Preparation of Papers for IEEE Sponsored Conferences & Symposia*

Albert Author[1] and Bernard D. Researcher[2]

***Abstract*— This electronic document is a live template. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document.**

I. INTRODUCTION

This template provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. All standard paper components have been specified for three reasons: (1) ease of use when formatting individual papers, (2) automatic compliance to electronic requirements that facilitate the concurrent or later production of electronic products, and (3) conformity of style throughout a conference proceedings. Margins, column widths, line spacing, and type styles are built-in; examples of the type styles are provided throughout this document and are identified in italic type, within parentheses, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

II. PROCEDURE FOR PAPER SUBMISSION

A. Selecting a Template (Heading 2)

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size. It may be used for A4 paper size if the paper size setting is suitably modified.

B. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations

*This work was not supported by any organization

[1]Albert Author is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands `albert.author@papercept.net`

[2]Bernard D. Researcheris with the Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA `b.d.researcher@ieee.org`

III. MATH

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as 3.5-inch disk drive.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: Wb/m2 or webers per square meter, not webers/m2. Spell out units when they appear in text: . . . a few henries, not . . . a few H.
- Use a zero before decimal points: 0.25, not .25. Use cm3, not cc. (bullet list)

C. Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled. Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in

$$\alpha + \beta = \chi \tag{1}$$

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use (1), not Eq. (1) or equation (1), except at the beginning of a sentence: Equation (1) is . . .

D. Some Common Mistakes

- The word data is plural, not singular.
- The subscript for the permeability of vacuum ?0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter o.
- In American English, commas, semi-/colons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an inset, not an insert. The word alternatively is preferred to the word alternately (unless you really mean something that alternates).
- Do not use the word essentially to mean approximately or effectively.
- In your paper title, if the words that uses can accurately replace the word using, capitalize the u; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones affect and effect, complement and compliment, discreet and discrete, principal and principle.
- Do not confuse imply and infer.
- The prefix non is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the et in the Latin abbreviation et al..
- The abbreviation i.e. means that is, and the abbreviation e.g. means for example.

IV. USING THE TEMPLATE

Use this sample document as your LaTeX source file to create your document. Save this file as **root.tex**. You have to make sure to use the cls file that came with this distribution. If you use a different style file, you cannot expect to get required margins. Note also that when you are creating your out PDF file, the source file is only part of the equation. *Your TeX $\rightarrow$ PDF filter determines the output file size. Even if you make all the specifications to output a letter file in the source - if your filter is set to produce A4, you will only get A4 output.*

It is impossible to account for all possible situation, one would encounter using TeX. If you are using multiple TeX files you must make sure that the “MAIN“ source file is called root.tex - this is particularly important if your conference is using PaperPlaza’s built in TeX to PDF conversion tool.

A. Headings, etc

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named Heading 1, Heading 2, Heading 3, and Heading 4 are prescribed.

B. Figures and Tables

Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation Fig. 1, even at the beginning of a sentence.

TABLE I

AN EXAMPLE OF A TABLE

One	Two
Three	Four

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an document, this method is somewhat more stable than directly inserting a picture.

Fig. 1. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

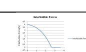

Fig. 2. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity Magnetization, or Magnetization, M, not just M. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write Magnetization (A/m) or Magnetization A[m(1)], not just A/m. Do not label axes with a ratio of quantities and units. For example, write Temperature (K), not Temperature/K.

V. Conclusions

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

VI. Outline

1-Introduction
2-Potential Field Based Approach
Burada Samitha Pubudu daki denklemleri nasil yeniden yazdigimizi anlatalim kisaca
3- Bubble Packing
Samitha Pubududaki artificial forcelari shape partitioning icin nasil kullandigimizi anlatip 4 lu algoritmayi verelim
4-Evaluation
Settling Time- Total Displacements
Dynamic formation- Bubble Packing deki gecikmeler

5-Conclusion

VII. Abstract

Formation control in robotics is a growing topic where research works are mainly geared towards heterogeneous swarm colonies under either decentralized control or limited centralization. Swarm robotics where decentralization is applied, nevertheless assume that the agents are capable of getting global information about the whole swarm.Moreover in the literature, formation control is generally done for known fixed shapes that can be defined mathematically. However no dynamically changing shapes are envisaged and no shape transitions are clearly handled in those works. In this project, we attempt to bring a clear impact to the literature by focusing on tracking and realizing formation shapes under dynamically changing formation shape demands. We have proposed a novel method named Bubble Packing method for dynamic formation control and compared the performance of this method with artificial forces method which is generally used in literature in formation shape generation problems.

