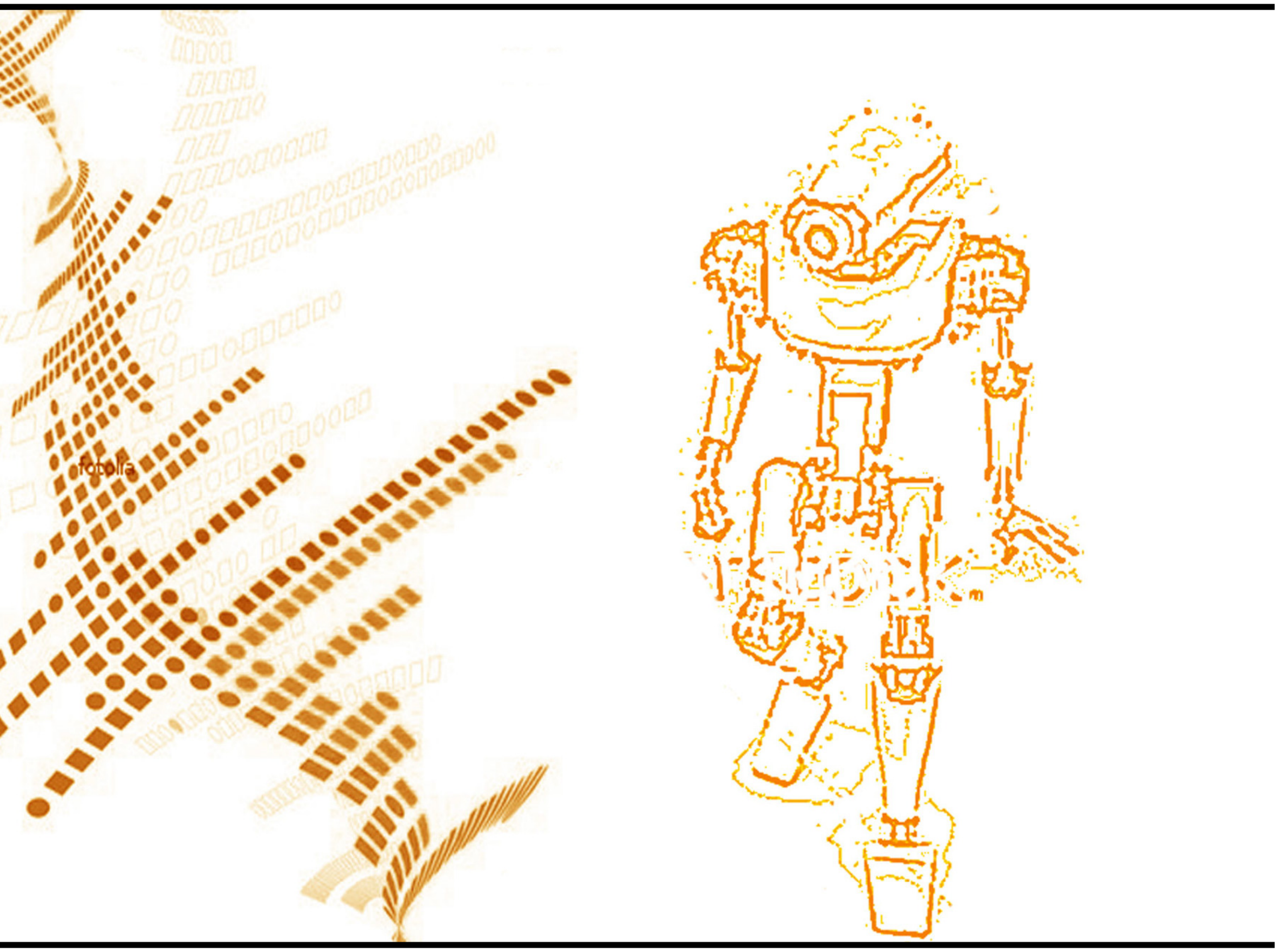
Volume 6 ▪ Issue 1 ▪ March / April 2015

INTERNATIONAL JOURNAL OF ROBOTICS AND AUTOMATION (IJRA)

ISSN : 2180-1312
Publication Frequency: 6 Issues / Year

CSC PUBLISHERS
http://www.cscjournals.org



INTERNATIONAL JOURNAL OF ROBOTICS AND AUTOMATION (IJRA)

VOLUME 6, ISSUE 1, 2015

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 2180-1312

I International Journal of Robotics and Automation (IJRA) is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJRA Journal is a part of CSC Publishers
Computer Science Journals
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF ROBOTICS AND AUTOMATION (IJRA)

Book: Volume 6, Issue 1, March / April 2015
Publishing Date: 31-03- 2015
ISSN (Online): 2180-1312

IJRA Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2015

EDITORIAL PREFACE

Robots are becoming part of people's everyday social lives - and will increasingly become so. In future years, robots may become caretaking assistants for the elderly or academic tutors for our children, or medical assistants, day care assistants, or psychological counselors. Robots may become our co-workers in factories and offices, or maids in our homes. It is the *First* Issue of Volume *Six* of International Journal of Robotics and Automation (IJRA). IJRA published six times in a year and it is being peer reviewed to very high International standards.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 6, 2015, IJRA appears in more focused issues. Besides normal publications, IJRA intends to organize special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

IJRA looks to the different aspects like sensors in robot, control systems, manipulators, power supplies and software. IJRA is aiming to push the frontier of robotics into a new dimension, in which motion and intelligence play equally important roles. IJRA scope includes systems, dynamics, control, simulation, automation engineering, robotics programming, software and hardware designing for robots, artificial intelligence in robotics and automation, industrial robots, automation, manufacturing, and social implications etc. IJRA cover the all aspect relating to the robots and automation.

The IJRA is a refereed journal aims in providing a platform to researchers, scientists, engineers and practitioners throughout the world to publish the latest achievement, future challenges and exciting applications of intelligent and autonomous robots. IJRA open access publications have greatly speeded the pace of development in the robotics and automation field. IJRA objective is to publish articles that are not only technically proficient but also contains state of the art ideas and problems for international readership.

In order to position IJRA as one of the top International journal in robotics, a group of highly valuable and senior International scholars are serving its Editorial Board who ensures that each issue must publish qualitative research articles from International research communities relevant to signal processing fields.

IJRA editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJRA. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJRA provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Robotics and Automation (IJRA)

Editorial Board Members

TABLE OF CONTENTS

Volume 6, Issue 1, March / April 2015

Pages

Flocking Control In Networks of Multiple VTOL Agents with Nonlinear and Under-actuated Features

Shulong Zhao jaymaths@163.com
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha,410073, China

Xiangke Wang xkwang@nudt.edu.cn
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha,410073, China

Tianjiang Hu t.j.hu@nudt.edu.cn
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha,410073, China

Daibing Zhang swimsword@163.com
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha,410073, China

Lincheng Shen lcshen@nudt.edu.cn
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha,410073, China

Abstract

In this paper, the problem of flocking control in networks of multiple Vertical Take-Off and Landing (VTOL) agents with nonlinear and under-actuated features is addressed. Compared with the widely used double-integrator model, the VTOL agents are distinguished with nonlinear and under-actuated dynamics and cannot be linearly parameterized. A unified and systematic procedure is employed to design the flocking controllers by using the backstepping technique to guarantee multi-agents to arrive at a fixed formation and converge in a desired geometric pattern whose centroid move along a desired trajectory. Finally, some numerical simulations are provided to illustrate the effectiveness of the new design.

Keywords: Flocking, Multi-agent System, Under-actuated, VTOL, Formation.

1. INTRODUCTION

Over the last decades, tremendous interests have been paid to the problem of flocking in biology, physics, and computer science. Engineering applications of flocking include search, rescue, coverage, surveillance, sensor networks, and cooperative transportation [1]. The analysis of flocking is inspired by scenes of animals in nature, such as birds, fishes, and bacteria. Compared with a single animal, multiple animals provide advantages over their monolithic counterparts for many reasons. Perhaps the greatest benefit is their resilience against entirety failures and their ability to adapt to unknown environments [2]. Thus, a variety of algorithms have been proposed for the coordination of multiple agent systems and the flocking control problem has been employed in many fields.

While, there remain several open problems to be solved before the complete of flocking controller. Work on control of multi-agent systems mainly focuses on linear systems with first-order [3] or second-order [4] dynamics, and the work on Vertical Take-Off and Landing (VTOL) is immediately inspired by the recent results in coordinated control of multi-agent systems. Some nonlinear or under-actuated models are also used for underwater vehicles [5] and wheeled robots [6]. Furthermore, the second-order nonholonomy in hovercraft systems of flocking is also considered [7].

However, only a little work has been done on nonlinear systems with nonlinear couplings and under-actuated systems. Under-actuated systems are those with fewer inputs than their degrees of freedom. Controllability, for instance, which is usually implied in systems with full control, is not easy to determine in an under-actuated system. Control synthesis for an under-actuated system is also more complex than that for a system with full actuated features [8][9]. Formation control of VTOL Unmanned Aerial Vehicles (UAVs) with communication delays are presented in [10] and formation control of VTOL UAVs without linear velocity measurements are required to track a desired reference linear velocity and maintain a desired formation [11]. In [12], a fault tolerant control approach is proposed for the formation control system of UAVs. In addition, several other methods have also been applied to multiple agents. In [13], a distributed smooth time-varying feedback control law is proposed for multiple mobile robots. In [14], formation control of several mobile robots is addressed with the dynamic feedback linearization technique.

In this paper, we propose a backstepping control framework for multiple VTOL agents on an undirected graph. We are particularly settling the following two issues: (1) flocking: VTOL agents cooperatively attain a desired formation shape with all their velocities converging to a common unspecified constant; (2) formation control: similar to the case in flocking, position and velocity converge to a desired timed trajectory. To the best of our knowledge, flocking control for multi-agents with nonlinear and under-actuated features has not been paid extensively attentions in the coordinated control problems. The main difficulty encountered lies in the fact that the under actuation renders many flocking/consensus results instead of deriving from simple agents with full-actuation and finding a suitable (nominal) Lyapunov function for the backstepping design. Compared with the results in [5]-[14], the proposed framework in this paper has the following advantages:

1. Each VTOL agent is a high-order system with nonlinear and under-actuated features, and flocking is achieved with proven stability via a designed controller by backstepping technique and the graph theory. It is more realistic than the linear or full actuated systems.
2. A systematic backstepping-based control law design method is obtained in this paper. With this design method, the control problems of multiple nonlinear and under-actuated systems can be fixed step by step.
3. The inputs of control are related to force and moment, which is more convenient and direct than velocity and angular velocity in [6][12].

In contrast, our result here fully takes high order nonlinear physics characters of agents into account rather than wheel robots [6] and second integration model [7]. With the challenge to achieve flocking control of multiple agents with under-actuated features, many consensus/flocking results inapplicable [5][6][13]. To address this challenge, we extend the consensus to explicitly incorporate with under-actuated feature via backstepping technique.

The remainder of this paper is organized as follows. In Section II, the flocking control model of VTOL and some preliminary assumptions are given. The main result, namely the flocking control scheme, is proposed via backstepping in Section III. To illustrate the effectiveness of the proposed control, we present in Section IV some simulation results. And finally, Section V concludes the presented results.

2. PROBLEM STATEMENT

2.1 VTOL System

Assume that there are a group of VTOL agents in the same structure. The VTOL aircraft depicted in Fig. 1 is a simplified planar model of a real vertical take-off and landing plane (e.g. the Harrier) [15].

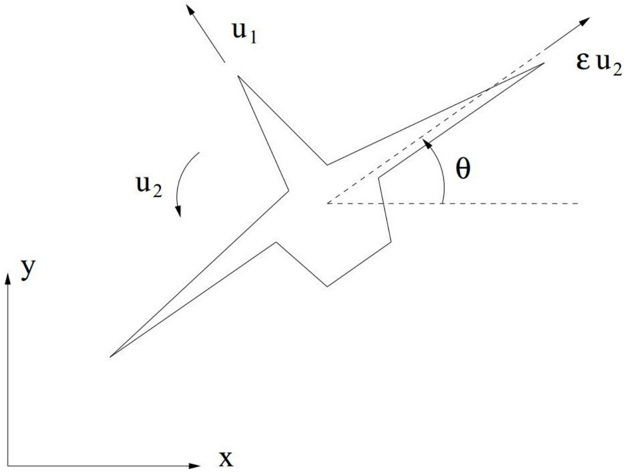


FIGURE 1: The VTOL aircraft.

In the figure, θ is the roll, the xy plane is spanned by the vertical axis and wing axis of aircraft, ε is a coefficient of moment and it is non-dimensional, andu_2 is the roll moment.

The dynamics of one of the VTOL aircraft agents is given in [15] as the following:

$$\begin{gathered} \dot{x}_1 = x_2 \\ \dot{x}_2 = u_1 \sin\theta + \varepsilon u_2 \cos\theta \\ \dot{y}_1 = y_2 \\ \dot{y}_2 = u_1 \cos\theta + \varepsilon u_2 \sin\theta - g \\ \dot{\theta} = \omega \\ \dot{\omega} = u_2 \end{gathered} \tag{1}$$

where(x_1, y_1) and (x_2, y_2) are the position and velocity of the aircraft, while θ and ω denote its angular and angular velocity, respectively. The VTOL aircraft is an under-actuated system with three degrees of freedom and two control inputs. In [15], it is assumed that $|\varepsilon|$ is relatively small and the VTOL aircraft is treated as a slightly non-minimum phase system.

