Connectivity maintenance with application to target search

Hiroaki Kata and Seiya Ueno

Graduate School of Engineering Science/Environment and Information Science, Yokohama National University, Yokohama, Japan

1. Introduction

Recent technological developments have made it possible to obtain high-performance unmanned vehicles at low cost. Therefore, system development using multiple unmanned vehicles is attracting attention. Various benefits can be obtained by using multiple vehicles, such as cost efficiency, robustness and scalability. A system using multiple unmanned vehicles is called a multi-agent system (MAS). In the MAS, the graph theory is often used [1–3]. In graph theory, each agent (unmanned vehicle) is replaced with a vertex, and a communication link for exchanging information between agents is replaced with an edge. Communication network by a group of unmanned vehicles is treated as a graph. A large number of vertices can make it difficult for a central computer to control all vertices. In that case, a decentralized control is more advantageous than a classical centralized control, according to [4]. In the decentralized control, each vertex processes using information about itself or its neighbours.

Vertex failures can cause network disconnection and disruption of information exchange. Algebraic connectivity, vertex connectivity, edge connectivity and others are used as an evaluation of network robustness, according to [5]. The algebraic connectivity is often used to the control law that maintains network connectivity. In [6], network connectivity is maintained by maximizing the algebraic connectivity. In [1], the generalized connectivity potential using the algebraic connectivity is introduced, and network connectivity is maintained by the gradient-like controller that does not fall below the lower limit of the generalized connectivity potential. Reference [1] is refined, and target search is performed in a decentralized system in [7]. However, as pointed out in [5], besides the fact that it is not intuitively clear which properties of the graph the algebraic connectivity expresses, as a measure for network robustness, it also has the problem that is not strictly increasing when an edge is added. This means that the use of the algebraic connectivity may not be suitable for some tasks. In [5], the authors give some evaluation of connectivity or network robustness. The natural connectivity is also one of the effective evaluations of robustness in [8]. In these evaluations, the vertex connectivity intuitively expresses fault tolerance in the sense that the connected network is maintained even if the unmanned vehicles fail and become unusable. In [9], a centralized control law that performs another task while maintaining the vertex connectivity is proposed. The algebraic connectivity, natural connectivity or vertex connectivity are the evaluation of network connectivity or robustness of the entire graph, and it is necessary to devise the usage in a decentralized system. We have introduced the partial vertex connectivity in a decentralized system, which gives the vertex connectivity for each vertex [10]. We showed that the robust \( k \)-connected network is realized by satisfying the reference value of the partial vertex connectivity of each vertex. This paper refines the past paper [10] and performs a target search in the presence of obstacles. Also, we add to our previous paper the idea of a critical edge that could reduce the vertex connectivity.
Figure 1. Definition of the robust network. (a) Non-robust and (b) robust.

The rest of this paper is organized as follows: in Section 2, we describe the robust $k$-connected network that is connected if less than $k$ unmanned vehicles fail. In Section 3, we introduce the definition and derivation of the partial vertex connectivity and application in a decentralized system. In Section 4, we explain the robust $k$-connected network control law and target search simulation, and Section 5 concludes the paper.

2. Problem statements

2.1. Dynamic network

Consider a network of $N$ unmanned vehicles connected by wireless communication and denoted by $(i, j)$ a communication link between vehicle $i$ and $j$. A network is given in the notion of a dynamic graph $G(t) = (V, E(t))$, where $V = \{1, \ldots, N\}$ denotes the vertex set indexed by the set of vehicles and $E(t) = \{(i, j) \mid i, j \in V\}$ denotes the time-varying edge set representing a communication link. It is assumed that a communication link can be enabled if two conditions are met. The first is that the Euclidean distance between two vehicles is less than the maximum communication range $R_{\text{comm}}$. The second is that there are no obstacles between two vehicles because it is difficult to obtain the relative position with the other vehicle by a camera or laser sensor. Any vertices $i$ and $j$ with a link $(i, j) \in E(t)$ are called neighbours or adjacent. Therefore, the neighbour set of vertex $i$ is defined by $N_i(t) = \{j \in V \mid (i, j) \in E(t)\}$.

