AUTHOR QUERY FORM

ELSEVIER	**Journal:** Robotics and Autonomous Systems **Article Number:** 1689	**Please e-mail or fax your responses and any corrections to:** **E-mail: corrections.esnl@elsevier.river-valley.com** **Fax: +44 1392 285879**

Dear Author,

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

Articles in Special Issues: Please ensure that the words 'this issue' are added (in the list and text) to any references to other articles in this Special Issue.

Uncited references: References that occur in the reference list but not in the text – please position each reference in the text or delete it from the list.

Missing references: References listed below were noted in the text but are missing from the reference list – please make the list complete or remove the references from the text.

Location in article	**Query / remark** **Please insert your reply or correction at the corresponding line in the proof**
Q1	TeX and pdf are different. In Tex one extra affiliation is given to the second author. Also third paragraph in Section 3.4.1 and biographies are different. We have followed the TeX file. Please check.

Electronic file usage

Sometimes we are unable to process the electronic file of your article and/or artwork. If this is the case, we have proceeded by:

☐ Scanning (parts of) your article ☐ Rekeying (parts of) your article ☐ Scanning the artwork

Thank you for your assistance.

Robotics and Autonomous Systems xx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Agent formations in 3D spaces with communication limitations using an adaptive Q-structure

Shuzhi Sam Ge [a,b,*], Cheng-Heng Fua [a], Khiang Wee Lim [c]

[a] Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576, Singapore
[b] Institute of Intelligent Systems and Information Technology (ISIT), University of Electronic Science and Technology of China, Chengdu, 610054, China
[c] Singapore Institute of Manufacturing Technology (SIMTech), 71 Nanyang Drive, Singapore 638075, Singapore

ARTICLE INFO

Article history:
Received 16 October 2006
Received in revised form
26 June 2009
Accepted 23 October 2009
Available online xxxx

Keywords:
3D-Formations
Queues
Convergence
Agents
Potential functions
Potential trenches

ABSTRACT

In this article, we further extend the Queue-formation structure (or Q-structure) in 3D spaces with additional features including: (i) specifying orientation information, (ii) a mechanism for forming sub-formations before the convergence into the final formation, and (iii) adapting the communication structure when communications are limited. The virtual Bobber-agents are used to guide each vehicle toward the appropriate queue, by acting as intermediate targets. In addition, virtual constellation-agents bias the motion of each vehicle to within a user-defined cone to the front of the vehicle so that abrupt direction changes are avoided as far as possible. The proposed scheme relies mainly on simple behaviors between embodied and virtual agents and is computationally inexpensive method. Extensive simulations show the effectiveness of the proposed method.

1. Introduction

Research on multi-robot systems has been extremely active in recent years, including topics in communications, high level decision making, and low level behavioral-based control mechanisms. A tiered approach is generally used for such complex systems, with deliberation protocols (such as [1–3]) higher up in the hierarchy passing commands to lower, motion level controls like those used in [4,5]. Specifically, formations are typically accomplished on two levels: (i) describing the formation, which may or may not change during runtime and (ii) determining desired points/paths for each vehicle in the system. This article will mainly focus upon decentralized formation approaches that are more suited for teams in dynamic, uncertain environments. The following issues are considered: (i) the change of agent numbers in large teams; (ii) the change of the communication structure due to the communication limitations; and (iii) obstacles avoidance. Our proposed decentralized formation approach facilitates scaling and flexibility of the formation with emphasis on the appearance of the formation, and allows adaptation of the communication structure itself, by leveraging on the fact that the Q-structure provides a convenient high level organization of the robot team in terms of short term information flow.

Most formations are described using concepts from graph theory [6]. Each agent is associated with a node in the graph, and formation maintenance involves the tracking of each node. This can be seen in virtual structure approaches [7,8], formation constrained functions [9], planning for formations [10], controller synthesis for non-holonomic vehicles using point-referenced formations [11], and also used in the methods for formation controller design proposed in [12,13]. Such a representation is also implicit in reactive approaches that require an agent to follow others located at connected nodes at a specific distances and bearings [14–17]. Several reactive approaches, including virtual leaders [18], social potentials [19] and pure behavior- based approaches [20], also use such a representation. Studies have also revolved around formation stability and convergence, such as leader-to-formation stability [21], in navigation functions [22–24] and in the presence of obstacles [25].

Graphs offer an instinctive method for describing formations, in which node/edges may be added and removed dynamically in response to the addition/removal of robots. Such representations become difficult to track dynamically when agent numbers change in large teams. An algorithm for generating formations that conform to specified 2D patterns was proposed in [26]. By using virtual bodies and artificial potentials, an approach to gradient estimation

* Corresponding author at: Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576, Singapore. Tel.: +65 6516 6821; fax: +65 6779 1103.
E-mail addresses: samge@nus.edu.sg, elegesz@nus.edu.sg (S.S. Ge).

0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.10.005

Nomenclature

Symbol	Description
$\mathcal{Q}$	The set of all the queues in a formation $\mathcal{F}$
$\mathcal{Q}_j$	j-th queue in the set Q
$\mathcal{V}_F(N_{tot})$	The set of formation vertices
V_i	Formation vertex
N_v, N_q	Number of formation vertices and queues, respectively
N_{tot}	Total number of vehicles
$\mathcal{S}_j$	A set of points describing the shape of the queue
$\mathcal{C}_j$	The capacity of the queue $\mathcal{Q}_j$
$\mathcal{O}_j$	The set of functions describe the orientation of agents at each point along the length of the queue
r_i	Agent i
r_{qvj}, q_{qvj}	Queue-vertex agent and its position, respectively
$q_{toq,i}$	Target-on-queue of robot i
d_{ir}	Acceptable distance between agents
d_c	Communication range of the agent
$\bar{R}_{c,i}$	A set of agents within communication range of r_i
R_{Q_j}	Sub-queue vertices from a set of ranked vertices belonging to $\mathcal{Q}_j$
R_{sos}	The set of agents broadcasting the distress flag
$Z_{cst,i}$	Cast-zone of agents r_i
$q_{ba,i}$, $q_{tg,i}$	Position of virtual bobber-agent and immediate target of r_i, respectively
N_{cs}, $N_{cs,k}$	The set of virtual constellation-agents around the vehicle and its subset, respectively
$r_{cs,i0}$	Virtual constellation-agent that lies on all the guidelines
$r_{cs,ika}$	The repulsive-distance between a vehicle r_i and each of its virtual constellation-agents
d_{cs}	Distance between virtual constellation-agents belonging to each subset $N_{cs,k}$
d_s	The safety distance a vehicle has to keep from any obstacle

and optimal formation geometry design and adaption were presented in [27]. Other methods such as that described in [28–31], while capable of supporting scaling, are more suited for flocking where mainly aggregation is considered for simple formations. In order to maintain a constant representation independent of team size, the Q-structure has been recently introduced [32] to facilitate scaling and flexibility in operating conditions with global communications. As with several methods mentioned above, direct wireless communications have often been used (e.g., in the works [33–35,1]), and the influence on agent cohesion and behavior have also been examined [36]. Global communications may not always be possible, and the convergence of a system based on the Q-structure has been examined in [37] when only limited communication is available. Since the Q-structure provides a convenient high level organization of the agent team in terms of short term information flow, in this article, it is further extended to allow adaptation of the communication structure itself. In addition, previously, only the problem of enabling agents to attain the shape of a specified formation, and have not paid much attention to the issue of orientation in the formation. All agents under the Q-formation scheme are made to follow the orientation of the virtual leader. However, depending on the application, more control over the final orientation of each agent may be desired, and is an important consideration in many cases. The desired orientation of each agent may be different, depending on each of their final position. In this article, our main contributions are as follows:

(i) The Q-formation scheme is extended into the 3D space and incorporates orientation information into the representation. Unlike our work in [37], the method proposed in this article exploits the organizational structure of the Q-structure to explicitly segregate short term information flow in the system and to adapt the short term communication structure according to communication ranges.

(ii) In contrast to our previous work in [32], we consider the limitations on the amount of direction changes each vehicle (or embodied agent) is capable of making at each instant, preferring to make gradual directional changes instead of abrupt turns. Constellation-agents are used by each vehicle to bias their motion to reflect such preferences.

(iii) Cast-zones and virtual bobber-agents are further used by each vehicle to generate suitable intermediate targets between the vehicle and their actual target on the queue. These intermediate targets are determined by the movement and convergence of the virtual bobber-agents in their associated cast-zones. The intermediate targets act as a more appropriate target for the vehicles by reducing the immediate need for sudden directional changes.

Remark 1. In this article, we are mainly concerned with formations involving embodied agents, which can be robots or autonomous vehicles, and would be referred to simply as 'agents' for the remainder of the article. This is distinguished from the virtual agents (i.e., virtual bobber-agents and virtual constellation-agents) that each of these vehicles is used for target determination and formation maintenance/tracking purposes.

Remark 2. In practice, there are two methods to get the postures and motions of other agents: (i) communication between agents, each agent broadcasts its states, such as velocity, position, and orientation, then other agents within its communication range can receive these information; and (ii) some sensors, such as sonar coupled with an infra-red (IR) sensor, laser scanner, camera and other motion detection sensors can be adopted to detect the postures and the motions of obstacles or other agents.

2. Q-structure representation

Formations are typically represented by graphs with each node corresponding to the exact position of a robot. On the other hand, the Q-structure puts emphasis on the appearance of the formation. It constrains the positions of robots in the formation, but does not dictate exact positions for them. A formation is described by the Q-structure, using queues and formation vertices as follows.

Definition 3 (*Formations [32]*). A formation is denoted by $\mathcal{F} = (\mathcal{Q}, \mathcal{V}_F(N_{tot}))$, where $\mathcal{Q}$ is the set of all the queues that make up the formation, and $\mathcal{V}_F(N_{tot})$ represents the set of formation vertices, V_i $(i = 1, \ldots, N_v)$, where N_{tot} is the total number of vehicles[1] and N_v is the number of formation vertices.

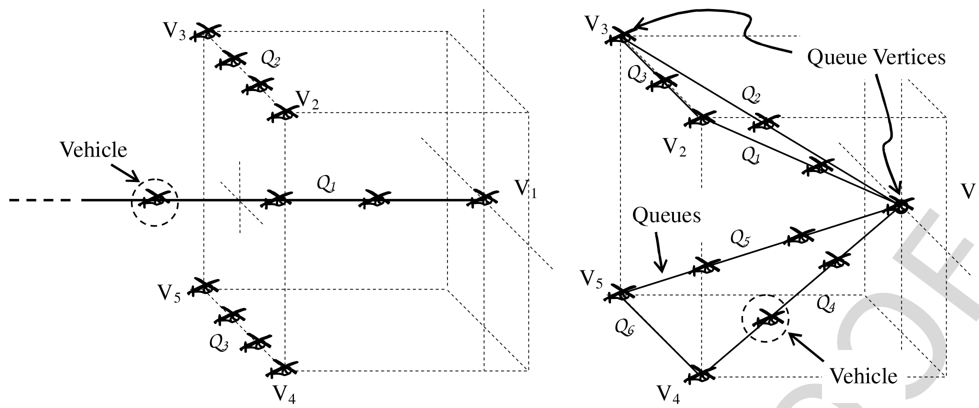
2.1. *Incorporation of orientation information*

In order to incorporate information regarding the desired orientation of agents in the final formation, an extra element, $\mathcal{O}_j$, is included in the definition of each queue. With this, each queue of a formation may be written as follows.

Definition 4 (*Queues*). A queue, $\mathcal{Q}_j \in \mathcal{Q}$, is denoted as $\mathcal{Q}_j = (\mathcal{V}_j, \mathcal{S}_j, \mathcal{C}_j, \mathcal{O}_j)$ and each of the elements that characterizes the queue is described as follows:

[1] Each formation vertex is defined in the coordinate frame of the target.

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005



(a) Formation with 5 formation vertices and 3 queues.

(b) Formation with 5 formation vertices and 6 queues.

Fig. 1. Examples of queues, and formation vertices (V_1 to V_5) in 3D space.

(i) $\mathcal{V}_j \subseteq \mathcal{V}_F(N_{tot})$ (Queue Vertices): a list of formation vertices through which $\mathcal{Q}_j$ passes.
(ii) $\mathcal{S}_j$ (Shape): a set of points following an equation in $\mathbb{R}^3$ that describes the spatial appearance of $\mathcal{Q}_j$.
(iii) $\mathcal{C}_j$ (Capacity): a fraction that refers to the proportion of all the agents in the formation it can hold, i.e., $\sum_{j=1}^{N_q} \mathcal{C}_j = 1$, where N_q is the total number of queues in the formation.
(iv) $\mathcal{O}_j$ (Orientation): consists of a set of functions that describe the orientation of agents at each point along the length of the queue.

To maintain scalability, the orientation information, $\mathcal{O}_j$, is defined as a function of the position along the length of the queue. This is different from nodes–edges approaches, which require the orientation of each agent in the formation to be explicitly defined. For orientation in 3D spaces, the orientation information at each point in the queue will typically be given by the roll, pitch and yaw (heading) angles. Thus, the orientation information may in general be written as $\mathcal{O}_j = \{\phi(d_{qj}), \theta(d_{qj}), \psi(d_{qj})\}$, where d_{qj} is the position along the length of $\mathcal{Q}_j$. Depending on the constraints on each agent, $\mathcal{O}_j$ may not contain all the three, but just one or a combination of two of the above functions. For instance, for a stationary formation with helicopters, only the yaw orientation may be defined by the user. By having the orientation as a function of the position along the length of the queue, instead of in absolute positions, results in easy scaling.