In order to simplify the issue, a variables transform is introduced.

$$\begin{gathered} z_1 = x_1 - \varepsilon \sin\theta \\ z_2 = x_2 - \varepsilon \cos\theta d\theta \\ \zeta_1 = y_1 + \varepsilon(\cos\theta - 1) \\ \zeta_2 = y_2 - \varepsilon \sin\theta d\theta \\ \xi_1 = \theta \\ \xi_2 = \omega \end{gathered} \tag{2}$$

The model of the aircraft agent can be described as:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= -\tilde{u}_1 \sin\xi_1 \\ \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= \tilde{u}_1 \cos\xi_1 - g \\ \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= u_2 \end{aligned} \tag{3}$$

where$\tilde{u}_1 = u_1 - \varepsilon\xi_2^2$. Under such circumstances, the system can be divided into two subsystems, namely a position subsystem and an angular subsystem.

2.2 Graph and Matrix

The undirected graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$ consists of a vertex set $\mathcal{V} = \{1 \dots m\}$ and an edge set$\varepsilon \in \mathcal{V} \times \mathcal{V}$, where an edge is an unordered pair of distinct vertices. The definition of adjacency matrix$\mathcal{A} = \mathcal{A}(\mathcal{G}) = [a_{ij}]$, an $m \times m$ matrix, is given by$a_{ij} = 1$, if$(i,j) \in \varepsilon$, and $a_{ij} = 0$ otherwise. In this paper, if$(i,j) \in \varepsilon$, then aircrafts i, j are adjacent. Furthermore, we considered j is a neighbor of$i$. The neighbors of aircraft i are donated byN_i. If there is a path between any two vertices of $\mathcal{G}$, then $\mathcal{G}$ is connected. The degree of vertex i is obtained by$d_i = \sum_j a_{ij}$.

The incidence matrix, $\mathcal{B} = \mathcal{B}(\mathcal{G}) = [b_{ij}]$ of an undirected graph is a$\{0,1\}$. Matrix with rows and columns indexed by the vertices and edges of$\mathcal{G}$, respectively, so that $b_{ij} = 1$ if the vertexes i and j are connected by an edge, and $b_{ij} = 0$ otherwise. The laplacian matrix of $\mathcal{G}$ is $L_n = L_n(\mathcal{G}) = diag(d_1, \dots, d_m) - \mathcal{A}$. For the connected graph $\mathcal{G}$, $L_n = L_n(\mathcal{G}) := \mathcal{B}\mathcal{B}^T$ is symmetric and positive semi-definite [16]. A directed tree is a digraph, where every node has exactly one parent except the root node. A root node is a node that has a directed path to every other node, without any parent. A directed spanning tree is a directed tree that contains all nodes of the digraph. A digraph has a spanning tree if there is a directed spanning tree as a subset of the digraph [8]. For more theories of graph, refer to [16].

In this paper, we consider a group of networked aircraft agents with a connected undirected graph, in which each aircraft is regarded as a vertex, and each existing control interconnection between agents is regarded as an edge.

3. CONTROLLER DESIGN

In order to allow our problem to be more reliable, it is essential to employ the following assumptions.

Assumption 1: For $i = 1, \dots, m$, the position (x_i, y_i), the velocity (v_{xi}, v_{yi}), the angular information (ξ_{1i}, ξ_{2i}), the positions to its neighbors (x_j, y_j), velocities of its neighbors (v_{xj}, v_{yj}), and angular information of its neighbors (ξ_{1j}, ξ_{2j}) of the ith aircraft for $j \in N_i$ are available for the design of the consensus protocol μ_i.

Assumption 2: For $i = 1, \dots, m$, $|v_{xd}| + |v_{yd}| > \rho$, $\rho > 0$, where ρ is a constant.

Assumption 3: The graph $\mathcal{G}$ has a spanning tree and the laplacian matrix of $\mathcal{G}$ is balanced.

Remark 1: The physical implication of assumption 1 is that the communications of each neighbor are reliable and delay and noise is not considered here. Assumption 2 is proposed that the velocity of every agent need to be positive and valid.

The objective of controller design is to address a distributed consensus protocol, which uses its own state $(x_i, y_i, v_{xi}, v_{yi}, \xi_{1i}, \xi_{2i})$, its neighbors' state $(x_j, y_j, v_{xj}, v_{yj}, \xi_{1j}, \xi_{2j})$, desired geometric pattern (p_{xi}, p_{yi}), the desired trajectory (x_d, y_d), and the desired velocity (v_{xd}, v_{yd}), so that:

$$\begin{array}{c}\lim_{t\to\inf\infty}((x_i, y_i) - (x_j, y_j)) \to (p_{xi}, p_{yi}) - (p_{xj}, p_{yj}) \\ \lim_{t\to\inf\infty}((v_{xi}, v_{yi}) - (v_{xj}, v_{yj})) \to (0, 0) \\ \lim_{t\to\inf\infty}((\frac{\sum x_i}{m}, \frac{\sum y_i}{m}) - (x_d, y_d)) \to (0, 0)\end{array} \tag{4}$$

Remark 2: (4) means that groups of aircraft agents reach an agreement that they converge in a given pattern at an identical velocity and the center of their formation moves along the desired trajectory.

We develop a non-linear consensus protocol by backstepping methodology. The system (3) is divided into two subsystems. The first subsystem with state (z_1, ζ_1) is called $z_1\zeta_1$-system, and the second subsystem with state ξ_1 is called ξ-system.

Step 1: $(\tilde{z}_1, \tilde{\zeta}_1)$-subsystem
For every aircraft agent, the desired trajectory (x_d, y_d) in (1) under transform (2) can be obtained:

$$\begin{array}{c}z_{1d} := x_d - \varepsilon \sin\theta \\ \zeta_{1d} := y_d + \varepsilon(\cos\theta - 1)\end{array} \tag{5}$$

Define$\tilde{z}_1(t) = z_{xd}(t) - z_1(t) - p_x$, $\tilde{\zeta}_1(t) = \zeta_{1d}(t) - \zeta_1(t) - p_y$, where$\tilde{z}_1 = [\tilde{z}_{11}(t), \tilde{z}_{12}(t), \ldots, \tilde{z}_{1m}(t)]^T$, $\tilde{\zeta}_1 = [\tilde{\zeta}_{11}(t), \tilde{\zeta}_{12}(t), \ldots, \tilde{\zeta}_{1m}(t)]^T$, $z_{xd} = [z_{xd1}(t), z_{xd2}(t), \ldots, z_{xdm}(t)]^T$, $z_{yd} = [z_{yd1}(t), z_{yd2}(t), \ldots, z_{ydm}(t)]^T$. This way of definition is so as to similar variables in this paper.

Then we can obtain:

$$\begin{array}{c}\dot{\tilde{z}}_1 = \dot{z}_{xd}(t) - \dot{z}_1 = \dot{z}_{xd}(t) - z_2 \\ \dot{\tilde{\zeta}}_1 = \dot{z}_{yd}(t) - \dot{\zeta}_1 = \dot{z}_{yd}(t) - \zeta_2\end{array} \tag{6}$$

Theorem 1: The $(\tilde{z}_1, \tilde{\zeta}_1)$-subsystem (6) with z_2 is taken as its control input. For any constant $\beta_{1i} \geq 0$,$\sum_{i=0}^{m} \beta_{1i} \geq 0$,$a_{ij} \geq 0$. We define the Lyapunov function candidates as:

$$V_1(\tilde{z}_1) = \frac{1}{2}\tilde{z}_1^T \tilde{z}_1 \tag{7}$$

Distributed control input $z_{2i} = \dot{z}_{xdi} + \beta_{1i}\tilde{z}_{1i} + \sum_{i\in m} a_{ij}(\tilde{z}_{1i}(t) - \tilde{z}_{ij}(t))$ can make the following satisfied.

$$\dot{V}_1(\tilde{z}_1) \leq 0 \tag{8}$$

Proof: All the edges are in graph $\mathcal{G}$with$i = 1, \ldots, m$. Under assumption 3, the laplacian matrix$L_n$of $\mathcal{G}$ is balanced. It can be obtained that $(L_n + L_n^T)$is positive semi definite by using graph theory.

Substituting z_{2i} into (6),

$$\dot{\tilde{z}}_{1i} = -\beta_{1i}\tilde{z}_{1i} - \sum_{i\in m} a_{ij}(\tilde{z}_{1i}(t) - \tilde{z}_{1j}(t)) \tag{9}$$

Differentiate (7) and substituting (8) into (9), then

$$\dot{V}_1 = -\sum_{i=1}^{m} \beta_{1i}\tilde{z}_{1i}^2 - \frac{1}{2}\tilde{z}_1^T(L_n + L_n^T)\tilde{z}_1 \tag{10}$$

Therefore $\dot{V}_1$is negative semi definite.

So as to $\tilde{\zeta}_1$-subsystem (6) with$\zeta_{2i} = \dot{z}_{ydi} + \beta_{2i}\tilde{\zeta}_{1i} + \sum_{i\in m} a_{ij}(\tilde{\zeta}_{1i}(t) - \tilde{\zeta}_{1j}(t))$, we can obtain the same result.

Step 2: $(\tilde{z}_2, \tilde{\zeta}_2)$-subsystem
Define $\tilde{z}_{2i} = z_{2i} - \dot{z}_{xdi} - \beta_{1i}\tilde{z}_{1i} - \sum_{i\in m} a_{ij}(\tilde{z}_{1i}(t) - \tilde{z}_{ij}(t))$, and $\tilde{\zeta}_{2i} = \zeta_{2i} - \dot{z}_{ydi} - \beta_{2i}\tilde{\zeta}_{1i} - \sum_{i\in m} a_{ij}(\tilde{\zeta}_{1i}(t) - \tilde{\zeta}_{ij}(t))$, then

$$\begin{array}{l} \dot{\tilde{z}}_{2i} = \dot{z}_{2i} - \beta_{1i}\dot{\tilde{z}}_{1i} - \sum_{i\in m} a_{ij}(\dot{\tilde{z}}_{1i}(t) - \dot{\tilde{z}}_{ij}(t)) \\ \dot{\tilde{\zeta}}_{2i} = \dot{\zeta}_{2i} - \beta_{2i}\dot{\tilde{\zeta}}_{1i} - \sum_{i\in m} a_{ij}(\dot{\tilde{\zeta}}_{1i}(t) - \dot{\tilde{\zeta}}_{ij}(t)) \end{array} \tag{11}$$

where, $\ddot{z}_{xd} = 0$, $\ddot{z}_{yd} = 0$.

Define$\delta_1 = -\tilde{u}_1 \sin\xi_1$, $\gamma_1 = \tilde{u}_1 \cos\xi_1$.

Theorem 2: The $(\tilde{z}_2, \tilde{\zeta}_2)$-subsystem with δ_1, γ_1 is taken as its control input. For any constant $\beta_{1i} \geq 0$, $\sum_{i=0}^{m} \beta_{1i} \geq 0$,$\beta_{2i} \geq 0$, $\sum_{i=0}^{m} \beta_{2i} \geq 0$, $a_{ij} \geq 0$, and $r_1(i) > 0$, $r_2(i) > 0$. We define the Lyapunov function candidates as:

$$V_2(\tilde{z}_1, \tilde{z}_2, \tilde{\zeta}_1, \tilde{\zeta}_2) = \frac{1}{2}\tilde{z}_1^T z_1 + \frac{1}{2}\tilde{\zeta}_1^T \zeta_1 + \frac{1}{2}\tilde{z}_2^T z_2 + \frac{1}{2}\tilde{\zeta}_2^T \zeta_2 \tag{12}$$

Distributed control input

$$\delta_{1i} = (\beta_{1i}(\dot{z}_{xdi} - z_{2i}(t)) + \sum_{i\in m} a_{ij}(\dot{\tilde{z}}_{1i}(t) - \dot{\tilde{z}}_{ij}(t)) - r_1(t)\sum_{i\in m} a_{ij}(\dot{\tilde{z}}_{2i}(t) - \dot{\tilde{z}}_{2j}(t))) \tag{13}$$

$$\gamma_{1i} = (\beta_{2i}(\dot{z}_{ydi} - \zeta_{2i}(t)) + \sum_{i\in m} a_{ij}(\dot{\tilde{\zeta}}_{1i}(t) - \dot{\tilde{\zeta}}_{ij}(t)) - r_2(t)\sum_{i\in m} a_{ij}(\dot{\tilde{\zeta}}_{2i}(t) - \dot{\tilde{\zeta}}_{2j}(t))) + g \tag{14}$$

can make the following satisfied.

$$\dot{V}_2(\tilde{z}_1, \tilde{z}_2, \tilde{\zeta}_1, \tilde{\zeta}_2) \leq 0 \tag{15}$$

Proof: Substituting$\dot{z}_2 = \delta_1$, $\dot{\zeta}_2 = \gamma_1$ into (11), we can obtain

$$\begin{array}{l} \dot{\tilde{z}}_{2i} = -r_1(t)\sum_{i\in m} a_{ij}(\dot{\tilde{z}}_{2i}(t) - \dot{\tilde{z}}_{2j}(t)) \\ \dot{\tilde{\zeta}}_{2i} = -r_2(t)\sum_{i\in m} a_{ij}(\dot{\tilde{\zeta}}_{2i}(t) - \dot{\tilde{\zeta}}_{2j}(t)) \end{array} \tag{16}$$

Differentiate (12) and substituting (15) into (16), then

$$\begin{array}{c} V_2 = -\sum_{i=1}^{m} \beta_{1i}\tilde{z}_{1i}^2 - \frac{1}{2}\tilde{z}_1^T(L_n + L_n^T)\tilde{z}_1 \\ -\frac{1}{2}\tilde{\zeta}_1^T(L_n + L_n^T)\tilde{\zeta}_1 - \frac{1}{2}\tilde{z}_2^T\Lambda_1(L_n + L_n^T)\tilde{z}_2 - \frac{1}{2}\tilde{\zeta}_2^T\Lambda_2(L_n + L_n^T)\tilde{\zeta}_2 \end{array} \tag{17}$$

where, $\Lambda_1 = diag(r_1(1), r_1(2), \ldots, r_1(m))$, and $\Lambda_2 = diag(r_2(1), r_2(2), \ldots, r_2(m))$.