VIII. Intro

Formation control problem have different subproblems like formation shape generation, formation reconfiguration&selection and formation tracking [1]. In formation shape generation, agents are expected to get a formation shape which can be defined by external commands or with some mathematical constraint functions [2]. One general approach is to consider artificial potential functions. Samitha and Pubudu [3] have presented an artificial potential function method based on the consideration of the problem as controlling the position of a swarm into a shape, bounded by a simple closed contour. This approach results in deploying uniformly of swarm agents within the contour. Their work provides analysis about the stability and robustness of the proposed system based on Lyapunov like functions. In their work, desired formation shapes are defined with some analytical expressions and individual control laws for agents are composed with artificial potential functions by using these analytical expressions. In real world applications, it may not be possible to have the analytical expressions of the desired formation shape. In our project, we have focused on designing control laws which do not depend on the analytical expressions of the formation shapes.

In some applications, it may be needed to change the formation shape or splitting and joining of the agents together due to either a change in coordinated task requirements or change in environmental conditions such as narrow corridors. In such a scenario, the swarm has to propagate in a narrow corridor before reaching the desired goal state and it is not possible to follow this path by keeping the final desired formation shape. Hence, swarm has to adapt itself to environmental conditions while following a predetermined trajectory. This task requires formation reconfiguration and dynamic task assignment of swarm agents to be dispatched. Hou et al. [4] have defined a method based on global objective functions to provide formation control of a swarm. In their approach it is possible to implement scaling and rotating functions into control laws to adapt the swarm to environmental conditions while achieving a specific task. But their work only covers dynamics adaptation of the formation shape with scaling and rotation rather than dynamically change the shape randomly without any rule. On the other hand, their approach also requires the analytical expressions of the desired formation shapes.

One of the subproblems studied in formation control is formation tracking. The main objective of this problem is to maintain a desired formation with a group of robots, while tracking or following a reference trajectory. The solutions for the formation tracking approaches can be classified into three basic strategies as leader-following, virtual structure and behaviour based approaches [1]. The most general strategy to provide a solution for this problem is leader-following swarm structures [5]. Other strategies have a basis on optimization and graph theory approaches [1]. Kumar et al. proposed a vision based formation control framework for this problem. This framework has a leader following infrastructure [5]. In leader following strategy, some of the agents in the swarm are the leaders to manage the rest of the swarm to achieve a desired specific task and the rest of the agents act as followers. This approach reduces the formation control problem into tracking control problem of individuals to follow the leader from a desired distance and bearing angle, thus the stability and convergence analysis of the formation can be done with the usage of single tracking controllers of members. This approach simplifies the tracking problem of a network of agents to a single agent. Kumar et al. at [5], proposed a control framework in which follower agents move along a trajectory afterwards the leader agent with a desired separation distance l_{ij} and desired relative bearing angle ψ_{ij}. In this approach it is hard to gather the agents in a certain shape. Another drawback is that, determining the separation and bearing angles for individual agents is getting harder as the number of agents in the swarm increases and this strategy is not fault tolerant to the absence of communication between agents.

In virtual structure approach, the formation is composed with a virtual rigid body. Formation control is applied to the whole virtual structure and then the individual agent control laws are determined with inverse dynamic solutions [1]. Lewis and Tan [6] proposed a virtual structure based method for formation control with a bidirectional flow control where robots move to keep the virtual structure when the swarm is following a trajectory and virtual structure adapt itself to the robots' current positions to compensate the relative errors at the end of a maneuver. In virtual structure strategy it is easy to achieve a coordinated behaviour for the group to maintain the formation during a trajectory tracking or a maneuvering, but it is not a suitable strategy to apply a formation control to achieve certain geometrical shape with the swarm agents.

