COORDINATED MOTION CONTROL OF SWARMS WITH DYNAMIC CONNECTIVITY IN POTENTIAL FLOWS

Guohua Ye [*] **Hua O. Wang** [*] **Kazuo Tanaka** [**]

[*] *Dept. of Aerospace and Mechanical Engineering*
Boston University, 110 Cummington St., MA 02215 USA
e-mail:ygh2003,wangh@bu.edu

[**] *Dept. of Mechanical Engineering and Intelligent Systems*
University of Electro-Communications, 1-5-1 Chougaoka,
Chofu, Tokyo 182-8585 Japan
e-mail:ktanaka@mce.uec.ac.jp

Abstract: This paper presents a general framework for the coordinated motion control of autonomous swarms in the presence of obstacles. The proposed framework judiciously combines concepts and techniques from potential flows, artificial potentials and dynamic connectivity to realize complex swarm behaviors. To begin with, existing concepts from potential flows in fluid mechanics are used to solve the single-agent navigation problem. As an extension, an analytical solution to the stagnation point problem is provided. The potential flow based framework is then modified significantly to facilitate the coordinated control of swarms navigating through multiple obstacles. Artificial potentials are employed for swarming as well as enhanced obstacle avoidance. A novel concept of dynamic connectivity is utilized to improve the performance of obstacle avoidance (Line of Sight Connectivity) and to organize diverse swarm behaviors (Probabilistic Connectivity). Simulation results with a set of developed algorithms are included to illustrate the viability of the proposed framework.

Keywords: Swarm Control, Potential Flow, Dynamic Connectivity.

1. INTRODUCTION

Recent years have witnessed a rising interest in the dynamics and control of group behaviors for vehicular swarms, i.e., systems of multiple autonomous and semi-autonomous vehicles. Swarm systems such as insects, birds, fish or mammals are very common in nature and have served as an inspiration in this research theme. Outcomes of this research can impact a wide variety of applications, especially in the fields of cooperative control of autonomous robots, unmanned air vehicles and mobile sensor networks.

In this paper, we focus on the problem of coordinated motion control of autonomous swarms, i.e., how to design the control algorithms to enable a group of cooperative agents to move from a starting location to a target location in the presence of multiple and possibly moving obstacles. There are a number of essential requirements for the swarm motion. First of all, the swarm motion should be collision-free, i.e., no inter-agent collision and no collision between any agents and obstacles. Secondly, the swarm should move in a formation or flocking mode, i.e., the agents should stay together and move together. Lastly, there may be additional optimality type of requirements.

The motion control problem of a swarm system has typically been divided into two subproblems (Ogren, 2003). The first is on path generation and navigation with obstacle avoidance, which deals with how to move an agent (e.g., a robot) from location A to location B in some efficient manner while avoiding the obstacles. The second is how to keep the agents as a swarm moving together based on the solution of the navigation problem, i.e., every agent in the swarm is coordinating with other agents to realize group motion without inter-agent collision.

Robot navigation is a well studied problem in systems and control. Typical approaches involve the use of artificial potential fields, (**APF**), road maps (**RM**) and so on. Among these approaches, the **APF** method has been used extensively for path planning of mobile robots. A fundamental problem in the application of APF method is how to deal with the local minima that may occur in a potential field environment.

In (Waydo and Murray 2003a), a method of using stream functions to generate smooth paths for vehicle motion planning is introduced. Concepts from hydrodynamic analysis are used to construct potential fields with no local extrema for vehicle guidance. Related work can also be found in (Waydo and Murray 2003b) and (Sullivan, *et. al.*, 2003). Despite the many positive attributes of stream function based methods, a possible problem may arise, i.e., the so-called stagnation point problem (**SP**). A stagnation point in fluid dynamics refers to a point at which the velocity of the fluid becomes zero. Once a robot moves onto a **SP**, it will stop there and can never reach the goal.

In this paper, we discuss how the stagnation point may affect the robot navigation, concepts from fluid mechanics are used to provide a solution to this problem. Based on the potential flow framework, additional concepts and techniques from artificial potentials and dynamic connectivity are incorporated to realize coordinated navigation of swarms. Specifically, artificial potentials are employed for swarming as well as enhanced obstacle avoidance. A novel concept of dynamic connectivity is utilized to improve the performance of obstacle avoidance (Line of Sight Connectivity) and to organize diverse swarm behaviors (Probabilistic Connectivity). Simulation results with a set of developed algorithms are included to illustrate the viability of the proposed framework.