Two very different formations may have the same $\mathcal{V}_F(N_{tot})$ as shown by the two formations in Fig. 1. The actual appearance of the formation is specified by both the queues and formation vertices. Fig. 1(b) shows a formation consisting of six queues. The formation vertices are labeled V_1 to V_5 and the queues are labeled $\mathcal{Q}_1$ to $\mathcal{Q}_3$. It is noticed that there is only one vertex V_1 in $\mathcal{Q}_1$ in Fig. 1(a). Such queues like $\mathcal{Q}_1$ with only one vertex are called open queues [32], which are able to extend to infinity starting from the formation vertex. A detailed discussion of the open queues and closed queues can be found in [32]. The attraction of agents to a queue in the 3D space is shown in Fig. 2. The contour rings radiating from 2 points along the plane perpendicular to the gradient of the queue at those points indicate the various levels of the potential trench.

In this work, we assume that each agent can broadcast the information such as its states, and other agents within its communications range can receive this information. It means that each agent can make the decision using only other agents' information which are within its communication. And, we assume that each agent has the same communication range d_c. In practice, most of the agents, such as mobile robots, marine surface vessels, and helicopters, always have some restrictions on their velocities and angular velocities, which means that they have their immediate regions where to move in each decision step.

Furthermore, in order to maintain connectivity between queue vertices when there are limited communication ranges, a system for dynamic inclusion of sub-queue vertices to bridge queue vertices that are too far apart is used. The sub-queue vertices form a set of ranked queue-vertices belonging to the queue $\mathcal{Q}_j$, given by R_{Qj}. Agents belonging to the same queue would have a direct communication link with a subset of agents in R_{Qj}, and will select the highest ranked vertex in the subset to follow. The dynamic addition of sub-queue vertices into the system is described in the following sections.

3. Target generation and determination of agent behavior

In this article, decision flow and communications take place on several levels as shown in Fig. 3. The communication levels include the low and high frequency scales. The control of the formation takes place on these two levels based on the information available on each scale. This reduces the amount of information that must be available to each robot for reactive decision making.

(i) *Low frequency, long term transfer:* This refers to the gradual multihop information transfer, through a weakly connected communication network, between robots that are not within the immediate vicinity of each other. The collection of information over a longer time period allows for intermittent information losses between links. Formation control on this level involves low frequency decisions regarding the (re)allocation of robots to different parts of a formation.
(ii) *High frequency, short term transfer:* This facilities time-critical and reactive decision making, such as interrobot collision avoidance and getting into formation. It only involves local communication range. Explicit controls governing the actual movements and paths of the agents occur at this level. Such decisions take place at a higher frequency when information is available.

Corresponding to the above two communication levels, the decision making level contains four stages. As for low frequency communication level,

(i) High level decision making/User level: The formation, including the objective, the shape of the formation, etc., is specified by users.

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005

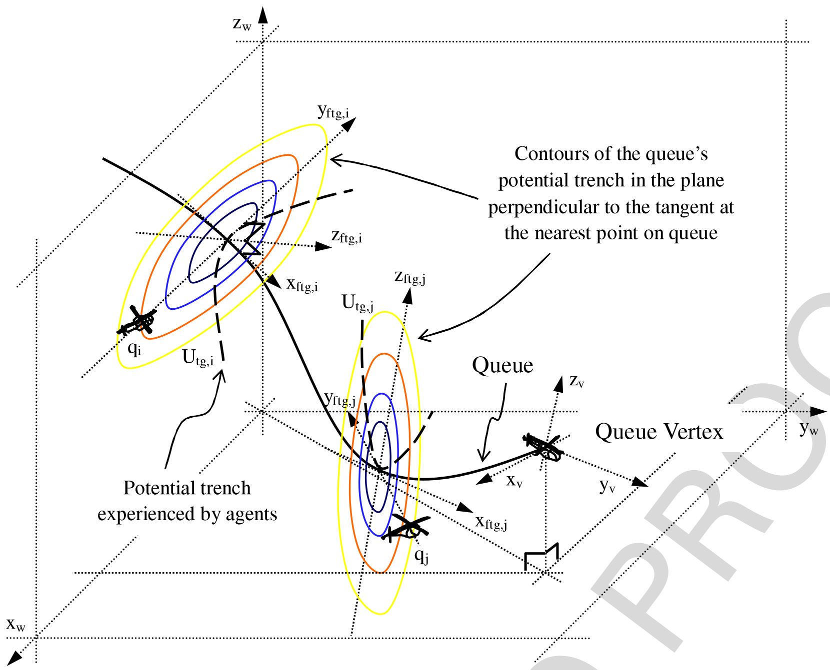


Fig. 2. Agents are attracted to their associated queues based on the potential trenches in the 3D space. The potential trenches experienced by each agent, given their positions, are also shown.

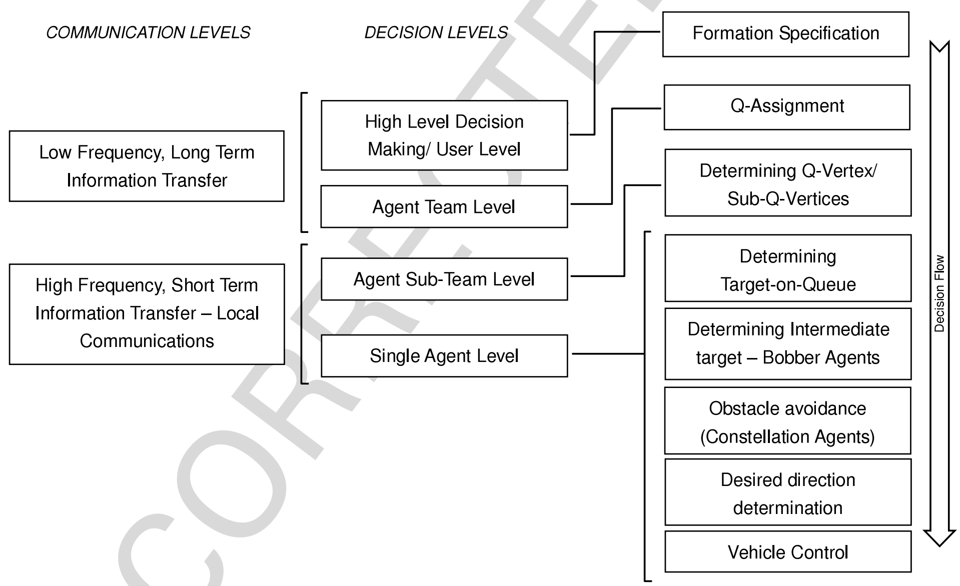


Fig. 3. Decision Flow within the system, as well as the different levels of communications and decision making.

(ii) Agent team level: All the agents in the team are distributed amongst the queues based on a greedy allocation mechanism that considers their distance from each queue in $\mathcal{Q}$ [37].

As for high frequency communication level,

(iii) Agent sub-team level: After Q-Assignment, Q-Vertex and Sub-Q-Vertex should be determined for each sub-team.

(iv) Single agent level: For all agents belonging to the same queue (e.g., $\mathcal{Q}_j$), one is chosen greedily to track $\mathcal{V}_j(1)$ based on distance to the queue vertex. Let this agent be r_{qvj}. The remaining agents in the queue determine their target-on-queue ($q_{toq,i}$) based only on communication with other agents within its communication range on the same queue by using Algorithm 1. The queue is formed with respect to the current position of r_{qvj} and $\mathcal{V}_j(1)$.

Each agent uses $q_{toq,i}$ to determine the equilibrium position of their individual virtual bobber-agent which they proceed to track. Obstacle avoidance and movement are constrained by the virtual constellation-agents. After the desired direction of the vehicle is determined, vehicle control will generate the real control input, such as forces and torques, for each vehicle according to the dynamic model to track the immediate target.

3.1. Generation of target-on-queue with limited communication ranges

The first phase involves the generation of the target-on-queue for each agent. In this work, the greedy assignment is made based on the shortest distance, such as those in [38,39]. If the agent is greedily assigned to be the leader in its queue, its target-on-queue

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005

is given by $\mathcal{V}_j$. This agent hence follows the formation keeping objective and tracks $\mathcal{V}_j$. Other agents determine their target based on Algorithm 1.

Due to limited communication ranges, a direct link may not be present between an agent and the queue-vertex agent r_{qvj}. For this, a tiered system of sub-queue vertices is generated to produce intermediate points of references along the queue. Sub-queue vertex agents have contact with at least another sub-queue vertex agent at a higher level of the hierarchy than itself.

The algorithm (lines 1–11) works as follows. An agent that detects no queue-vertex agents (of whatever tier) in its communication range will emit a distress signal. If an agent, with a direct link to its queue-vertex agent, detects that all agents further than itself from its queue-vertex agent are emitting the distress signal, it will take on the role of a queue-vertex agent a tier lower than the one it is following.

A queue-vertex agent broadcasts its position along the queue and the current queue orientation, acting as a reference point along the queue for other agents to follow. Therefore, with limited communications, information propagates implicitly through the set of agents through the sub-queue-vertices. If there are more than one agent in the same tier of the queue-vertex hierarchy, we use the greedy assignment based on shortest distance [38,39]. Then, the one closer to the next highest level queue-vertex will win and adopt the role. This is the simplest form of greedy allocation.

The second part of the algorithm is based on the distance of the agents on each queue. The n-th agent chooses its target to lie on the queue, at a distance of d_{ir} from the target of the $(n-1)$-th agent in the ordered list $R_{c,i}$.

Algorithm 1 Determining Target-on-Queue (by agent r_i)

1: Let $\bar{R}_{c,i} \in R_N$ be the set of agents within communication range of r_i and on the same queue as r_i, i.e., belonging to $\mathcal{Q}_j$.
2: Let $R_{Qj} \in R_N$ be the set of agents that are on $\mathcal{Q}_j$ and belongs to the hierarchy of queue vertices.
3: **if** $\bar{R}_{c,i} \cap R_{Qj} = \varnothing$ **then**
4: // r_i do not have a direct link to a queue-vertex.
5: Broadcast a 'distress' flag signalling this state.
6: **else**
7: Set $q_{vtx,i}$ to be the highest ranked vertex in $\bar{R}_{c,i} \cap R_{Qj}$.
8: Let $R_{sos} \in \bar{R}_{c,i}$ be the set of agents broadcasting the distress flag.
9: **if** $(\forall q_{xi} \in \bar{R}_{c,i}\ s.t.\ \|q_{xi} - q_{vtx,i}\| \geq \|q_i - q_{vtx,i}\|, q_{xi} \in R_{sos})$ **then**
10: // All agents in $\bar{R}_{c,i}$ that are further from $q_{vtx,i}$ than r_i do not have direct links to $q_{vtx,i}$.
11: r_j is included into the lowest tier of R_{Qj}. It broadcasts (i) its position on the queue, and (ii) the queue's orientation, given by the orientation of $q_{vtx,i}$.
12: Let $R_{c,i} \in R_N$ be an ordered set of agents (according to increasing Euclidean distance from q_{qvj}) within communication range of r_i and on the same queue as r_i, i.e., belonging to $\mathcal{Q}_j$.
13: Suppose, r_i is the n-th agent in the list $R_{c,i}$.
14: **if** $n = 1$ **then**
15: Set $q_{toq,i} = \mathcal{V}_j$.
16: **else**
17: Let $r_k \in R_{c,i}$ be the $(n-1)$-th agent in the list, and d_{ir} be the desired distance between any two agent on $\mathcal{Q}_j$.
18: Set $q_{toq,i} = \arg\min_{q\in Q} \|q - q_{qvj}\|$ where $Q = \{q \in \mathcal{Q}_j \mid \|q - q_{tg,j}\| = d_{ir}$ and $\|q - q_{qvj}\| > \|q_{tg,k} - q_{qvj}\|\}$, and q_{qvj} is the position of r_{qvj}.

For better understanding the algorithm, we take a formation example as shown in Fig. 4. Now we consider the generation of target-on-queue on $\mathcal{Q}_1$, which contains two formation vertex,

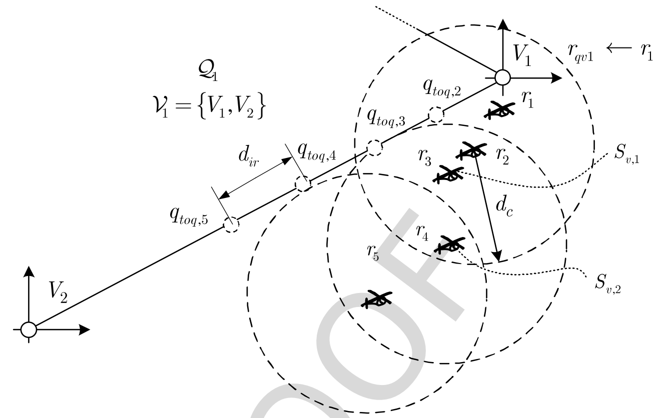


Fig. 4. Generation of target-on-queue.