For that $r_1(i) > 0$, $r_2(i) > 0$, Λ_1 and Λ_2 are positive define. With assumption 3 hold, $(L_n + L_n^T)$ is positive semi definite. Then$\Lambda_1(L_n + L_n^T)$ and $\Lambda_2(L_n + L_n^T)$ are positive semi define.

Therefore, $\dot{V}_2$ is negative semi definite.

Then we can obtain

$$\tilde{u}_1 = \sqrt{\delta_1^2 + \gamma_1^2} \tag{18}$$

as the control input in (3).

Define$\bar{\xi}_1 = \arctan \frac{-\delta_1}{\gamma_1 + g} + 2k\pi$, $k = 1, 2, \ldots, n$. In order to obtain the ultimate control input, we use backstepping methodology (refer to [17]). With the control inputs (13), the system in (11) can be written as

$$\begin{aligned}
\begin{pmatrix} \dot{\tilde{z}}_{2i} \\ \dot{\tilde{\zeta}}_{2i} \end{pmatrix} &= \begin{pmatrix} \dot{z}_2 - \beta_{1i}\dot{\tilde{z}}_{1i} - \sum_{i\in m} a_{ij}(\dot{\tilde{z}}_{1i}(t) - \dot{\tilde{z}}_{ij}(t)) \\ \dot{\zeta}_2 - \beta_{2i}\dot{\tilde{\zeta}}_{1i} - \sum_{i\in m} a_{ij}(\dot{\tilde{\zeta}}_{1i}(t) - \dot{\tilde{\zeta}}_{ij}(t)) \end{pmatrix} \\
&= \begin{pmatrix} -\delta_{1i} - \beta_{1i}\dot{\tilde{z}}_{1i} - \sum_{i\in m} a_{ij}(\dot{\tilde{z}}_{1i}(t) - \dot{\tilde{z}}_{ij}(t)) \\ \gamma_{1i} - \beta_{2i}\dot{\tilde{\zeta}}_{1i} - \sum_{i\in m} a_{ij}(\dot{\tilde{\zeta}}_{1i}(t) - \dot{\tilde{\zeta}}_{ij}(t)) - g \end{pmatrix} \\
&\quad + \begin{pmatrix} -\tilde{u}_{1i}\sin\xi_{1i} + \delta_{1i} \\ \tilde{u}_{1i}\cos\xi_{1i} - \gamma_{1i} \end{pmatrix} \\
&= \begin{pmatrix} A \\ B \end{pmatrix} + \tilde{u}_{1i} \begin{pmatrix} \sin\bar{\xi}_{1i} & -\cos\bar{\xi}_{1i} \\ \cos\bar{\xi}_{1i} & \sin\bar{\xi}_{1i} \end{pmatrix} \begin{pmatrix} \frac{\sin\bar{\xi}_{1i}}{\tilde{\xi}_{1i}} & \frac{\cos\bar{\xi}_{1i} - 1}{\tilde{\xi}_{1i}} \end{pmatrix}^T \tilde{\xi}_{1i}
\end{aligned} \tag{19}$$

where, $\tilde{\xi}_1 = \xi_1 - \bar{\xi}_1$, $A = -r_1(t)\sum_{i\in m} a_{ij}(\tilde{z}_{2i}(t) - \tilde{z}_{2j}(t))$, and $B = -r_2(t)\sum_{i\in m} a_{ij}(\tilde{\zeta}_{2i}(t) - \tilde{\zeta}_{2j}(t))$.

Then,

$$\dot{\tilde{\xi}}_1 = \dot{\xi}_1 - \dot{\bar{\xi}}_1 = \xi_2 - \dot{\bar{\xi}}_1 \tag{20}$$

Firstly, we calculate$\dot{\bar{\xi}}_1$.

Differentiating $\delta_1$andγ_1, it can be obtained that

$$\begin{aligned}
\dot{\delta}_1 &= \dot{\tilde{u}}_1 \sin\bar{\xi}_1 + \tilde{u}_1\dot{\bar{\xi}}_1 \cos\bar{\xi}_1 \\
\dot{\gamma}_1 &= \dot{\tilde{u}}_1 \cos\bar{\xi}_1 - \tilde{u}_1\dot{\bar{\xi}}_1 \sin\bar{\xi}_1
\end{aligned} \tag{21}$$

Then we can obtain

$$\dot{\bar{\xi}}_1 = \frac{\dot{\delta}_1 \cos\bar{\xi}_1 - \dot{\gamma}_1 \sin\bar{\xi}_1}{\sqrt{\delta_1^2 + \gamma_1^2}} \tag{22}$$

Remark 3: Under assumption 2, the desired velocity of every aircraft would not converge to zero at the same time. Furthermore, if (4) is reliable, $\delta_1$and γ_1 will not to be zero at the same time.
Step 3: $\tilde{\xi}_1$-subsystem

Theorem 3: The $\tilde{\xi}_1$-subsystem with ξ_2 is taken as its control input. For any constant$\beta_{3i} \geq 0$, $\sum_{i=0}^{m} \beta_{3i} \geq 0$. We define the Lyapunov function candidates as

$$V_3(\tilde{\xi}_1) = \frac{1}{2}\sum_{i\in m} \tilde{z}_{1i}^2 + \frac{1}{2}\sum_{i\in m} \tilde{z}_{2i}^2 + \frac{1}{2}\sum_{i\in m} \tilde{\zeta}_{1i}^2 + \frac{1}{2}\sum_{i\in m} \tilde{\zeta}_{2i}^2 + \frac{1}{2}\sum_{i\in m} \tilde{\xi}_{1i}^2 \tag{23}$$

Distributed control input

$$\xi_{2i} = -\begin{pmatrix} z_{2i} & \zeta_{2i} \end{pmatrix} \begin{pmatrix} \sin\bar{\xi}_{1i} & -\cos\bar{\xi}_{1i} \\ \cos\bar{\xi}_{1i} & \sin\bar{\xi}_{1i} \end{pmatrix} \begin{pmatrix} \frac{\sin\tilde{\xi}_{1i}}{\tilde{\xi}_{1i}} \\ \frac{\cos\tilde{\xi}_{1i}-1}{\tilde{\xi}_{1i}} \end{pmatrix} \tilde{u}_{1i} - \beta_{3i}\tilde{\xi}_{1i} + \dot{\bar{\xi}}_{1i} \tag{24}$$

can make the following to be satisfied.

$$\dot{V}_3(\tilde{\xi}_1) \leq 0 \tag{25}$$

Proof: Substituting (23) into (18) and (19), we have

$$\dot{\tilde{\xi}}_{1i} = -\begin{pmatrix} z_{2i} & \zeta_{2i} \end{pmatrix} \begin{pmatrix} \sin\bar{\xi}_{1i} & -\cos\bar{\xi}_{1i} \\ \cos\bar{\xi}_{1i} & \sin\bar{\xi}_{1i} \end{pmatrix} \begin{pmatrix} \frac{\sin\tilde{\xi}_{1i}}{\tilde{\xi}_{1i}} \\ \frac{\cos\tilde{\xi}_{1i}-1}{\tilde{\xi}_{1i}} \end{pmatrix} \tilde{u}_{1i} - \beta_{3i}\tilde{\xi}_{1i} \tag{26}$$

Differentiate (22) and substituting (25) into (23), then

$$\begin{aligned} V_3 = -\textstyle\sum_{i=1}^{m} \beta_{1i}\tilde{z}_{1i}^2 - \frac{1}{2}\tilde{z}_1^T(L_n+L_n^T)\tilde{z}_1 - \frac{1}{2}\tilde{\zeta}_1^T(L_n+L_n^T)\tilde{\zeta}_1 - \frac{1}{2}\tilde{z}_2^T\Lambda_1(L_n+L_n^T)\tilde{z}_2 \\ -\tfrac{1}{2}\tilde{\zeta}_2^T\Lambda_2(L_n+L_n^T)\tilde{\zeta}_2 - \textstyle\sum_{i=1}^{m}\beta_{3i}\tilde{\xi}_{1i}^2 \end{aligned} \tag{27}$$

Therefore $\dot{V}_3$ is negative semi definite.

Step 4: $\tilde{\xi}_2$-subsystem
Define$\tilde{\xi}_{2i} = \xi_{2i} + \beta_{3i}\tilde{\xi}_{1i} - \dot{\bar{\xi}}_{1i}$, then

$$\dot{\tilde{\xi}}_{2i} = \dot{\xi}_{2i} + \beta_{3i}\dot{\tilde{\xi}}_{1i} - \ddot{\bar{\xi}}_{1i} = u_{2i} - \ddot{\bar{\xi}}_{1i} + \beta_{3i}(\xi_{2i} - \dot{\bar{\xi}}_{1i}) \tag{28}$$

Theorem 4: The $\tilde{\xi}_2$-subsystem with u_2 is taken as its control input. For any constant $\beta_{4i} \geq 0$, $\sum_{i=0}^{m}\beta_{4i} \geq 0$. We define the Lyapunov function candidates as

$$V_4(\tilde{\xi}_2) = \frac{1}{2}\sum_{i\in m}\tilde{z}_{1i}^2 + \frac{1}{2}\sum_{i\in m}\tilde{z}_{2i}^2 + \frac{1}{2}\sum_{i\in m}\tilde{\zeta}_{1i}^2 + \frac{1}{2}\sum_{i\in m}\tilde{\zeta}_{2i}^2 + \frac{1}{2}\sum_{i\in m}\tilde{\xi}_{1i}^2 + \frac{1}{2}\sum_{i\in m}\tilde{\xi}_{2i}^2 \tag{29}$$

Distributed control input

$$u_{2i} = \ddot{\bar{\xi}}_{1i} - (\beta_{3i} + \beta_{4i})\xi_{2i} - \beta_{3i}\dot{\bar{\xi}}_{1i} \tag{30}$$

can make the following satisfied.

$$\dot{V}_4(\tilde{\xi}_2) \leq 0 \tag{31}$$

Proof: Substituting (29) into (27), and differentiating (28), we have

$$\begin{aligned} V_4 = -\textstyle\sum_{i=1}^{m} \beta_{1i}\tilde{z}_{1i}^2 - \frac{1}{2}\tilde{z}_1^T(L_n+L_n^T)\tilde{z}_1 - \frac{1}{2}\tilde{\zeta}_1^T(L_n+L_n^T)\tilde{\zeta}_1 - \frac{1}{2}\tilde{z}_2^T\Lambda_1(L_n+L_n^T)\tilde{z}_2 \\ -\tfrac{1}{2}\tilde{\zeta}_2^T\Lambda_2(L_n+L_n^T)\tilde{\zeta}_2 - \textstyle\sum_{i=1}^{m}\beta_{3i}\tilde{\xi}_{1i}^2 - \sum_{i=1}^{m}\beta_{3i}\tilde{\xi}_{2i}^2 \end{aligned} \tag{32}$$

Therefore $\dot{V}_4$ is negative semi definite.

Remark 4: With the control input (30) and (18), the system can be globally asymptotically stable and (4) is reliable. In another word, consensus protocol is achieved.

Remark 5: The controller proposed in this paper is general that can also wide extend to other nonlinear under-actuated systems with the same structures, such as under-actuated underwater vehicles, mobile robots [6], UAVs [11].

4. SIMULATION

In order to verify the distributed controller designed in this paper, some numerical simulations are given in this section. There are five aircraft agents and their state information could be obtained without delay. The desired trajectory is

$$(x_d, y_d) = (5, 2.5\sin(t) + 10) \tag{33}$$

and the desired velocity is

$$(v_{xd}, v_{yd}) = (0, 2.5\cos(t)) \tag{34}$$

Pattern is given in Fig.2, the initial positions of them are $p_{10} = [2,4]$, $p_{20} = [4,0]$, $p_{30} = [-2,2]$, $p_{40} = [0,4]$, and $p_{50} = [-4,-2]$. The given patterns are $p_1 = [-1.1756, -1.618]$, $p_2 = [-1.902, 0.618]$, $p_3 = [1.1756, -1.618]$, $p_4 = [0,2]$, and $p_5 = [1.9021, 0.618]$.