2. BACKGROUND

As in (Waydo and Murray 2003a), this section gives a brief introduction of some important concepts from hydrodynamic analysis. For detailed information, please refer to (Milne-Thomson, 1968) and (Currie, 1993).

2.1 Potential Flows and Complex Potential

Potential Flows and velocity potential: If the flow of an ideal fluid around a body originates in an irrotational flow, then the flow will remain irrotational even near the body. That is, the vorticity vector ω will be zero everywhere in the fluid ($\omega = \nabla \times u_f = 0$). Since $\nabla \times \nabla\phi = 0$ holds for any scalar function ϕ, the condition of irrotationality can then be satisfied identically by choosing $u_f = \nabla\phi$. ϕ is called *velocity potential*, and flow fields which are irrotational, and so can be represented in the form of $u_f = \nabla\phi$ are referred to as *potential flows*. Since the velocity u_f can be expressed as $u_f = u + iv$, we have

$$u = \frac{\partial\phi}{\partial x}, v = \frac{\partial\phi}{\partial y} \tag{1}$$

For an ideal flow, the equation of continuity can be expressed as: $\nabla \cdot u_f = 0$. Substitute this expression for u_f into $u_f = \nabla\phi$ gives $\nabla^2\phi = 0$. So the velocity potential ϕ satisfies the Laplace's equation.

Stream Function: In cartesian coordinates, the continuity equation can be expressed as $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$. Introducing a function ψ which is defined as

$$u = \frac{\partial\psi}{\partial y}, v = -\frac{\partial\psi}{\partial x} \tag{2}$$

The function ψ is then called *stream function*, and by virtue of its definition it is valid for all two-dimensional flows, both rotational and irrotational (Currie, 1993). If the flow is irrotational, which means $\nabla^2\psi = 0$, then the stream function will also satisfies the Laplace's equation.

Complex Potential: The complex potential ω of an irrotational two-dimensional flow of an inviscid flow is defined by

$$\omega(z) = \phi + i\psi \tag{3}$$

here $z = x + iy$, ϕ and ψ are the *velocity potential* and *stream function* respectively. Then by equating the velocity components from (1) and (2) gives the Cauchy-Riemann equation: $\frac{\partial\phi}{\partial x} = \frac{\partial\psi}{\partial y}, \frac{\partial\phi}{\partial y} = -\frac{\partial\psi}{\partial x}$.

Instantaneous streamlines are determined by $\frac{dx}{\partial\psi/\partial y} = \frac{dy}{-\partial\psi/\partial x}$. which is equivalent to $d\psi = 0$, so that along any streamlines $\psi =$constant. The *complex velocity* is expressed as (Currie, 1993)

$$\omega'(z) = \frac{\partial\phi}{\partial x} - i\frac{\partial\phi}{\partial y} \equiv u - iv \tag{4}$$

The uniform flow, sink and vortex are the most important flow types to be implemented for modeling the navigation of single robot, and their complex potentials can be expressed as: $f_u = Uz$, $f_s = -Cln(z)$, $f_v = Ciln(z)$ respectively.

3. NAVIGATION WITH OBSTACLE AVOIDANCE

3.1 Avoidance of a Single Obstacle

Circular obstacle in a uniform flow: First, consider in an uniform flow with strength U ($f_u = Uz$) a single stationary obstacle of radius a is placed at (b_x, b_y), let $b = b_x + ib_y$, apply the Circle Theorem (Milne-Thomson, 1968) gives the complex potential:

$$\omega = Uz + U(\frac{a^2}{z-b} + \bar{b}) \tag{5}$$

For simplicity, suppose (b_x, b_y) is at the origin $(0,0)$, then the complex potential becomes $\omega = Uz + U\frac{a^2}{z}$, and the imaginary part gives the stream function of the flow:

$$\psi = Uy(1 - \frac{a^2}{x^2+y^2}) = Uy(1 - \frac{a^2}{r^2}) \tag{6}$$

Note that $\psi = 0$ on the boundary of the obstacle there $x^2 + y^2 = a^2$, this shows the flow is tangent to the boundary of the obstacle. The complex velocity $\omega'(z) = U - U\frac{a^2}{z^2} = U[1 - \frac{a^2}{r^4}(x^2 - y^2 - i2xy)]$. i.e. $u = U[1 - \frac{a^2}{r^4}(x^2 - y^2)], v = U\frac{a^2}{r^4}2xy$. Here $r^2 = (x - b_x)^2 + (y - b_y)^2$. Use x_1 and x_2 to replace x and y, we have:

$$\dot{x}_1 = U[1 - \frac{a^2}{r^4}(x_1^2 - x_2^2)], \dot{x}_2 = U\frac{a^2}{r^4}2x_1x_2 \tag{7}$$

a plotting is given as Fig.1 where the red lines represent the streamlines and the circle centered at the origin with radius 2 is the obstacle. It can be seen that none of the streamlines goes into the obstacle.