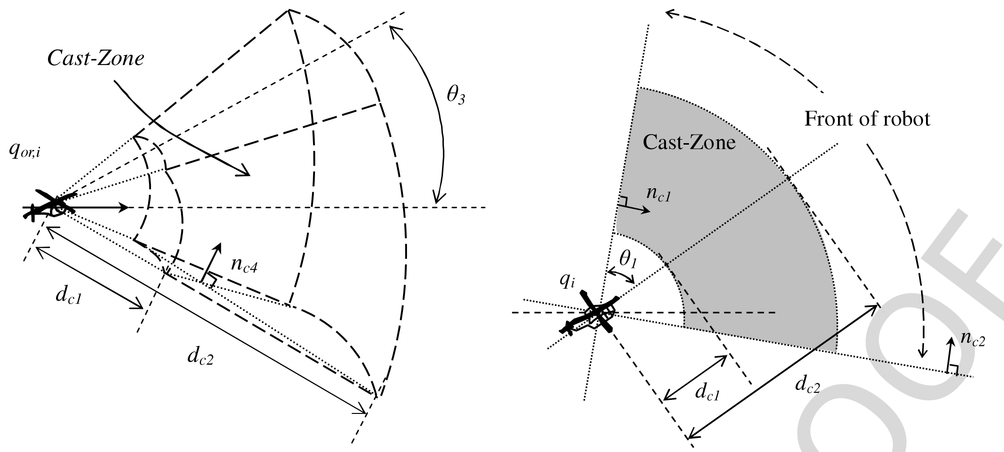
i.e., $\mathcal{V}_1 = \{V_1, V_2\}$ as shown in the figure. Assuming that r_1 is greedily assigned to be the leader in $\mathcal{Q}_1$, its target-on-queue will be given by V_1 and r_1 will be selected as the queue-vertex agent. Both r_2 and r_3 lie in the communication range of r_1, then they can receive the information from r_1 directly. From the figure, we can find that r_4 and r_5 do not have direct link with r_1 due to limited communication range d_c. Then, r_4 and r_5 will emit distress signal. Compared with r_2, r_3 is farther from r_1 and it can detect the distress signal emitted by r_4, then it takes the role of sub-queue vertex agent with a tier lower than r_1, denoted as $S_{v,1}$ in the figure. Similarly, r_4 will act as a sub-queue vertex with a tier lower than $S_{v,1}$ as it can detect the distress signal emitted by r_5.

In the formation, r_1 broadcasts its position along the queue and the current queue orientation. As the sub-queue vertex at level $S_{v,1}$, r_3 will broadcast the information to the next highest level queue-vertex, r_4. According to Algorithm 1, r_2 and r_3 will take $q_{toq,2}$ and $q_{toq,3}$ as its target-on-queue, respectively, as r_3 is far from r_1 than r_2. Accordingly, r_4 and r_5 will take $q_{toq,4}$ and $q_{toq,5}$ as their target-on-queue with distance d_{ir} between any neighbor target-on-queue, which form the shape defined by $\mathcal{S}_1$. It is noted that with distance varying between agents, the sub-queue vertex and the target-on-queue change accordingly.

The main difference between this algorithm and that described in [37] is that queue-keeping is defined with respect to the dynamically chosen r_{qvj}. This has two advantages over the previous algorithm:

(i) It results in clustering and pre-formation of agents belonging to the same queue prior to the actual convergence into formation. There is an explicit partitioning of the team and formation into distinct sub-formations on the planning and movements layers, in addition to the communications layer. It facilitates split-and-rejoin maneuvers by systematically segmenting the formation into sub-formations. Split-and-rejoin maneuvers can be initiated by a separate decision layer that manipulates the positions of the queue vertices by different methods such as those presented in [38]. Again, such manipulation saves on computation because of the reduced number of nodes to control.

(ii) The procedure for changing queues is initiated based on communications between the queue-leaders. Therefore, there is no need for global information at all. Queue vertices maintain communication with other agents on the same queue and with other queue leaders. Other agents maintain communication with only those in their immediate vicinity (local communication) and their associated queue vertex. Sub-queue vertices by design have direct links to at least one (sub)-queue vertex on the same queue and are not required to maintain specific direct links to queue vertices belonging to other queues. This architecture reduces the average amount of information required by each agent for decision making.



(a) 3D view of cast-zone. (b) Top-down view of the cast-zone.

Fig. 5. Cast-zone of an agent located at q_i.

Due to the dynamic addition of (sub)-queue vertices, the structure of the formation, including these sub-queue vertices, is different and can change during runtime depending on the communication range. This is at the agents' level and has no bearing on the actual formation description $\mathcal{F}$, which is set on the (higher) user level.

Remark 5. If the acceptable interrobot distance for robots on the same queue equals to the communication range of the robot, i.e., $d_{ir} = d_c$, each agent will become a sub-queue vertex. In such cases, each queue becomes a collection of sub-queues each containing one (sub)-queue vertex and one other member which is a sub-queue vertex of another queue. This results in a system which is similar to the formation maintenance techniques (such as [14,15]) based on the formations described using traditional graph theory.

3.2. Virtual bobber-agents

Due to the velocity and angular velocity constraints, each agent has an immediate region that it can move to, which is denoted as cast-zone $Z_{cst,i}$ for the agent r_i as shown in Fig. 5. Each agent initializes a virtual-agent, called the virtual bobber-agent in the cast-zone, which provides the agent with a series of intermediate points between its current position and its target-on-queue. This point is determined based on vehicle orientation and motion constraints through the cast-zone, and does not require the agent to make large orientation changes. The use of the virtual bobber-agent narrows the path planning of each agent to its immediate vicinity. Therefore, it requires only communications between an agent and its local neighbors. This is in contrast to other planning algorithms (for omni-directional or non-holonomic vehicles) that uses only on one final target and where convergence and collision avoidance for the entire path must take into account the paths and positions of all other vehicles, even those out of communication range.

For simplicity, this can be a conical or pyramidal region with the axis along the current orientation of the vehicle. The virtual bobber-agent is initialized to a point within the cast-zone and is subjected to a number of forces that keep it within the cast-zone and attracts it to the point in the cast-zone (referred to as the virtual bobber-point) that will cause the vehicle orientate itself as much to the direction of the actual target as possible.

3.2.1. Behavior of virtual bobber-agent

The behavior of a virtual bobber-agent is determined by the attractive force toward the bobber-point, and repulsive forces that restrict its movement to the cast-zone. At the start of each cycle, the embodied agent/vehicle 'casts' the virtual bobber-agent into the cast-zone, and do not impose commands on the virtual bobber-agent, using the equilibrium position of the agent to guide it towards its target-on-queue. The behavior of a virtual bobber-agent associated with agent r_i is mainly governed by two sets of potential functions U_{tcast} and U_{obcast} and the overall potential field is given by

$$U_{cast,i} = U_{tcast,i} + U_{obcast,i}. \tag{1}$$

The potential field, $U_{tcast,i}$, acts like an attractive force that pulls the virtual bobber-agent towards the bobber-point, and is described by for cases when $q_{tg,i} \in Z_{cst,i}$

$$U_{tcast,i} = \arccos\left(\frac{(q_{ba,i} - q_i)^{\mathrm{T}}(q_{tg,i} - q_{ba,i})}{\|q_{ba,i} - q_i\|\|q_{tg,i} - q_{ba,i}\|}\right) \tag{2}$$

where $q_{ba,i}$ is the position of the virtual bobber-agent associated with agent r_i. The second set of potential functions repulses the virtual bobber-agent from the perimeter of the cast-zone so as to enclose the agent within the cast-zone. For a conical region as described above, the potential can be designed as:

$$U_{obcast,i} = \sum_{k=1}^{2} \frac{1}{2(\|q_{ba,i} - q_i\| - d_{ck})^2} + \sum_{k=1}^{4} \frac{1}{(q_{ba,i} - q_i)^{\mathrm{T}} n_{ck}} \tag{3}$$

where d_{c1} and d_{c2} are as shown in Fig. 5 and $d_{c1} < \|q_{ba,i} - q_i\| < d_{c2}$, and n_{ck} ($k = 1, \dots, 4$) are the unit vectors perpendicular to the four planes bounding the (pyramidal-shaped) cast-zone. Note that these quantities may vary between different agents, which will result in different sizes for the respective cast-zones. The angle that each plane that bounds the cast-zone makes with the current orientation of the vehicle ($q_{or,i}$) is defined by the user, and denoted by θ_k (for $k = 1, 2, 3, 4$ corresponding to the planes to the left, right, top and bottom of $q_{or,i}$, respectively). With these values of θ_k, the values of n_{ck} at each time given $q_{or,i}$ can be obtained as follows:

$$n_{ck} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 \end{bmatrix} q_{or,i}, \quad \text{for } k = 1, 2, \text{ and}$$

$$\phi = (-1)^{k+1}\left(\frac{\pi}{2} - \theta_k\right) \tag{4}$$

and

$$n_{ck} = \begin{bmatrix} x_{or,i}\cos\phi \\ y_{or,i}\cos\phi \\ \sin\phi \end{bmatrix}, \quad \text{for } k = 3, 4, \text{ and}$$

$$\phi = (-1)^{k}\left(\frac{\pi}{2} - \theta_k\right). \tag{5}$$

The virtual bobber-agent is treated as a simple fully actuated point mass, and its overall behavior is generated from the negative gradient of the overall potential field U_{cast}, i.e., we have

$$\dot{q}_{ba,i} = -\nabla U_{cast,i} = -\nabla(U_{tcast,i} + U_{obcast,i}). \tag{6}$$

The gradient of the attractive and repulsive functions in (2) and (3) are computed as follows.

$$\nabla U_{tcast,i} = \frac{-1}{\sqrt{1-g_i^2}} \frac{\partial g_i}{\partial q_{ba,i}} \tag{7}$$

where $g_i = \frac{(q_{ba,i}-q_i)^{\mathrm{T}}(q_{tg,i}-q_{ba,i})}{\|q_{ba,i}-q_i\|\|q_{tg,i}-q_{ba,i}\|}$. We have

$$\begin{aligned}\frac{\partial g_i}{\partial q_{ba,i}} &= \frac{-2q_{ba,i}+q_i+q_{tg,i}}{\|q_{ba,i}-q_i\|\|q_{tg,i}-q_{ba,i}\|} \\ &\quad - \frac{(q_{ba,i}-q_i)^{\mathrm{T}}(q_{tg,i}-q_{ba,i})(q_{ba,i}-q_i)}{\|q_{ba,i}-q_i\|^3\|q_{tg,i}-q_{ba,i}\|} \\ &\quad + \frac{(q_{ba,i}-q_i)^{\mathrm{T}}(q_{tg,i}-q_{ba,i})(q_{tg,i}-q_{ba,i})}{\|q_{ba,i}-q_i\|\|q_{tg,i}-q_{ba,i}\|^3}.\end{aligned} \tag{8}$$

For the repulsive forces from the boundaries of the cast-zone, we have

$$\begin{aligned}\nabla U_{obcast,i} &= -\sum_{k=1}^{2} \frac{(q_{ba,i}-q_i)}{\|q_{ba,i}-q_i\|(\|q_{ba,i}-q_i\|-d_{ck})^3} \\ &\quad - \sum_{k=1}^{4} \frac{n_{ck}}{((q_{ba,i}-q_i)^{\mathrm{T}} n_{ck})^2}.\end{aligned} \tag{9}$$

For each virtual bobber-agent (BA) cycle, the virtual bobber-agent uses the information regarding q_i and $q_{tg,i}$ at the start of the cycle to compute its movement through the cast-zone. Algorithm 2 describes the process for the determination of potentials within the cast-zone, $Z_{cst,i}$, in particular $U_{cast,i}$, and the movement the virtual bobber-agent toward the equilibrium point in the cast-zone. The vehicle r_i will then use the equilibrium position of the virtual bobber-agent as an intermediate target or waypoint for moving toward its actual target on the formation queue.

Algorithm 2 Determination of Cast-Zone Potentials and Bobber-Agent Movement

if $q_{tg,i} \in Z_{cst,i}$ **then**
  Set $q_{ba,i} = q_{tg,i}$.
else if $q_{or,i}^{\mathrm{T}}(q_{tg,i} - q_i) > 0$ and $(q_{tg,i} - q_i)$ passes through the cast-zone **then**
  // Use a normal attractive potential to target.
  Set $U_{tcast,i} = 1/2\|q_{ba,i} - q_{tg,i}\|$.
  Compute $\dot{q}_{ba,i}$ according to (6)
else
  // A turn is necessary to orientate the vehicle correctly.
  Set
  $$U_{tcast,i} = \arccos\left(\frac{(q_{ba,i}-q_i)^{\mathrm{T}}(q_{tg,i}-q_{ba,i})}{\|q_{ba,i}-q_i\|\|q_{tg,i}-q_{ba,i}\|}\right) \tag{10}$$
  (as described in (2).
Compute $\dot{q}_{ba,i}$ according to (6)

3.3. Convergence of bobber-agents towards minimum point in the cast-zone

To examine the behavior of the virtual bobber-agent in the cast- zone, first, let us consider only the potential $U_{tcast,i}$. Given that $U_{tcast,i}$ is as described in (2), a virtual bobber-agent under the influence of this field will be stable and converge toward the critical points of the surface defined by $U_{tcast,i}$, i.e., we have

$$\dot{U}_{tcast,i} = \nabla U_{tcast,i}\dot{q}_{ba,i} \tag{11}$$

where $\nabla U_{tcast,i} = \frac{\partial}{\partial q_{ba,i}} U_{tcast,i}$. Thus, by choosing the movement of the virtual bobber-agent, under only the influence of $U_{tcast,i}$, to be

$$\dot{q}_{ba,i} = -C\nabla U_{tcast,i} \tag{12}$$

where C is a positive definite matrix, $\dot{U}_{tcast,i}$ is strictly negative. Furthermore, at the critical point of $U_{tcast,i}$,

$$\nabla U_{tcast,i} = 0. \tag{13}$$

This is satisfied only at the midpoint between the straight line joining q_i and $q_{tg,i}$. At this point, q_{ecast}, the first terms of individual components of $\nabla U_{tcast,i}$ becomes zero, $-2q + q_i + q_{tg,i} = 0$, where q is a point in space. In addition, we observe that at q_{ecast}, $\|q_{ba,i} - q_i\| = \|q_{tg,i} - q_{ba,i}\|$ and the second and third term of the components cancels each other. Thus, a virtual bobber-agent influenced by $U_{tcast,i}$, will converge toward the global minima of the surface.