In this study, the model of unmanned vehicles is represented by a first-order integral system. The state of vehicles with unique id $i = \{1, \ldots, N\}$ and the velocity input at discrete time $t$ are represented by $x_i(t)$ and $u_i(t)$, respectively, and the state at time $t + 1$ is given by the following:

$$x_i(t + 1) = x_i(t) + u_i(t). \quad (1)$$

2.2. Robust $k$-connected network

There is a case that the whole network is divided by the failure of vehicles as shown in Figure 1(a). The divided state is said to be disconnected, and information cannot be exchanged in the entire network. For that reason, the robust network that is resistant to vehicle failure is desirable, as shown in Figure 1(b).

The minimum number of vertices that need to be removed to disconnect the graph is called the vertex connectivity. The graph that is still connected after removing any $k - 1$ vertex is called the $k$-connected graph. Considering an unmanned vehicle network, the robust network that is not disconnected even if any $k - 1$ vehicles fail is called the robust $k$-connected network.

The purpose of this study is to realize the robust $k$-connected network in the decentralized method.

3. Partial vertex connectivity

In this section, we introduce the evaluation of graph connectivity and show the definition and derivation of the proposed connectivity evaluation. We use symbol $v$ to indicate a specific vertex, similar to $i$ or $j$, and the graph $G = (V, E)$ at a specific time $t$.

3.1. Definition of partial vertex connectivity

The vertex connectivity is used as an evaluation of the connectivity of the graph in [5]. For a subset $S \subset V$ of the vertex set $V$ of the graph $G$, $S$ is called the “vertex cut set” if removing them leaves the remaining subgraph $G - S$ disconnected. The minimum set of the vertex cut set $S$ is called the “vertex connectivity” $\kappa(G) = \kappa = |S|$. If there is no path from the vertex $s$ to the vertex $t$ in the subgraph $G - S$, $s \neq t$ for two different vertices $s, t \in V$, the subset $S_{s,t} \subseteq V - \{s, t\}$ is called the “$s, t$ cut set.” That is, a vertex set needs to be removed to separate vertices $s$ and $t$ and include them in another connected component. The minimum set of the $s, t$ cut set is called the “$s, t$ vertex connectivity” $\kappa_{s,t}^P(G) = \kappa_{s,t} = |S_{s,t}|$. The vertex connectivity is an evaluation for the graph, and $s, t$ vertex connectivity is an evaluation of robustness for two vertices $s, t$.

In this study, the vertex connectivity is defined for each vertex in the same way. For each vertex $v$, the $s, t$ cut set that contains vertex $v \in S_{s,t}$ with adjacent vertices $s, t$ such that $(s, v), (t, v) \in E$ but $(s, t) \notin E$ is called the “partial vertex $s, t$ cut set” $S_{s,t,v} \subseteq V - \{s, t\}$. The subset $S_{s,t,v}$ with the minimum set in the combination of vertices $s, t$ is called the “partial vertex cut set” $S_v \ni \{v\}$. The number of elements of subset $S_v$ is called the “partial vertex connectivity” $\kappa^P(v) = \kappa' = |S_v|$. The partial vertex connectivity indicates the partial robustness and importance of each vertex. In other words, it implies the effect on the whole system when the vertex fails. Figure 2 shows the image of the vertex connectivity, the $s, t$ vertex connectivity, and the partial vertex connectivity. The ellipse represents the subset of the vertices $V$, and the line represents the sub-edge set of $E$. (b) The $s, t$ cut set $S_{s,t}$ is (a) the vertex cut set $S$ that cuts two vertices $s, t$, and (c) the partial vertex cut set $S_v$ of the vertex $v$ is (b) the $s, t$ cut set $S_{s,t,v}$ that contains $v$ as an element.
3.2. Derivation of partial vertex connectivity

A method using the maximum flow problem is used to derive the vertex connectivity in [11]. The maximum flow problem is to find the maximum value of the total flow from the inflow point \( s \) to the outflow point \( t \) when a weighted graph is given. The \( s, t \) vertex connectivity \( \kappa_{st} \) can be obtained by the following procedure in global manner.