Circular obstacle in a sink flow: Similarly, consider a single stationary obstacle of radius a is placed in a sink flow with strength C. Detailed analysis for this scenario can be found in (Waydo and Murray 2003b). A plot of the streamlines passing through the obstacle in both situations is in Fig.1

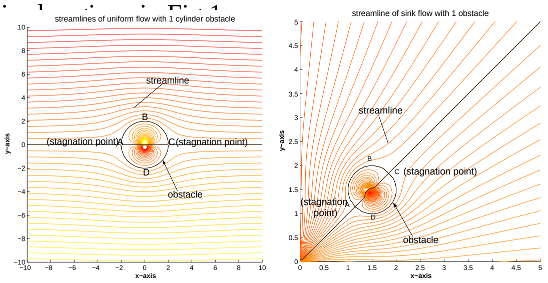

Fig. 1. *Circular Obstacle in Different Flows*

3.2 *Avoidance of Multiple Obstacles*

If there be multiple circular obstacles in the fluid, then we need to solve the Laplace's equation with multiple boundary conditions. This is undoable analytically. But basic ideas from single obstacle avoidance can still be implemented by using method called addition and thresholding, detailed information can be found in (Waydo and Murray 2003a) and (Waydo and Murray 2003b). An example is given in Fig.2. In the simulation, three circular obstacles are placed in an uniform flow and a sink is induced to act as the goal.

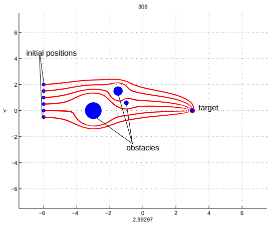

Fig. 2. *Avoidance of Multiple Obstacles*

4. STAGNATION POINTS

As discussed in (Waydo and Murray 2003a) and (Waydo and Murray 2003b), one main advantage of using stream functions is the absence of local extrema, which means the situations of robots stop at local minima when using the **APF** method would not happen. But there is another problem which still needs to pay attention to: the *Stagnation Points* (**SP**). As at any **SP**, the velocity of the fluid becomes zero and if a robot happens to get onto a stagnation point, it will forever stay there.

Here is a simple example of **SP**: from (7), suppose the right side of both equations equal to zero, i.e., the velocity of the fluid becomes zero: $u = 0$ and $v = 0$. Then by solving the equations: $U[1-\frac{a^2}{r^4}(x^2-y^2)] = 0, U\frac{a^2}{r^4}2xy = 0$, we can find the solutions: $x = -a, y = 0$ and $x = a, y = 0$, which are point A and C in the left plot of Figure 1. So at these points the fluid will come to rest. Similarly, it's easy to show in the right plot of Fig.1 A and C are also stagnation points. Since at **SP**s the velocity of the fluid becomes zero, the equation for calculating stagnation points can be expressed as (Currie, 1993):

$$\frac{d\omega}{dz} = 0 \tag{8}$$

A remark is in order here. When applying the stream function method, the dimension of the circular obstacle is typically chosen to be bigger than that of the real obstacle for the sake of safety. So if any robot happens to get onto a **SP**, although it will stay but it will not collide with the obstacles. An example is given in the left plot of Fig.3, one of the robots (the middle one) stopped at one of the **SP**s. To solve the problem, methods such as adopting certain random walking algorithms when reaching a **SP** can be incorporated. However, in this paper, we will use concepts from hydrodynamics to reach a solution for this problem.