The repulsive forces, $U_{obcast,i}$, restrain the movement of the virtual bobber-agent within the cast-zone, which is predefined as described in the sections above. This results in the possible existence of local minima within the cast-zone. Since $U_{tcast,i}$ essentially depends on the angle between the vectors $q_{ci} = q_{ba,i} - q_i$ and $q_{tgc} = q_{tg,i} - q_{ba,i}$, it can be observed that in the 2D case, the potential is symmetric about the line joining q_i and $q_{tg,i}$, as shown in Fig. 6(a). For the 3D case, the potential is the same for points lying along circular rings with centers along $q_i - q_{tg,i}$, i.e., points belonging to the set of points $Z_{cst,i} \in \mathbb{R}^3$ such that for all $q \in Z_{cst,i}$,

$$(q_{ba,i} - q_i)^{\mathrm{T}}(q_{tg,i} - q_i) = d_{u1} \tag{14}$$

$$\left\|(q_{ba,i} - q_i) - d_{u1}\left(\frac{(q_{tg,i} - q_i)}{\|q_{tg,i} - q_i\|}\right)\right\| = d_{u2} \tag{15}$$

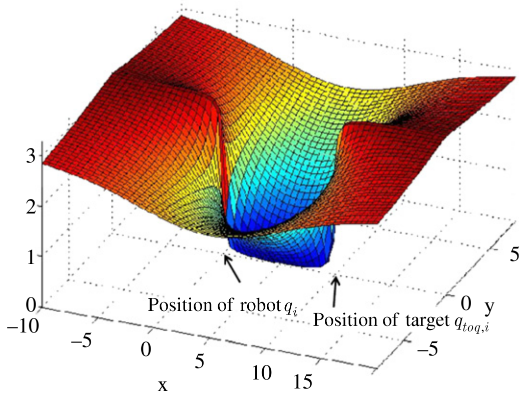
where d_{u1} and d_{u2} are the distance from q_i and the perpendicular distance to the vector $(q_{tg,i} - q_i)$, respectively. Furthermore, the vertex of the cone describing the cast-zone is always at q_i, and local minima in the cast-zone exist only when the line through $q_{tg,i}$ and q_i passes through the cast-zone and such that

$$q_{ci}^{\mathrm{T}}(q_{tg,i} - q_i) < 0. \tag{16}$$

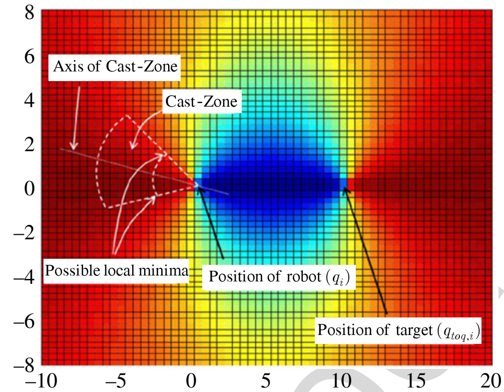
Thus, the virtual bobber-agent will be initialized on the 'wrong side' of the cast-zone. One simple way of preventing it from converging to the local minima is by initializing it along the axis of the cast-zone that is shown in Fig. 6.

3.4. Generation of desired trajectories for individual agents

Formation control takes place on a higher level involving the generation of a desired trajectory. Separate control laws, depending on the system involved [40,41], can then be generated to track the trajectory. In this article, the desired orientation of the vehicle along this trajectory will also be considered. Specifically, the desired orientation of each vehicle in the x–y plane (yaw). For instance, for helicopter formations, which while able to hover and rotate at a point, it may be desirable to produce a desired trajectory that involves less of such maneuvers. As described in several previous works on helicopter control (e.g., the articles [42–44]), helicopters belong to the class of feedback linearizable systems. The main objective is to produce a path for each vehicle, such that the required change in heading (in the x–y plane) when moving to the next position is kept within a certain user-defined range as much as possible.



(a) 3D view of potential.



(b) Top-down view of potential and the cast zone of virtual bobber-agent.

Fig. 6. Attractive potential ($U_{tcast,i}$) in the cast-zone.

At any time instant, the intermediate targets, $q_{tg,i}$, are determined by the virtual bobber-agents using Algorithms 1 and 2. The target of each vehicle is determined by the algorithm for each cycle, and thus for each iteration, the targets ($q_{tg,i}$) are taken to be constant. A new cycle begins when the vehicle reaches its current intermediate target, and the virtual bobber-agent is used again to determine the next intermediate target. The desired trajectory, in an n_w-dimensional space, from the current position to the intermediate target is given by the path of q_i which follows

$$\dot{q}_i = u_i \tag{17}$$

where u_i is based on a potential field that will be described in the following.

For each cycle, each agent considers only those others within its communications range d_c. It is assumed that the communications range of each agent is the same. The overall potential function is

$$U = U_{tg} + U_{ob}. \tag{18}$$

The first part, U_{tg}, describes the attractive potentials between the vehicles and their targets, and may be written as:

$$U_{tg} = \frac{1}{2}\sum_{i=1}^{N} \|q_i - q_{tg,i}\|^2 \tag{19}$$

where N is the number of agents in the team. The function U_{ob} is chosen such that it is equal to infinity when collisions occur and minimum when the agents are at their intermediate targets. In this article, U_{ob} is given by

$$U_{ob} = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} U_{ob,ij} \tag{20}$$

where $U_{ob,ij}$ is a function of U_{ij} and $U_{tg,ij}$, which are given by

$$U_{ij} \;=\; \frac{1}{2}(\|q_i - q_j\| - d_{ob,ij})^2 \tag{21}$$

$$U_{tg,ij} \;=\; \frac{1}{2}\|q_{tg,i} - q_{tg,j}\|^2 \tag{22}$$

where $d_{ob,ij}$ is the minimum acceptable distance between i and its neighbor j (which can be a virtual constellation- agent, described in the subsequent section) and $U_{ob,ij}$ is chosen to be

$$U_{ob,ij} = \frac{1}{1 + \exp(a_t(U_{ij} - U_{tg,ij})^3)}\left(\frac{U_{ij}}{U_{tg,ij}^2} + \frac{1}{U_{ij}}\right) \tag{23}$$

where a_t is a user-defined constant and such that

(a) $U_{ob,ij} = \infty$, if $U_{ij} = 0$.
(b) $U_{ob,ij} > 0$, if $U_{ij} \neq 0$.
(c) $U'_{ob,ij} = \frac{\partial U_{ob,ij}}{\partial U_{ij}} = 0$, if $U_{ij} = U_{tg,ij}$.
(d) $U''_{ob,ij} = \frac{\partial^2 U_{ob,ij}}{\partial U_{ij}^2} \geq 0$, if $U_{ij} = U_{tg,ij}$.
(e) $U_{ob,ij} \approx 0$, if $U_{ij} \geq 0.5\bar{d}_c^2$

where $\bar{d}_c$ is an user-defined value. For the rest of this section, we first examine the stability and convergence of the system under the potential fields described above. Then, it is shown that choosing $\bar{d}_c \leq d_c - 4d_{c2}$ and an appropriate d_{c2} will allow each agent, for each cycle, to only consider other agents that are (i) within its communications range at the start of each cycle, and (ii) which stays within its communications range for that cycle. This further reduces information requirements by considering only selected agents within range.

3.4.1. Virtual constellation-agents and vehicle movements

To bias the movement of the vehicle in a certain general direction given by a cone around its current orientation, a vehicle interacts not just with other physical vehicles around it, but also with a set of virtual agents (referred to as 'virtual constellation-agents' for the rest of the article). In comparison with β-agents presented in [29], the virtual constellation-agents consists of a group of interacting agents, while β-agents operate as a single entity. The virtual constellation-agents interact based on the guiding-line. In addition to providing a repulsive force from obstacles (like the β-agent), the virtual constellation-agents also act as a guide for the direction of movement for the agent.

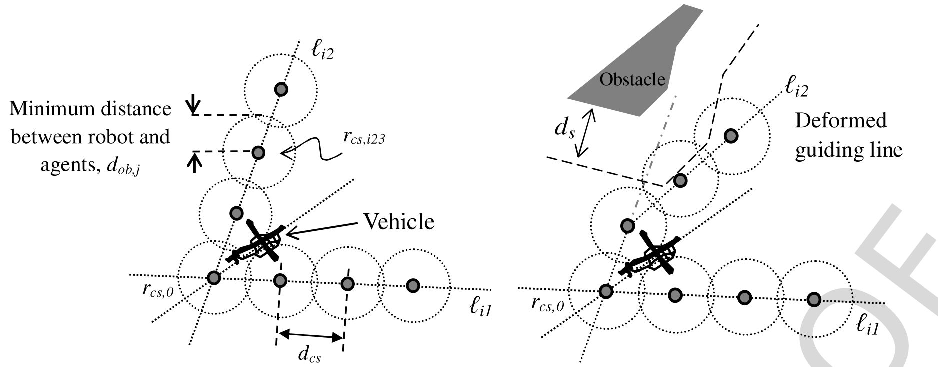
Each vehicle has a set of virtual constellation-agents around it, which is denoted as N_{cs}. The virtual constellation-agents are divided into $n_{cs,sb}$ subsets, each denoted by $N_{cs,k}$ (for $k = 1, 2, \ldots, n_{cs,sb}$), such that we have

$$\bigcup_{k=1}^{n_{cs,sb}} N_{cs,k} = N_{cs} \tag{24}$$

$$N_{cs,k1} \bigcap N_{cs,k2} = r_{cs,i0}, \quad \forall k_1, k_2 = 1, 2, \ldots, n_{cs} \text{ and } k_1 \neq k_2 \tag{25}$$

where $r_{cs,i0}$ is the virtual constellation-agent that lies on all the guiding-lines (let each guiding-line be $\ell_{i,k}$), as shown in Fig. 7(a). The repulsive-distance between a vehicle r_i and each of its virtual constellation-agents, $r_{cs,ika}$,[2] is given by $d_{ob,ik}$, and let the distance between virtual constellation- agents belonging to each subset $N_{cs,k}$ be d_{cs}.

[2] The subscripts i, k, and a represent the vehicle, subset of virtual constellation-agent, and position on the guiding-line, that this virtual constellation-agent is associated with.



(a) A set of virtual constellation-agents.

(b) Deformation of guiding-line in the presence of an obstacle.

Fig. 7. Constellation-agents around a vehicle.

The virtual constellation-agents move together with the vehicle, and they restrict the region which the vehicle can move to in the next cycle. The presence of the virtual constellation-agents prevents the vehicle from moving beyond the boundaries of the cast-zone, by acting like physical agents in the operating space. Furthermore, the inclusion of virtual constellation-agents allows consideration of physical obstacles by through the deformation of the guiding-lines. The deformation of guiding-lines causes the displacement of not just the virtual constellation-agent that is immediately influenced by the obstacle, but also of other virtual constellation-agents that lie further along the line. Compared with a simple repulsion from the nearest point on the obstacle, guiding-line deformation causes the vehicle to perform wider maneuvers in anticipation of possible obstacle surfaces it may encounter. Fig. 7(b) illustrates the scenarios with guiding-line deformation.

Each virtual constellation-agent has a default position on its associated guiding-line, and will change this position according to the procedure described in Algorithm 3, with $d_{ob,ika}$ as the distance from the vehicle to the nearest point on an obstacle in the direction $q_{ob,ika} = q_{ika} - q_i$, and d_s is the safety distance a vehicle have to keep from any obstacle.

Algorithm 3 Determining Position/Target of Constellation-Agent, $r_{cs,ika} \in N_{cs,k}$

if $r_{cs,ika}$ is the first or second along the guiding-line (i.e., a = 0 or 1) **then**
  Let $d_t = \min(d_{ob,ika} - d_s, \|q_{ob,ika}\|)$.
  Set $q_{ika,new} = q_i + d_t q_{ob,ika}$.
else if $(d_{ob,ika} - d_s < \|q_{ik(a-1)} + d_{cs}(q_{ik(a-1)} - q_{ik(a-2)}) - q_i\|)$ **then**
  Set $q_{ika,new} = q_i + (d_{ob,ika} - d_s)q_{ob,ika}$.
else
  Set $q_{ika,new} = q_{ik(a-1)} + d_{cs}(q_{ik(a-1)} - q_{ik(a-2)})$.