Algorithm: Calculation of the \( s, t \) vertex connectivity

Input: Graph \( G = (V, E) \), non-adjacent vertex pair \( s, t \)

Output: \( \kappa_{st} \)

1. Replace each undirected edge with directed arcs opposite to each other and call it a digraph \( D \).
2. For all vertices \( u \) in \( V \) of the digraph \( D \) except \( s, t \), replace \( u \) with two new vertices \( u_1, u_2 \) and add a directed arc from \( u_1 \) to \( u_2 \). Connect all the arcs that were coming to \( u \) in \( D \) to \( u_1 \) and all the arcs that were going out of \( u \) to \( u_2 \), and let the remaining digraph be \( H \).
3. Assuming that the weight of all edges is 1, find the maximum flow \( f \) from \( s \) to \( t \) of the digraph \( H \).
4. Let maximum flow \( f \) be \( \kappa_{st} \).

As can be seen from Figure 2, the vertex connectivity \( \kappa \) is the minimum value of the \( s, t \) vertex connectivity of any two vertices \( s, t \).

\[
\kappa = \min \{ \kappa_{st} \mid s, t \in V \}. \tag{2}
\]

Similarly, the partial vertex connectivity \( \kappa^\nu \) of the vertex \( \nu \) is given by the following equation:

\[
\kappa^\nu = \min \{ \kappa_{st} \mid s, t \in N_{\nu}, (s, t) \notin E \}. \tag{3}
\]

Here, \( N_{\nu} \subset V \) is a neighbour set of the vertex \( \nu \). The condition \( (s, t) \notin E \) in Equation (3) is the condition that \( \nu \) is included in the partial vertex cut set due to the nature of the maximum flow problem. In some cases, there are no vertices \( s, t \) such that \( s, t \in N_{\nu} \land (s, t) \notin E \), in that case, \( \kappa^\nu = N - 1 \) for the vertex \( \nu \) in the calculation. \( N \) is the number of vertices. As can be seen from

![Figure 2. Image of connectivities.](image)

3.3. Computing partial vertex connectivity in a decentralized system

In a decentralized system, each vertex communicates only with neighbours and can use the information that the neighbours have. The \( l \)-hop neighbour graph \( G_{lj} = (V_{lj}, E_{lj}) \) of vertex \( i \) is constructed by exchanging \( l-1 \) hop neighbour information \((l-1 \text{ hop NI})\) from the neighbour of vertex \( i \). \( l-1 \) hop NI stores the vertex numbers in \( l-1 \) hop and their adjacent relations, that is the edge. By constructing the \( l \)-hop neighbour graph \( G_{lj} \) from \( l-1 \) hop NI of the adjacent vertices and executing algorithm (calculation of the \( s, t \) vertex connectivity) with the graph \( G_{lj} \) as an input, the estimated value of the \( s, t \) vertex connectivity \( \tilde{\kappa}_{st} \) and the partial vertex connectivity \( \tilde{\kappa}^\nu = \min \{ \tilde{\kappa}_{st} \mid s, t \in N_{\nu}, (s, t) \notin E \} \) is calculated. The estimated value \( \tilde{\kappa}^\nu \) may be smaller than the actual value \( \kappa^\nu \) (clear from \( G_{lj} \subset G \)) but has no negative effect when considering robustness. The larger the value of \( l \), the more information is required and the processing load increases. If \( l \) is a small value, the amount of information is small and a connectivity cannot be obtained properly. In Figure 3, (a) is the \( G_{3j} \) network with \( \tilde{\kappa}^\nu = 3 \), (b) is the \( G_{2j} \) network with \( \tilde{\kappa}^\nu = 2 \). The dashed line is the edge where \( G \) is not included in \( G_{3j} \) or \( G_{2j} \). As can be intuitively understood from Figure 3, \( I = k \) is appropriate for realizing a robust \( k \)-connected network.