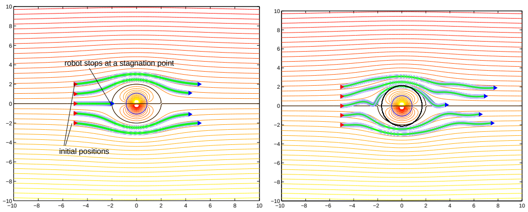

Fig. 3. *Stagnation Point Problem*

In fluid mechanics, the complex potential of vertex $f_v = Ciln(z)$ applies to the circulation motion of fluid between two concentric cylinders. Adding this to the complex potential of a circular obstacle in certain types of flow, it will change the positions of stagnation points (Milne-Thomson, 1968). Here is a brief analysis based on the earlier example of a circular obstacle in a uniform flow: Adding $Ciln(\frac{z}{a})$ to $\omega = Uz + U\frac{a^2}{z}$,

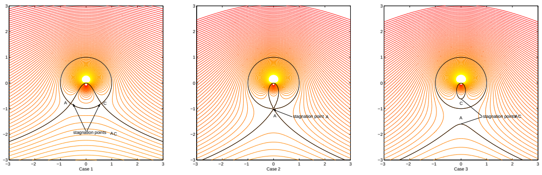

Fig. 4. Stagnation Points Shifting

the new complex potential becomes $\omega = Uz + U\frac{a^2}{z} + iCln(\frac{z}{a})$. As can be seen when $z = ae^{i\theta}$, $\omega = 2Ua$, the imaginary part of ω is constant 0. which means the boundary of the cylinder is still part of the streamline. To find the new positions of the stagnation points, by applying (8) we can get $\frac{z^2}{a^2} + \frac{z}{a}\frac{iC}{aU} - 1 = 0$. The solution can then be found as

$$z = a(-\frac{iC}{2aU} \pm \sqrt{1-\frac{C^2}{4a^2U^2}}) \tag{9}$$

So the positions of stagnation points will be decided by the relationships between C and aU (Currie, 1993).

Case 1: If $C < 2aU$, suppose $\frac{C}{2aU} = sin\beta$. Then $z = a(-i\sin\beta \pm \cos\beta)$, so the stagnation points lie on the cylinder below the center.

Case 2: If $C = 2aU$, then $\beta = \frac{\pi}{2}$, this time the stagnation points coincide at the bottom of the cylinder.

Case 3: If $C > 2aU$, suppose $\frac{C}{2aU} = \cosh\beta$,then $z = ai(-\cosh\beta \pm \sinh\beta) = -aie^{\pm\beta}$, calling the two solutions z_1 and z_2, then $|z_1 z_2| = a^2$. this time the stagnation points are inverse points on the y-axis(imaginary axis), and one of the SP is inside the obstacle cylinder. Plottings of all three cases are shown in Fig.4

It can be seen from above analysis that for those added vertex flows with different strength C the positions of corresponding **SP**s are also be different, so if a vertex flow with its strength a function of time is added, then the **SP**s would keep changing with time. Therefore once a robot gets onto a **SP**, next step when it updates its position, it will be out of the **SP**. This will help the robots which happen to get onto **SP**s to get out of them. An example is given in the right plot of Fig.3, there a vertex flow with its strength a sine function is added: $C = Ksin(\omega t)$ with $K = 1.5aU$. It can be seen the robot which is supposed to stop at **SP** now has no trouble to pass the obstacle.

5. SWARM NAVIGATION USING STREAM FUNCTIONS

In the real world, phenomena of insects or birds aggregating and flocking in swarms are very common. Swarm systems can exhibit diverse adaptable behaviors such as split, rejoin and squeezing maneuvers. In this paper the scenario of interest is the navigation of a swarm such as a school of fish passing through a water course with reefs to their spawning place. Similar research can be found in (Saber and Murray, 2003) and (Saber, 2004), therein models of nets and flocks are discussed based on the graph theory and different types of agents (α, β and γ) are designed to solve the problem of flocking in the presence of multiple obstacles. Here we present a general framework that judiciously combines the stream function based method with dynamic swarm models for coordinated swarm navigation.

5.1 Swarm Modeling

The basic idea to model a swarm system is to express the mutual attractive and repulsive effects between every agent in the swarm. So far many methods have been brought forward and in this paper the model developed in (Gazi and Passino, 2004) and (Gazi and Passino, 2003) will be used. In (Gazi and Passino, 2004) the model considers a swarm of M individuals in an n-dimensional space. Here we will consider the situation when $n = 2$.

