi_0(\tau,t,\mathbf{Z})\|_2)\frac{\phi_0^T}{\|\phi_0\|_2}(\frac{\partial\phi_0}{\partial t}+\frac{\partial\phi_0}{\partial\mathbf{Z}}F_0\mathbf{Z})\,d\tau \\ &\quad + \sum_i \int_t^{t_1} G'(\|\phi_i(\tau,t_i,\mathbf{Z}_i)\|_2)\frac{\phi_i^T}{\|\phi_i\|_2}(\frac{\partial\phi_i}{\partial t}+\frac{\partial\phi_i}{\partial\mathbf{Z}}F_0\mathbf{Z})\,d\tau \end{aligned} \tag{2.59}$$

where F_0 represents the LTI system dynamics between the times t and t_1. The summation term is zero because $\frac{\partial\phi_i}{\partial t} = \frac{\partial\phi_i}{\partial\mathbf{Z}} = 0, \forall i \ge 1$. In the second term, due the asymptotically stable LTI nature of the dynamics, it is trivial to prove that $\frac{\partial\phi_0}{\partial t}+\frac{\partial\phi_0}{\partial\mathbf{Z}}F_0\mathbf{Z}$ is uniformly zero between t and t_1. Thus,

$$\frac{\partial V}{\partial t}+\frac{\partial V}{\partial \mathbf{Z}}f(\mathbf{Z}) \;=\; -G(\|\mathbf{Z}\|_2). \tag{2.60}$$

Since G is a class $\mathcal{K}$ function, V satisfies equation (2.49). ∎

Corollary 4 *Let the perturbation function $\Psi(t,\mathbf{Z})$ be piecewise continuous in t and locally*

Lipschitz in $\mathbf{Z}$*. Let a be as defined in Corollary 3. If* Ψ *satisfies the uniform bound*

$$\|\Psi(t,\mathbf{Z})\| \;\leq\; \delta \;<\; \frac{\theta\alpha_3(\alpha_2^{-1}(\alpha_1(a)))}{\alpha_4(a)} \tag{2.61}$$

for all $t > 0$*, all* $\mathbf{Z} \in \mathcal{B}_a$ *and some positive constant* $\theta < 1$*, then for all* $\|\mathbf{Z}(t_0)\| < \alpha_2^{-1}(\alpha_1(a))$*, the solution* $\mathbf{Z}(t)$ *of the perturbed system satisfies*

$$\|\mathbf{Z}(t)\| \leq \begin{cases} \beta(\|\mathbf{Z}(t_0)\|, t-t_0), & \forall\; t_0 \leq t < t_1 \\ \rho(\delta), & \forall\, t \geq t_1 \end{cases} \tag{2.62}$$

for some class $\mathcal{KL}$ *function* $\beta(\cdot,\cdot)$ *and some finite time* t_1*, where* $\rho(\delta)$ *is a class* $\mathcal{K}$ *function of* δ *defined by*

$$\rho(\delta) = \alpha_1^{-1}\left(\alpha_2\left(\alpha_3^{-1}\left(\frac{\delta\alpha_4(a)}{\theta}\right)\right)\right). \tag{2.63}$$

Proof: Since our unperturbed system is uniformly asymptotically stable around the origin, the corollary follows from Lemma 5.3 in [54].

CHAPTER III

APPLICATION OF RESULTS TO DESIGN OF FORCE-LAWS

In this chapter, we employ the results derived in the previous chapter to design control laws for networks. To do so we must derive the functions $\alpha_1, \ldots, \alpha_4$. To do so we note that the response of the linear system can be bounded as

$$\|\mathbf{Z}_0\| \, e^{-H(\tau-t)} \;\leq\; \|\phi(\tau,t,\mathbf{Z})\| \;\leq\; K\,\|\mathbf{Z}_0\| \, e^{-H_l(\tau-t)} \;\; \textit{and} \;\; \|\phi_{\mathbf{Z}}(\tau,t,\mathbf{Z})\| \;\leq\; e^{H(\tau-t)}. \quad (3.1)$$

If we assume

$$G(r) = r^{\frac{1+Q}{Q}} \quad \textit{where} \quad Q = \frac{H_L}{pH}, \;\; p > 1, \tag{3.2}$$

we trivially derive

$$\alpha_1(\|\mathbf{Z}\|) \;=\; \frac{Q}{H(1+Q)} \, \|\mathbf{Z}\|^{\frac{1+Q}{Q}}, \tag{3.3}$$

$$\alpha_2(\|\mathbf{Z}\|) \;=\; \frac{K^{\frac{1+Q}{Q}} Q}{H_L(1+Q)} \, \|\mathbf{Z}\|^{\frac{1+Q}{Q}}, \tag{3.4}$$

$$\alpha_3(\|\mathbf{Z}\|) \;=\; \|\mathbf{Z}\|^{\frac{1+Q}{Q}}, \tag{3.5}$$

$$\alpha_4(\|\mathbf{Z}\|) \;=\; \frac{1+Q}{Q} \frac{K^{\frac{1}{Q}}}{H(p-1)} \, \|\mathbf{Z}\|^{\frac{1}{Q}}. \tag{3.6}$$

We apply these results to three different force-laws. First we consider a force-law which acts like a spring of unstreched length d attached between neighboring nodes. As shown in chapter II the force in the x-direction is

$$F_{x_i} = -g \sum_j (x_i - x_j) + gd \sum_j \frac{x_i - x_j}{r_{i,j}}. \tag{3.7}$$

Our results allow us to determine limits on the second term (Ψ) so that the states of the system do not exceed design limits. We consider a network of three nodes to reduce computational costs. Starting from a initial condition of zero velocity and positions of $(0,0)$,

$(35,0)$, $(5,30)$, the response to the linear force can be bounded by the parameters $H = 0.53$, $H_L = 0.4$ & $K = 1.34$. Now, if we wish to limit our system states by $\|\mathbf{Z}(t)\|_\infty \leq 35$, we can use corollary 4 to calculate that if $\|\mathbf{Z}_0\|_\infty \leq 24.082$ and $\|\Psi\|_\infty \leq 0.613$ then the design limit can be maintained. As shown in the Figure 5 the network state norm is maintained when the

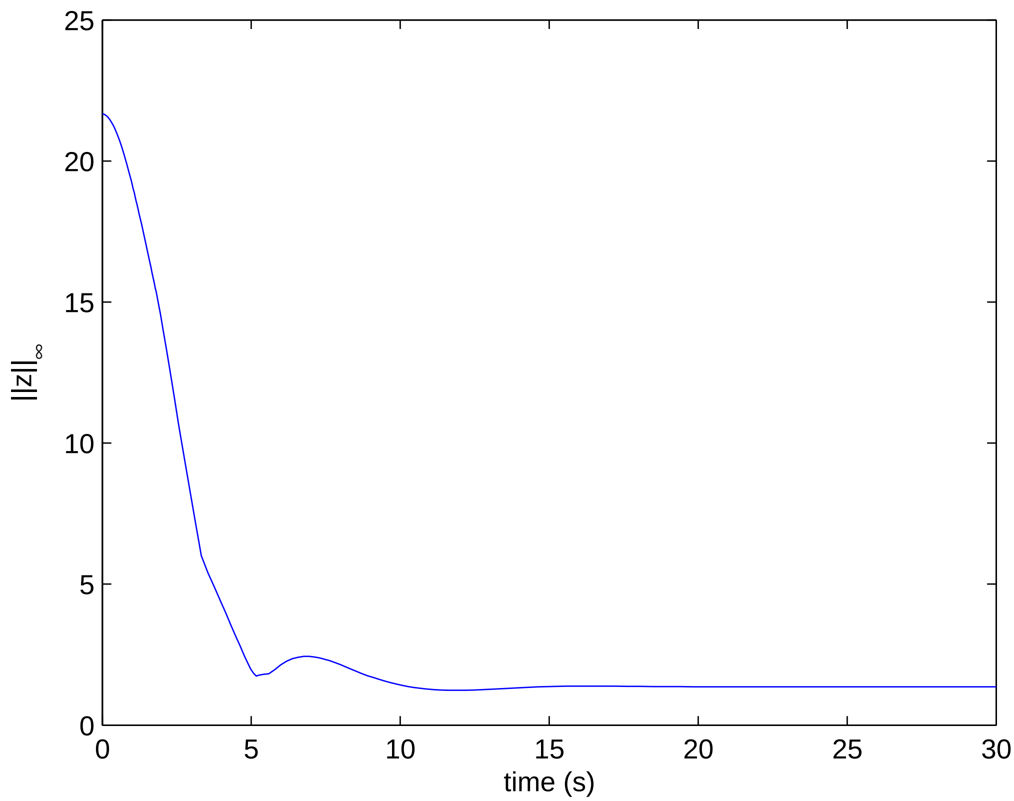

Fig. 5. State norm for system controlled by a spring-like force under constraints designed.

non-linear part is constrained by the limit calculated. The limit is admittedly conservative.