A graph of five aircraft agents is given in Fig.2.

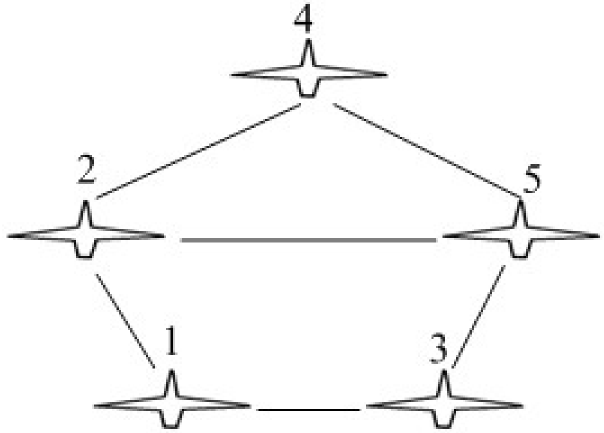


FIGURE 2: Graph of Five Aircraft Agents.

The incident matrix B and the Laplace matrix L_n of the graph in Fig. 2 are

$$B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{pmatrix}$$

$$L_n = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & 0 & -1 & -1 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

The calculation results are$|B| = 2.149$, $\lambda_2(L_n) = 1.382$, $|L_{n1}B| = 4.5826$, $|L_{n2}B| = 7.0711$, $|L_{n3}B| = 4.5826$, $|L_{n4}B| = 4.4721$, and $|L_{n5}B| = 7.0711$. Choose$\beta_{1i} = 1.1$, $\beta_{2i} = 1.1$, $\beta_{3i} = 0.5$, $\beta_{4i} = 0.5$, $\beta_{1i} = 1.1$, $r_1(i) = 1$, $r_2(i) = 1$, $i = 1, \ldots, 5$.

Under the condition that every aircraft agent can obtain the state information as shown in figure 2, and the desired trajectory is available. The distributed controller was designed in section III. The trajectories of agents were presented in figure3.

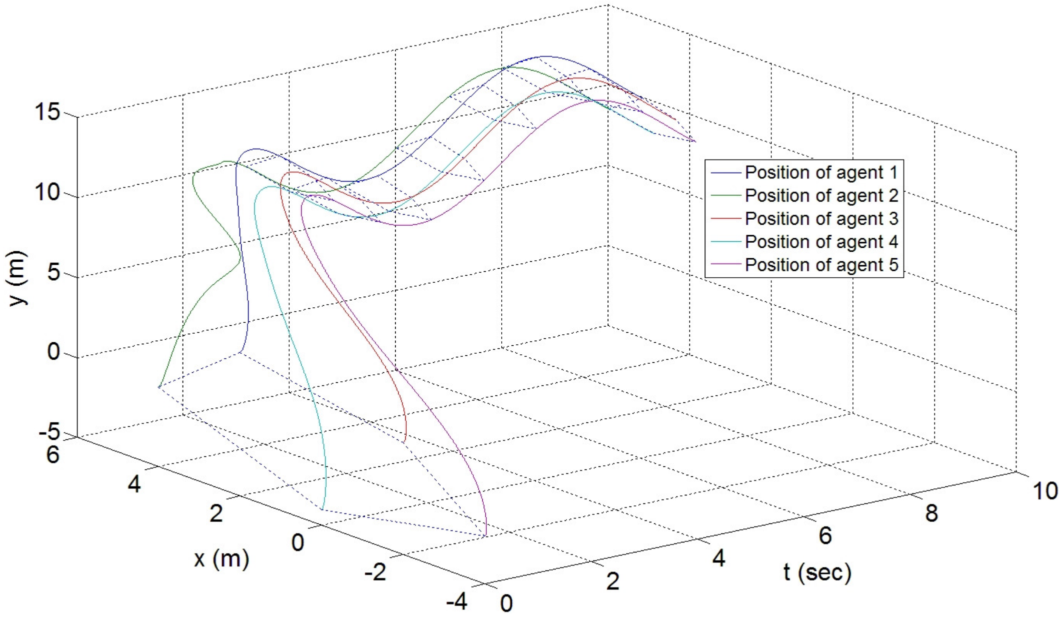


FIGURE 3: The Trajectories of Agents.

Through simulation, two objectives are solved: flocking, and flocking with centroid tracking. The aircraft agents initiate and keep the desired formation, and they follow the desired trajectories after about 5s. Then, the absolute distance between any two aircrafts converges exactly to the desired formation size. As presented in Fig.3, five aircraft agents follow the desired path. Fig.4 shows the center of pattern and the desired trajectory, and Fig.5 shows the formation error between the center of pattern and the desired trajectory. The controls provide convergence of the desired trajectory after a short transience corresponding to the initialization of the formation and the results also prove the effectiveness of the proposed techniques.

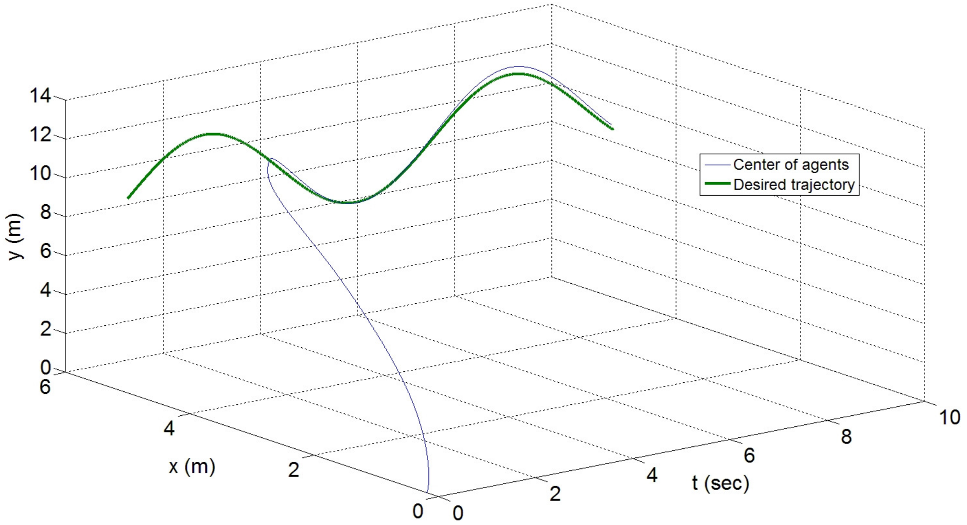


FIGURE 4: The Center of Pattern and Desired Trajectory.

After staying at the initial position for the first second, the aircraft agents initiate the formation following the desired trajectory. The aircrafts achieve perfect positioning after a smooth transience, and the position errors reach a constant in finite time for some tracking error in y axis.

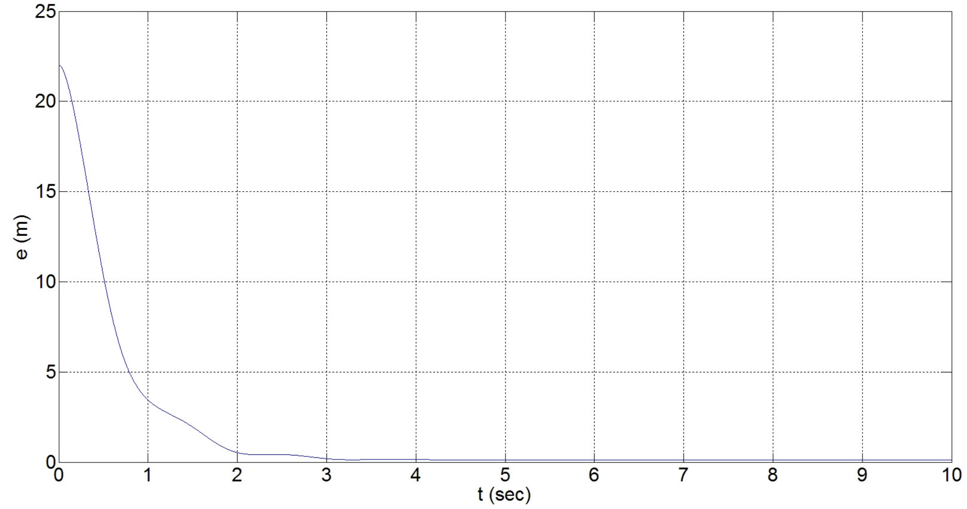


FIGURE 5: The Formation Error.

Fig.6 and Fig.7 show the velocities of agents in axisx, y , respectively. From those figures we can see that the velocities of agents in each axis converge to the center of velocity after 5s, and the positions of agents in each axis maintain a fixed distance at the same time.

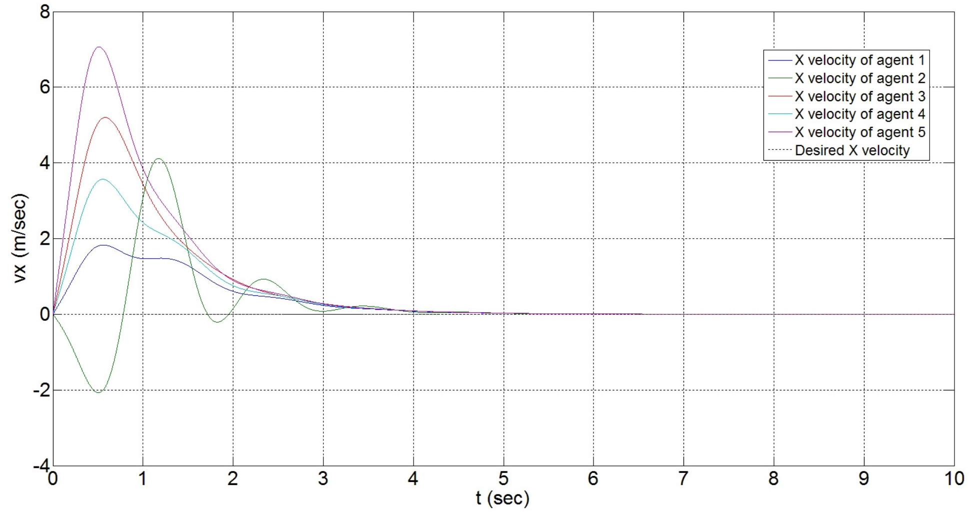


FIGURE 6: Velocities of Agents in axis$x$.

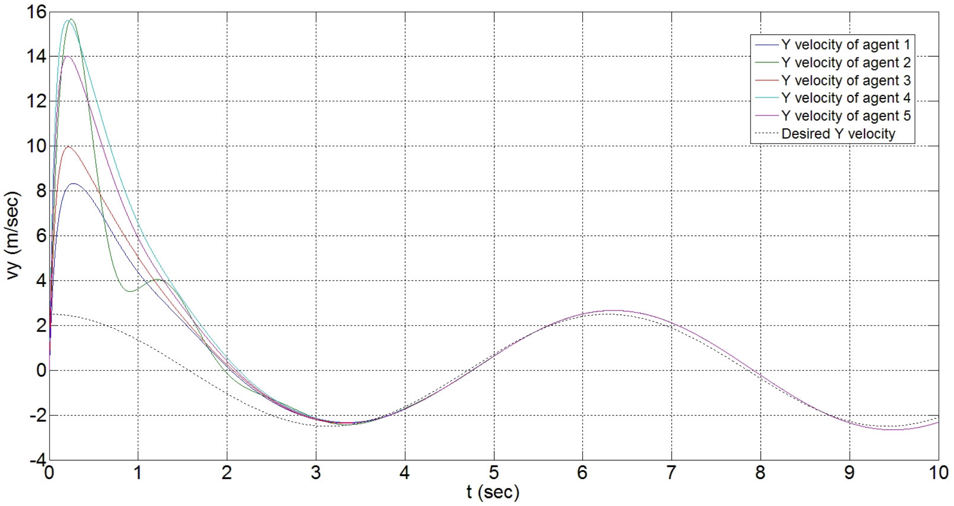


FIGURE 7: Velocities of Agents in axis$y$.

5. CONCLUSION

In this paper, a novel disturbed flocking controller was applied to the formation flight of multiple VTOL agents. Flocking is achieved with designed controller based on the backstepping technique and the graph theory. Simulation results show that the controller exhibits smooth and continuous effects under arbitrary initial conditions and the tracking stage.

The future work include consider the delay between neighbors and apply the controller to actuator malfunction. In addition, the controllers in this paper are full state measurements and it is essential to obtain proper controller using partial states. So it is worthwhile to obtain a control law without the requirement of angular velocity and linear velocity measurements. It is also meaning to extend to other classes of under-actuated systems.

6. REFERENCES

[1] X. K. Wang, J. H. Qin, C. B. Yu, “ISS Method for Coordination Control of Nonlinear Dynamical Agents Under Directed Topology”, IEEE Transactions on Cybernetics, Volume:44, Issue:10, pp. 1832-1845. 2014.

[2] Y. P. Tian and Y. Zhang, “High-order consensus of heterogeneous multi-agent systems with unknown communication delays”, Automatica, vol. 48, no. 6, pp. 1205-1212, 2012.

[3] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control”, IEEE Control Syst. Mag., vol. 27, no. 2, pp. 71-82, Apr. 2007.

[4] R. O. Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems”, IEEE, vol. 95, no. 1, pp. 215-233, Jan. 2007.