At each time instant, each vehicle moves along the negative gradient of the potential function U. Due to the presence of virtual constellation-agents, the overall potential function includes the associated repulsive potentials as well. The time derivative of the overall potential function U in (18) is given by

$$\dot{U} = \sum_{i=1}^{N}(q_i - q_{tg,i})^{\mathrm{T}}u_i + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} U'_{ob,ij}(q_i - q_j)^{\mathrm{T}}(u_i - u_j)$$
$$= \sum_{i=1}^{N}\left((q_i - q_{tg,i})^{\mathrm{T}} + \sum_{j\neq i}^{N} U'_{ob,ij}q_{ij}^{\mathrm{T}}\right)u_i$$
$$= \sum_{i=1}^{N}\Omega_i^{\mathrm{T}}u_i \quad (26)$$

where $N = N_R + N_R N_{cstl}$, $q_{ij} = q_i - q_j$, and Ω_i is defined as

$$\Omega_i = (q_i - q_{tg,i}) + \sum_{j\neq i}^{N} U'_{ob,ij}q_{ij}. \quad (27)$$

This implies that a choice of

$$u_i = -C\Omega_i \quad (28)$$

where $C \in \mathbb{R}_{+}^{n_w \times n_w}$ is a symmetric, positive definite matrix, which is chosen as $C = c\mathbf{I}_{n_w \times n_w}$ where $c > 0$, will result in

$$\dot{U} = -\sum_{i=1}^{N}\Omega_i^{\mathrm{T}}C\Omega_i \quad (29)$$

and the closed loop dynamics of a single vehicle r_i in the team is then given by

$$\dot{q}_i = -C\Omega_i. \quad (30)$$

If the vehicles are at different positions (i.e. non-colliding) at an initial time t_0, and the target of each vehicle is different as well, these conditions may be written as

$$\|q_i(t_0) - q_j(t_0)\| \geq \epsilon_1 \quad (31)$$

where ϵ_1 is a strictly positive constant, and R is the set of vehicles comprising the team. In addition, Algorithm 1 guarantees that if the condition in (31) is satisfied, the targets for each cycle do not collide, i.e., $\|q_{tg,i} - q_{tg,j}\| \geq \epsilon_2,\ \forall i, j \in R$, where ϵ_2 is strictly positive. It is thus desired that, under such conditions, each vehicle will converge toward their targets, and at the same time avoiding collisions, i.e.

$$\lim_{t\to\infty}(q_i(t) - q_{tg,i}) = 0$$
$$\|q_i(t) - q_j(t)\| \geq \epsilon_3, \quad \forall i, j \in R \text{ and } \forall t \geq t_0 \geq 0 \quad (32)$$

where ϵ_3 is a strictly positive number representing the minimum acceptable inter-vehicular distance.

Lemma 6. *Under the conditions stated in (31) and Algorithm 1, the control input to each vehicle, given in (28), with the vehicle target determined by the virtual bobber-agent as in Algorithm 2, each vehicle will converge in finite time to their virtual bobber-agent targets, and such that:*

(i) *The target at q_{tg} is located at an asymptotically stable equilibrium point of (30), and*
(ii) *The critical points of the system other than that at q_{tg} are unstable equilibrium points.*

Proof. Please refer to Appendix. □

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005

Lemma 7. *Under the condition that the intermediate targets are asymptotically stable equilibrium points, d_{c2} can be chosen such that the effect of agents initially out of each other's communication ranges at the start of a cycle will be negligible, i.e., information from such agents is not necessary in the generation of u_i.*

Proof. It has been shown in Lemma 6 that the intermediate targets are asymptotically stable. For each cycle, with $q_{tg,i}$ a distance of $d_{tg,i}$ away from q_i at the beginning of that cycle, the movements of the agent i will be constrained to a sphere of radius $d_{tg,i}$ around $q_{tg,i}$. Furthermore, a cast-zone $Z_{cst,i}$ with a maximum range of d_{c2} implies a maximum radius of the sphere to be d_{c2}.

Consider an agent j initially outside the communication range of agent i, i.e., at the start of the cycle $\|q_j - q_i\| \geq d_c$. The movements of both agents is confined to a sphere with at most a radius of d_{c2}, at any time t within the cycle, the minimum distance between agents i and j is given by $d_c - 4d_{c2}$, occurring when j is just outside the communication radius of i.

Hence, choosing $d_{c2} \leq 0.25(d_c - d_x)$ where $d_x > 0$, will ensure that $\|q_j - q_i\| \geq d_x$, and setting $\bar{d}_c = d_x$, from the design of $U_{ob,ij}$ (condition (e)), the contribution of the information from j to u_i is negligible. □

In comparison with other graph-based formation approaches, the proposed method builds upon the advantages provided by the Q-structure in terms of representation consistency and scalability described in detail in [37]. Further, from the preceding sections, only directly communicated information is necessary for immediate action determination. If $N_{c,i}$ is the number of agents within the set $R_{c,i}$, a maximum of $N_{c,i}$ direct links are established. This is improved upon the scheme proposed in [37] in which an agent must also maintain adequate information flow with the main queue vertex, i.e., communication links can reach $N_{c,i} + 1$ for large but simple formations with very few queue vertices. Formation maintenance schemes such as [14,15] also require a maximum of $N_{c,i} + 1$ links for an agent to effectively follow a leader or the preceding node in the formation's graphical representation, while approaches such as virtual structures require significantly more communication links up to a maximum of the total number of agents in the system given by $N \geq N_{c,i}$.

Remark 8. In practice, the communication range is generally large when compared with the desired inter-agent distance or the minimum obstacle avoidance distance, i.e., $d_c \gg d_{ir} \geq d_{ob,ij}$. The value of d_{c2} is also chosen such that $d_x > d_{ir}$ so that agents within collision range of each other are considered. In addition, it can be seen that d_{c2} decreases with smaller d_c, meaning that with a reduced communications range, the intermediate targets for consecutive cycles are closer together. Intuitively, this means that the agent is more careful, taking smaller steps per cycle, when communication range is small and less direct information is available.

3.5. Overall vehicle target and behavior generation

The overall process undertaken by each vehicle r_i during each BA-cycle is described in Algorithm 4. After the determination of the target-on-queue, each vehicle computes its desired behavior based on the locations of other vehicles around it. This gives the desired direction of motion for the vehicle which takes into account obstacle avoidance behaviors between vehicles as well. However, given the restricted region the vehicle can move in given its current orientation, the vehicle may not be able to move in that direction without first reorienting itself. In such circumstances, the vehicle casts the virtual bobber-agent into the cast-zone to determine a possible intermediate target that is immediately reachable. Using this intermediate target, the vehicle again determines the desired velocity ($u_{d,i}$) to reach it. Only when this target is not immediately reachable (i.e., r_i is unable to move in the direction given by $u_{d,i}$), the vehicle will reorientate itself at its current position to the direction given by u_i. The use of virtual bobber-agents as intermediate targets hence help reduce the need for reorientation, especially for non-omni-directional vehicles, thus, making movements smoother.

Algorithm 4 Movement of Vehicle r_i via Bobber-Agents

1: Determine the target-on-queue using Algorithm 1
2: Using $q_{tg,i}$, determine the Cast-Zone, $Z_{cst,i}$, and the potential, $U_{tcast,i}$, for the virtual bobber-agent.
3: Obtain the equilibrium position, $q_{eba,i}$, of the virtual bobber-agent. This will be used as the intermediate target of the vehicle, i.e., r_i sets its target as $q_{eba,i}$
4: The required movement of r_i in response to $q_{eba,i}$ is computed with (28) and is given by u_i.
5: **if** (r_i is unable to move in the direction u_i) **then**
6:     The vehicle r_i will remain at its current position and orientate itself to u_i. Thus, reorientate-and-move maneuvers will only be used if a vehicle should find itself trapped either between obstacles or other vehicles.
7: **else**
8:     Move in the direction given by u_i.

Lemma 9. *Given the target determination processes and the convergence of vehicles towards their targets per BA-cycle, the vehicles will converge towards their positions on their respective queues within finite time.*

Proof. Consider a BA-cycle that is within the time interval $[t_{bas}, t_{bae}]$. From Algorithm 2, it can be seen that, if the target-on-queue $q_{toq,i}$ is 'to the front' of the vehicle (see Fig. 5), the equilibrium position of the virtual bobber-agent for the BA-cycle will be such that

$$\|q_{eba,i} - q_{toq,i}\| < \|q_i(t) - q_{toq,i}\|, \quad \text{where } t_{bas} \leq t < t_{bae} \tag{33}$$

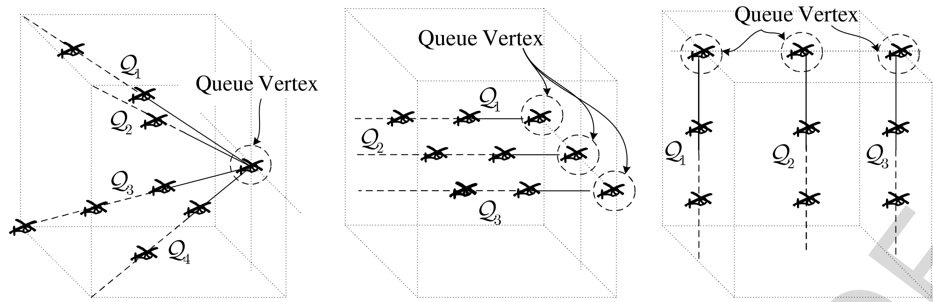
where $\|q_i(t) - q_{toq,i}\| - \|q_{eba,i} - q_{toq,i}\| = d_{cz}$ with d_{cz} being the length of the cast-zone. Since the vehicle will converge toward $q_{eba,i}$ (by Lemma 6), and the BA-cycle ends when $q_i = q_{eba,i}$, we see that $\|q_{eba,i} - q_{toq,i}\|$ (and hence $\|q_i(t) - q_{toq,i}\|$) will decrease after each BA-cycle. This continues until $q_{tg,i} \in Z_{cst,i}$ and $q_{eba,i} = q_{toq,i}$ (lines 1 to 2 of Algorithm 2). Therefore, $q_{eba,i}$ converges to the desired target-on-queue in a finite number of BA-cycles, and implies that r_i will converge toward $q_{toq,i}$ in the final BA-cycle.

For the case when $q_{toq,i}$ is not 'to the front' of the vehicle, i.e., $\|\theta_{itoq}\| > \theta_{cz,i}$, Algorithm 2 (lines 8 and 9) results in $q_{eba,i}$ such that, the angle between vectors $q_{eba,i} - q_i$ and $q_{toq,i} - q_i$ is less than θ_{itoq}. This implies that $\theta_{itoq}(t_{bae}) < \theta_{itoq}(t_{bas})$ and θ_{itoq} decreases over a finite number of BA-cycles to a value such that $\|\theta_{itoq}\| \leq \theta_{cz,i}$. The convergence towards $q_{toq,i}$ then follows that described for the case described earlier. □

4. Simulation studies

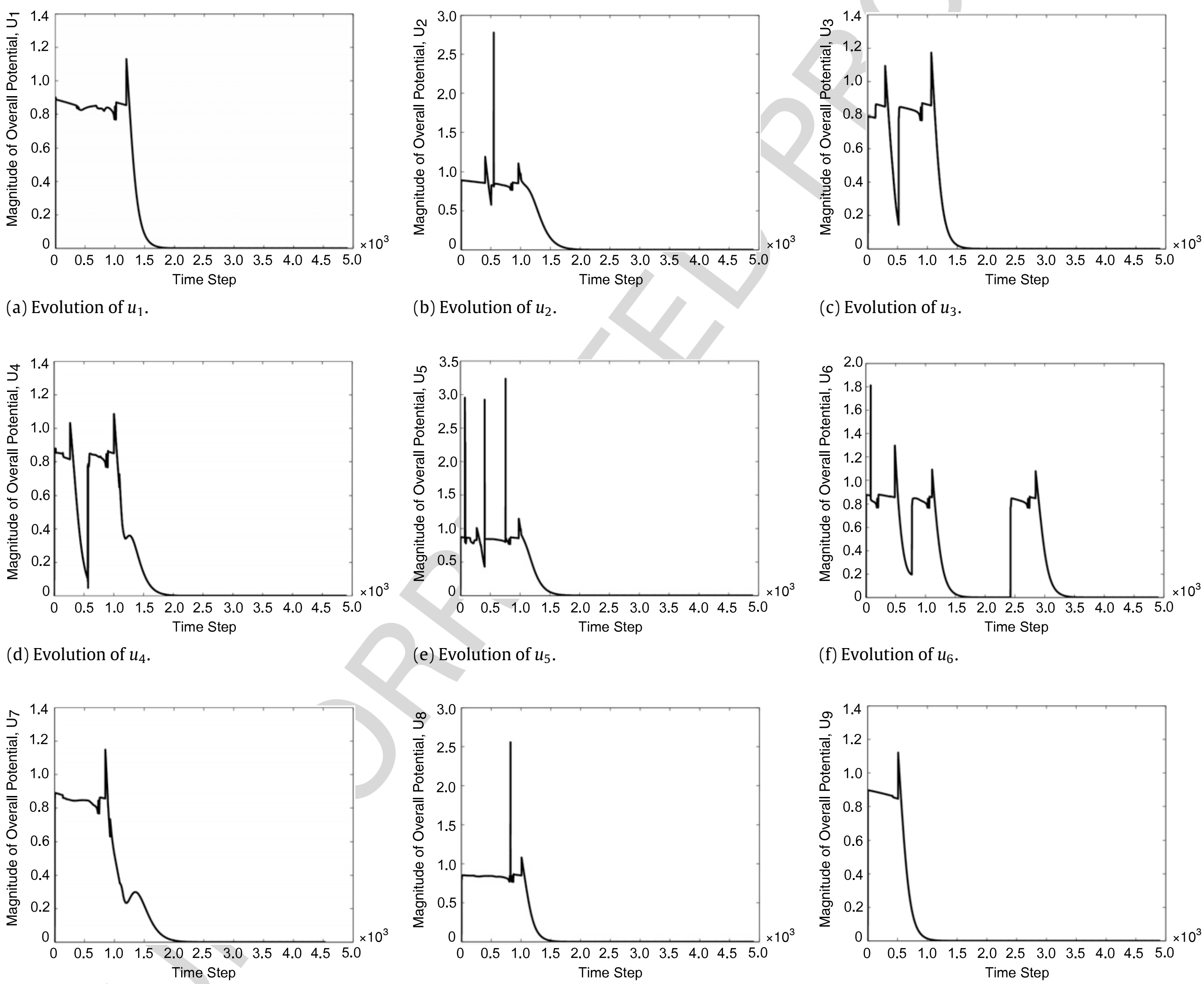
Simulations were carried out to observe the reactions of the agents as they move into formations, while interacting with other agents in the team, mainly their obstacle avoidance abilities and the amount of orientation change experienced by each vehicle. It is assumed that each robot is able to localize itself in the global frame. Furthermore, each robot is equipped with a laser scanner (180°) and 16 sonar range sensors arranged in a ring around the circular robots for obstacle avoidance. The simulation parameters are as follows: $d_c = 6$, $d_{ir} = 1$, $d_{ob,ij} = 0.25$ and $d_{c2} = 0.5$.