Behaviour based strategies model every agents' behaviours to achieve specific tasks with swarm. These behaviours may be very simple like randomly walk and avoidance of obstacles in the environment or they may be defined in a very complicated manner in order to achieve complex formation shapes with the entire swarm while for example optimizing the overall energy consumption depending upon the implementation of the controller structures. One of the main usage of this strategy is artificial potential field based implementations . Cheng and Nagpal have introduced a robust and self repairing formation control method for swarms [7]. In this approach, individual control laws for the agents are defined with the artificial forces between agents themselves (to avoid collisions) and between the desired formation shape and agents. This solution provides robustness to the agent losses in the swarm during formation control and the rest of the swarm has the ability to refiil their absence in real time without changing the dynamics and the parameters of the formation controller. Because individual control laws are not dependent on the other member of the swarm. Each agent can calculate its own control input at an instant time with current formation shape and current members of the swarm and the whole swarm converge another equilibrium with current members [7]. One of the main disadvantage of the artificial potential based approaches is that , the control forces applied to individual agents are determined instantaneously in accordance with that agent's and the other agents' positions and they cannot guarantee the optimization of the total distance travelled by the agents. Another drawback related with this type of solution is that, there is a possibility to have local minimas in the solution where an agent reaches an undesired point in configuration space under the equilibrium of different types of artificial force components. In that case the total control input acting on the agent will be zero because of cancelling force vectors which has opposite directions to each other generated by formation shape and obstacles etc. In this strategy, the solution may converge slowly to the steady state due to absence of generalized goal states for individual agents in the final formation shape. Because, there are no specific goal states for the individual agents to reach at the final configuration and they are expected to get a global equilibrium with the help of different artificial force components.

Another approach is to define mathematical constraints and objective functions to achieve a specific formation shape by controlling the shape of the swarm colony while following a trajectory. Kumar and Belta [8] presented an abstraction method of configuration space to a manifold defined as $A = GxS$ where G is a lie group representing the position and the orientation of the swarm and S represents the shape of the manifold. They provide individual control laws which can be separately handled to manipulate the lie group G to achieve formation tracking and orientation control and to manipulate the shape S to achieve different geometrical shapes.Their method defines the desired formation shape with shape manifold equations and control the orientation and the scale of this shape with lie group. Similarly Cheah and Slotine [4] proposed a method based on objective functions. Common drawback for these researches, they can only implement a limited number of simple geometrical shapes because the desired formation shapes must be analytically identified in order to compose the related objective functions or shape manifolds. Even if it is possible to define a simple geometrical shape and to control the rotation and the scaling of this shape dynamically, there may be need for more complex and dynamically changing formation shapes rather than scaling and rotation maneuvers in real time applications.

Specific tasks including different missions, requires agents with different capabilities and this kind of tasks may not be achieved with swarms composed of homogeneous agents [9]. In our work, one of the objectives is to implement a formation control system with heterogeneous agents. Furthermore, proposed solutions for formation control problem generally includes a decision making process which is executed by an individual agent or a central server. This kind of approach creates a single point of failure system and if this decision maker unit fails during mission, swarm cannot achieve the desired task. In this thesis work, it is aimed to implement a solution in which every agent is responsible of contributing on decisions and reaching a global consensus.

In this paper, we propose a solution named Bubble Packing method, to achieve the objectives discussed in this section. We have compared the performance of this method with the Artificial forces based method which is commonly used in formation control problems. We have implemented algorithms which can adapt itself to the dynamically changing formations for both of these methods.

IX. Artificial Forces Method

In Artificial Forces method we have implemented potential fields over each agent arised from the interactions between agents, formation shape and environment. The final positions of the agents at the formation shape are determined with local equilibrium of the swarm in which every agent is at balance under the total force acting from the environment. Basically we have implemented three different kinds of artificial forces named; intermember forces representing the forces created by the other agents in the swarm to achieve collision avoidance, the attractive forces representing the forces created by the desired formation shape to attract the agent into the shape and repulsive forces created by the formation shape to keep agents inside the desired formation shape, obstacle forces to provide collision avoidance with workspace obstacles. We have updated the contour integral equations for these potential fields [3] to be calculated on complex formations shapes and obstacles which do not have analytical expressions as following.