[5] W. B. Dunbar, R. M. Murray, “Model predictive control of coordinated multi-vehicles formations”, Proc. of the 41stIEEE Conference on Decision and Control, Las Vegas, Nevada, USA (pp.4631-4636),2002.

[6] W. Dong, “Flocking of multiple mobile robots based on backstepping”, IEEE Trans. Syst., Man Cybern. B, Cybern, vol. 41, no. 2, pp. 414-424, Apr. 2011.

[7] J. Qin, W. X. Zheng, and H. Gao, “Consensus of multiple second-order vehicles with a time-varying reference signal under directed topology”, Automatica, vol. 47, no. 9, pp. 1983-1991, 2011.

[8] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach", IEEE Trans. Autom.Control, vol. 57, no. 1, pp. 33-48, Jan. 2012.

[9] Z. Daniel, F. Antonio, A. Frank, et al, "Rigidity Maintenance Control for Multi-Robot Systems", Robotics: Science and System (RSS) 2012, Sydney, Australia.

[10] A. Tayebi, "Formation control of VTOL Unmanned Aerial Vehicles with communication delays", Automatica, Volume 47, Issue 11,November , 2011.

[11] A. Abdessameud, A. Tayebi, "Formation Control of VTOL UAVs", Motion Coordination for VTOL Unmanned Aerial Vehicles Advances in Industrial Control 2013, pp 105-127.

[12] M. Qian, B. Jiang, D. Xu, "Fault tolerant control scheme design for the formation control system of unmanned aerial vehicles", Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering August 9, 2013.

[13] H. Yamaguchi, "A distributed motion coordination strategy for multiple nonholonomic mobile robots in cooperative hunting operations", Robot. Auton. Syst., vol. 43, no. 4, pp. 257-282, Jun. 2003.

[14] J. Lawton, R. W. Beard, and B. Young, "A decentralized approach to formation maneuvers", IEEE Trans. Robot. Autom., vol. 19, no. 6, pp.933-941, Dec. 2003.

[15] J. Hauser, S. Sastry, and G. Meyer, "Nonlinear control design for slightly non-minimum phase systems", Automatica, vol. 28:pp.665-679,1992.

[16] J. H. Qin, C. B. Yu, H. J. Gao, X. K. Wang, "Leaderless consensus control of dynamical agents under directed interaction topology", Decision and Control and European Control Conference(CDC-ECC), 2011.

[17] Y. Zhu, Y. R. Cong, X. K. Wang, "Consensus labialization for Continuous-Time High-Order Multi-agent Systems with Time-Varying Delays", Mathematical Problems in Engineering Volume 2013.

A Path Planning Technique For Autonomous Mobile Robot Using Free-Configuration Eigenspaces

Shyba Zaheer *s.shyba@gmail.com*
Department of Electrical & Electronics Engineering
T.K.M. College of Engineering
Kerala, India

Tauseef Gulrez *gtauseef@ieee.org*
Virtual and Simulations of Reality (ViSOR) Lab,
Department of Computing, Macqaurie University
2109 NSW, Sydney, Australia.

Abstract

This paper presents the implementation of a novel technique for sensor based path planning of autonomous mobile robots. The proposed method is based on finding free-configuration eigen spaces (FCE) in the robot actuation area. Using the FCE technique to find optimal paths for autonomous mobile robots, the underlying hypothesis is that in the low-dimensional manifolds of laser scanning data, there lies an eigenvector which corresponds to the free-configuration space of the higher order geometric representation of the environment. The vectorial combination of all these eigenvectors at discrete time scan frames manifests a trajectory, whose sum can be treated as a robot path or trajectory. The proposed algorithm was tested on two different test bed data, real data obtained from Navlab SLAMMOT and data obtained from the real-time robotics simulation program Player/Stage. Performance analysis of FCE technique was done with existing four path planning algorithms under certain working parameters, namely computation time needed to find a solution, the distance travelled and the amount of turning required by the autonomous mobile robot. This study will enable readers to identify the suitability of path planning algorithm under the working parameters, which needed to be optimized. All the techniques were tested in the real-time robotic software Player/Stage. Further analysis was done using MATLAB mathematical computation software.

Keywords: Free-configuration Space, Eigenvector, Motion Planning, Trajectory Planning.
.

1. INTRODUCTION

Motion planning is one of the most important tasks in intelligent control of an autonomous mobile robot (AMR). It is often decomposed into path-planning and trajectory planning. Path planning is referred as to generate a collision free path in an environment with obstacles. Whereas, trajectory planning schedule the movement of a mobile robot along the planned path. Based on the availability of information about environment, the path-planning algorithms are divided into two categories, namely offline and online. Offline path planning of robots in environments uses complete information about stationary obstacles and trajectory of moving obstacles, which are known in advance. This method is also known as global path planning. When complete information about environment is not available in advance, the mobile robot gets information through sensors, as it moves through the environment. This is known as online or local path planning. Essentially, online path planning begins its initial path offline but switches to online mode when it discovers new changes in obstacle scenario. Classical approaches used in online path planning are Potential Filed approach (PF), collision–cone approach, and vector field histogram (VFH) method. Khatib [1] proposed the Artificial Potential Field (APF) approach which is popular in mobile robotics. This approach is known for its mathematical elegance and simplicity as the path is found with very little computation. However, the drawback of this algorithm is that the robot may become stagnant or trapped when there is a cancellation of equal magnitudes of attractive and repulsive forces. Till date many variants of the potential field approach like escape-force algorithm [2], trap recovery model, adaptive virtual target algorithm etc. have been proposed. Path planning problems can also be solved by VFH approach [3]. At every instant, a polar histogram is generated to represent the polar density of obstacles around a robot. The robot's steering direction is chosen based on the least polar density and closeness to the goal. In a given environment, the polar histogram must be regularly regenerated for every instant and hence this method is suited for environments with sparse moving obstacles. Another commonly used online approach is based on the collision cone concept [4]. The Collision of a robot can be averted if the relative velocity of it with respect to a particular obstacle falls exterior to the collision cone. Another online approach for obstacle avoidance is dynamic windows approach [5]. The dynamic window contains the feasible linear and angular velocities taking into consideration the acceleration capabilities of a robot. Then the velocity at the next instant is optimized for obstacle avoidance subject to vehicle dynamics. With classical techniques, the optimum result requires more computational time due to incomplete information of the environment. This classical approach can be combined with heuristic approaches like genetic algorithm (GA) and particle swarm optimization (PSO) [6]. Another class of online path planning algorithms are sampling based path planning algorithms like rapidly evolving random trees (RRT) and probabilistic roadmap methods PRM [7]. The idea of connecting points sampled randomly from the state space is essential in both RRT and PRM approaches.

A limitation of the classical planning algorithm is that a complete model of the environment is needed before the planner can proceed. A solution to this problem is a sensor based path planner. Sensor based planner allows robots to work autonomously in unknown environments. Specifically the robot is able to move to a given configuration without prior knowledge of the environment. The motion of the robot is generated step by step while more and more knowledge about the environment is accumulated incrementally. A great variety of sensors that can be used for robots includes tactile sensor, vision sensor, sonar sensor etc. A well-known sensor based technique is "Bug" family. These path planners incorporates sensing and path planning as an integrated solution. Bug algorithms assume only local knowledge of the environment and a global goal. The most commonly used sensor based robot path planners are Bug1, Bug2, TangentBug, DistBug, and VisBug [8] .Bug 1 and Bug 2 uses tactile sensors while tangent bug, and Distbug uses range sensors. These techniques require its own position by using odometry, goal position and range sensor. In this paper, a novel sensor based path planning technique is proposed, namely, free-configuration eigenspace (**FCE)**. This approach tends to find principal components (Eigenvectors) spanning the low dimensional space (Eigenspace) of high order scanning data. Integrating the highest eigenvector in time steps can produce a collision free trajectory.

In this paper, we have compared our proposed FCE path planning approach with other well-known path planning techniques. Also, we have analyzed the path produced by any path planner based on the following parameters. **Path Length:** distance of the path from start to finish. **Computation time:** algorithm's total execution time excluding time spent during driving (i.e. from start to goal). **Turning:** the amount of turning which is performed along the path from start to finish. **Memory requirements:** the amount of global memory reserved by the algorithm. For good path planner it is assumed that all these parameter should be as small as possible.

This paper is organized as follows, in **section 2**, the Problem formulation and proposed technique is presented. In **Section 3** the materials and methods used are explained. **Section 3.3** describes the detection of eigensapce and trajectory generation algorithms. **Section 4** explains about the experimental set up and implementation of the FCE algorithm. In **section 5** a detailed performance analysis done with existing four planning algorithms namely, **APF, A*, RRT** and **PRM** and **section 6** with conclusion.

2. PROBLEM FORMULATION

2.1 Free-Configuration Space Concept

Finding a collision free path problem can be formulated as a search in a configuration space [9]: Let A be a robot, moving in a Euclidean space. $W = R^N$, where N = 2 or 3. Let $B_1, B_2, \ldots, B_n$ be fixed rigid bodies distributed in ***W*** are called as obstacles and are the closed subsets of ***W***. A configuration of ***A*** is a specification of the position of every point in ***A*** with respect to F_W, where F_W is a Cartesian coordinate system. The configuration space of ***A*** is the space denoted by C, with all possible configurations of ***A.*** The subset of ***W*** occupied by ***A*** at configuration *q* is denoted by ***A(q).*** A path from an initial configuration q_{init} to a goal configuration q_{goal} is a continuous map with $\tau:[0,1] \rightarrow C$ $\tau(1) = q_{goal}$ and $\tau(0) = q_{init}$. The workspace contains a finite number of obstacles denoted by $B_1, B_2, \ldots, B_n$. Each obstacle maps in ***C*** to a region:

$$C(B_i) = \{q \in C \mid A(q) \cap B_i \neq 0\} \tag{1}$$

Which is called as $C_{obstacle}$. The union of all $C_{obstacle}$ is the region $\bigcup_{i=1}^{n} C(B_i)$ and the set

$$C_{free} = C - \bigcup_{i=1}^{n} C(B_i) \tag{2}$$

A collision free path between two configurations is any continuous path $\tau:[0,1] \rightarrow C_{free}$

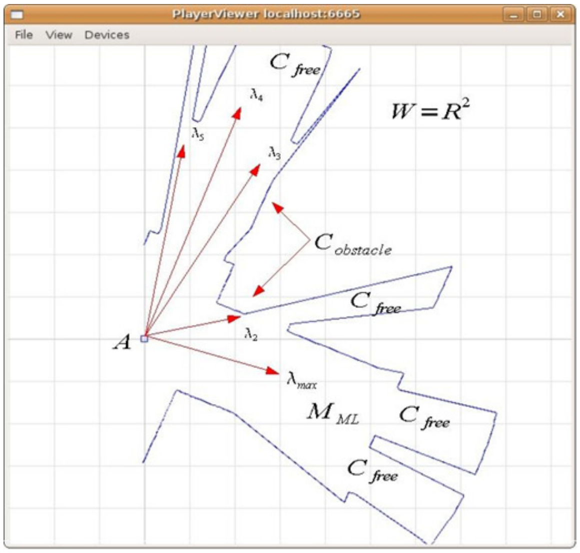


FIGURE 1: Configuration Space 2D Model - The blue line represents the laser scan area, whereas the red arrows inside the scan area are the eigenvectors and the red arrows outside the laser scan region show the obstacle region.