Suppose the position of an individual agent i can be described as $x^i \in R^2$. The equation of motion for each individual agent i is (Gazi and Passino, 2004) :

$$\dot{x}^i = -\nabla_{x^i}\sigma(x^i) + \sum_{j=1, j\neq i}^{M} g(x^i - x^j), i = 1, ..., M \tag{10}$$

$\sigma : R^2 \rightarrow$ R represents the attractant/repellent profile of the environment. $g(\cdot)$ represents the function of mutual attraction or repulsion between individuals and is an odd function of the form: $g(x) = -x[g_a(\|x\|) - g_r(\|x\|)]$. The function in (Gazi and Passino, 2004) is $g(x) = -x[a - bexp(\frac{-\|x\|^2}{c})]$ and it will also be used in this paper. To avoid confusion with (9), we rewrite it as: $g(x) = -x[k_1 - k_2exp(\frac{-\|x\|^2}{k_3})]$. Detailed analysis of $g(x)$ can be found in (Gazi and Passino, 2004) and (Gazi and Passino, 2003).

5.2 Swarm Navigation Based on Stream Functions

As motivated by swarm phenomena in nature, in this paper we assume each robot only interact with those that are in front of it along the navigating direction (as indicated in Fig.5) and each robot has a limited sensor range.

Simple Superposition: A natural (and naive) scheme to facilitate swarm navigation based on stream functions would be a superposition: Let the streams "carry" every robot to the "catchment area" while at the same time apply the interaction forces between neighbor robots to keep the group as a swarm.

For example, when considering a swarm navigating in an uniform flow with only one obstacle in the origin, the model can be expressed as: $\dot{x}_1^i = U[1 - \frac{a^2}{r^4}(x_1^2 - x_2^2)] + \sum_{j=1}^{N_i} g(x_1^i - x_1^j)$ and $\dot{x}_2^i = U\frac{a^2}{r^4}2x_1x_2 + \sum_{j=1}^{N_i} g(x_2^i - x_2^j)$. Here N_i is the number of robot within the sensor ranger of agent i. A more general expression can be written as:

$$\dot{x}^i = \dot{x}^i_{flow} + \dot{x}^i_{swarm} \tag{11}$$

A problem with this strategy is that the robots may collide with the obstacles due to the extra "pushing or pulling effects" among every robot in the effort to stay together in a swarm. See Fig.7 for such an example.

Simple Superposition with Switching: To solve the problem of the preceding section, one strategy is to introduce switching control, i.e., once a robot i gets close to any obstacle, stop the swarm control for all the robots. Then since every robot will now only keep navigating along streamlines, no collision with the obstacles will happen. Simulation result as shown in Fig.8 indicates that when using this switching method no collision happens.

Adding Repellent Profile for Obstacles: Using the switching method can help avoid collision with obstacles, but it introduces added complexity in the control algorithm and may also lead to nonsmooth motions. For instance, in Fig.8, it can be seen robot 3 follows its own streamline and becomes separated from other robots.

As the reason for a robot to collide with an obstacle when using the simple superposition is due to the pushing and pulling effects from other robots, therefore another method is to add repellent effects for all the obstacles, once a robot k gets close to a obstacle, then the repulsive effect from the obstacle will try to balance the interactions on k from other robots, and thus avoid the obstacle-agent collision. To do this, we just need to add the $-\nabla_{x_i}\sigma(x^i)$ term back to (10). But at this time, the added term will only be used to represent obstacles. In this paper, the Gaussian type function from (Gazi and Passino, 2004) is used: $\sigma(x) = -\frac{A_\sigma}{2}\exp(-\frac{\|x-c_\sigma\|^2}{l_\sigma}) + b_\sigma$. Now (10) can be rewritten as

$$\dot{x}^i = \dot{x}^i_{flow} + \dot{x}^i_{swarm} + \dot{x}^i_{obs} \tag{12}$$

As the reason for adding the repellent profile for an obstacle is to balance the pushing or pulling effects from other robots, so the scope for this term to be effective should be confined within a limited range. If the radius of the circular obstacle is a, then the range for $\nabla_x\sigma(x)$ would be $R_{rep} = ma$. Usually $1 \leq m \leq 1.5$. The simulation results is shown in Fig.9.

Navigation with Connectivity Testing: In (Gazi and Passino, 2004 and 2003), the algorithm assumes that every robot will interact with all the other robots, i.e., the robots are fully connected. It can be seen this assumption to some extent overlooks the information of obstacles when building the inter-robot connections. In this paper, we introduce a more natural algorithm called navigation with connectivity testing or navigation with line of sight (**LOS**) connectivity to take into account the presence of obstacles.