Let $f(z)$ is a complex function in a domain D in the complex plane and let C be simple closed contour contained in D with initial point a and terminal point b. It is possible to take the integral of $f(z)$ along the contour C [10]

$$\oint_C f(z)dz = \int_a^b f(z(t))\frac{dz(t)}{dt}dt \tag{1}$$

where

$$\frac{dz(t)}{dt} = \frac{dx(t)}{dt} + i\frac{dy(t)}{dt},\ a \leq t \leq b \tag{2}$$

To simplify this equation, one can write $f(z) = u(x, y) + iv(x, y)$ and $dz = dx + idy$ into the statements,

$$\begin{aligned}\oint_C f(z)dz &= \oint_C udx - vdy + i\oint_C udy + vdx \\ &= A + iB\end{aligned} \tag{3}$$

where

$$\begin{aligned}A &= \int_a^b \left[u(x(t), y(t))\frac{dx(t)}{dt} - v(x(t), y(t))\frac{dy(t)}{dt}\right]dt \\ B &= \int_a^b \left[u(x(t), y(t))\frac{dy(t)}{dt} + v(x(t), y(t))\frac{dx(t)}{dt}\right]dt\end{aligned} \tag{4}$$

Discrete representation of the (4)

$$\begin{aligned}A &= \sum_{k=1}^{K} u(x_k, y_k)(x_{k+1} - x_k) - v(x_k, y_k)(y_{k+1} - y_k) \\ B &= \sum_{k=1}^{K} u(x_k, y_k)(y_{k+1} - y_k) - v(x_k, y_k)(x_{k+1} - x_k)\end{aligned} \tag{5}$$

where

$$\begin{aligned}\|z_k - z_{k-1}\| = \|z_{k+1} - z_k\|, \\ \forall k,\quad k = 1, 2, ..., K \quad when \quad K \to \infty\end{aligned} \tag{6}$$

The assumption of $K \to \infty$ makes it possible to calculate the integral of Cauchy winding number with a small error with large number of K which can be achieved by partitioning the desired formation shape into small pieces with equal $p2$ norms of Δz. We have used this approach to provide discrete domain representations of the contour integrals given in potential field equations in [3].

X. BUBBLE PACKING METHOD

In Bubble Packing method, we have reduced down the formation control problem into two subproblems. The first part of the solution is to partition the desired formation shape into potential goal states according to the agent types to cover the desired formation shape homogeneously. The second part of the solution is the decision process to assign the agents to these goal states continuously to minimize the total displacement of swarm while achieving the desired formation shape. During this decision process, the cost of reaching different goal states is defined with the displacement on the shortest path while reaching that goal state. Our algorithm tries to reduce down the total cost value by assigning each agent to proper goal states.

A. Partitioning Formation Shape Into Goal States

This process basically depends on covering the formation shape surface with a proper number of bubbles by packing them tightly. [11]. The algorithm places the bubbles with their initial conditions in the environment and apply them interbubble forces which imitates the Van der Waals forces between the molecular bonds to distribute the bubbles homogeneously inside the shape. Here, the main idea is to generate a mesh for a surface with identical bubbles to mimic a regular Voronoi diagram with the vertices represented by the centers of these bubbles. We have defined coverage circles as circles with minimum radius which cover the whole collision surface of an agent in 2D workspace. We have implemented the algorithm by representing the agents in the swarm as bubbles with the radius of their coverage circles and create a mesh by using these bubbles.

The bubbles are distributed homogeneously with this process under two kinds of forces, interbubble forces and shape repulsive forces. The interbubble forces are proximity-based forces so that a system of bubbles is in equilibrium when bubbles are distributed over the whole formation shape. The implemented force equation is given

$$f_i(l) = \begin{cases} al^3 + bl^2 + cl + d & \text{when } 0 \leq 1 \leq l_0 \\ 0 & 1 > l_0 \end{cases} \tag{7}$$

where l is the distance between the centers of the related bubbles and a, b, c, d and l_0 are the variables to tune the force acting on the bubbles.

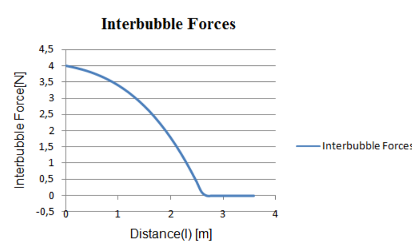

Fig. 3. Interbubble Forces

The shape repulsive forces have the same characteristics with the repulsive artificial forces in [3].