![Figure 3. Comparison of \( l \)-hop neighbour graphs with \( l = 2 \) and \( l = 3 \). (a) \( G_{3j} \) with \( l = 3 \) and (b) \( G_{2j} \) with \( l = 2 \).](image)

Equations (2) and (3), the minimum value of the partial vertex connectivity at each vertex is the value of the vertex connectivity:

\[
\kappa = \min_{\nu \in V} \{ \kappa^\nu \}. \tag{4}
\]

3.4. Numerical example

Figure 4 shows a numerical example of connectivities. The number at the lower right of each vertex is the value of the partial vertex connectivity \( \kappa^\nu \) and the number at the upper right is the estimated value of the partial vertex connectivity \( \tilde{\kappa}^\nu \) in \( l = 2 \), and has the same value. Vertices with \( \kappa^\nu = 9 \) do not match the definition of the partial vertex connectivity and set \( \kappa^\nu = N - 1 \). Even if the vertex with \( \kappa^\nu = 9 \) is deleted, the partial vertex connectivity of neighbouring vertices is not changed, so
the partial vertex connectivity following equations:

\[ \text{Gl} \]

where \( \kappa \) is the reference value of the vertex connectivity obtained from the \( l \) hop neighbour graph \( G_{ij} \): 

\[ C_i = \max \left( 0, \frac{\kappa_v - \tilde{\kappa}_v}{\kappa_v} \right). \]  

(5)

Here, \( \kappa_v \) is the reference value of the vertex connectivity, that is, the value \( k \) of \( k \)-connectivity \( (\kappa_v = k) \) is an integer), and \( \tilde{\kappa}_v \) is the estimated partial vertex connectivity \( \tilde{\kappa}_v \) of vehicle \( i \). Each vehicle determines the velocity input according to the adjacent vehicle cost \( C_i \). The velocity input of each vehicle is given by the following equations:

\[ u_i = \arg \min_{u \in U} f_i(x_i + u_i), \]  

(6)

\[ f_i(x_i) = f_i^0(x_i) = -k_v \sum_{j \in N_i} C_j x_{ij}, \]  

(7)

where \( k_v > 0 \) is the coefficient, \( x_{ij} \) is the relative position from vehicle \( j \) to \( i \) and \( N_i \) is the neighbour set of vehicle \( i \). And \( U \) represents the control constraint:

\[ U = \{ u_i | u_i^2 \leq v_{max}^2 \}, \]  

(8)

with maximum velocity \( v_{max} \). The state of \( C_i = 0 \) means that the partial vertex connectivity is equal to or higher than the reference value, and there is no need for an adjacent vehicle to approach. The state of \( C_i > 0 \) means that the partial vertex connectivity is smaller than the reference value, and the nearby vehicles need to approach and improve the partial vertex connectivity. If the partial vertex connectivity of all vehicles satisfies the reference value, all vehicles will stop. This is intended to avoid excessive concentration of all vehicles. As many vehicles converge in one region, the connectivity evaluation such as the vertex connectivity or the algebraic connectivity improves, resulting in a robust network. But given another task such as the target search, over-concentration can be a disadvantage. Many control laws so far have been overly concentrated. This control law avoids excessive concentration and can advantageously perform another task.

4.2. Triangulation input

In addition to the velocity input based on the partial vertex connectivity described in the previous subsection, the input for collision avoidance and advances in robustness, named “triangulation input,” is added. Vehicles can collide with other vehicles, obstacles or wall. Vehicles form communication links or edges if and only if there is no obstacle between them, and they can be seen directly, that is, within the communication range. Vehicles move in the direction to prevent cutting off the edge by obstacles. In addition, it is thought that the minimum thickness of the network shape, that is, the network width, is not too small, which contributes to the advances in robustness. This is because if the network width is extremely small, multiple edges will be cut by obstacles, and the network will be disconnected. Figure 5 is an example that vehicles moved from the initial position in Figure 4 using the input of Equations (6) and (7). The network width near the centre is small, and the network may be disconnected due to the cutting of multiple edges.