Navigation with connectivity testing means for any robot i, other robots within its sensor range are to be tested for suitable connectivity, i.e., only when the connecting line between robots i and j does not go into any obstacles, can robot j be considered as a neighbor for i. This means any robot will only interact with the robots which are within its sensor range as well as light on sight (i.e., no obstacles between interacting robots). The idea of connectivity testing stems from the so-called probabilistic Road Map method (**PRM**)in which testing the connectivity between randomly generated nodes is a very important process. More information can be found in (Kavraki and Latombe, 1998), and (Guang, *et. al.*, 2003). A definite advantage of connectivity testing is that for every robot the chance of being pushed or pulled to obstacles is greatly reduced. Suppose the set of robots within the sensor range of robot i is $\mathscr{K}_1^i$ and the robots within sensor range of i but the connecting lines with i will goes into obstacles is $\mathscr{K}_2^i$, then when calculating $\dot{x}^i_{swarm}$, only $x^j \in K_1^i - K_2^i$ will be considered. For example, in Fig.5, $\mathscr{K}_1^i = \{1,2,4,5\}$, $\mathscr{K}_2^i = \{2,4\}$, so for robot i only robot 1 and 5 will be considered. If $K_2^i = K_1^i$, then $x^j \in \emptyset$, if robot i is not within any effective range of the obstacles, then the governing equation will be simplified as $\dot{x}^i = \dot{x}^i_{flow}$, as no streamlines will go into obstacles, so the robot can safely keep marching along the streamline till it find other robots.

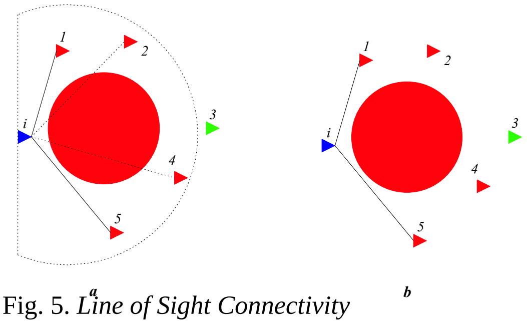

Fig. 5. *Line of Sight Connectivity*

Navigation with Probabilistic Connectivity: Again looking for inspiration from nature, for example in marathon, the most possible action for an athlete to take is to catch up with the nearest runner in front of him. So another algorithm, namely ”Navigation with Probabilistic Connectivity **PC**”, is designed as follows: Suppose for robot i the set of robots which are within the sensor range of i and also have suitable connectivity with i is $\{\mathscr{S} \mid {s_1}^i, {s_2}^i, \ldots {s_n}^i\}$, the distances to i is $\{\mathscr{D} \mid {d_1}^i, {d_2}^i, \ldots {d_n}^i\}$, then the probability for agent j to be chosen as a partner for robot i to follow is expressed as:

$$Pr(choose\ m) = \frac{\frac{1}{{d_m}^i}}{\sum_{k=1}^{n}\left(\frac{1}{{d_k}^i}\right)} \qquad (13)$$

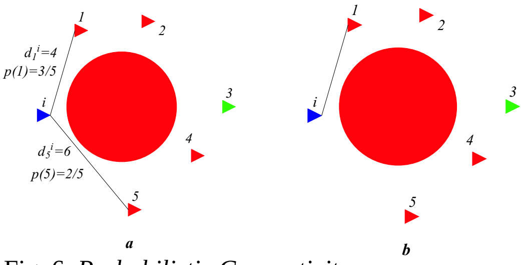

Fig. 6. *Probabilistic Connectivity*

Every time when agent i update its position, it will calculate the $\dot{x}^i_{swarm}$ term only by choosing $x^j = x^m$ with probability $Pr(choose\ m)$. For example, in Fig.5, the neighbors for robot i are robot 1 and 5. The distances ${d_1}^i = 4$ and ${d_5}^i = 6$, using (13), the probability for robot 1 and 5 to be considered are $Pr(choose\ 1) = 3/5$ and $Pr(choose\ 5) = 2/5$ respectively, now robot i generates a sample of a random variable P_{dec} which is uniformly distributed between $[0,1]$; for example, if 0.45 is the number generated, as it is within $[0,0.6]$, so robot 1 will be selected; if the number is 0.8, since it is within $[0.61]$, then robot 5 would be selected. The designed algorithm means any robot would put more emphasis upon closer neighbors, far away ones will be also be considered but with a smaller possibility. This algorithm is more natural and it further decreases the possibility for a robot to collide with obstacles since in most situations the effect between two "connected" robots will not likely go through obstacles.