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005



(a) Pyramidal wedge formation.   (b) 3 parallel horizontal lines formation.   (c) 3 parallel vertical lines formation.

Fig. 8. Three different shapes of formation used in simulation.



(a) Evolution of u_1.   (b) Evolution of u_2.   (c) Evolution of u_3.

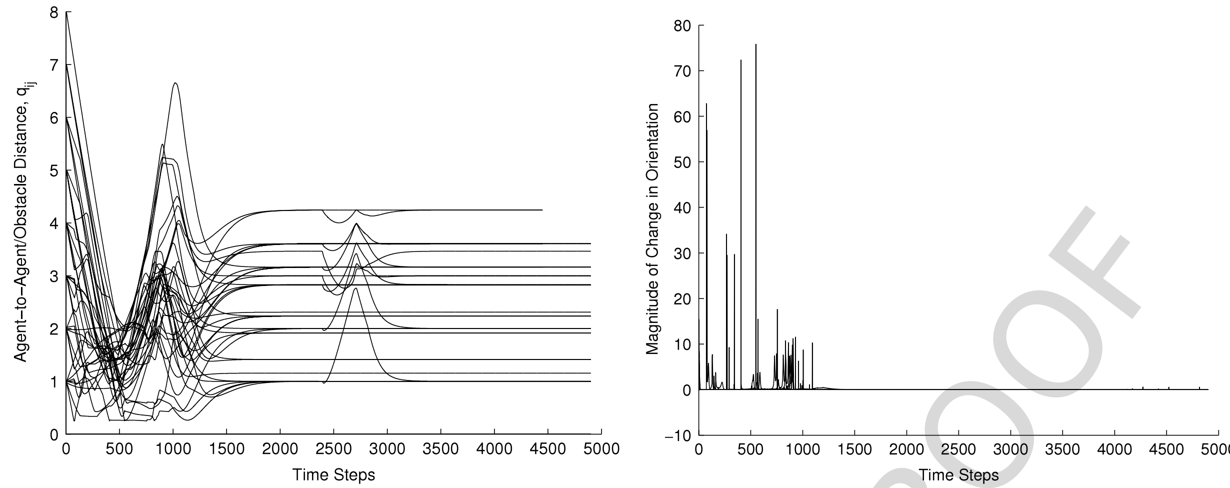
(d) Evolution of u_4.   (e) Evolution of u_5.   (f) Evolution of u_6.

(g) Evolution of u_7.   (h) Evolution of u_8.   (i) Evolution of u_9.

Fig. 9. Convergence of formation with 9 agents.

The efficient convergence of the agent team into the required formation is also examined. Each agent is assumed to be able to move only to points that require less than 45° change in yaw, and are also assumed to be able to hover and move vertically. If the agent is required to change its direction by more than 45°, it will do a on-the-spot rotation.

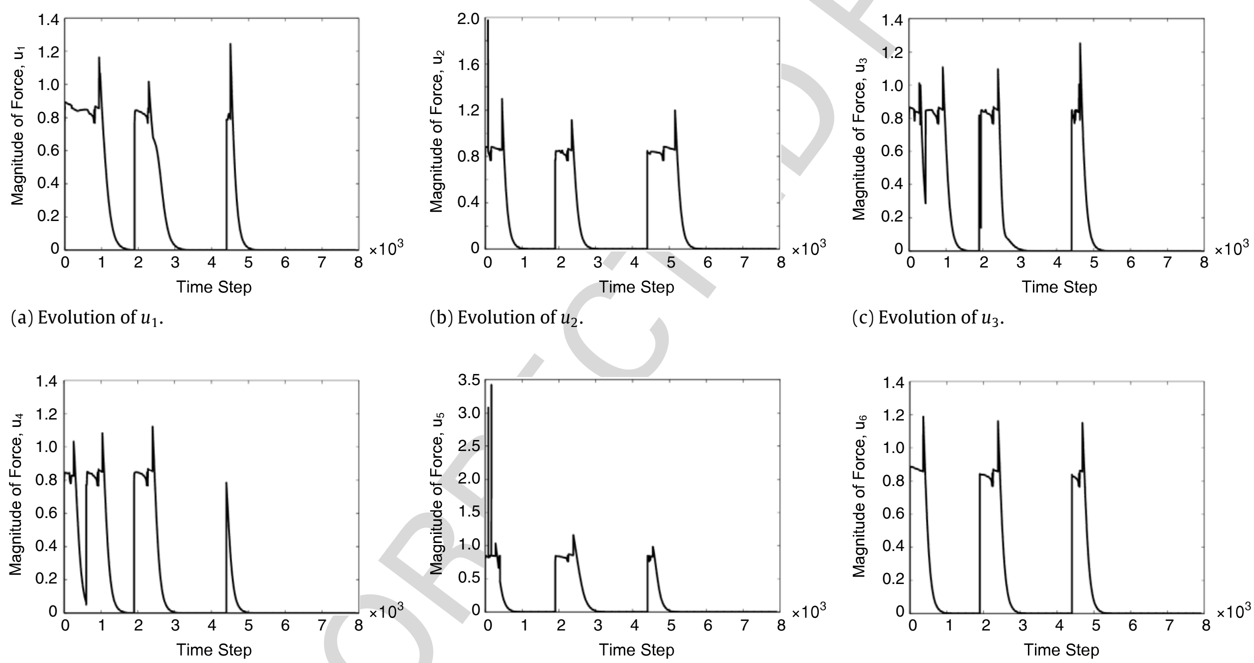
The results for a simple simulation with 9 agents is shown in Figs. 9 and 10. The desired formation is a pyramidal wedge formation as shown in Fig. 8(a), which contains 4 open queues. As observed from the magnitude of the force experienced by each vehicle over time, each of these forces decay to zero in a relatively short time, indicating that each agent reaches its target (as determined dynamically in Algorithm 1) in time. Changes in target manifest themselves in the form of sharp spikes in the graphs, caused by the non-continuous movement of an agents target when the target changes in response to Algorithm 1. Fig. 10(a) shows the distance between any two agents over time. It can be observed that the agents are always a minimum of 0.25 m apart. Convergence of



(a) Absolute distances between any two vehicles in the team.

(b) Absolute change in agent heading (in degrees).

Fig. 10. Inter-agent collision avoidance and agent directional changes.



(a) Evolution of u_1.

(b) Evolution of u_2.

(c) Evolution of u_3.

(d) Evolution of u_4.

(e) Evolution of u_5.

(f) Evolution of u_6.

Fig. 11. Convergence of formation with 6 agents changing between formations.

agents into formation can also be seen from the graphs as the distances between agents stabilizes when they enter their final positions in formation. The required direction/orientation change for each agent from one time step to the next is shown in Fig. 10(b). From the graphs, the required directional change is for the most part, limited to less than 45°, except for a few instances when the agents are starting to get into formation. These spikes can be attributed to the proximity of agents when they start out, and reorientation may be necessary in certain situations to prevent collisions. In addition, simulations were done to observe agent convergence and formation stability as the agent team switches between different formations.

The results in Figs. 11 and 12 show the forces, and inter-agent distances as the team of 6 agents switches between (i) a pyramidal wedge, (ii) 3 parallel horizontal lines, and (iii) 3 parallel vertical lines.

Finally, we examine the ability of the team of 6 agents to maneuver through an obstacle field with a pyramidal wedge formation. A total of 10 obstacles are strewn at random around the area that the formation will move through as shown in Fig. 13. The results are shown in Fig. 14. It can be observed from the graphs that the agents are able to form into and move in formation despite the presence of obstacles. Furthermore, the agents are able to successfully negotiate through the obstacle field without collisions, as can be seen from the graphs in Fig. 15, where the minimum distance between any agent and any obstacle is always above 0.5. From Fig. 15, we can also observe the movement of the formation as it moves into the obstacle field, where the agent-to-obstacle distances starts decreasing, and also where the distance starts increasing as the team moves out of the obstacle field.

5. Conclusions

In this article, the Q-structure has been extended to include orientation information of the final formation in 3D spaces, and to

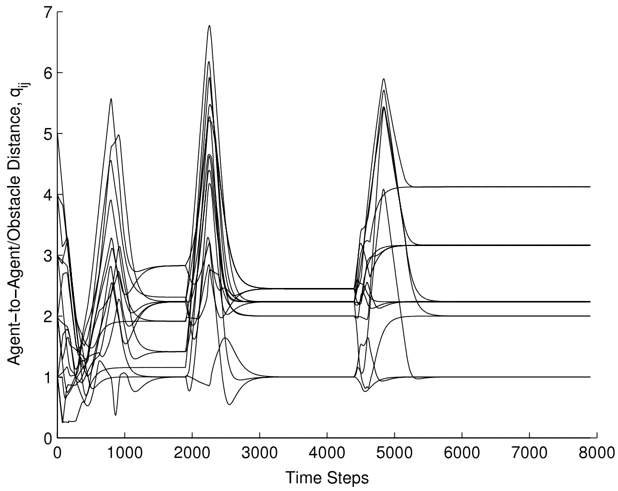


Fig. 12. Inter-agent separation as the team changes between formations.

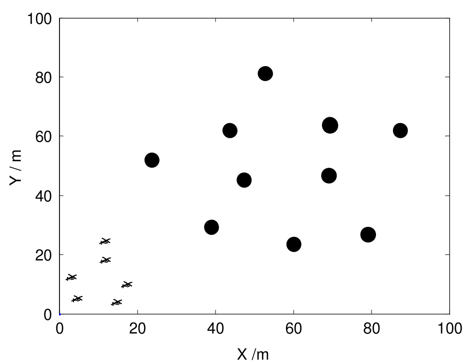


Fig. 13. Six agents maneuvering through an obstacle field.

include a mechanism for adapting the communication structure that overlays the Q-structure to account for limited communication ranges. This allows restriction of short term information flows