The bubbles are distributed homogeneously under the influence of these two forces and they get an equilibrium state in which the total net forces acting on individual bubbles reaches zero. Figure 4 shows a simulation output of Bubble Packing algorithm. The coverage circles are homogeneously distributed in the formation shape with the help of inter-bubble and shape repulsive forces. The final equilibrium states of the bubbles determine the potential goal states $g_i \in G$ of the agents in the swarm to cover the formation shape.

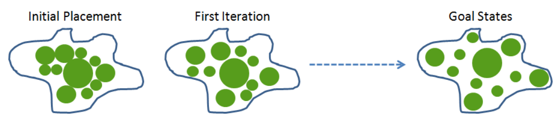

Fig. 4. Bubble Packing Algorithm

B. Decision Process on Goal States

In Section X-A, we have reduced down the formation control problem in which every agent is expected to decide where to position in a given set of possible goal states $g_i \in G$. During this decision process, the cost of reaching different goal states will be the main criteria for each agent. Given goal states and cost values to these goal states, each agent must decide where to position in the formation. This process must be held to optimize the utility of swarm with a collaboration. It is obvious that some of the agents may want to choose the same goal point to reach, so the swarm must reach a global consensus on target points and cases including conflictions must be handled. Our main approach to provide a solution for this problem is to make each agent to calculate the costs of its own to reach the goal states $g_i \in G$ and then reach a global consensus with the other agents to minimize the overall displacements of the swarm.

The algorithm which handles the assignment process of the agents to the goal states, is implemented in four stages. At the first stage, each agent calculates its own free configuration space. Secondly, agents calculate their visibility graphs by using their free configuration space. At the third stage, agents calculate and report their costs to reach each of the goal states $g_i \in G$ with the help of Dijkstra's algorithm. These cost values are defined as the displacements to reach a goal state on shortest path. Finally, agents are assigned to the goal states with the help of Hungarian algorithm which uses the cost values reported by the agents. Hungarian algorithm handles the assignment process to minimize the overall cost(i.e. total displacement) of the swarm.

1) Configuration Space:
We have defined the shortest paths to the goal states of $g_i \in G$ in the free configuration space to avoid collisions with the workspace obstacles [12]. In our implementation, each agent calculates its free configuration space by extracting their forbidden spaces from the configuration space itself. Forbidden space is calculated by augmenting the workspace obstacles with the help of Minkowski sums and we have implemented this process as follows.

Assume an environment with set of obstacles $S = \{P_1, P_2, ..P_t\}$. Configuration for agent i can be described with the position of the center of its coverage circle with $R = \{x_i, y_i\}$. Configuration space of i^{th} agent is the workspace itself and represented by $C(R_i)$. This configuration space is composed of two subspaces; free configuration space and forbidden configuration space of agent i [12].

$$C(R_i) = C_{free}(R_i, S) + C_{forb}(R_i, S) \tag{8}$$

In our implementation each agent calculates its forbidden space by augmenting the workspace obstacles with the help of Minkowski sum method. Minkowski sum method is implemented as follows:

Let a single obstacle is described with a point set of S_1 and the agent is described with a point set of S_2. The Minkowski sum of these two sets $S_1 \subset R^2$ and $S_2 \subset R^2$ can be calculated with the following [12],

$$S_1 \oplus S_2 := \{p + q : p \subset S_1, q \subset S_2\} \tag{9}$$

where $p + q$ denotes the vector sum of the vectors p and q.

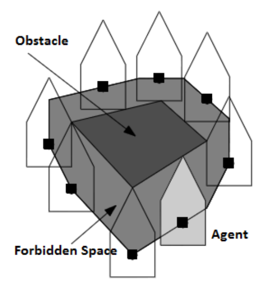

Fig. 5. Forbidden Space Caused by an Obstacle [12]

Figure 5 shows a forbidden space related with an obstacle. Forbidden space for agent i, $C_{forb}(R_i, S)$, is the sum of the forbidden spaces calculated for each obstacle in the environment [12]. Agents extract their forbidden space $C_{forb}(R_i, S)$ from the configuration space itself to calculate their free configuration spaces.