We use this method for another force-law. We use the one proposed by Tanner et. al. [36, 37]. They proposed a force-law consisting of a linear term and an non-linear term given by

$$F_{ij} = \frac{1}{r^2} - \frac{2\log r}{r}, \;\; r = \text{distance between nodes i and j.} \qquad (3.8)$$

As the Figure 6 shows, the state norm stays within limit.

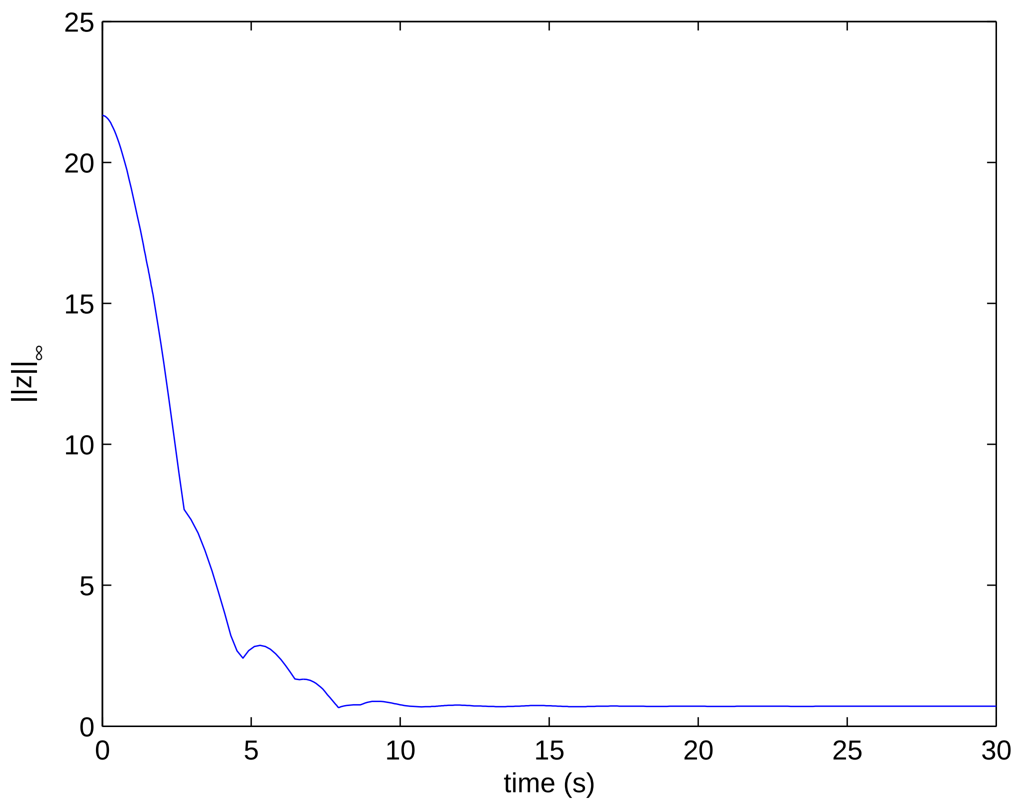

Fig. 6. State norm for system controlled by force proposed by Tanner et al.

We also test this method on a third force-law.

$$F_{ij} = -(3r^2 + 300r - 10000)/1500, \;\; r = \text{distance between nodes i and j.} \qquad (3.9)$$

$$\therefore F_{xij} = -0.2(x_i - x_j) + \left(\frac{10000 - 3r^2}{1500}\right)\left(\frac{x_i - x_j}{r}\right). \qquad (3.10)$$

Again, as the Figure 7 shows, the state norm stays within limit.

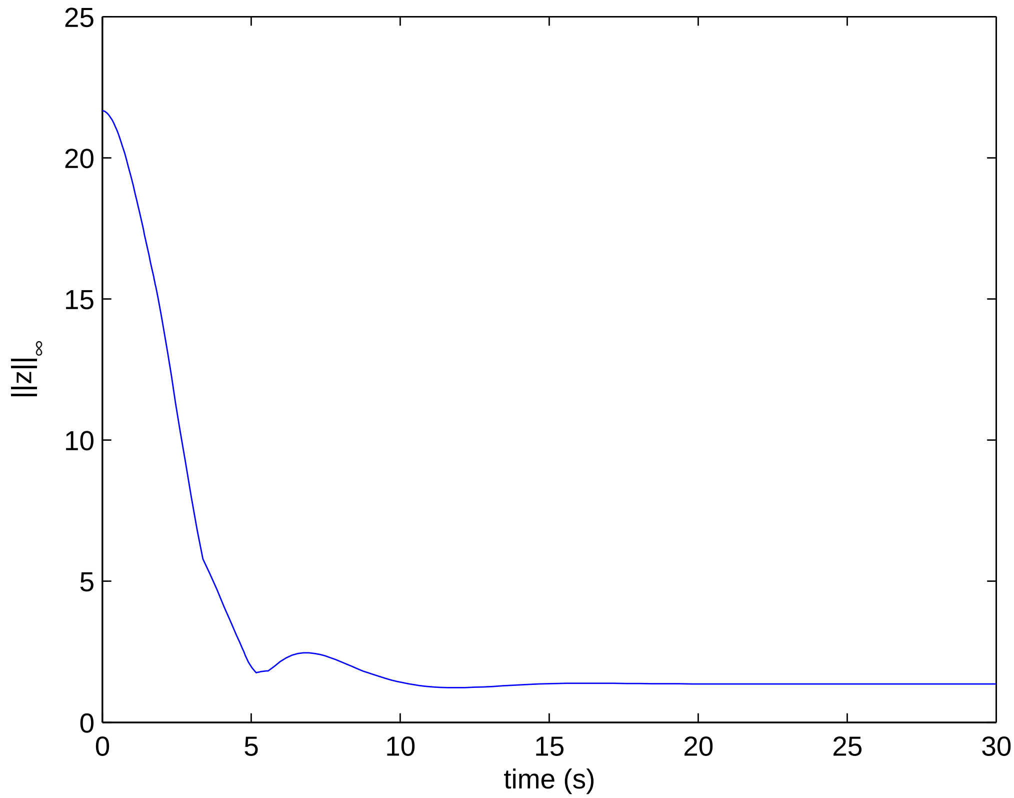

Fig. 7. State norm for system controlled by force proposed above.

We now also try to apply the method to a network of 6 nodes under the spring-like force. Inspite of the increased number of nodes, the method is not very difficult to implement. The linear response of the system can be bounded by the parameters $H = 0.53$, $H_L = 0.35$ & $K = 1.4$. We can then calculate that if $\|\mathbf{Z}_0\|_\infty \leq 33$ and $\|\Psi\|_\infty \leq 0.1$ then the design limit of $\|\mathbf{Z}(t)\|_\infty \leq 35$ can be maintained (Figure 8).

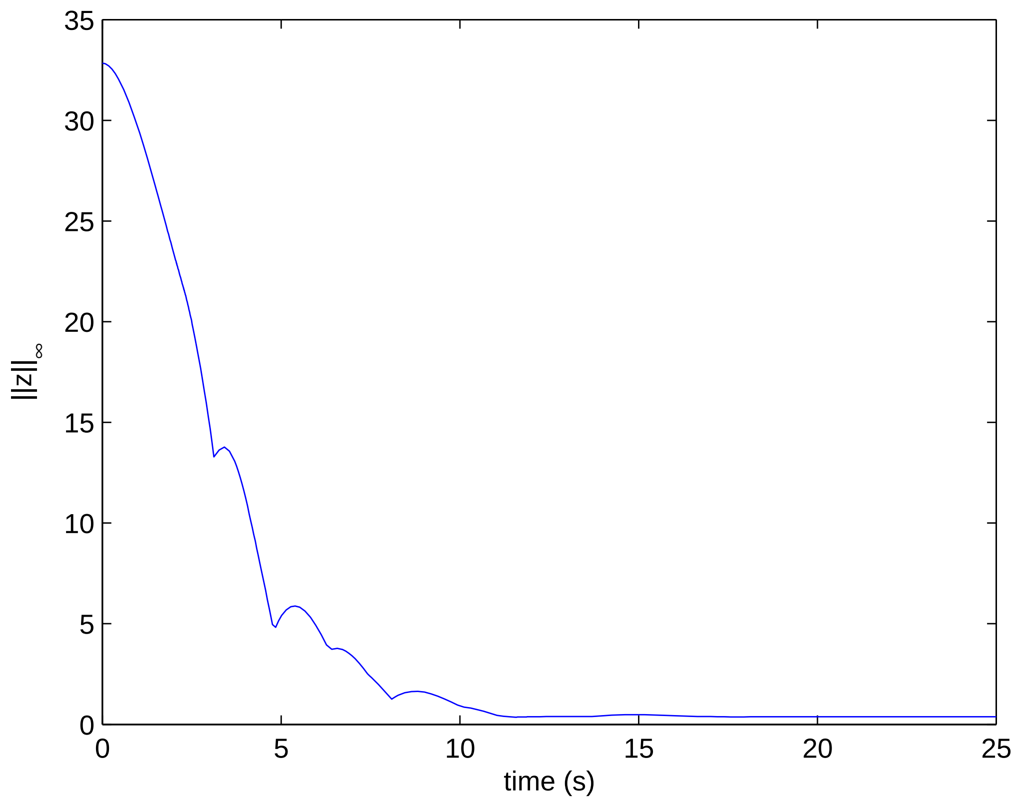

Fig. 8. State norm for system controlled by a spring-like force and 6 nodes.