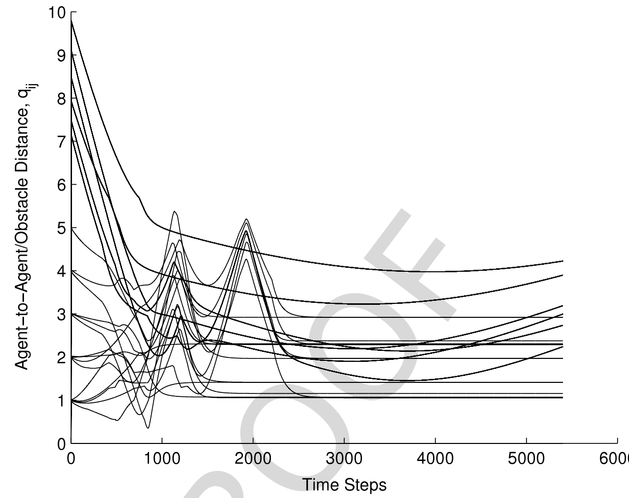


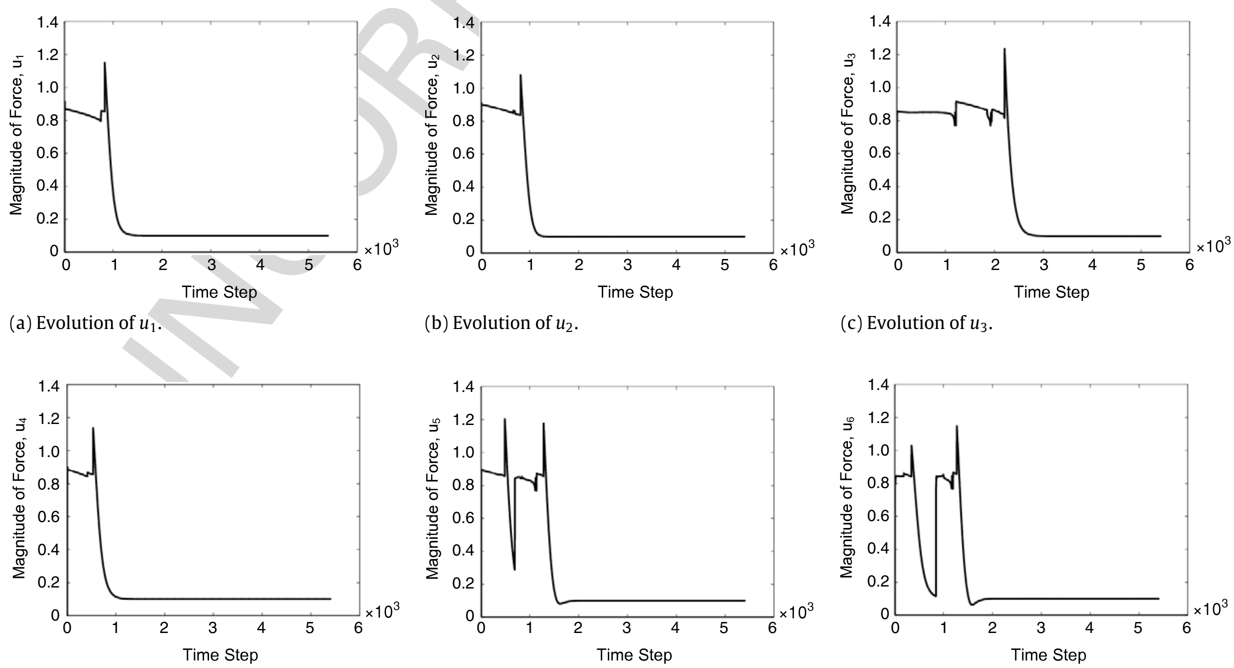
Fig. 15. Distance between different agents, and the distance between agents and obstacles.

to strictly between agents with direct communication links with each other. Furthermore, the mechanism allowed the agents to gather into sub-formations before converging into the final formation. Bobber-agents has been used for the generation of intermediate targets to reduce dependence on global information for short term decisions. Lower level obstacle avoidance and constraining of excessive orientation changes has been achieved with the use of virtual constellation-agents. Finally, the effectiveness of our approach has been verified with realistic simulations.

Appendix. Proof of Lemma 6

Integrating both sides of (29) from t_0 to t, we obtain

$$U_{tg}(t) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} U_{ob,ij}(t) \leq U_{tg}(t_0) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} U_{ob,ij}(t_0) \tag{A.1}$$



(a) Evolution of u_1. (b) Evolution of u_2. (c) Evolution of u_3.

(d) Evolution of u_4. (e) Evolution of u_5. (f) Evolution of u_6.

Fig. 14. Convergence of formation with 6 agents through an obstacle field.

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005

where

$$U_{tg}(t) = \frac{1}{2}\sum_{i=1}^{N} \|q_i(t) - q_{tg,i}\|^2$$

$$U_{ob,ij}(t) = \frac{U_{ij}(t)}{U_{tg,ij}^2} + \frac{1}{U_{ij}(t)}. \tag{A.2}$$

From the conditions in (31), $U_{ij}(t_0)$ and $U_{tg,ij}$ are strictly larger than some positive constants. The right hand side of (A.1) is bounded by some positive constant (the value of which depends on the initial conditions at t_0). Hence, the left hand side is also bounded, which in turn implies that $U_{ij}(t)$ must be strictly larger than some positive constant for all $t \geq t_0 \geq 0$. From (A.2), $\|q_i(t) - q_j(t) - d_{ob,ij}\|$ will, therefore, always be larger than some strictly positive constant, and there will be no collisions, i.e., the the distance between vehicle r_i and its neighbor j will always be greater than $d_{ob,ij}$. The boundedness of the left hand side of (A.1) also implies that of $\|q_i(t)\|$ for all $t \geq t_0 \geq 0$, and the solutions of the closed loop system in (30) exist.

By setting $\Omega_i = 0$, we obtain the root sets (critical points) of the system in (30), which are given by $q = q_{tg}$ (due to Property (c) of $U_{ob,ij}$) and $q = q_c$ (representing the remaining critical points), where $q = [q_1^{\mathrm{T}}, \ldots, q_N^{\mathrm{T}}]^{\mathrm{T}}$ and $q_{tg} = [q_{tg,1}^{\mathrm{T}}, \ldots, q_{tg,N}^{\mathrm{T}}]^{\mathrm{T}}$ and $q_c = [q_{c,1}^{\mathrm{T}}, \ldots, q_{c,N}^{\mathrm{T}}]^{\mathrm{T}}$.

The behavior of the equilibrium points is examined by considering the relative distances between agents. To convert the dynamics of each agent (given in (30)) to inter-agent dynamics, we define $q_{ij} = q_i - q_j$ and $q_{tg,ij} = q_{tg,i} - q_{tg,j}$ for all $i, j \in R$ for each i, and arranging i and j such that $i < j$. This yields the dynamics of q_{ij} as

$$\dot{q}_{ij} = -C\Omega_{ij} \tag{A.3}$$

where

$$\Omega_{ij} = \Omega_i - \Omega_j \tag{A.4}$$

$$= (q_{ij} - q_{tg,ij}) + \sum_{\ell \neq i}^{N} U'_{ob,i\ell} q_{i\ell} - \sum_{\ell \neq j}^{N} U'_{ob,j\ell} q_{i\ell}$$

$$= (q_{ij} - q_{tg,ij}) + 2U'_{ob,ij} q_{ij} + \sum_{\ell \neq i, \ell \neq j}^{N} \left(U'_{ob,i\ell} q_{i\ell} - U'_{ob,j\ell} q_{j\ell}\right). \tag{A.5}$$

The closed loop system in (A.3) may then be written as

$$\dot{\bar{q}} = -\bar{C}F(\bar{q}, \bar{q}_{tg}) \tag{A.6}$$

with

$$\bar{q} = [q_{12}^{\mathrm{T}}, q_{13}^{\mathrm{T}}, \ldots, q_{ij}^{\mathrm{T}}, \ldots, q_{N-1N}^{\mathrm{T}}]^{\mathrm{T}} \tag{A.7}$$

$$\bar{q}_{tg} = [q_{tg,12}^{\mathrm{T}}, q_{tg,13}^{\mathrm{T}}, \ldots, q_{tg,ij}^{\mathrm{T}}, \ldots, q_{tg,N-1N}^{\mathrm{T}}]^{\mathrm{T}} \tag{A.8}$$

$$\bar{q}_c = [q_{12c}^{\mathrm{T}}, q_{13c}^{\mathrm{T}}, \ldots, q_{ijc}^{\mathrm{T}}, \ldots, q_{(N-1)(N)c}^{\mathrm{T}}]^{\mathrm{T}} \tag{A.9}$$

$$\bar{C} = \mathrm{diag}(C, \ldots, C), \quad \text{(comprising } E \text{ number of } C \text{ along its diagonal)} \tag{A.10}$$

$$F(\bar{q}, \bar{q}_{tg}) = [\Omega_{12}^{\mathrm{T}}, \Omega_{13}^{\mathrm{T}}, \ldots, \Omega_{ij}^{\mathrm{T}}, \ldots, \Omega_{N-1N}^{\mathrm{T}}]^{\mathrm{T}} \tag{A.11}$$

where E is the total number of communication links that can exist between robots if global communications exist

Linearizing (A.6) at the critical points $\bar{q}_e$ results in

$$\frac{\mathrm{d}(\bar{q} - \bar{q}_e)}{\mathrm{d}t} = -\bar{C}\left.\frac{\partial F(\bar{q}, \bar{q}_{tg})}{\partial \bar{q}}\right|_{\bar{q}=\bar{q}_e} (\bar{q} - \bar{q}_e) \tag{A.12}$$

where the general gradient of $F(\bar{q}, \bar{q}_{tg})$ with respect to $\bar{q}$ is

$$\frac{\partial F(\bar{q}, \bar{q}_{tg})}{\partial \bar{q}} = \begin{bmatrix} \frac{\partial \Omega_{12}}{\partial q_{12}} & \frac{\partial \Omega_{12}}{\partial q_{13}} & \cdots & \cdots & \frac{\partial \Omega_{12}}{\partial q_{N-1N}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \Omega_{ij}}{\partial q_{12}} & \cdots & \frac{\partial \Omega_{ij}}{\partial q_{ij}} & \cdots & \frac{\partial \Omega_{ij}}{\partial q_{N-1N}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \Omega_{N-1N}}{\partial q_{12}} & \cdots & \cdots & \cdots & \frac{\partial \Omega_{N-1N}}{\partial q_{N-1N}} \end{bmatrix} \tag{A.13}$$

where $i, j \in R$, and

$$\frac{\partial \Omega_{ij}}{\partial q_{ij}} = \mathbf{I}_{n_w \times n_w} + 2U'_{ob,ij} + 2U''_{ob,ij} q_{ij} q_{ij}^{\mathrm{T}} \tag{A.14}$$

$$\frac{\partial \Omega_{ij}}{\partial q_{i_* j_*}} = \sigma U'_{ob,i_* j_*} + \sigma U''_{ob,i_* j_*} q_{i_* j_*} q_{i_* j_*}^{\mathrm{T}} \tag{A.15}$$

in which $\mathbf{I}_{(n_w \times n_w)}$ is an n_w-dimensional identity matrix, and $(i_*, j_*) \neq (i, j)$, $i_* \neq j_*$ and σ can either be 1 or -1 depending on the values of i, j, i_* and j_*.

To investigate the properties of the equilibrium $\bar{q}_e$, consider the following Lyapunov function candidate

$$V_{\bar{q}_e} = (\bar{q} - \bar{q}_e)^{\mathrm{T}}(\bar{q} - \bar{q}_e) \tag{A.16}$$

whose derivative along the solution of (A.16) satisfies

$$\dot{V}_{\bar{q}_e} = -2c\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(q_{ij} - q_{e,ij})^{\mathrm{T}}\left(\mathbf{I}_{n_w \times n_w} + N\mathbf{I}_{n_w \times n_w} U'_{ob,ij}\big|_{q_{ij}=q_{e,ij}} + N\,U''_{ob,ij}\big|_{q_{ij}=q_{e,ij}} q_{e,ij} q_{e,ij}^{\mathrm{T}}\right)(q_{ij} - q_{e,ij}). \tag{A.17}$$

Since $U'_{ob,ij}\big|_{q_{ij}=q_{tg,ij}} = 0$ and $U''_{ob,ij}\big|_{q_{ij}=q_{tg,ij}} \geq 0$, substituting $\bar{q}_e = \bar{q}_{tg}$ into (A.17) gives

$$\dot{V}_{\bar{q}_{tg}} = -2c\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(q_{ij} - q_{tg,ij})^{\mathrm{T}} \times \left(\mathbf{I}_{n_w \times n_w} + N\,U''_{ob,ij}\big|_{q_{ij}=q_{tg,ij}} q_{tg,ij} q_{tg,ij}^{\mathrm{T}}\right)(q_{ij} - q_{tg,ij})$$
$$\leq -2c\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(q_{ij} - q_{tg,ij})^{\mathrm{T}}(q_{ij} - q_{tg,ij}) \tag{A.18}$$

which clearly indicates that $\bar{q}_{tg}$ is asymptotically stable.