6. SIMULATION RESULTS

In this section, simulation results are shown to illustrate the effectiveness of the algorithms discussed in the proceeding section. Fig.7 to Fig.9 are the snapshots of simulation results of simple superposition, simple superposition with switching, and adding repellent profile for obstacles, respectively. For all three simulations, there are two circular obstacles with radius 1 and centered at $(0,2)$ and $(0,-2)$ in an uniform flow with strength $U = 2$. The initial positions of all the robots are same for all these simulations: $(-3,5)$, $(-3,0.5)$, $(-2.5,-6)$, $(-2.5,0)$ and $(-2,-1)$. For simulation result in Fig.9, the parameters for the added repellent profiles are: $A_{\sigma 1} = A_{\sigma 2} = 65$, $c_{\sigma 1} = (0,2)$, $c_{\sigma 2} = (0,-2)$, $l_{\sigma 1} = l_{\sigma 2} = 1.1$. Fig.10 and Fig.11 are snapshots of simulation results of navigation with connectivity testing and navigation with probabilistic connectivity. In both simulations, there are five obstacles located at $(-6,0)$, $(-5,5.6)$, $(-2,-3.5)$, $(5,5)$ and $(5,-5)$ with radius 2.2, 1.3, 1.6, 4 and 4. The initial positions of 15 robots are randomly generated but for comparison they are copied and used in both simulations. The strength of the uniform flow is $U = 16$, sensor range for every robot is 18. The added profile for the obstacles are $A_\sigma = (380,380,380,2980,2980)$, c_σ is just the center of obstacles and l_σ equals 1.5 times the radius of every obstacle.

7. CONCLUSIONS

In this paper, we extend the stream function based navigation method to a framework for coordinated motion control of autonomous swarms. The stagnation point problem associated with stream functions is identified and a hydrodynamics based analytical solution is provided. For swarm navigation, novel concepts such as navigation with connectivity testing and probabilistic