CHAPTER IV

ROBOT LOCALIZATION USING FUZZY LOGIC

In this chapter we use fuzzy logic to effect global position estimation and local position tracking in mobile nodes. This scheme involves dividing the space in which the node operates, using a grid. At each of these grid points, the node specifies a number between zero and one which represents its confidence that it is located on that grid point. Two scenarios can be envisaged here.

A. Global Position Estimation

It is assumed that the node will have on it sensors with which it can ascertain its position. The node has *a priori* information about the accuracy and precision of these sensors. i.e. if the node gets a reading from the sensor, it has the means to construct a map of the confidence, the sensor reading provides, in the robot's location at every grid point. As an example consider the AmigoBot which is being used in our laboratory. Given a map of the environment, the robot can use the sonar sensors mounted on it to measure the distance from various walls. Depending on the distance, these measurements are usually accurate to within $\pm 5\%$. Having made a measurement, this knowledge about its accuracy can be used to convert it into a confidence function for the node position. In addition the node may have some *a priori* confidence in its position on the map. This confidence may be due to the node having kept track of its past movements. It may also be pre-provided to the node. In the case that it knows the environment map but has no idea about its own position in the environment, it may set the confidence level at every grid point to a single low value (say 0.1) to represent its ignorance of its own position. The node then combines these two confidence functions using heuristic Fuzzy rules. To do this, the confidence level at each grid point is fuzzified into three categories, high, medium and low. The membership

Table I. Fuzzy rule set for one sensor.

Node Conf.	Sensor Confidence		
	High	*Medium*	*Low*
High	Very High	Medium High	Medium
Medium	Medium High	Medium	Medium Low
Low	Medium	Medium Low	Very Low

functions for these categories are shown in Figure 9.

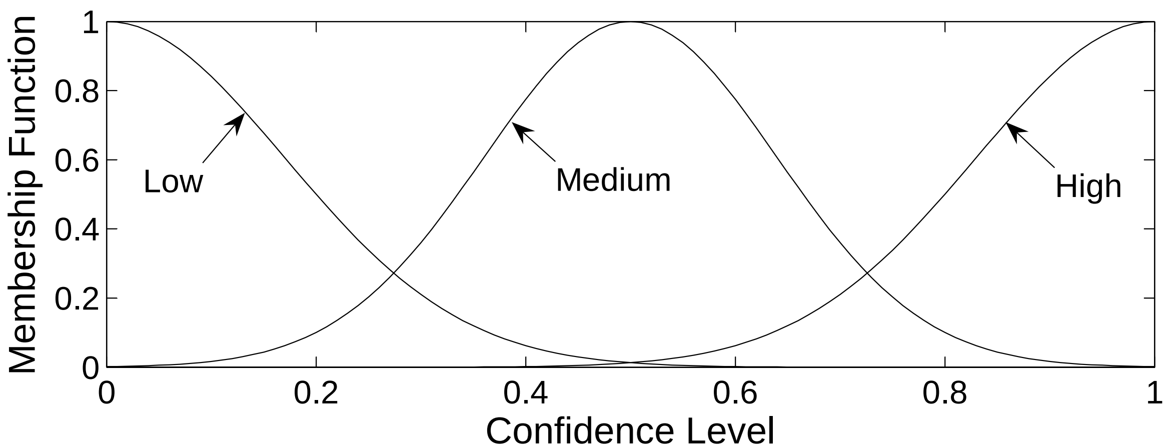

Fig. 9. Membership functions for input confidence levels.

The new or output confidence level of the node at each grid-point is divided into finer categories as shown in Figure 10. The fuzzy engine combines the original node confidence with the sensor confidence using the rule set represented in Table I. The results of these fuzzy rules are described and discussed in the next section. The effect of repeatedly obtaining sensor measurements is also displayed.

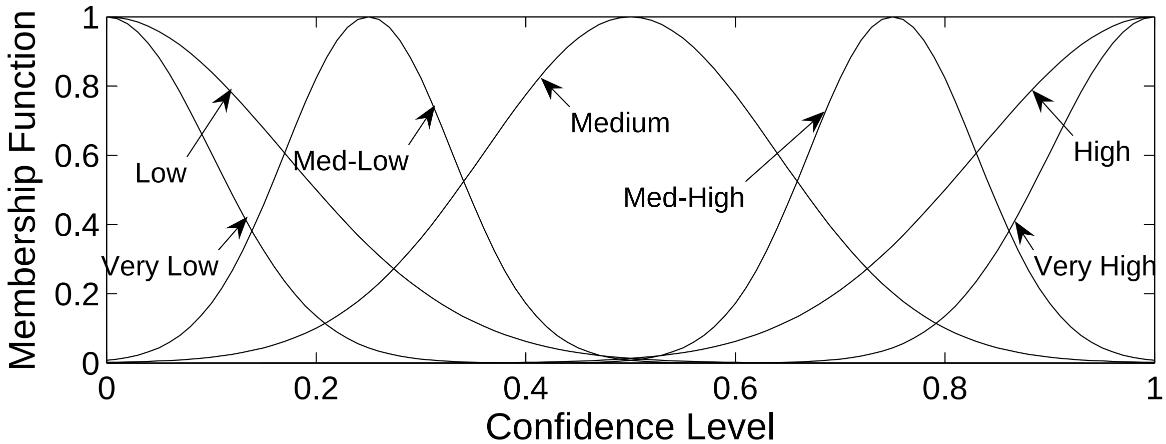

Fig. 10. Membership functions for output confidence levels.

Further, the case that a robot may use multiple sensors to gauge its location is considered. In the simulations, only two sensors are considered but the concept can easily be extended to many sensors. In this case, if both sensors give the robot more or less identical confidence levels at the grid points, the results are similar to the previous part. The interesting case occurs if the one of the sensors gives inaccurate results or if the sensor readings are correct but the robot's own original confidence in its location is misplaced. In this case, the fuzzy rules could be set up so that the 'wrong' data could be filtered out. This can be done by setting up rules so that if two of the three confidence measures match, the third could be given less weightage, eg. *If, at a grid-point, Node Confidence is High and Sensor1 Confidence is High and Sensor2 Confidence is Low, new node confidence is High*.The results for these kind of rule-sets are also displayed in the next section.

B. Local Position Tracking

The case where the node moves is now tackled. In general, the node's motion control devices should provide the robot with information as to how far it has travelled in which direction. The accuracy of this information is again a function of the hardware used and should be known *a priori*. For example, a specific motion controller, when commanded to travel a feet in the x-direction will be able to achieve the the same to an accuracy of $a \pm \delta$ feet and $0 \pm \theta$ degrees. This capacity (the values of δ and θ) of the controller is known *a priori*. Using this information, a confidence function representing the motion of the robot can be constructed as displayed in Figure 11.