2.2 Proposed FCE Technique

To obtain a collision free path, we have utilized our paradigm [13] of "***Free-Configuration Eigenspaces***" which explains the underlying *low-dimensional manifolds of laser scanning data, represented by an eigenvector which corresponds to the free-configuration space and the vectorial combination of all these eigenvectors at discrete time scan frames manifests a trajectory.* To realize the above methodology of path planning, we have set forth the basis of two critical hypotheses:

1. **H1**: In the exploration process there lies a free-configuration space, such that the discrete paths which lie in that space can form an exploratory-trajectory

2. **H2:** The free-configuration space path shows a distinct pattern in the sensor data (surrounded by obstacles) and can be learnt to produce a better quality exploratory map.
According to the hypothesis:

- The single vector formulation of the trajectory point can be described as;

$$C_{free} \cap M_{ML} = C_{free}(E_{\lambda \max}) \tag{3}$$

- Consequently the vectorial sum of all the free-configuration eigenspaces;

$$\sum_{i=1}^{n} C_{free} \cap M_{ML} = C_{free}(E_{\lambda \max 1} + E_{\lambda \max 2} + E_{\lambda \max 3}) \tag{4}$$

3. MATERIALS AND METHODS

3.1 Laser Sensor Model

Range sensing is a crucial element of any obstacle avoidance system. Sensors suitable for obstacle detection are 2-D LADAR (i.e., a laser that scans in one plane). 2-D laser scanners are widely used sensor for obstacle detection. The SICK LMS is a laser scanner based on the measurement of time-of-flight (TOF) as shown in Fig.2 (a). The Laser Cartesian space is defined by its range and bearing, with the resolution of 0.5^0 , having 180^0 of frontal AMR scan with the range of 5 meters. Where the center position of the AMR is represented as (AMR_x, AMR_Y) in terms of the world coordinates frame. The incoming data are numbered as r_j , where $j = 0,1,\ldots,180$ are the distance value from the central position of the laser range finder to the object. The object position (Obj_x, Obj_y) is determined by measuring distance from the central position of the laser scanner to the object. The object position (Obj_x, Obj_y) is determined by the following where r_d is the distance from the centre of the SICK LMS to that of the mobile robot and θ_c is the angle of the mobile robot measured counter clockwise from the positive x-axis.

$$Obj_x = AMR_x + r_j \cos\left(\frac{j\Pi}{180} + \theta_c + \frac{\Pi}{2} - r_d \cos\theta_c\right) \tag{5}$$

$$Obj_y = AMRy - r_j \sin\left(\frac{j\Pi}{180} - \theta_c + \frac{\Pi}{2} - r_d \sin\theta_c\right) \tag{6}$$

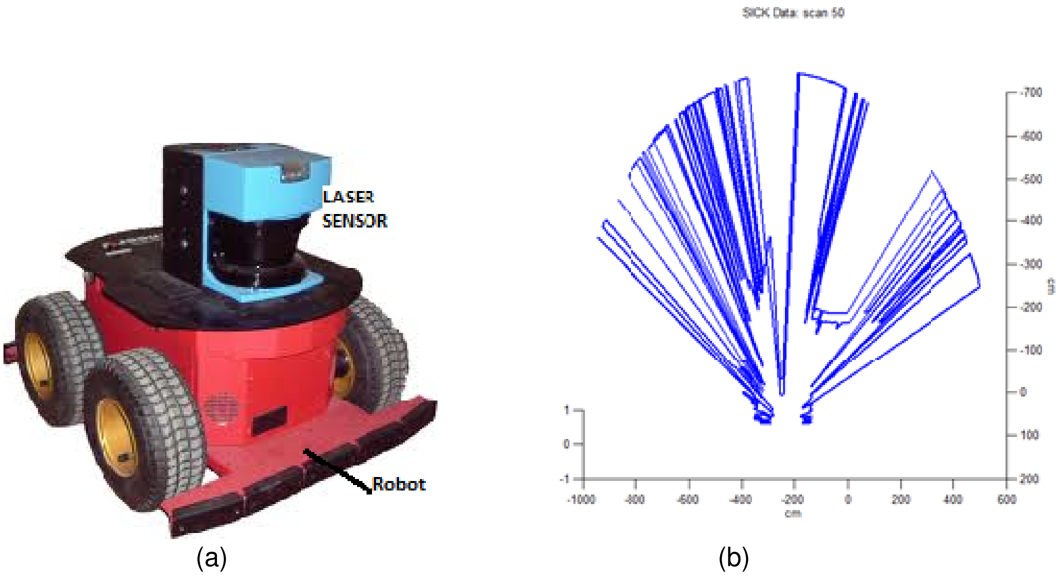


FIGURE 2: (a)The Pioneer 2AT autonomous mobile robot (AMR) carrying an onboard SICK laser scanner (b) A single laser scan plot output.

3.2 Dimensionality Reduction Technique

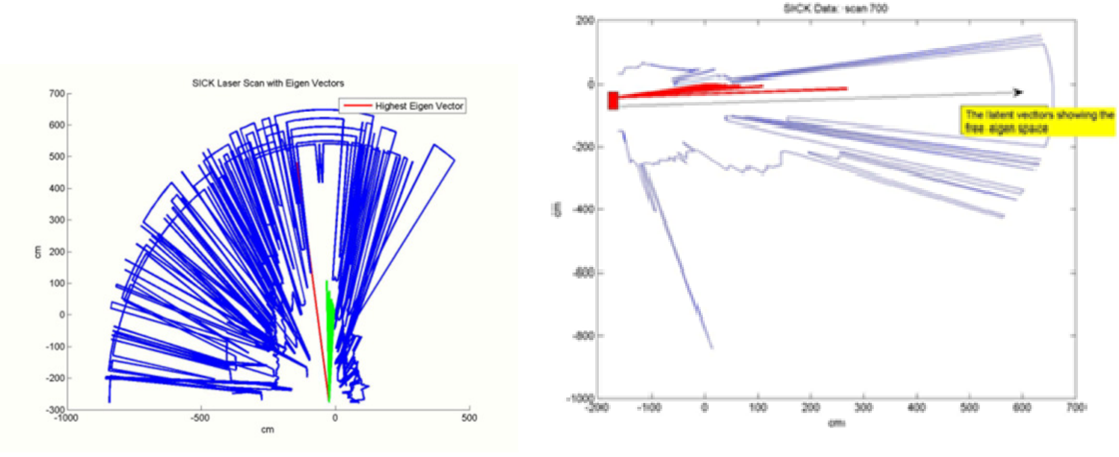
Using the SICK-laser a large number of point features can be obtained and all of them correspond to the same environmental structure. The dimensionality reduction technique enables to extract higher-order features in lower dimensional manifold representations which capture the data patterns. Principal Components Analysis (PCA) is a powerful tool that computes a set of bases functions that can be linearly combined to represent a collection of data [16]. PCA has been used extensively to cluster sets of point features in a map. We can extract line segments to represent each cluster of point features, generating very efficient representations, e.g., a representation using just the four parameters of the endpoints of 2-d line segment. In this case, the highest eigenvector is to be found by applying PCA to the sensor data which can be integrated in discrete time scan to get a trajectory.

3.3 Eigenspace Detection Method

It is possible to produce a trajectory as vectorial combination of highest eigenvectors in discrete time scan frame of sensor scanning data. But to ensure the easy manoeuvring of the robot along this direction shown by the eigenvector, one must make sure that there is enough space for the vehicle to pass through. Hence it can be ascertained by plotting the all the eigenvectors obtained by the dimensionality reduction technique, with reference to the vehicle pose at that point along with the sensor data. The pose of the mobile robot is denoted by the position and orientation (x, y, θ) in the map. In the beginning we compute the visibility model, which is the data2D laser sensor, which is commonly used for constructing range data maps in robotics. In order to construct such map, sensor's position and robot's orientation, while obtaining laser scans should be estimated accurately. The algorithm for eigenspace detection is described below. From the Fig. 3 it is evident that the highest eigenvector shows the direction of maximum free space through which the vehicle can easily manure without hitting the obstacle and hence this space we have named as free-configuration eigenspace.

Algorithm 1: Finding Free-Configuration Space
Input: The Pose data (x, y, theta), sensor scanning data (matrix)
Output: Eigenspace (Eigenvectors)

```
Begin
|  Initialize Pose (x, y, theta), n, p
|       M = [ ]
|   for Time = 1 to n
|   |   for angle =-90 : 0.05: 90
|   |       | M(time, angle) = d(Time, angle )
|   |       end
|   end
|   Calculate the Covariance of M
|
|   Calculate the Eigenvalues (V) and Vectors (E) of the Covariance of M
|     for i = 0,1,…,m
|       |   for angle =-90 : 0.5:90
|       |   |   $E_{x\lambda} = E_{\lambda\max} \times \cos(angle)$
|       |   |   $E_{y\lambda} = E_{\lambda\max} \times \sin(angle)$
|       |   |
|       |   end
|     end
|   Plot the eigenvectors from the corresponding vehicle pose,
End
```



(a) (b)

FIGURE 3: (a) Principle components extracted from the laser data, the green vectors are the smaller eigenvectors, whereas the red vector is the highest principal component, (b) the RED Vector indicating towards maximum the free area.

3.4 Autonomous Robot Trajectory Generation Method

According to the hypotheses in low-dimensional manifolds of laser scanning data, there lies an eigenvector which will represent free area or obstacle area. Assuming this to be free area, the highest eigenvector in discrete time can be integrated to get an exploratory trajectory .The following algorithm describes trajectory formation using eigenvectors. The obtained eigenvector trajectory and the actual trajectory with laser scans for Navlab test bed data are shown in Fig.4.

Algorithm 2: Trajectory connectivity in Free-Configuration Space

```
Input: Robot Pose data, sensor measurement data M
Output: Highest Eigenvector   Sensor data in discrete time scan
 Begin
    Initialise Pose (x, y, theta), n, p
    M = [ ]
    for  Time = 1 to n
        for angle =-90 : .05:90
            M(time, angle) = d(Time, angle )
        end
    end
      Calculate the Covariance of M
      Calculate the Eigenvalues (V) and Vectors (E) of the Covariance of M
      Sort the diagonal of E in descending order
      Taking the Maximum of E and its corresponding V
      Plot E and V from the initial Pose(x,y,theta) as follows

        $E_{x\lambda} = E_{\lambda \max} \times \cos( angle )$
        $E_{y\lambda} = E_{\lambda \max} \times \sin( angle )$
     Plot the eigenvectors from the corresponding vehicle pose  ,Pose(x,y,theta)
 End
```

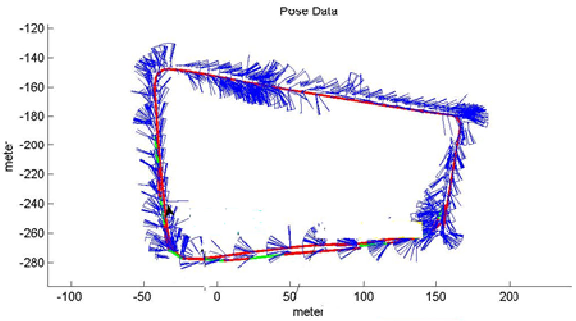
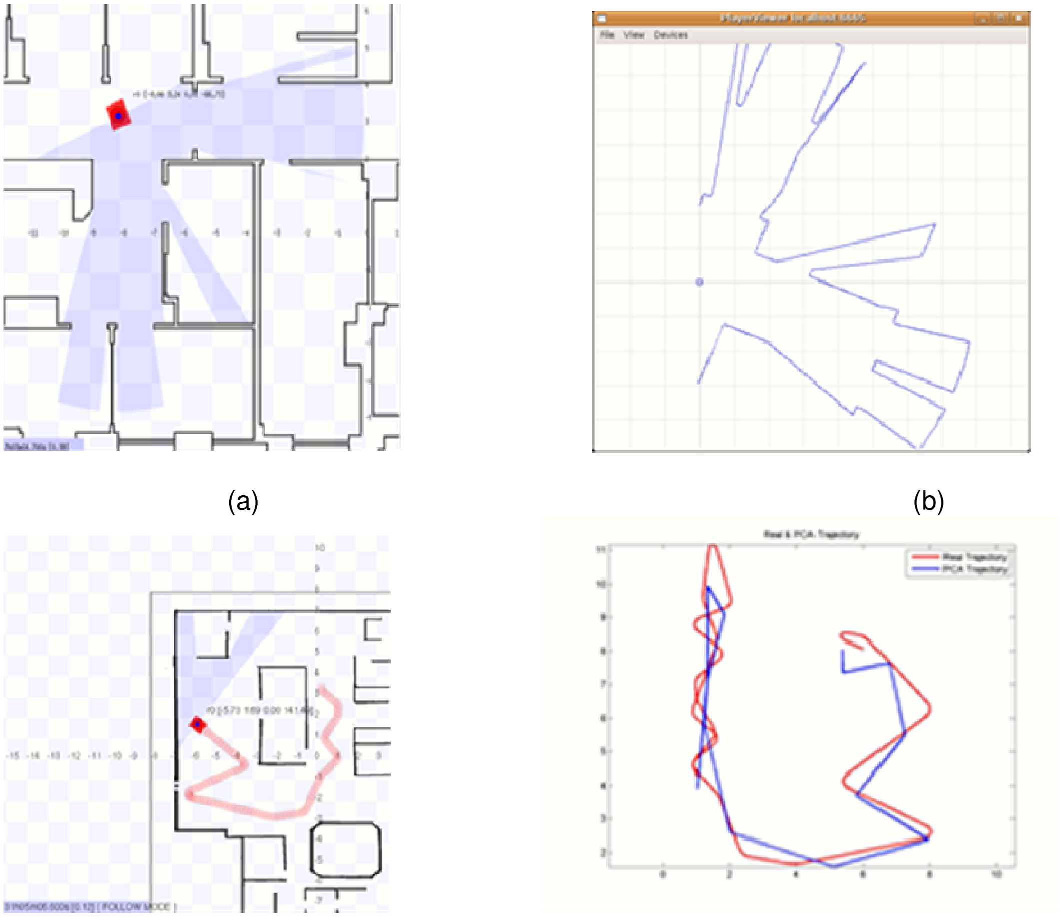


FIGURE 4: Green & red lines showing the trajectory formed by the Eigenvectors in discrete time scan of laser data.