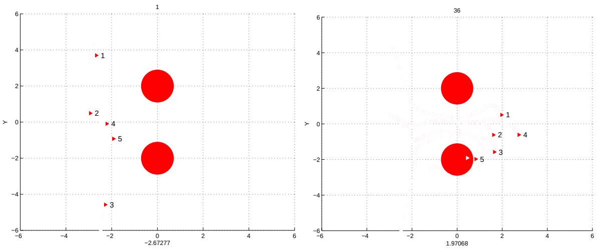

Fig. 7. *Simple Superposition*

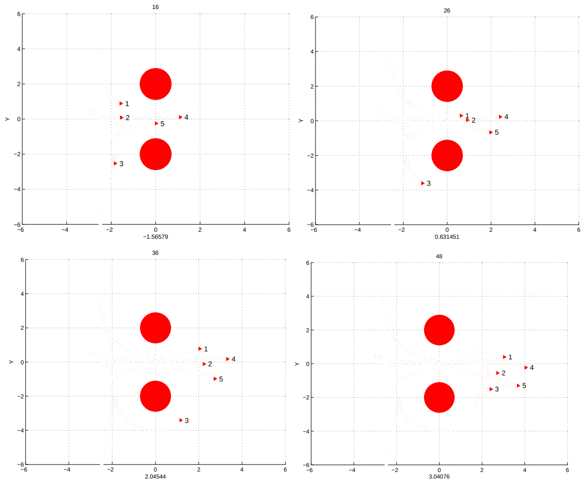

Fig. 8. *Simple Superposition with Switching*

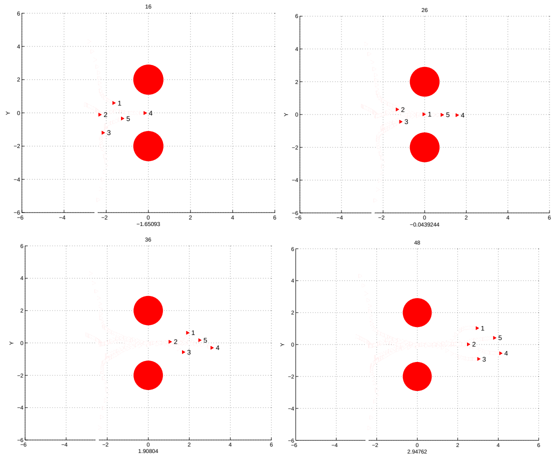

Fig. 9. *Adding Repellent Profile for Obstacles*

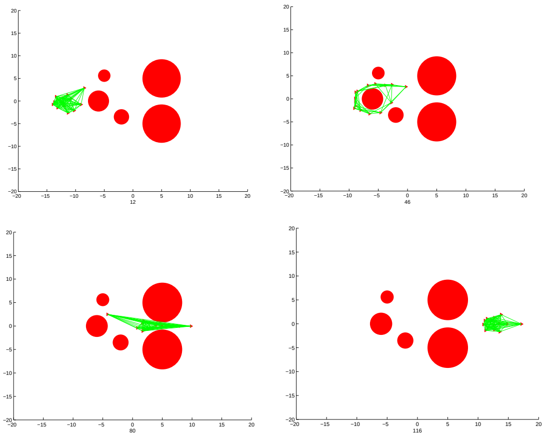

Fig. 10. *Swarm Navigation with Connectivity Testing*

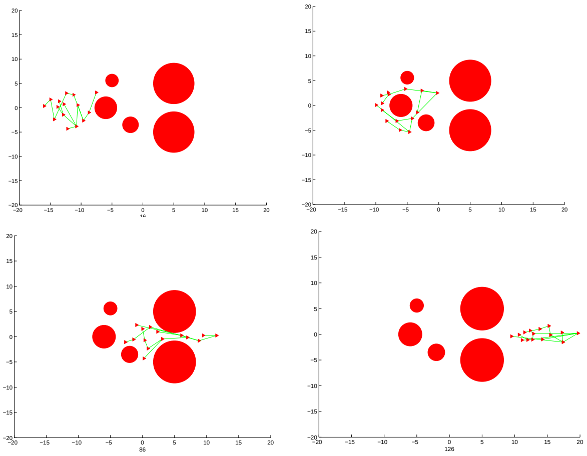

Fig. 11. *Swarm Navigation with Probabilistic Connectivity*

connectivity are introduced. Extensive simulation results illustrate the effectiveness of the proposed framework. Research is underway for further in-depth analysis of the proposed framework.

REFERENCES

Waydo, S., and R.M. Murray, "Vehicle Motion Planning Using Stream Functions," *Proc. IEEE Int. Conf. on Robot. and Autom.*, 2003.

Waydo, S., and R.M. Murray, "Vehicle Motion Planning Using Stream Functions," CDS Technical Report 2003-001, California Institute of Technology, 2003. http://caltechcdstr.library.caltech.edu/.

Sullivan, J., S. Waydo and M. Campbell, "Using Stream Functions for Complex Behavior and Path Generation," *Proc. AIAA Guidance, Navigation and Control Conference*, 2003.

Milne-Thomson, L.M., *Theoretical Hydrodynamic*, Macmillan Company, 5th edition, 1968.

Currie, I.G. *Fundamental Mechanics of Fluids*, McGraw-Hill inc., 2th edition, 1993.

Gazi, V., and K.M. Passino, "Stability Analysis of Social Foraging Swarms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol.34, no.1, pp. 539-557, 2004.

Gazi, V., and K.M. Passino, "Stability Analysis of Social Foraging Swarms," *IEEE Trans. on Automatic Control*, vol.48, no.4, pp. 692-697, 2003.

Saber, R.M. Murray, "Flocking with Obstacle Avoidance: Cooperation with Limited Communication in Mobile Networks", *Proc. 42nd IEEE Conf. on Decision and Control*, pp. 2022-2028, 2003.

Saber, R.O. "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," CDS Technical Report 2004-005, California Institute of Technology, 2004. http://caltechcdstr.library.caltech.edu.

Ogren, P., *Formations and Obstacle Avoidance in Mobile Robot Control*, Ph.D. thesis, Royal Insitute of Technology, 2003.

Kavraki, L.E., and J.C. Latombe, "Probabilistic Roadmaps for Robot Path Planning," *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, K. Gupta and A. del Pobil (eds), John Wiley, pp. 33-53, 1998.

Guang, S., S. Thomas and N.M. Amato "A General Framework for PRM Motion Planning," *Proc. IEEE Int. Conf. on Robot. and Autom.*, pp. 21-26, 2003.