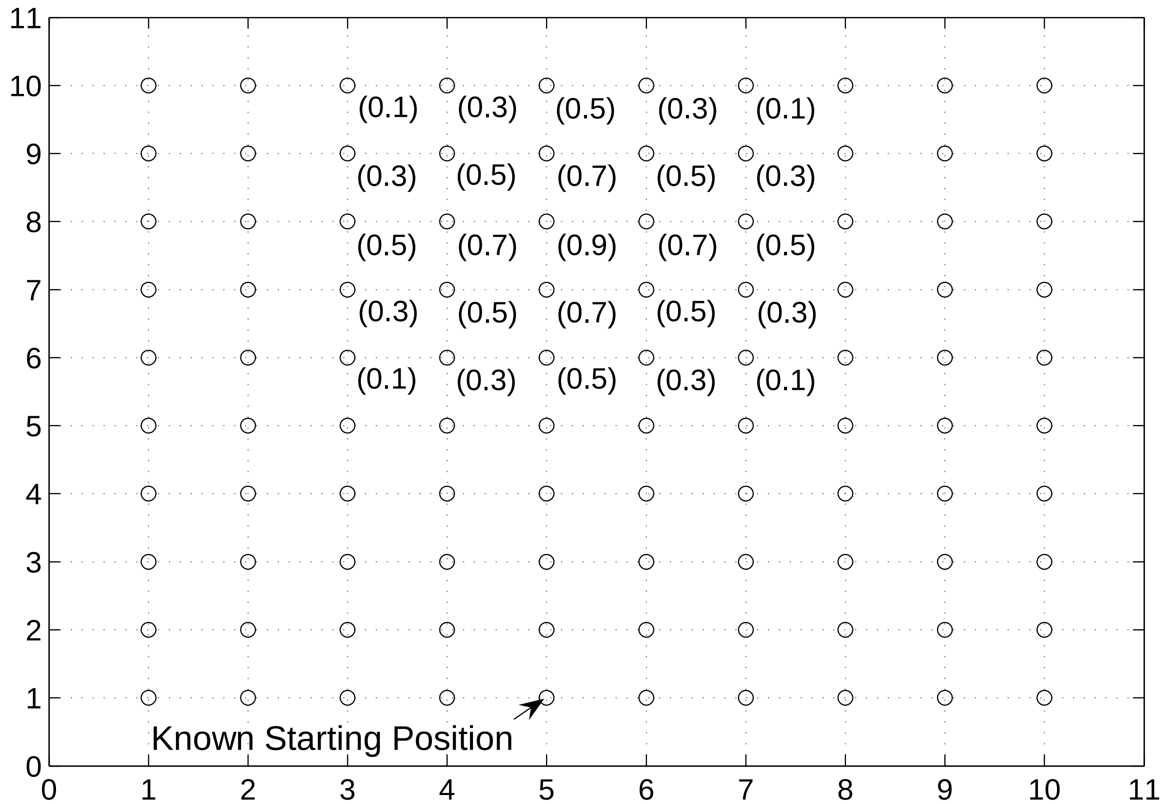

Fig. 11. Confidence levels at grid-points after motion.

In the figure, it is assumed that the starting position of the node is known exactly. The controller is commanded to move 7 units in the y-direction. Since the accuracy of the controller is known, we can develop confidence levels in the node's position at specific grid-points. The numbers in the brackets denote this confidence. The confidence levels at

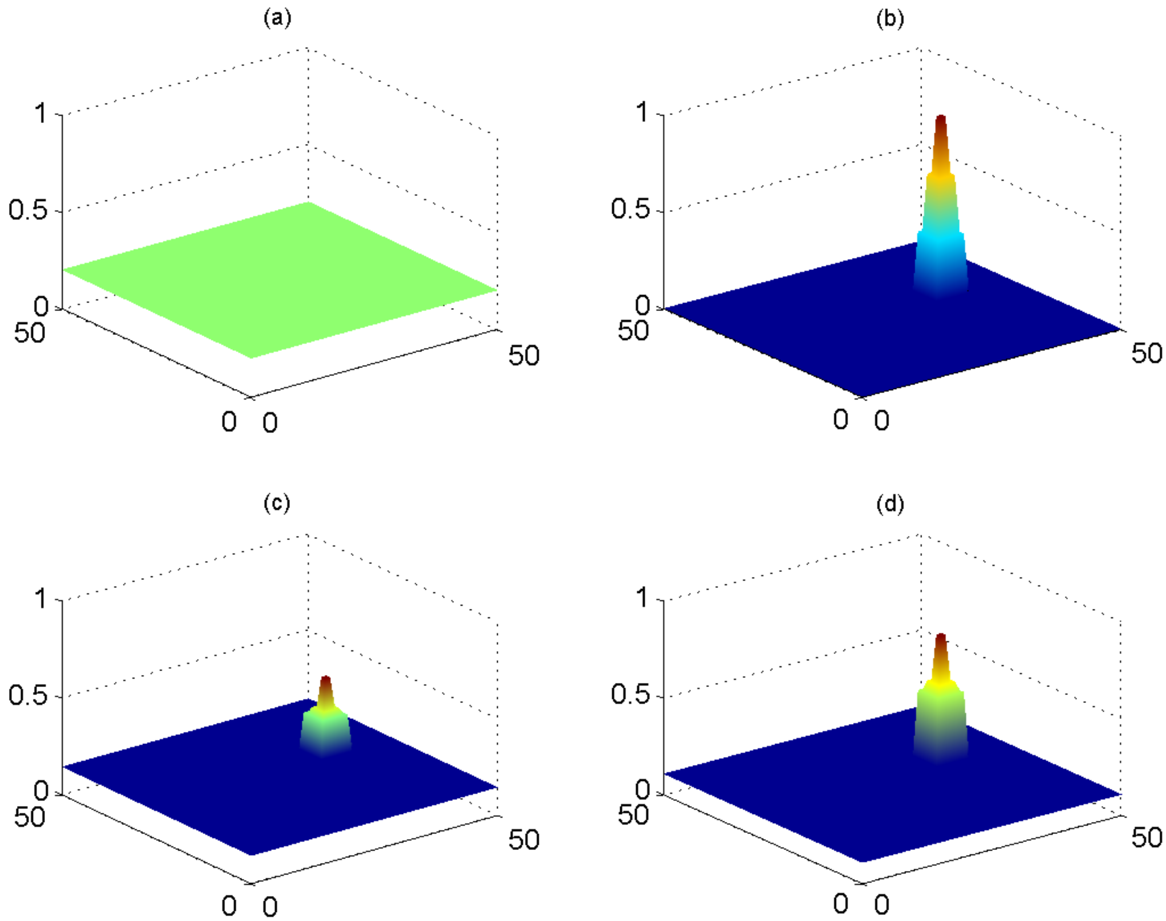

Fig. 12. (a)Initial node confidence, (b)Sensor measurement confidence, (c)Resultant node confidence, (d)Node confidence after a repeat sensor measurement.

the rest of the grid-points are zero.

Once such a confidence function can be constructed, the node can heuristically combine its confidence level at each grid-point with this function to obtain a series of confidence levels at each grid-point. It will then choose the maximum of these confidence levels. The fuzzy rules for these combinations will be the same as outlined in Table I.

One big concern here is the amount of computation that will be needed. To reduce this computation, the authors propose a simple method. We define a threshold value $t > 1$. We compute the maximum confidence level the robot has at any grid-point. The confidence level at each grid-point is then compared to the maximum and only those for which the ratio, of the maximum to the current grid-point confidence level, is less than the threshold are considered in the computation. In effect we are trying to reduce the computation cost by considering only those grid-points where the confidence of the robot in its position is somewhat significant. Thus by controlling the value of t, we can control the computation cost. The lower the value of t, the lesser the number of grid-points used in the computation. Note that for this method to be effective, the node must have at least some prior knowledge of its position. If it has no prior knowledge, all grid-points will end up being considered and thus no reduction in computation will occur. The better the node has localized itself already in the environment,the more it will be able to reduce the computations required for position tracking.

C. Simulation Results and Discussion

To demonstrate this method in simulations, we have assumed a grid of 50 by 50 units. To demonstrate global position estimation, it is assumed that the node initially has no idea of its position. So its confidence is set at a constant low level (of 0.2) initially. This constant level is a matter of preference and can be set to any value as long as it is low, representing the robot's ignorance about its location. The results are shown in Figure 12.

As can be seen the fuzzy engine combines the sensor and node confidences and computes the new confidence satisfactorily. If repeat sensor measurements are taken and give the same result (as can be expected from a precise sensor) the node confidence in its position progressively increases. But this also means that if the sensor is giving it wrong information, repeated measurements worsen the situation.

It is possible that the robot may have more than one sensors to sense its position. Figure 13 illustrates the case that the sensors are working ok. The results are similar to the ones above. Once again repeated measurements have the same advantage as above.