2) Visibility Graphs and Dijkstra's Algorithm:
We have provided collision avoidance while travelling towards to the goal states $g_i \in G$ by defining the shortest paths in the free configuration spaces of the agents. In [12], an additional constraint for the shortest path is defined as follows :

> The shortest path between p_{start} and p_{goal} among a set S of augmented polygonal obstacles consists of the arcs of the visibility graph $\gamma_{vis}(S^*)$ where $S^* := S \cup \{p_{start}, p_{goal}\}$

A visibility graph, $\gamma_{vis}(S^*)$, is a graph which is set of interior nodes representing the vertices of the set of obstacles, S, in the environment and edges which represents visible (which are not intersecting interior region of an obstacle) connections between these nodes [12]. In this project we have updated this constraint in our algorithm. We have inserted all of the goal states $g_i \in G$ in the visibility graphs of each agent to calculate the shortest paths to all of these goal states rather than a specific p_{start} and p_{goal} points given in the definition. In the implementation phase, we have used the obstacles which are augmented with the Minkowski sums to calculate the free configuration space. Let these set of augmented polygonal obstacles represented with $S_i \subset S$. Algorithm to calculate the visibility graph of $S^* := S \cup \{g_i \in G\}$

Data: Set of Vertices Included in S^*
Result: Visibility Graph of S^*
Initialize a graph $\gamma = (V, E)$ where V is the set of all vertices in S^* and $E = \oslash$
for *¡all vertices* $v \subset V$*¿* **do**
    W = VISIBLE_VERTICES(v,S);
    Add edges W to list E;
end

Algorithm 1: VISIBILITY_GRAPH

where $VISIBLE\_VERTICES(v, S)$ algorithm checks whether line segments drawn from v to all vertices in S is intersecting an interior area of any obstacle in the environment and returns the non-intersecting edges. By using this $VISIBILITY\_GRAPH$ algorithm, $SHORTEST\_PATH$ algorithm is defined as follows:

Data: A set S of disjoint polygonal obstacles, position of agent, p_{start}, and all goal states $g_i \in G$
Result: The shortest collision-free paths from p_{start} to all goal states $g_i \in G$
* Assign γ = VISIBILITY_GRAPH(S^*)
* Assign each arc (v, w) in γ_{vis} a weight, which is the Euclidian length of line segment drawn from v to w
for *¡Each goal state* $g_i \in G$*¿* **do**
    * Use Dijkstra's algorithm to compute a shortest path between p_{start} and $g_i \in G$ in γ_{vis}
end

Algorithm 2: SHORTEST_PATH

Figure 6 shows a simulation output executed with 5 workspace obstacles and 6 goal states in desired formation shape. In this algorithm, agent first calculates its own visibility graph by adding goal states $g_i \in G$ in graph as nodes. The visibility graph is illustrated with red edges in the figure. Then it calculates the shortest paths to the goal states, which are given with green edges in the figure. We have used Dijkstra algorithm to compute the shortest path between two nodes in graph with multiple edges, each having a non-negative weight. In our work, the weights of the edges in γ_{vis} , are calculated with the Euclidian distance between nodes in the workspace. Dijkstra algorithm is a tree search algorithm and time complexity of the original algorithm is

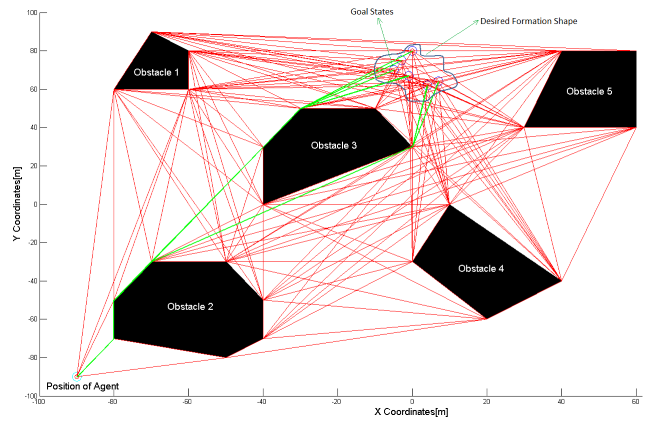

Fig. 6. Shortest Paths to Goal States $g_i \in G$ in a Visibility Graph

$O(n^2)$ where n is the number of the nodes in the graph. With the usage self balancing binary search tree, the algorithm requires $O(k + nlogn)$ time in the worst case where k is the number of edges in the graph. The algorithm for the Dijkstra's is implemented as follows [12]:

Data: γ_{vis} , Position of the agent "$source\_node$"
Result: Shortest Distance to goal states $g_i \in G$ from the $source\_node$ in γ_{vis}
for *¡Each Vertex* $v \subset \gamma_{vis}$*¿* **do**
    Distance[v] := ∞ ;
    Previous[v] := $undefined$;
end
Distance[$source\_node$] :=0 ;
Q:= The set of all vertices $g_i \subset \gamma_{vis}$;
while *¡*$Q \neq null$*¿* **do**
    u:= Node in Q with smallest distance to $source\_node$;
    remove u from Q;
    for *¡Each Neighbor v of u¿* **do**
        alt:= Distance[u] + Cost Between u and v nodes;
        if *alt¡Distance[v]* **then**
            Distance[v] :=alt;
            Previous[v] :=u;
        end
    end
end
return Distance[ ], Previous[ ];

Algorithm 3: DIJKSTRA_ALGORITHM

In algorithm above $Distance[x]$ function call, calculates the total cost from the $source\_node$ to the x vertex, and $Previous[x]$ function call, returns the previous node in optimal path from $source\_node$. This algorithm calculates the shortest paths from the position of an agent to all available goal states. If there is a line segment between the position of an agent and a goal state(i.e. agent can reach that goal state directly without colliding with any workspace obstacle), cost of reaching that goal state is the length of that line segment. These cost values are used to assign the agents to the goal states by minimizing the total displacement.

3) Decision Process of Final Goal States:
We have provided an algorithm to calculate the costs to the goal states $g_i \in G$ with the help of Visibility Graphs and Dijkstra's algorithm in previous sections. These costs are defined as displacements on the shortest paths to the goal states in the visibility graphs for each agent. In this project, our aim is to minimize the total displacement of the individuals while achieving the desired formation shape. For this purpose we have implemented an algorithm to minimize the overall displacement of whole swarm while achieving a formation shape. The problem related with this process can be defined as follows:

We have n number of agents in our swarm and n number of goal states $g_i \in G$ placed in desired formation shape. Each agent has reported their costs (i.e. minimum displacements) to reach each of these goal states and we have to implement an algorithm to assign the agents to these goal states by minimizing the total displacement of the swarm. This is a generalized assignment problem and we have used Hungarian algorithm which is a combinational optimization algorithm that solves this assignment problem. To implement this algorithm, a complete bipartite graph $G = (S, T, E)$ with $n \in S$ agents and $g_i \in T$ goal states is constructed.In this graph, each agent has a cost which is defined by the shortest path to the destination in the workspace for different goal points.

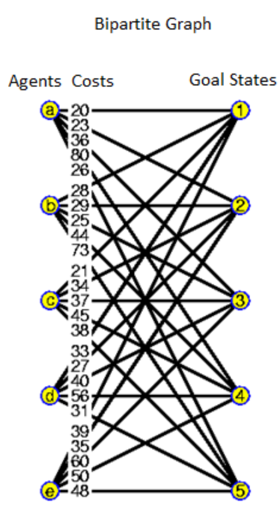

Fig. 7. Sample Bipartite Graph Used in Assignment Problem [13]

We have defined a cost matrix C to implement the Hungarian algorithm. The dimensions of the cost matrix is nxm in which each element represents the cost of assigning the goal state m to the agent n. Since we have equal number of agents and goal states, the cost matrix will be a square nxn matrix. The algorithm for the assignment process as follows:

Data: Cost Matrix , C
Result: Assignment Array of Agents to Goal States
Label1 ;
for *Each Row, R, in C* **do**
  Find the smallest element and subtract it from every element
end
Label 2 ;
if *A column,K, contains more than one zero* **then**
  Repeat Label1 for each column, K
end
Label 3 ;
Select element in columns for which a distinct minimum weight has been determined and add to solution
Label 4 ;
If it is not possible to reach the full solution, flag rows without solutions. Flag all columns in flagged rows that contain a zero. Flag all rows with a previously determined solution in previously flagged columns.
Label 5 ;
From elements remaining in flagged rows and unflagged rows, determine the element which has smallest value and assign this value to γ. Subtract γ from every unflagged element and add γ to every element that has been flagged twice.
Label6 ;
Goto Label3 until full solution has been achieved.

Algorithm 4: HUNGARIAN_ALGORITHM

Hungarian algorithm returns a vector of $nx1$ and each row in this vector represents the goal state that the related agent is assigned. With this assignment, the total cost (i.e. total displacement) of the swarm is minimized.