To show that the remaining critical points of the system i.e., $\bar{q}_c$ are unstable equilibrium points, consider the following.

$$\bar{q}_c^{\mathrm{T}} F(\bar{q}_c, \bar{q}_{tg}) = 0 \tag{A.19}$$

$$\Rightarrow \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(q_{c,ij}^{\mathrm{T}}(q_{c,ij} - q_{tg,ij}) + N\,U'_{ob,ij}\big|_{q_{ij}=q_{c,ij}} q_{c,ij}^{\mathrm{T}} q_{c,ij}\right) = 0$$

$$\Rightarrow \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(1 + N\,U'_{ob,ij}\big|_{q_{ij}=q_{c,ij}}\right) q_{c,ij}^{\mathrm{T}} q_{c,ij}$$

$$= \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_{c,ij}^{\mathrm{T}} q_{tg,ij}. \tag{A.20}$$

Consider $q_{c,ij}^{\mathrm{T}} q_{tg,ij}$. It gives the position of i relative to j, and for all intents and purposes, j can be seen as an obstacle situated at $q_{ij} = 0$. Furthermore, this point must lie between the points $q_{ij} = q_{tg,ij}$ and $q_{ij} = q_{c,ij}$, and such that these 3 points are colinear. Thus, the term $\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_{c,ij}^{\mathrm{T}} q_{tg,ij}$ is strictly negative and there exists at least one pair (i,j) denoted by (i^*,j^*) such that

$$1 + N\,U'_{ob,i^*j^*}\big|_{q_{i^*j^*}=q_{c,i^*j^*}} \le -b \tag{A.21}$$

where b is a strictly positive constant. Substituting $\bar{q}_e = \bar{q}_c$ into (A.17) gives

$$\begin{aligned}
\dot{V}_{\bar{q}_c} &= -2c\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(q_{ij}-q_{c,ij})^{\mathrm{T}}\Big(\mathbf{I}_{n_w\times n_w} + N\mathbf{I}_{n_w\times n_w}\,U'_{ob,ij}\big|_{q_{ij}=q_{c,ij}} \\
&\quad + N\,U''_{ob,ij}\big|_{q_{ij}=q_{c,ij}} q_{c,ij}q_{c,ij}^{\mathrm{T}}\Big)(q_{ij}-q_{c,ij}) \\
&\ge 2cb(q_{i^*j^*}-q_{c,i^*j^*})^{\mathrm{T}}(q_{i^*j^*}-q_{c,i^*j^*}) \\
&\quad - 2c\sum_{i=1,i\neq i^*}^{N-1}\;\sum_{j=i+1,j\neq j^*}^{N}(q_{ij}-q_{c,ij})^{\mathrm{T}}\Big(\mathbf{I}_{n_w\times n_w} \\
&\quad + N\mathbf{I}_{n_w\times n_w}\,U'_{ob,ij}\big|_{q_{ij}=q_{c,ij}}\Big)(q_{ij}-q_{c,ij}) \\
&\quad - 2c\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(q_{ij}-q_{c,ij})^{\mathrm{T}} \\
&\quad \times \Big(N\,U''_{ob,ij}\big|_{q_{ij}=q_{c,ij}} q_{c,ij}q_{c,ij}^{\mathrm{T}}\Big)(q_{ij}-q_{c,ij}).
\end{aligned} \tag{A.22}$$

Considering a subspace such that $q_{ij} = q_{c,ij}\ \forall(i,j) \in \{1,\ldots,N\}$, $(i,j) \neq (i^*,j^*)$ and $(q_{ij}-q_{c,ij})^{\mathrm{T}}q_{c,ij}q_{c,ij}^{\mathrm{T}}(q_{ij}-q_{c,ij}) = 0,\ \forall(i,j)\in\{1,\ldots,N\}$. In this subspace, the following holds

$$V_{\bar{q}_c} = (q_{i^*j^*}-q_{c,i^*j^*})^{\mathrm{T}}(q_{i^*j^*}-q_{c,i^*j^*}) \tag{A.23}$$

$$\dot{V}_{\bar{q}_c} \ge 2bc(q_{i^*j^*}-q_{c,i^*j^*})^{\mathrm{T}}(q_{i^*j^*}-q_{c,i^*j^*}) \tag{A.24}$$

which indicates that $\bar{q}_c$ is unstable. □

References

[1] L.E. Parker, ALLIANCE: An architecture for fault tolerant multirobot cooperation, IEEE Transactions on Robotics and Automation 14 (2) (1998) 220–240.

[2] B.P. Gerkey, M.J. Matarić, Sold!: Auction methods for multirobot coordination, IEEE Transactions on Robotics and Automation 18 (5) (2002) 758–768.

[3] C. Fua, S.S. Ge, COBOS: Cooperative back-Off adaptive scheme for multi-robot task allocation, IEEE Transactions on Robotics 21 (6) (2005) 1168–1178.

[4] S.S. Ge, Y.J. Cui, Dynamic motion planning for mobile robots using potential field method, Autonomous Robots 13 (2002) 207–222.

[5] A. Howard, H. Seraji, B. Werger, Global and regional path planners for integrated planning and navigation, Journal of Robotic Systems 22 (12) (2005) 767–778.

[6] J.P. Desai, A graph theoretic approach for modeling mobile robot team formations, Journal of Robotic Systems 19 (11) (2002) 511–525.

[7] W. Ren, R.W. Beard, A decentralized scheme for spacecraft formation flying via the virtual structure approach, AIAA Journal of Guidance, Control, and Dynamics 27 (1) (2004) 73–82.

[8] M. Tillerson, L. Breger, J.P. How, Distributed coordination and control of formation flying spacecraft, Proceedings of the American Control Conference 2 (2003) 1740–1745.

[9] M. Egerstedt, X. Hu, Formation constrained multi-agent control, IEEE Transactions on Robotics and Automation 17 (6) (2001) 947–951.

[10] Y. Hao, S.K. Agrawal, Planning and control of UGV formations in a dynamic environment: A practical framework with experiments, Robotics and Autonomous Systems 51 (2005) 101–110.

[11] T.D. Barfoot, C.M. Clark, Motion planning for formations of mobile robots, Robotics and Autonomous Systems 46 (2004) 65–78.

[12] W. Kang, N. Xi, Control and adaptation of multi vehicle formations, in: Proc. International Conference on Intelligent Robots and Systems, 1999, pp. 1155–1160.

[13] W. Kang, N. Xi, Y. Zhao, J. Tan, Y. Wang, Formation control of multiple autonomous vehicles: Theory and experimentation, IFAC 15th Triennial World Congress, 2002, pp. 1155–1160.

[14] A.K. Das, R. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer, C.J. Taylor, A vision based formation control framework, IEEE Transactions on Robotics and Automation 18 (2002) 813–825.

[15] J.P. Desai, J.P. Ostrowski, V. Kumar, Modeling and control of formations of nonholonomic mobile robots, IEEE Transactions on Robotics and Automation 17 (2001) 905–908.

[16] H. Yamaguchi, A distributed motion coordination strategy for multiple nonholonomic mobile robots in cooperative hunting operations, Robotics and Autonomous Systems 43 (4) (2003) 257–282.

[17] H. Yamaguchi, T. Arai, G. Beni, A distributed control scheme for multiple robotic vehicles to make group formations, Robotics and Autonomous Systems 36 (2001) 125–147.

[18] N.E. Leonard, E. Fiorelli, Virtual leaders, artificial potentials and coordinated control of groups, in: Proc. IEEE Conference on Decision and Control, vol. 3, December,2001, pp. 2968–2973.

[19] T. Balch, M. Hybinette, Social potentials for scalable multi robot formations, in: Proc. IEEE International Conference on Robotics and Automation, April 2000, pp. 3–80.

[20] T. Balch, R.C. Arkin, Behavior-based formation control for multirobot teams, IEEE Transactions on Robotics and Automation 14 (1998) 926–939.

[21] P. Song, V. Kumar, A potential field based approach to multi robot manipulation, in: Proc. IEEE International Conference on Robotics and Automation, May 2002, pp. 1217–1222.

[22] E. Rimon, D.E. Koditschek, Exact robot navigation using artificial potential functions, IEEE Transactions on Robotics and Automation 8 (5) (1992) 501–518.

[23] H. Tanner, A. Kumar, Towards decentralization of multi-robot navigation functions, in: Proc. IEEE Int. Conf. Robotics and Automation, April 2005, pp. 4143–4148.

[24] K.D. Do, Bounded controllers for formation stabilization of mobile agents with limited sensing ranges, IEEE Transactions on Automatic Control 52 (3) (2007) 569–576.

[25] P. Ögren, N.E. Leonard, Obstacle avoidance in formation, in: Proc. IEEE Int. Conf. Robotics and Automation, September 2003, pp. 2492–2497.

[26] M.A. Hsieh, V. Kumar, Pattern generation with multiple robots, in: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006, pp. 2442–2447.

[27] P. Ogren, E. Fiorelli, N.E. Leonard, Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment, IEEE Transactions on Automatic Control 49 (8) (2004) 1292–1302.

[28] X. Yun, G. Alptekin, O. Albayrak, Line and circle formation of distributed physical mobile robots, Journal of Robotic Systems 14 (2) (1997) 63–76.

[29] R.O. Saber, Flocking for multi-agent dynamic systems: Algorithms and theory, IEEE Transactions on Automatic Control 51 (3) (2006) 401–420.

[30] J. Lee, S. Venkatesh, M. Kumar, Formation of a geometric pattern with a mobile wireless sensor network, Journal of Robotic Systems 21 (10) (2004) 517–530.

[31] K.N. Krishnanand, D. Ghose, Formations of minimalist mobile robots using local-templates and spatially distributed interactions, Robotics and Autonomous Systems 53 (2005) 194–213.

[32] S.S. Ge, C. Fua, Queues and artificial potential trenches for multi-robot formations, IEEE Transactions on Robotics 21 (3) (2005) 646–656.

[33] J. Fredslund, M.J. Matarić, Robot Formations using only local sensing and control, in: IEEE International Symposium on Computational Intelligence in Robotics and Automation, July, August 2001, pp. 308–313.

[34] S. Carpin, L.E. Parker, Cooperative leader following in a distributed multi-robot system, in: Proceedings of International Conference on Robotics and Automation, May 2002, pp. 2994–3001.

[35] P. Kostelnik, M. Šamulka, M. Jánošík, Scalable multi-robot formations using local sensing and communication, in: Third International Workshop on Robot Motion and Control, November 2002, pp. 319–324.

[36] Y. Liu, K.M. Passino, M. Polycarpou, Stability analysis of one-dimensional asynchronous swarms, IEEE Transactions on Automatic Control 48 (10) (2003) 1848–1854.

[37] C.-H. Fua, S.S. Ge, K.D. Do, K.W. Lim, Multi-robot formations based on the queue-formation scheme with limited communications, IEEE Transactions on Robotics 23 (6) (2007) 1160–1169.

[38] H. Axelsson, A. Muhammad, M. Egerstedt, Autonomous formation switching for multiple mobile robots, in: IFAC Conf. Analysis and Design of Hybrid Systems, sant-Malo, Brittany, France, June 2003.

[39] R. Olfati-Saber, R.M. Murry, Distributed structrual stabilization and tracking for formations of dynamic multi agents, in: Proc. IFAC Conf. Decis. Hybrid Sys., Sant-Malo, Brittany, France, June 2003.

[40] A. Fujimori, M. Teramoto, P.N. Nikiforuk, M.M. Gupta, Cooperative collision avoidance between multiple mobile robots, Journal of Robotic Systems 17 (7) (2000) 347–363.

[41] S.S. Ge, X. Lai, A.A. Mamun, Boundary following and globally convergent path planning using instant goals, IEEE Transactions on Systems, Man and Cybernetics—Part B 35 (2) (2005) 240–254.

[42] T.J. Koo, S. Sastry, Design of a helicopter model based on approximate linearization, in: Proc. IEEE Conf. Decision and Control, December 1998, pp. 3635–3640.

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005

[43] S. Zelinski, T.J. Koo, S. Sastry, Hybrid system design for formations of autonomous vehicles, in: Proc. IEEE Conf. Decision and Control, vol. 1, December,2003, pp. 1–6.
[44] A. Pant, P. Seiler, T.J. Koo, K. Hedrick, Mesh stability of unmanned aerial vehicle clusters, in: Proc. American Control Conference, vol. 1, June,2001, pp. 62–68.

Shuzhi Sam Ge is a IEEE Fellow, IET Fellow, Ph.D., DIC, B.Sc., P.Eng, the Director, Institute of Intelligent Systems and Information Technology (ISIT), University of Electronic Science and Technology of China, and is founding Director of Social Robotics Lab of Interactive Digital Media Institute, and Professor of the Department of Electrical and Computer Engineering, the National University of Singapore. He has (co)-authored three books: Adaptive Neural NetworksControl of Robotic Manipulators (World Scientific, 1998), Stable Adaptive Neural NetworksControl (Kluwer, 2001) and Switched Linear Systems: Control and Design (Springer-Verlag, 2005), edited a book: Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications (Taylor and Francis, 2006), and over 300 international journal and conference articles.

He is the founding Editor-in-Chief, International Journal of Social Robotics, Springer. He has served/been serving as an Associate Editor for a number of flagship journals including IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, and Automatica. He also serves as a Book Editor of the Taylor & Francis Automation and Control Engineering Series. At IEEE Control Systems Society, he served/serves as Vice President for Technical Activities, 2009–2010, Member of Board of Governors of IEEE Control Systems Society, 2007–2009. His present research interests include social robotics, multimedia fusion, adaptive control, intelligent systems and artificial intelligence.

Cheng-Heng Fua received his B.Eng. degree in Electrical and Computer engineering from the National University of Singapore in 2003, and his Ph.D. from the NUS Graduate School for Integrative Sciences and Engineering in 2008. He is presently a postdoctoral fellow at Northwestern University, USA. His thesis work focused on embodied and situated multi-robotic systems, specifically aimed at robust operation (in teams comprising non-competitive members) in noisy environments e.g., search and rescue, and de-mining applications. More recently, his research involves multi-agent interaction in social settings, in particular, modeling the biological mechanisms that underly affect and personality, how these are manifested in different agent behaviors. He is the recipient of the A*STAR Graduate Scholarship, awarded by the Agency for Science, Technology, and Research (A*STAR), Singapore.

Khiang-Wee Lim received his graduate degree in electrical engineering from the University of Malaya, Kuala Lumpur, Malaysia, and his D.Phil. degree in engineering science from the University of Oxford, Oxford, UK, in 1978 and 1982, respectively.

He has been with the National University of Singapore, Singapore, and the University of New South Wales, Sydney, New South Wales, Australia. Presently, he is an Executive Director at the Institute of Materials Research and Engineering, Singapore. Concurrently, he is the Programme Director for Research at Singapore Science and Engineering Research Council, Agency for Science, Technology, and Research (A*STAR), Singapore, and the Programme Director for the Singapore International Graduate Award (SInGA). He is presently convening the Working Group for Standardization, Risk Assessment, and Safety of the Asian Nano Forum as well as leading the Singapore National International Standards Organization (ISO)/TC 229 Working Group that is part of the ISO effort to develop standards in nanotechnology.

Please cite this article in press as: S.S. Ge, et al., Agent formations in 3D spaces with communication limitations using an adaptive Q-structure, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.10.005