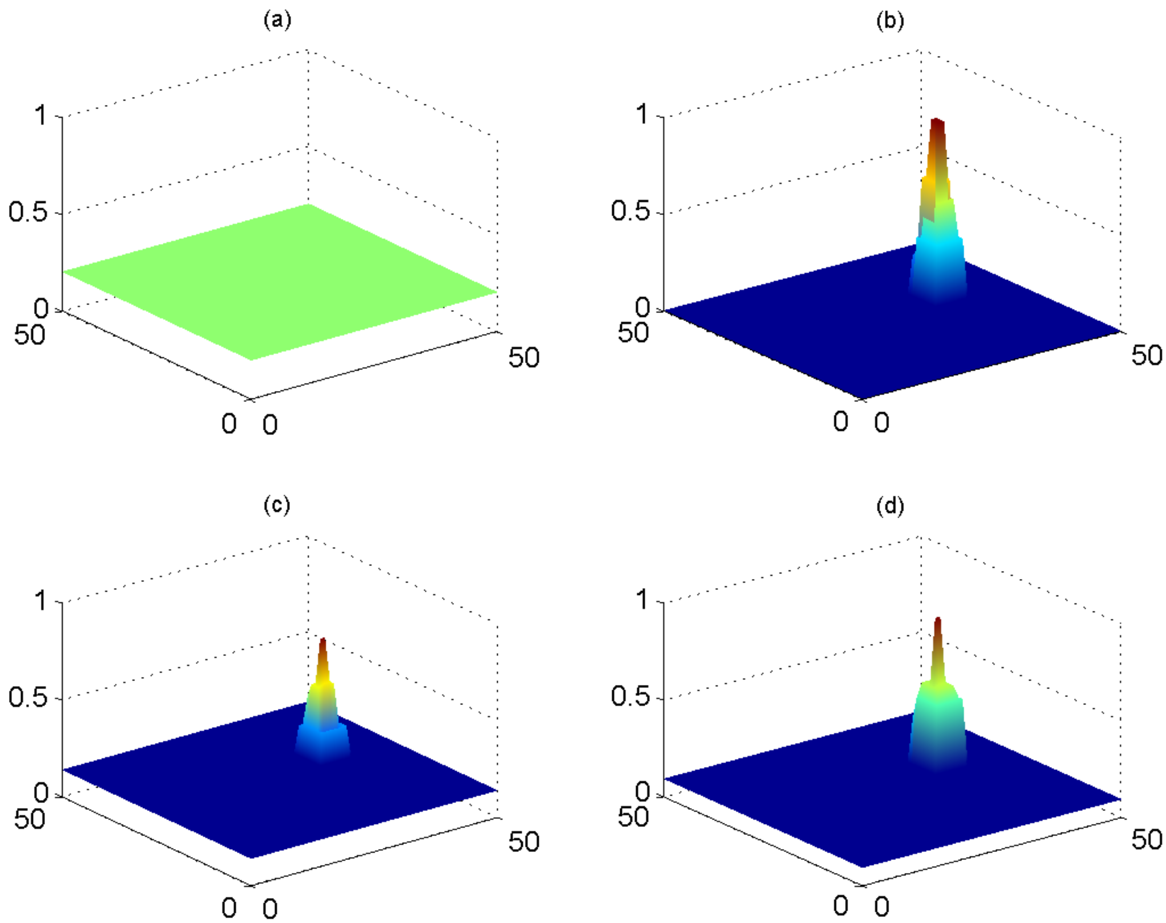

Fig. 13. (a)Initial node confidence, (b)Sensor measurement confidence, (c)Resultant node confidence, (d)Node confidence after four repeat sensor measurements.

As discussed above, the case that one of the sensors is not supplying accurate data is now considered. Here, for the filtering effect to take place it is necessary that the node has some prior idea of its location. The results are shown in Figure 14.

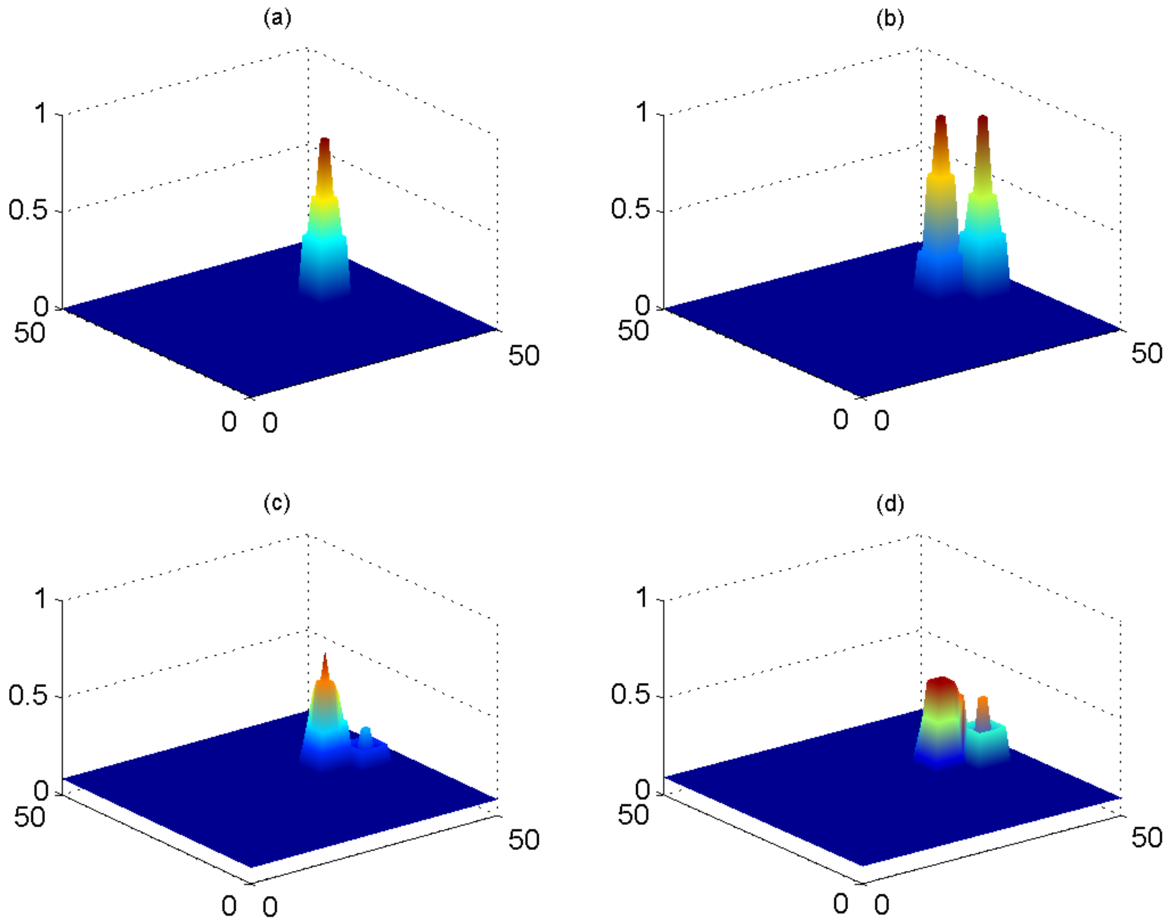

Fig. 14. (a)Initial node confidence, (b)Sensor measurement confidence, (c)Resultant node confidence, (d)Node confidence after a repeat sensor measurement.

The filtering effect of the rules can be clearly seen in part (c). But what was an advantage in the previous two cases becomes a disadvantage here. As can be seen in part (d), repeated measurements severely reduce the node's confidence in its position.

Local position tracking, when the node moves, is now considered. As can be expected, uncertain motion reduces the node's confidence in its position by increasing the area over which the the node could possibly be located. Figures 15 and 16 are two cases where node motion is demonstrated. The node is moving 20 units in the negative sideways direction.

In Figure 15, the parameter t is set to 4 so that a fewer number of grid points are taken into consideration. In Figure 16, t is set to 5. Note that higher the value of t, more will be the computation required as more grid-points will be included in the calculation. Also, the higher the value of t, the more spread the final position estimate will have.However, higher the value of t more accurate will the position estimate be.

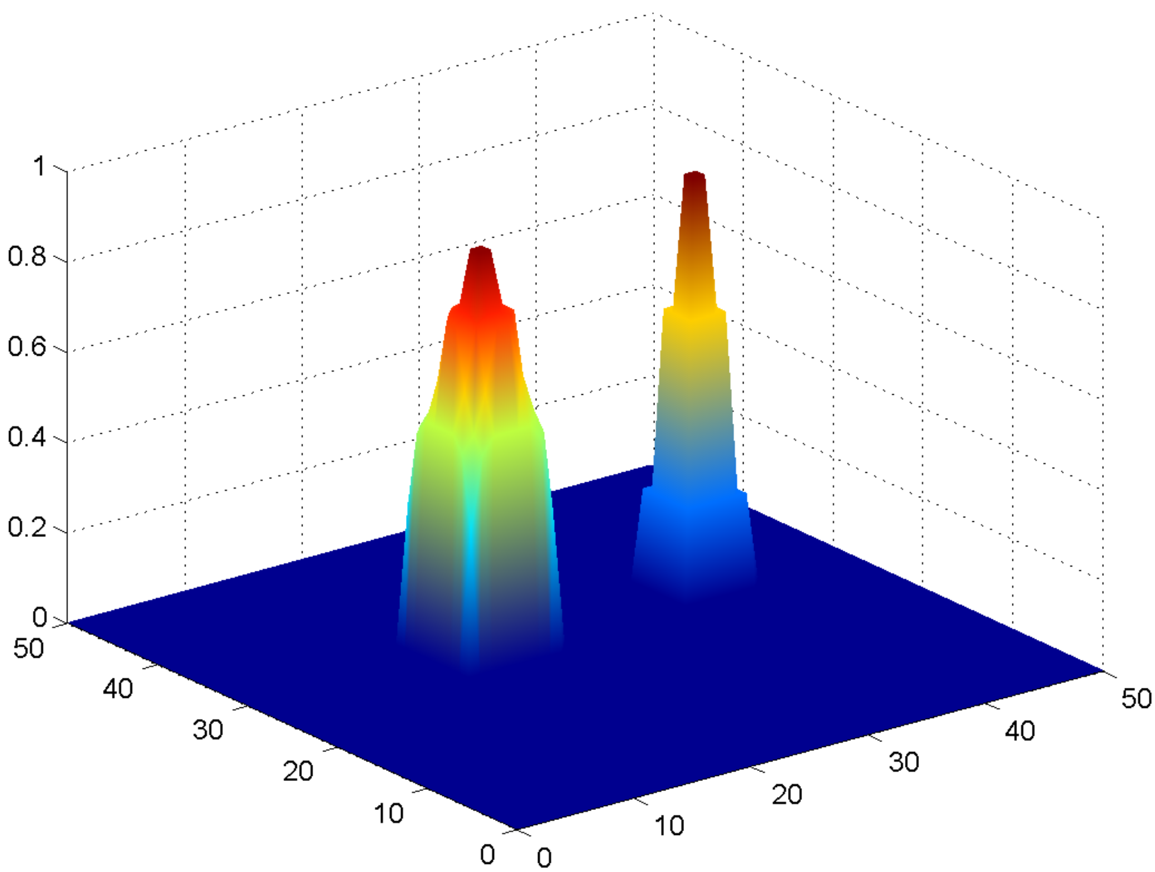

Fig. 15. Node confidences before (right) and after (left) the move. $t = 4$.