APPENDIX

Appendixes should appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word acknowledgment in America is without an e after the g. Avoid the stilted expression, One of us (R. B. G.) thanks . . . Instead, try R. B. G. thanks. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

REFERENCES

[1] Kiattisin Kanjanawanishkul. Formation control of mobile robots: Survey. n.d.
[2] Magnus Egerstedt and Xiaoming Hu. Formation constrained multi-agent control. *IEEE Transactions On Robotics And Automation*, 17:947–951, 2001.
[3] Samitha Ekanayake and Pubudu Pathirana. Formations of robotic swarm: An artificial force based approach. *International Journal of Advanced Robotic Systems*, 7:173–190, 2010.
[4] S.P. Hou, C.C. Cheah, and J.J.E. Slotine. Dynamic region following formation control for a swarm of robots. In *ICRA*, 2009.

[5] Aveek Das, Rafael Fierro, Vijay Kumar, James Ostrowski, John Spletzer, and Camilla Taylor. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation*, 18:813–825, 2002.
[6] Anthony Lewis and Kar-Han Tan. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, 4:387–403, 1997.
[7] Jimming Cheng, Winston Cheng, and Radhika Nagpal. Robust and self-repairing formation control for swarms of mobile agents. In *AAAI 20th National Conference on Artificial Intelligence*, 2005.
[8] Calin Belta and Vijay Kumar. Abstraction and control for groups of robots. *IEEE Transactions on Robotics and Automation*, 20:865–875, 2004.
[9] Dorigo et al. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics* & *Automation Magazine*, 20:60–71, 2013.
[10] Wikipedia. Methods of contour integration — Wikipedia, the free encyclopedia, 2016. [Online; accessed 10-February-2016].
[11] Kenji Shimada and David Gossard. Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing. In *ACM Symposium on Solid Modeling and Applications*, 1995.
[12] Mark Berg, Otfried Cheong, Kreveld. Marc, and Marc Overmars. *Computational Geometry*. Springer, 1998.
[13] Jop Frederik Sibeyn. Graph algorithms. https://www8.cs.umu.se/ jopsi/dinf504/chap14.shtml, last visited on April 2016.

REFERENCES

[1] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.
[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.
[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.
[4] B. Smith, An approach to graphs of linear forms (Unpublished work style), unpublished.
[5] E. H. Miller, A note on reflector arrays (Periodical styleAccepted for publication), IEEE Trans. Antennas Propagat., to be publised.
[6] J. Wang, Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleSubmitted for publication), IEEE J. Quantum Electron., submitted for publication.
[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style), IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].
[9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.
[10] J. U. Duncombe, Infrared navigationPart I: An assessment of feasibility (Periodical style), IEEE Trans. Electron Devices, vol. ED-11, pp. 3439, Jan. 1959.
[11] S. Chen, B. Mulgrew, and P. M. Grant, A clustering technique for digital communications channel equalization using radial basis function networks, IEEE Trans. Neural Networks, vol. 4, pp. 570578, July 1993.
[12] R. W. Lucky, Automatic equalization for digital communication, Bell Syst. Tech. J., vol. 44, no. 4, pp. 547588, Apr. 1965.
[13] S. P. Bingulac, On the compatibility of adaptive controllers (Published Conference Proceedings style), in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 816.
[14] G. R. Faulhaber, Design of service systems with priority reservation, in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 38.
[15] W. D. Doyle, Magnetization reversal in films with biaxial anisotropy, in 1987 Proc. INTERMAG Conf., pp. 2.2-12.2-6.
[16] G. W. Juette and L. E. Zeffanella, Radio noise currents n short sections on bundle conductors (Presented Conference Paper style), presented at the IEEE Summer power Meeting, Dallas, TX, June 2227, 1990, Paper 90 SM 690-0 PWRS.
[17] J. G. Kreifeldt, An analysis of surface-detected EMG as an amplitude-modulated noise, presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, Narrow-band analyzer (Thesis or Dissertation style), Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.
[19] N. Kawasaki, Parametric study of thermal and chemical nonequilibrium nozzle flow, M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.
[20] J. P. Wilkinson, Nonlinear resonant circuit devices (Patent style), U.S. Patent 3 624 12, July 16, 1990.