4. EXPERIMENTAL SETUP and SYSTEM DESCRIPTION

4.1 Player/Stage Data Simulation

The proposed technique was applied to the robotic simulator running under real-time operating system (RTOS) Ubuntu 8.04 Hardy–heron on 2.0 GHz Intel dual core processor having 3 GB of RAM. The environment was created in real time (distributed under GNU) 2D autonomous mobile robotic software, Player/Stage as shown in Fig. 5. A configuration definition XML file was created that contain positions of environment floor plan, a pioneer 2DX autonomous mobile robot, laser sensor, odometry sensor. The mobile robot is defined as a non-holonomous (pioneer2DX) , which can move around in the environment and can sense the obstacle by measuring the laser distance from its Centre point of gravity. TCP/IP sockets are used to for the communication between the mobile robotic agent and the robotic software server. For experimental purposes, the mobile robot's Laser data and the odometer readings was recorded using player/stage data commands, processed/ reduced online as shown in the Fig. 5.The proposed trajectory detection methodology was implemented in MATLAB (product of Mathworks Inc.). The actual trajectory was divided into segments and each segment's consecutive laser sensor readings were taken. The latent values were found for these chunks of laser data using PCA analysis. The latent values (highest eigenvector) were integrated in time and the new trajectory was obtained. The scenario of the robot and environment is shown in the Figure.5(c) and the actual trajectory (in red) and the eigenvector trajectory in blue



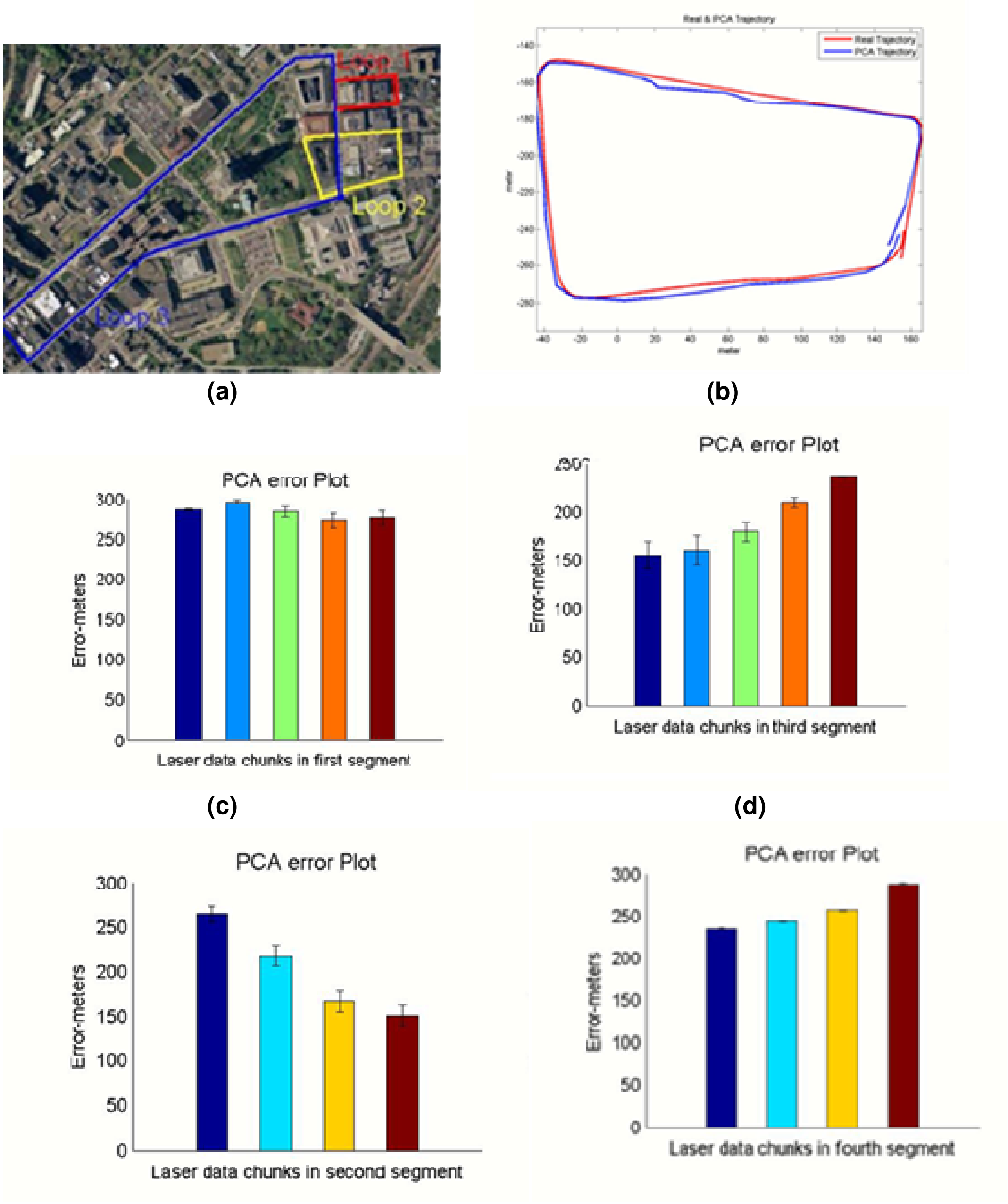
(a) (b)

(c) (d)

FIGURE 5: (a) 2Dview of the environment with the laser mounted autonomous mobile robot. (b) Simulated laser plot. (c) Graphical User Interface client program "playerv" that visualizes laser data from a player server, (c) The Actual trajectory & learned trajectory (red).

4.2 Real Time Data Simulation

The scenario of Navlab SLAMMOT Datasets, Carnegie Mellon University USA as shown in Fig. 6 in yellow Loop-2. The laser data for the entire loop was divided in to four segments and within each segment consecutive laser sensor readings was taken. The latent values were found using PCA analysis and the latent values (highest eigenvector)were integrated in time and the new trajectory was obtained .This new trajectory and the actual trajectory was compared and an error graph was plotted as shown in Fig. 6 (c , d, e, g) . It was concluded from the Fig. 6(b) that at the corners, the direction of eigenvector obtained deviated from the actual trajectory where as in a straight line segment, the eigenvector is very much close to the actual trajectory.



(a) (b)

(c) (d)

(e) (f)

FIGURE 6: (a)The Aerial Photo is downloaded from and the copyrighted property of GlobeXplorer, LLC. (“GlobeXplorer”). The trajectory (Loop-2) in yellow, was used for laser data-set collection. (b)The trajectory obtained after applying PCA to the laser data-set. (c).(d),(e) (f)error-plots show the respective trajectory segments vs laser-data chunks and their error bounds for PCA with real-trajectory.

4.3 Experimental Result Analysis

We have tested the developed algorithm on real time and simulated laser data set. From the result plots for the real time data, as shown in Fig. 6(b) & Fig. 6(d), it is evident that the constructed eigenvector trajectory (in red) is perfectly matching on the longer segments of the actual trajectory. But, failed to produce good results on the corners. In order to get a clear picture about the corner misalignment of the trajectory, we plotted the Navlab data trajectories with in the 2D map of the laser data. It is also evident from Fig. 7, at the corners the eigenvector trajectory is deviating away from the actual trajectory of the robot.

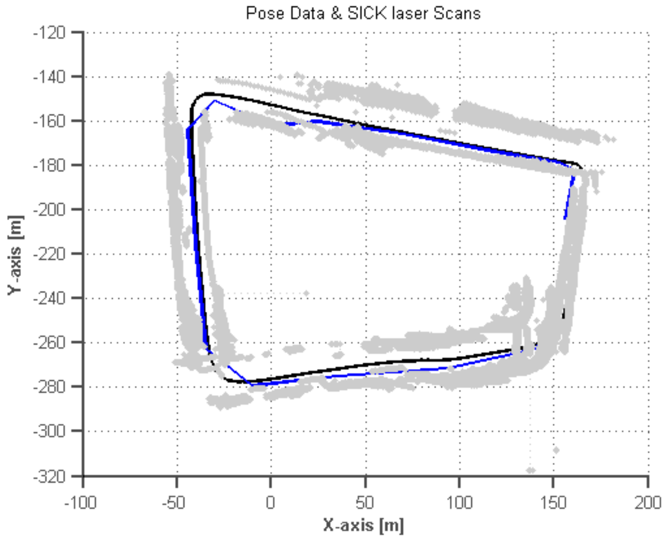
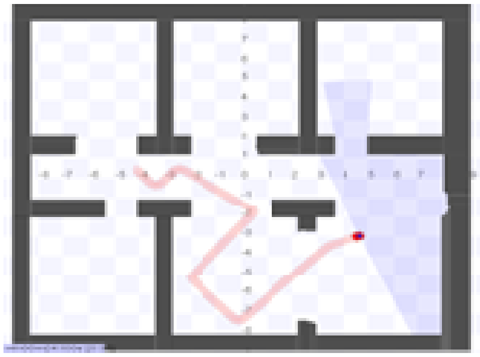


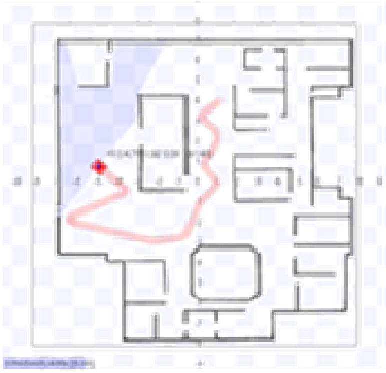
FIGURE 7: The 2D map obtained from plotting the outliers of laser data and the actual trajectory (black) and the eigenvector trajectory(blue).

5. PERFOMANCE ANALYSIS

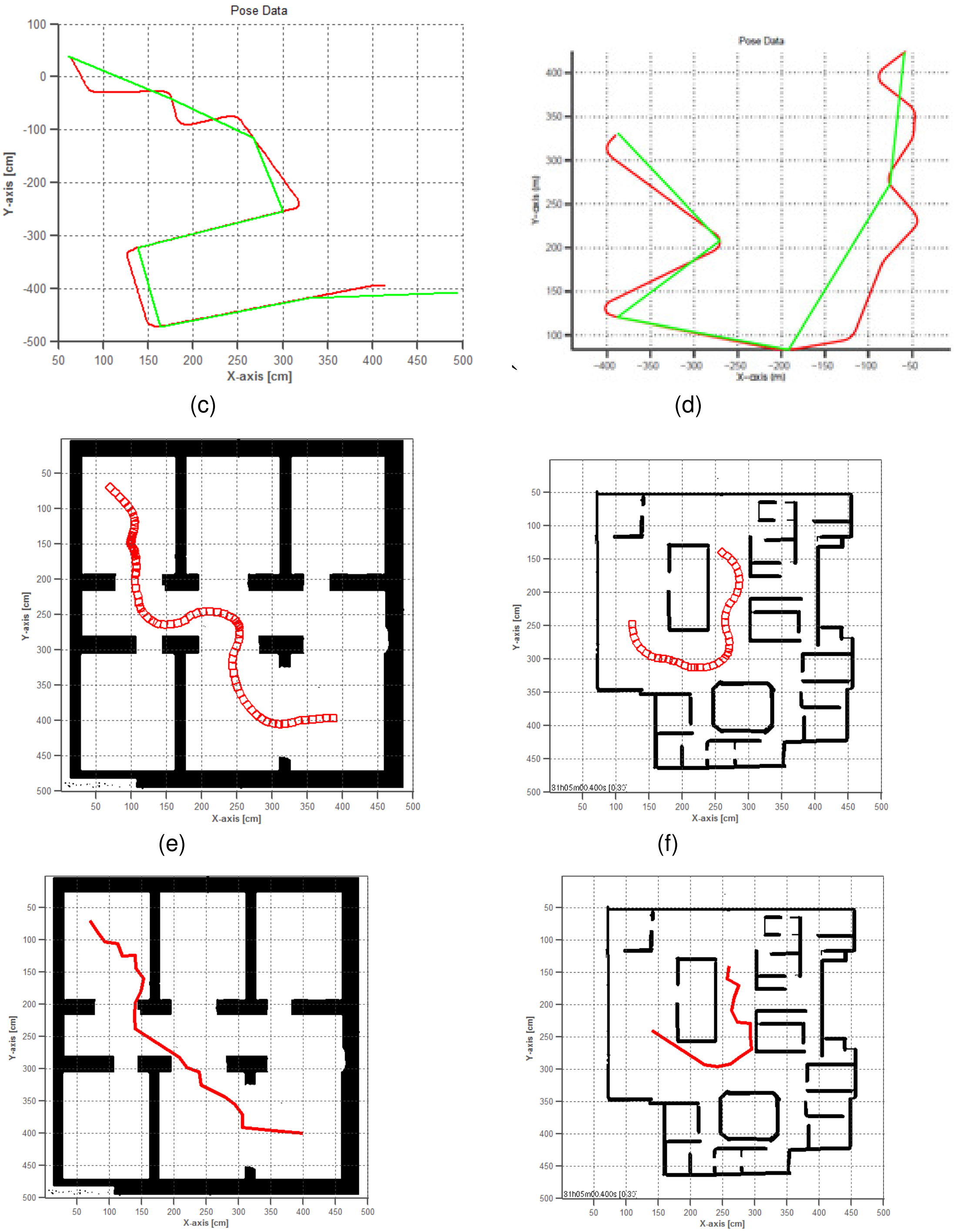
The path produced by any path planner can be analysed on the following parameters. **Path Length**: distance of the path from start to finish. **Computation time:** algorithm's total execution time excluding time spent driving. **Turning**: the amount of turning which is performed along the path from start to finish. **Memory requirements**: the amount of global memory reserved by the algorithm. The following section illustrate the result performance analysis done existing path planners **APF**, **RRT**, **PRM** and **A*** with the proposed **FCE** method. We have considered two scenarios Room1 and Room2 and found the estimated values for all the parameters under consideration. The analysis was done using MATLAB mathematical computation software. The below tables summaries the results obtained:



(a)



(b)