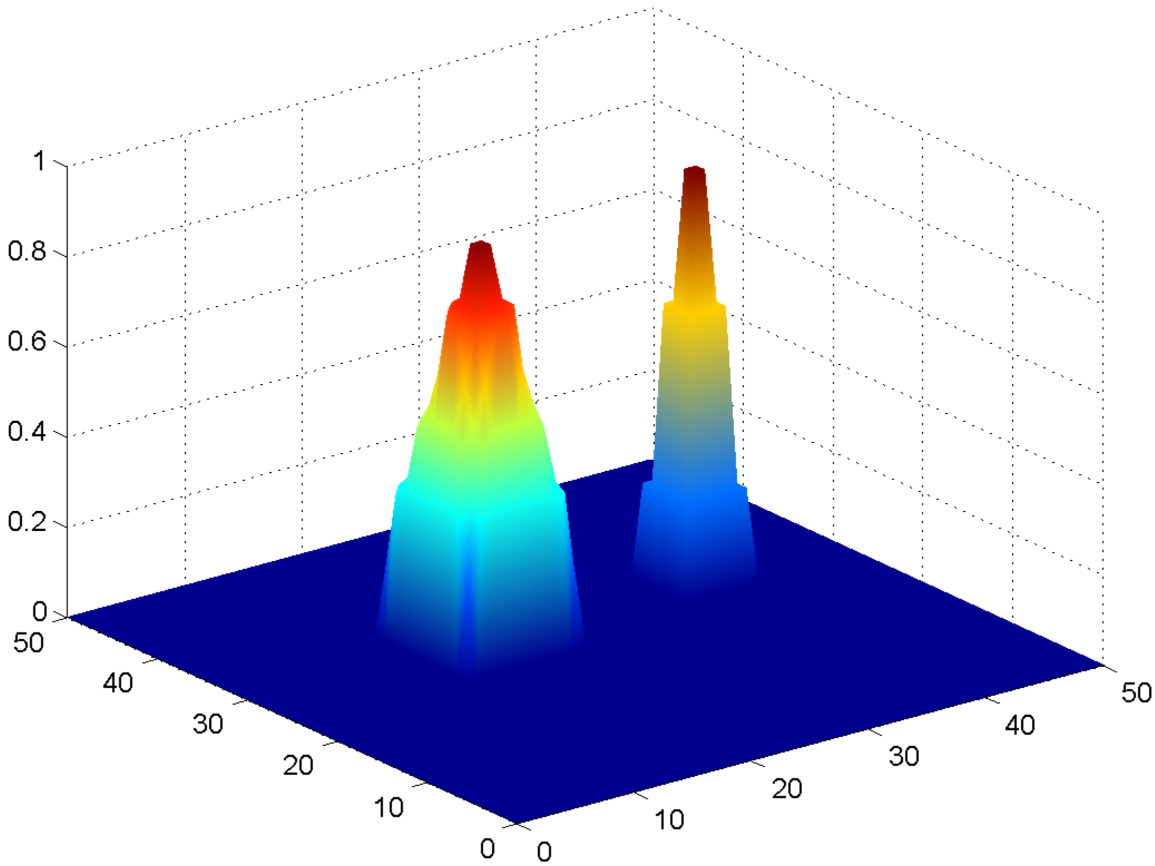

Fig. 16. Node confidences before (right) and after (left) the move. $t = 5$.

D. Experimental Results

In our lab we conducted experiments with a DIRRS (Digital Infra-Red Ranging Sensor). It is a compact, self-contained Infra-Red distance measuring system. The sensor measures distance between 10 and 80 cm. It can output an 8-bit digital range measurement approximately 20 times per second. A picture of the same is seen in Figure 17.

The sensor is mounted on a stepper motor which allows it to turn through 360^o. It has an accuracy of about 90%. The sensor is placed in an environment measuring 40x60 cms. It knows its initial orientation and takes four measurements which allow it to localize itself

Fig. 17. DIRRS sensor.

in the environment. The confidence function constructed from the measurements is shown in Figure 18.

The node is then moved in the 45^o direction by 2 cms. Since the motion is so small, it can be safely assumed that no errors exist in the motion. The robot will use the position confidence function it obtained previously to track its motion. The old and new confidence functions are shown in Figure 19.

In order to verify the working of the algorithm, the node is made to estimate its position again at the new position and this is compared with the computed confidence above. The result is shown in Figure 20. As can be seen, the robot tracks its motion quite well.

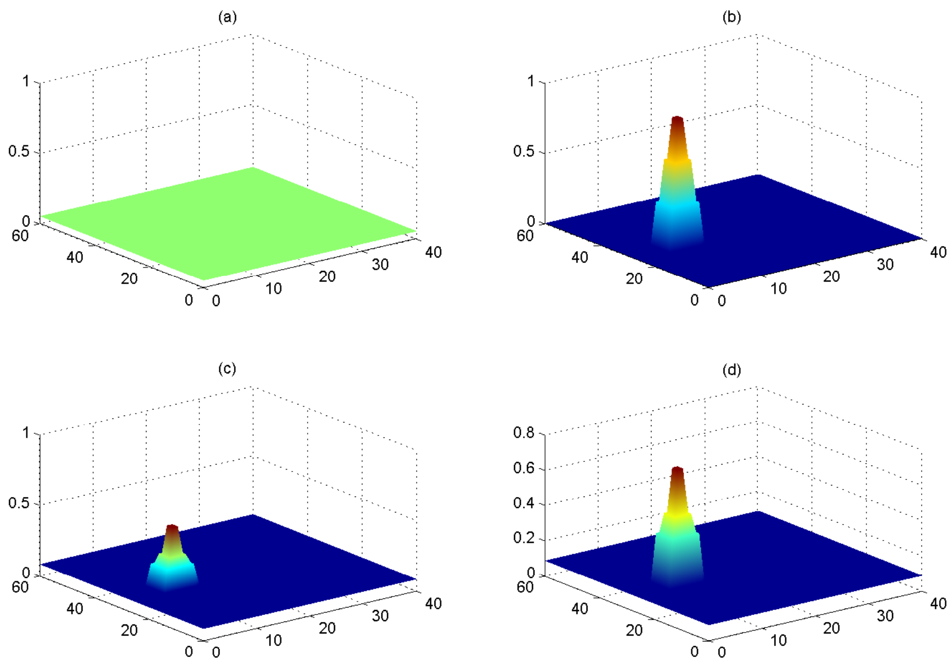

Fig. 18. Membership functions for output confidence levels.

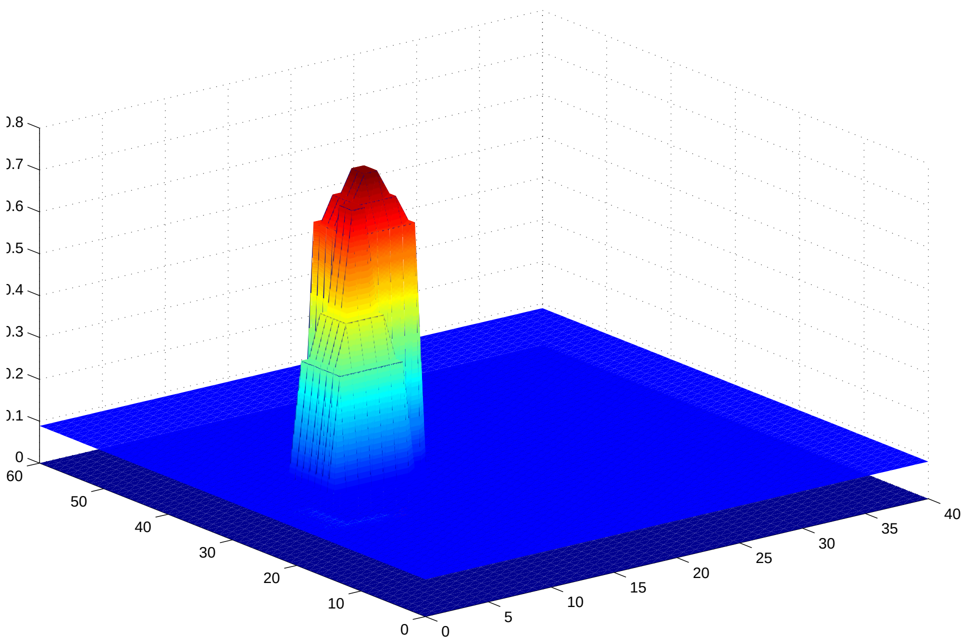

Fig. 19. Node position confidences before and after the move.

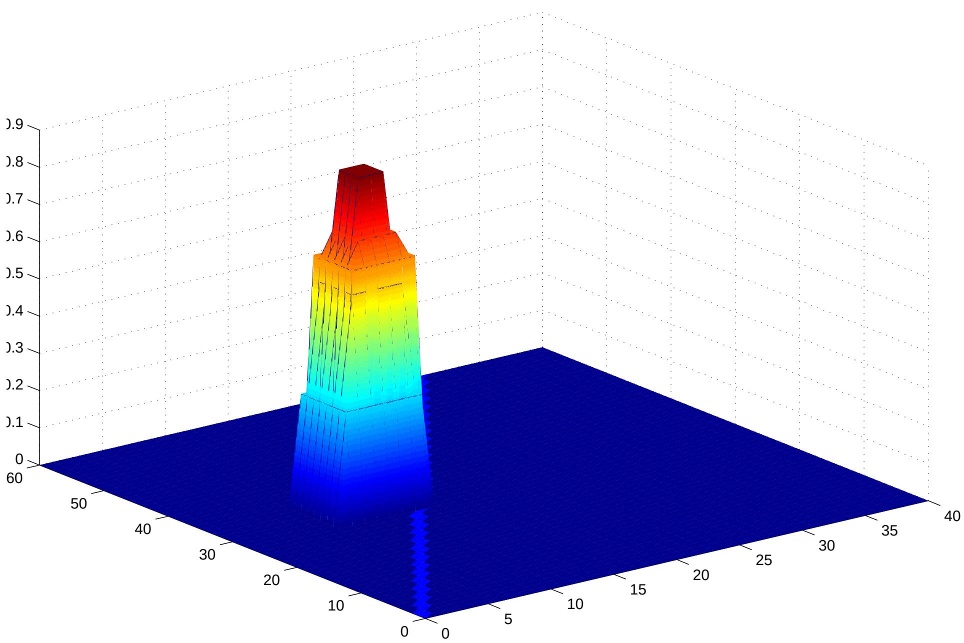

Fig. 20. Computed and measured node position confidences.

CHAPTER V

CONCLUSION

Two relevant issues in Autonomous Mobile Sensor Networks, namely, Design of Force-Laws and Localization of Robots are considered in this work. For the first issue, a stability analysis is done for the network operating under a set of force-laws having a specific structure. This set consists of forcing functions which can be divided into two parts, a linear and non-linear function of the system states, as shown in equation (2.12). This class of force-laws is quite wide and is frequently used in the literature to control mobile sensor networks. Forcing functions simulating a spring connecting two neighboring nodes fall into this category as do forcing functions derived from polynomial Potential Functions. Also, since the linear part of the forcing function considered in this work is a solution to the consensus problem, there are examples of forcing functions which are a combination of a linear function (to achieve consensus) and a second function to achieve another goal, say collision avoidance, and thus belong to the set of functions we are considering.

In the analysis, the non-linear part of the forcing function is treated as a perturbation on the system and the linear part as the input. The system is discontinuous in nature as the analysis is not limited to a constant communication topology. However, an assumption is made that the communication topology, while changing, remains connected at all times. Filippov's calculus of differential equations with discontinuous right hand sides is used in the analysis. The stability analysis is then used to come up with converse Lyapunov theorems for the discontinuous system which place bounds on the states of the system while under the influence of bounded perturbations. A bound on the norm of the states of the system is calculated as a function of the norm of the initial state of the system and the bound on the norm of the perturbation function. These theorems are then used to design force-laws for the network. A few examples of designing a force-law, by calculating the bounds on the force-law to satisfy design limits on the norm of the system states, using the above method are presented.

On the second issue, a fuzzy logic based range-free localization scheme is proposed for robots in a network. Range-free localization schemes are based on probability theory. However, their usability depends on the computational power available and assumptions on the structure of probability density functions. Here a grid-based approach is demonstrated which is expected to be more computationally cheap than one based on strict probability theory and doesn't depend on assumptions about the structure of probability density functions. It is validated using simulations and experiments.

Drawbacks and Future Work: A drawback of the design method developed above, a common one among Lyapunov based methods, is that it is very conservative. However, the work done in analyzing the network can be extended by considering force-laws which have a different structure than the one considered here. One can also, by considering more and more restrictive sets of force-laws, make the results obtained less conservative. Better results can also be obtained by considering the structure and properties of the non-linear part of the force-law. For example, by imposing conditions such as smoothness and differentiability, it could be possible to derive better results. The work can also be used as a framework for the analysis of specific forcing functions and design the parameters of the force-laws to satisfy design requirements. Another potential drawback of the results derived in this work is that even though they apply to any number of nodes present in the network, the computational power required will naturally grow with the number of nodes involved. Thus more powerful computers will be required to apply these results to networks with large number of nodes. The assumption made about the network being connected is necessary. Without this assumption, the network will split into multiple networks with no interaction between them.

REFERENCES

[1] A. Makarenko, T. Kaupp and H. Durrant-White, “Scalable Human-Robot Interactions in Active Sensor Networks,” IEEE Pervasive Computing, vol. 2, no. 4, pp. 63-71, 2003.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A survey on sensor networks,” IEEE Commun. Mag., vol. 40, no.8, pp. 102-114, 2002.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in Proc. Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom’2000), Boston, August 2000.

[4] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, “Instrumenting the world with wireless sensor networks,” in Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), Salt Lake City, Utah, May 2001.

[5] K. Dantu, M. H. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, “Robomote: Enabling mobility in sensor networks,” in IEEE/ACM Fourth International Conference on Information Processing in Sensor Networks, 2005, pp. 404-409.

[6] B. Liu, P. Brass, O. Dousse, P. Nain and D. Towsley, “Mobility Improves Coverage of Sensor Networks,” in Proc. MobiHoc 2005, Urbana-Champaign, 2005, pp. 25-28.

[7] P. Corke, R. Peterson and D. Rus, “Localization and Navigation Assisted by Cooperating Networked Sensors and Robots,” International Journal of Robotics Research, vol. 24, no. 9, 2005.

[8] Kansal, E. Yuen, W. J. Kaiser, G. J. Pottie and M. B. Srivastava, “Sensing Uncertainty reduction using low complexity actuation,” in ACM Third International Symposium on Information Processing in Sensor Networks (IPSN), 2004, pp. 388-395.

[9] K. M. Krishna, H. Hexmoor, S. Pasupuleti and S. Chellapa, “A Surveillance System based on Multiple Mobile Sensors,” in Proceedings of FLAIRS, 2004, pp. 128-134.

[10] A. R. Girard, J. B. de Sousa and J. K. Hedrick, “An overview of emerging results in networked, multi-vehicle systems,” in Proceedings of the 40th IEEE Conference on Decision and Control, Orlando, Florida, 2001, vol. 2, pp. 1485-1490.

[11] A. D. Ryan, D. L. Nguyen and J. K. Hedrick, “Hybrid Control for UAV-Assisted Search and Rescue,” in Proc. of IMECE , Orlando, Nov 2005.

[12] J. R. Spletzer and C. J. Taylor, “Dynamic Sensor Planning and Control for Optimally Tracking Targets,” International Journal of Robotics Research, vol. 22, no. 1, pp. 7-20, Jan. 2003.

[13] A. Makarenko, E. Nettleton, B. Grocholsky, S. Sukkarieh, and H. Durrant-White, “Building a decentralized active sensor networks,” in 11th International Conference on Advanced Robotics, Coimbra, Portugal, 2003.

[14] M. Sznaier and O. I. Camps, “Control issues in active vision: open problems and some answers,” in Proc. IEEE Conf. on Decision and Control, Dec 1998, vol. 3, pp. 3238-3244.

[15] L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne and J. Schutter, “Active Sensing for Robotics - A Survey,” in Proc. 5th International Conference on Numerical Methods and Applications, Borovets, Bulgaria, August 2002, pp. 316-324.

[16] R. Murphey, “Scout and Surveyor: Research Challenges in Autonomous Aircraft Guidance Navigation and Control,” Seminar, Department of Mechanical Engineering, TAMU, College Station, March 22^{nd}, 2006.