Pose Data
Y-axis [cm]
X-axis [cm]
100
0
-100
-200
-300
-400
-500
50
100
150
200
250
300
350
400
450
500
Pose Data
Y-axis [m]
X-axis [m]
400
350
300
250
200
150
100
-400
-350
-300
-250
-200
-150
-100
-50
Y-axis [cm]
X-axis [cm]
Y-axis [cm]
X-axis [cm]
81h05m00.400s [0.30']
Y-axis [cm]
X-axis [cm]
Y-axis [cm]
X-axis [cm]
81h05m00.400s [0.30']

(c)

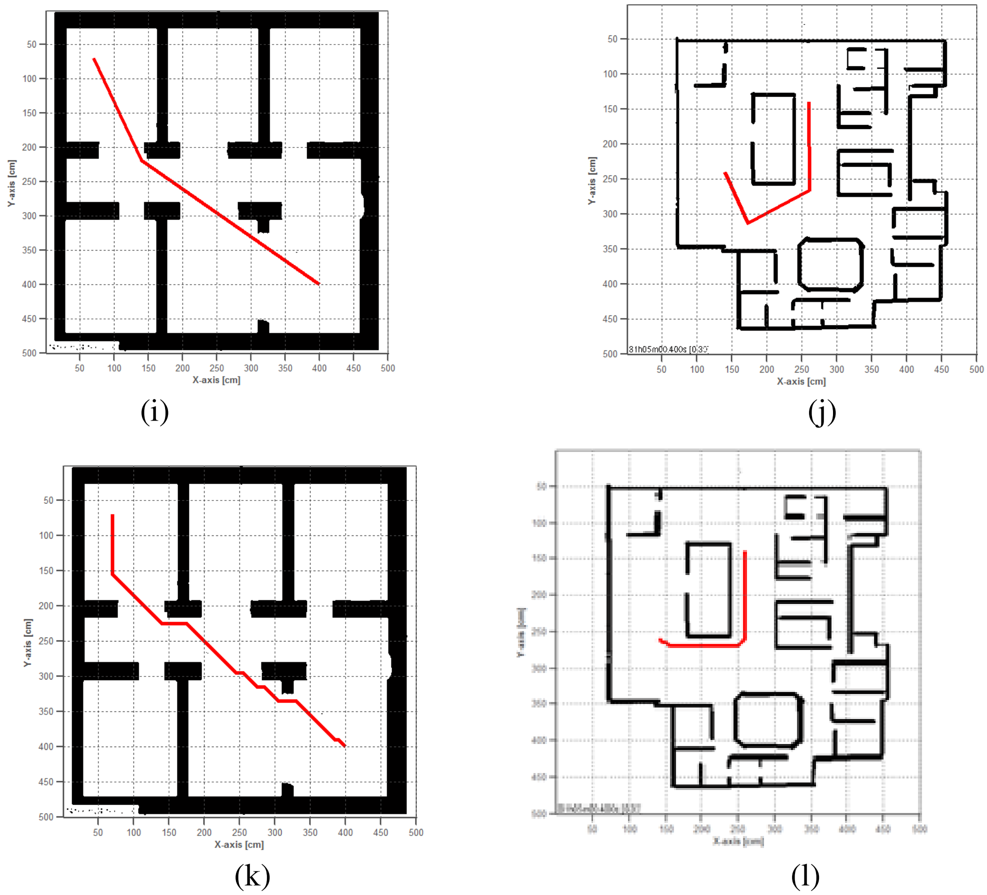
(d)

(e)

(f)

(g)

(h)



(i)

(j)

(k)

(l)

FIGURE 8: Trajectories obtained from different Path-planning alogrithms. a) Scenario of Room-1. b) Scenario of Room-2. c) Trajectories obtained from FCEalgorithm from scenario of Room-1. The red is normal way points trajectory, where as green is the FCE algorithm trajecorty. d) Trajectories obtained from FCE algorithm from scenario of Room-2. The red is normal way points trajectory, where as green is the FCE algorithm trajecorty. e) Trajectory obtainedfrom APF method for scenario of Room-1. f) Trajectory obtained from APF method forscenario of Room-2. g) Trajectory obtained from RRT method fromscenario of Room-1. h) Trajectory obtained from RRT method from scenario of Room-2. i) Smoothed trajectory obtained from PRM method from scenarioof Room-1. j) Smoothed trajectory obtained from PRM method from scenario of Room-2. k) Trajectory obtained from A* method from scenario of Room-1.l) Trajectory obtained from A* method from scenario of Room-2.

Source = [70 70]; in Y, X format.
Goal = [400 400] in Y, X format.

Technique	Scenario	Path length(cm)	Processing time	Max-turn(cm$^{-1)}$
RRT	Room1	5.646315e+002	1.513718e+002	0.1412
PRM	Room1	4.491235e+002	1.848314e+001	-0.0019
APF	Room1	5.889301e+002	1.846913e+000	3.00
A*	Room1	4.740559e+002	4.430567e+001	0.1474
FCE	Room1	10.91235e+002	1.98314e+001	1.7471

TABLE 1: Result for Room1.

Source = [140 260]; in Y, X format.
Goal = [260 140] in Y, X format.

Technique	Scenario	Path length(cm)	Processing time	Max-turn(cm$^{-1)}$
RRT	Room2	3.347247e+000	3.012290e+000	0.0004
PRM	Room2	2.806335e+002	3.273510e+000	0.0854
APF	Room2	3.234583e+002	1.037137e+000	-0.0204
A*	Room2	2.482843e+002	5.420330e+001	0.1574
FCE	Room2	6.234583e+002	2.012290e+000	2.900

TABLE 2: Result for Room2.

5.1 Result Analysis

As shown in Table-1 & 2, performance analysis on different path planning algorithms shows that the PRM technique performs better in terms of turning and path length. But it is probabilistic complete. But RRT is faster as compared to PRM and produce fine path with minimum turning. Even though A*shows an optimal path, the computational cost is high and the clearance space from the obstacle is low. The APF algorithm requires less computational time but the path length depends on the set value of the potential forces and it suffers from local-minima problem. In case of FCE, the path length and turning value are comparatively larger than all other methods.A good path is relatively short, keeps some clearance distance from the obstacles, and is smooth. Result analysis shows APF and proposed FCE technique is better on this attributes.

6. CONCLUSIONS

This paper proposes a novel path planning problem of an autonomous mobile robot navigating in a structured unknown environments, using free-configuration eigenspaces (FCE). The results are very much consistent with the hypothesis we laid for our research in the beginning, i.e. trajectory formation in correspondence to the highest eigenvector of the free-configuration laser data always results into a better exploration of the unknown area. Consequently by adapting to the similar trajectory formations, the autonomous mobile robot has better tendency towards the map-building process. The trajectories maximizing the map building process could be formed by the intramural property of the laser sensor dually responsible for the map building process, the vector sum of all these highest vectors obtained from the laser data result into the trajectory which maximizes the information of the map. These individual eigenvectors were machine learned and the predicted new trajectory vectors facilitated the autonomous robot to maneuver at considerably higher speed. A limitation of our proposed approach is the assumption of simple obstacle geometry. This can be tackled by situation aware sensing descriptors, which may result into dimensionality scaling and require algorithms that scale well.

7. REFERENCES

[1] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” The international journal of robotics research, vol. 5, no. 1,pp. 90–98, 1986.

[2] P. Vadakkepat, T. H. Lee, and L. Xin, “Application of evolutionary artificial potential field in robot soccer system,” in IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, pp. 2781–2785, IEEE, 2001.

[3] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots,” IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp. 278–288, 1991.

[4] A. Chakravarthy and D. Ghose, “Obstacle avoidance in a dynamic environment:A collision cone approach,” IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 28, no. 5, pp. 562–574, 1998.

[5] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23–33, 1997.

[6] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, “Evolutionary artificial potential fields and their application in real time robot path planning,”in Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, vol. 1, pp. 256–263, IEEE, 2000.

[7] D. Gallardo, O. Colomina, F. Fl´orez, and R. Rizo, “A genetic algorithm for robust motion planning,” in Tasks and Methods in Applied Artificial Intelligence, pp. 115–121, Springer, 1998.

[8] S. Tang, W. Khaksar, N. Ismail, and M. Ariffin, “A review on robot motion planning approaches,” Pertanika Journal of Science and Technology, vol. 20, no. 1, pp. 15–29, 2012.

[9] H. M. Choset, Principles of robot motion: theory, algorithms, and implementation. MIT press, 2005.

[10] E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, and E. Kavraki, “Sampling-based roadmap of trees for parallel motion planning,” IEEE Transactions on Robotics, vol. 21, no. 4, pp. 597–608, 2005.

[11] R. Al-Hmouz, T. Gulrez, and A. Al-Jumaily, “Probabilistic road maps with obstacle avoidance in cluttered dynamic environment,” in Intelligent Sensors, Sensor Networks and Information Processing Conference, pp. 241–245, IEEE, 2004.

[12] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[13] T. Gulrez, S. Zaheer, and Y. Abdallah, “Autonomous trajectory learning using free configuration-eigenspaces,” in IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 424– 429, IEEE, 2009.

[14] T. Gulrez and A. Tognetti, “A sensorized garment controlled virtual robotic wheelchair,” Journal of Intelligent & Robotic Systems, vol. 74, no. 3-4, pp. 847–868, 2014.

[15] T. Gulrez, A. Tognetti, and D. De Rossi, “Sensorized garment augmented 3d pervasive virtual reality system,” in Pervasive Computing, pp. 97– 115, Springer, 2010.

[16] I. Jolliffe, Principal component analysis. Wiley Online Library, 2005.

[17] A. Al-Odienat and T. Gulrez, “Inverse covariance principal component analysis for power system stability studies,” Turkish Journal of Electrical Engineering & Computer Sciences, vol. 22, no. 1, pp. 57–65, 2014.

[18] M. Tipping and C. Bishop, “Probabilistic principal component analysis, “Journal of the Royal Statistical Society”, Series B, vol. 61, pp. 611–622, 1999.

[19] “Player stage.” http://playerstage.sourceforge.net/.

[20] T. Chaudhry, T. Gulrez, A. Zia, and S. Zaheer, “Bezier curve based dynamic obstacle avoidance and trajectory learning for autonomous mobile robots,” in Proceedings of the 10th International Conference on Intelligent Systems Design and Applications, ISDA’10, 2010, pp. 1059–1065.

[21] S. Zaheer, T. Gulrez, “Beta-eigenspaces for autonomous mobile robotic trajectory outlier detection,” in 2011 IEEE Conference on Technologies for Practical Robot Applications, TePRA, pp. 31–34, 2011.

[22] P. Raja and S. Pugazhenthi, Review Optimal path planning of mobile robots: International Journal of Physical Sciences Vol. 7(9), 23, pp. 1314 – 1320, February, 2012

[23] R. Kala, K. Warwick Multi-Vehicle Planning using RRT-Connect. Paladyn Journal of Behavioural Robotics, 2(3): 134-144, 2011

INSTRUCTIONS TO CONTRIBUTORS

Robots are becoming part of people's everyday social lives - and will increasingly become so. In future years, robots may become caretaking assistants for the elderly or academic tutors for our children, or medical assistants, day care assistants, or psychological counselors. Robots may become our co-workers in factories and offices, or maids in our homes.

The International Journal of Robotics and Automation (IJRA), a refereed journal aims in providing a platform to researchers, scientists, engineers and practitioners throughout the world to publish the latest achievement, future challenges and exciting applications of intelligent and autonomous robots. IJRA is aiming to push the frontier of robotics into a new dimension, in which motion and intelligence play equally important roles. IJRA scope includes systems, dynamics, control, simulation, automation engineering, robotics programming, software and hardware designing for robots, artificial intelligence in robotics and automation, industrial robots, automation, manufacturing, and social implications.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJRA.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 6, 2015, IJRA appears in more focused issues. Besides normal publications, IJRA intends to organize special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJRA LIST OF TOPICS
The realm of International Journal of Robotics and Automation (IJRA) extends, but not limited, to the following:

- Automation Control
- Automation Engineering
- Autonomous Robots
- Biotechnology & Robotics
- Emergence of The Thinking Machine
- Forward Kinematics
- Household Robots and Automation
- Inverse Kinematics
- Jacobian and Singularities
- Methods for Teaching Robots
- Nanotechnology & Robotics (Nanobots)
- Orientation Matrices
- Robot Controller
- Robot Structure and Workspace
- Robotic & Automation Software Development
- Robotic Exploration
- Robotic Surgery
- Robotic Surgical Procedures
- Robotic Welding
- Robotics Applications
- Robotics Programming
- Robotics Technologies
- Robots Society and Ethics
- Software and Hardware Designing for Robots
- Spatial Transformations
- Trajectory Generation
- Unmanned (Robotic) Vehicles

CALL FOR PAPERS

Volume: 6 - **Issue: 2**

i. Paper Submission: May 31, 2015

ii. Author Notification: June 30, 2015

iii. Issue Publication: July 2015

CONTACT INFORMATION

Computer Science Journals Sdn BhD
B-5-8 Plaza Mont Kiara, Mont Kiara
50480, Kuala Lumpur, MALAYSIA
Phone: 00603 6204 5627
Fax: 00603 6204 5628
Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2015
COMPUTER SCIENCE JOURNALS SDN BHD
B-5-8 PLAZA MONT KIARA
MONT KIARA
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6204 5627
FAX: 006 03 6204 5628
EMAIL: cscpress@cscjournals.org