[17] W. M. Spears, D. F. Spears, J. C. Hamann and R. Heil, “Distributed, Physics-Based Control of Swarms of Vehicles,” Autonomous Robots, vol. 17, pp. 137162, 2004.

[18] E. Justh and P. Krishnaprasad, “Equilibria and steering laws for planar formations, Systems and Control Letters, vol. 52, no. 1, pp. 2538, 2004.

[19] J. Fax and R. Murray, “Information flow and cooperative control of vehicle formations,” IEEE Trans. on Automatic Control, vol. 49, no. 9, pp. 14651476, 2004.

[20] W. Ren and R. Beard, “Formation feedback control for multiple spacecraft via virtual structures,” IEEE Proceedings - Control Theory and Applications, vol. 151, no. 3, pp. 357368, 2004.

[21] E. Justh and P. Krishnaprasad, “A simple control law for UAV formation flying,” ISR, U. Maryland, Tech. Report 2002-38, 2002.

[22] E. H. Abed, “Formation Control with Virtual Leaders and Reduced Communications,” Workshop on Swarming in Natural and Engineered Systems, Napa Valley, CA, Aug. 2005.

[23] R. Sepulchre, D. Paley, and N. Leonard, “Stabilization of Planar Collective Motion with Limited Communication,” submitted to IEEE Transactions on Automatic Control, 2006.

[24] L. Chaimowicz, V. Kumar, and F. M. Campos, “A paradigm for dynamic coordination of multiple robots,” Autonomous Robots, vol. 17, no. 1, pp. 721, 2004.

[25] Y. Cao, A. Fukunaga, A. Kahng, and F. Meng, "Cooperative mobile robotics: Antecedents and directions," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 226234, 1995.

[26] M. Earl and R. DAndrea, "Multi-vehicle cooperative control using mixed integer linear programming," submitted to IEEE Trans. on Robotics.

[27] A. Howard, M. J. Matarić and G. S. Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," in Proc. of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02) Fukuoka, Japan, Jun. 2002.

[28] F. E. Scheider, D. Wildermuth and H. L. Wolf, "Motion coordination in formations of multiple mobile robots using a potential field approach," Distributed Autonomous Robotics Systems, vol. 4, pp. 305-314, 2000.

[29] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," International Journal of Robotics Research, vol. 5, no. 1, pp. 90-98, 1986.

[30] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," IEEE Trans. on Robotics and Automation, vol. 20, no. 2, pp. 243255, 2004.

[31] G. Sukhatme, "Controlling Sensor Network Deployment for Coverage and Connectivity," in Workshop on Swarming in Natural and Engineered Systems, Napa Valley, CA, Aug. 2005.

[32] R. Olfati-Saber and R. M. Murray, "Agreement Problems in Networks with Directed Graphs and Switching Topology," in Proc. Conf. Decision and Control, Hawaii, 2003.

[33] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Trans. on Automatic Control, vol. 49, no. 9, pp. 15201533, 2004.

[34] R. Olfati-Saber, "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," California Institute of Technology, Technical Report CIT-CDS 2004-005.

[35] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," IEEE Trans. Automatic Control, vol. 48, pp. 9881001, 2002.

[36] H. G. Tanner, A. Jadbabaie and G. J. Pappas, "Stable flocking of mobile agents, part I: Fixed topology," in Proc. IEEE Conf. Decision and Control, pp. 20102015, 2003.

[37] H. G. Tanner, A. Jadbabaie and G. J. Pappas, "Stable flocking of mobile agents, part II: Dynamic topology," Proc. IEEE Conf. Decision and Control, pp. 20162022, 2003.

[38] A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," IEEE Trans. Automatic Control, vol. 49, pp. 1465-1476, Sep. 2004.

[39] G. Lafferriere, A. Williams, J. Caughman and J. J. P. Veerman, "Decentralized Control of Vehicle Formations," Systems and Control Letters, vol. 54, no. 9, pp. 899-910, Sep. 2005.

[40] L. B. Cremean and R. M. Murray, "Stability Analysis of Interconnected Nonlinear Systems Under Matrix Feedback," in Proc. Conf. Decision and Control, Hawaii, 2003.

[41] L. Moreau, "Time-dependent unidirectional communication in multi-agent systems," IEEE Trans. Automatic Control, June 2003.

[42] L. Moreau, "Stability of Multiagent Systems With Time-Dependent Communication Links," IEEE Trans. Automatic Control, vol. 50, no. 2, Feb. 2005, pp. 169-182.

[43] D. Angeli and P. Bliman, “Stability of leaderless multi-agent systems. Extension of a result by Moreau,” arXiv:math.OC/0411338 v1 15 Nov 2004, www.arxiv.org

[44] T. Vicsek, A. Czirok, E. Ben Jacob, I. Cohen and O. Schochet, “Novel type of phase transitions in a system of self-driven particles,” Physical Review Letters, vol. 75, pp. 1226-1229, 1995.

[45] A. S. Morse, “Analysis of Vicseks System with Integer-Valued Measurement Delays,” Workshop on Swarming in Natural and Engineered Systems, Napa Valley, CA, Aug. 2005.

[46] Hatano and Mesbahi, “Agreement over Random Networks,” in Proc. Conf. Decision and Control, pg 2010-2015, 2004.

[47] M. Mesbahi, “Dynamic systems over random networks,” Workshop on Swarming in Natural and Engineered Systems, Napa Valley, CA, Aug. 2005.

[48] V. Gazi and K. M. Passino, “Stability Analysis of Swarms,” IEEE Trans. Automatic Control, vol. 48, no. 4, pp. 692-697, Apr. 2003.

[49] D. Swaroop and J. K. Hedrick, “String Stability for a Class of Nonlinear Systems,” IEEE Trans. Automatic Control, vol. 41, no . 3, pp. 349-357, Mar. 1996.

[50] A. Pant, P. Seiler and K. Hedrick, “Mesh Stability of Look-ahead Interconnected Systems,” in Proc. Conf. Decision and Control, pg.3009-3013, 2001.

[51] F. Xiao and L. Wang, “State consensus for multi-agent systems with switching topologies and time-varying delays,” Int. Journal of Control, vol. 79, no. 10, pp. 1277-1284, 2006.

[52] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*, Kluwer Academic Publishers, Norwell, MA, 1988.

[53] D. Shevitz and B. Paden, “Lyapunov Stability Theory of Nonsmooth Systems,” IEEE Trans. Automatic Control, vol. 39, no.9, pp. 1910-1914, Sep. 1994.

[54] H. Khalil, *Nonlinear Systems*, Prentice Hall, NJ, 1996.

[55] C. Karlof and D. Wagner, “Secure Routing in Sensor Networks: Attacks and Countermeasures,” in 1st IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.

[56] J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing,” IEEE Computer, vol. 34, no. 8, pp. 57-66, Aug. 2001.

[57] L. Doherty, K. Pister and L. Ghaoui, “Convex Position Estimation in Wireless Sensor Networks,” in Proc. IEEE InfoCom 2001, pp. 1655-1663, Apr. 2001.

[58] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, “Localization from Mere Connectivity,” in Proc. MobiHoc 2003, Jun. 2003.

[59] D. Niculescu and Badri Nath, “Ad Hoc Positioning System (APS) Using AoA,” Proc. IEEE InfoCom, 2003.

[60] N. Patwari and A. O. Hero III, “Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks,” IEEE/ACM 2^{nd} Workshop on Wireless Sensor Networks and Applications, Sep. 2003.

[61] P. Maybeck, *Stochastic Models, Estimation and Control, Volume 1*, Academic Press, NY, 1979.

[62] W. Burgard, D. Fox, D. Hennig and T. Schmidt, “Estimating the absolute position of a mobile robot using position probability grids,” in Proc. 13^{th} National Conference on Artificial Intelligence, Portland, pp. 896-901, Aug. 1996.

[63] R. Ueda, T. Fukase et al, "Uniform Monte Carlo Localization - Fast and Robust Self-localization Method for Mobile Robots," in Proc. 2002 IEEE lntemational Conference on Robotics and Automation, Washington D.C., pp.1353-1358, May 2002.

[64] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," in Proc. MobiCom '04, pp. 45-57, 2004.

[65] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Information, and Control*, Prentice-Hall, NJ, 1999.

[66] A. G. Dharne, J. Lee and S. Jayasuriya, "Using Fuzzy Logic for Localization in Mobile Sensor Networks: Simulations and Experiments," in Proc. American Control Conference 2006, Minneapolis, MN, Jun. 2006.

VITA

This dissertation is written by Avinash Gopal Dharne. He completed his Bachelors Degree in mechanical engineering in May 1999 from Sardar Patel College of Engineering, University of Mumbai, Mumbai, India and his Masters Degree in mechanical engineering from Texas A&M University, College Station, TX. He can be reached at the Department of Mechanical Engineering, c/o Dr. Suhada Jayasuriya, TAMU, College Station TX 77843.