The method of constructing an equilateral triangle for collision avoidance and advances in robustness is proposed. The triangular relationship between any three vehicles is a relationship in which two vehicles adjacent to one vehicle are adjacent. The obstacle avoidance relationship is a relationship formed by the two adjacent vehicles and the obstacle if the distance between the edge and the obstacle is smaller than threshold distance \( d_{obs} \). An image of the triangular relationship and the obstacle avoidance relationship is shown in Figure 6. Each vehicle in the triangular relationship or the obstacle avoidance relationship moves...
so that its relationship approaches an equilateral triangle shape. That is, if the vehicle $i$ forms the triangular relationship or the obstacle avoidance relationship with vehicles or obstacle, the following triangulation potential is given:

$$f_{tri}^i = k_d f_i^d + k_f f_i^f + k_s f_i^s,$$  \hspace{1cm} (9)

where $k_d, k_f, k_s$ are coefficients related to the following. The first term $f_i^d$ is a potential that brings the distance $d_{i1}, d_{i2}$ from partners $j1, j2$ closer to the reference distance $d_{ref}$. The second term $f_i^f$ is a potential that brings the angle $\theta_i$ formed with partners $j1, j2$ closer to the reference value $\frac{\pi}{3}$. The third term $f_i^s$ is a potential that brings the area $S$ of the triangle closer to the reference value $S_{opt}$. That is, it is given by the following equations:

$$f_i^d = (d_{ref} - d_{i1})^2 + (d_{ref} - d_{i2})^2,$$

$$f_i^f = \left(\frac{\pi}{3} - \theta_i\right)^2,$$

$$f_i^s = (S_{opt} - S)^2 = \left(\frac{1}{2} d_{ref}^2 \sin \frac{\pi}{3} - S\right)^2.$$  \hspace{1cm} (10)

4.3. Critical edge

When vehicles performed another task such as target search, the edge can be cut due to excessive communication distance or obstacle. For each vertex, the edge that the vertex connectivity of the $l$-hop neighbour graph becomes smaller than the reference value when the edge is deleted is called the critical edge. Figure 7 shows the $l$-hop neighbour graph $G_{li}$ and the critical edge $(i, j1)$ of vertex $i$ when the reference value of connectivity is 2 ($l = r^l = 2$). If the edge $(i, j1)$ is removed, the vertex connectivity $\kappa(G_{li})$ of the graph $G_{li}$ decreases ($2 \rightarrow 1$) and becomes smaller than the reference value. For vertex $j1$ not only the edge $(j1, i)$ but also the edge $(j1, j2)$ is the critical edge. The minimization function in Equation (7) is partially modified so that the critical edge is not cut:

$$f_i^c(x_i) = -k_v \sum_{j \in N_i} f_{ij}^c,$$

$$f_{ij}^c = \begin{cases} C_j x_{ij} & \text{if } \bar{e}_c, \\ \frac{a}{d^2_c - \|x_{ij}\|^2} + b \frac{x_{ij}}{\|x_{ij}\|} & \text{if } e_c, \end{cases} \hspace{1cm} (11)$$

where $d_c$ is the maximum communication distance. $\|x_{ij}\|$ is the Euclidean distance between vehicle $j$ and $i$. In Equation (11), the potential is given for each edge in neighbour set $N_i$ by two kinds of conditions. (I) $\bar{e}_c$: If it is not the critical edge or it is the critical edge but the edge length is within the safe distance $d_s$, the potential is the same as Equation (7). (II) $e_c$: If the edge is the critical edge and the edge length is greater than safe distance, the potential increment increases as the edge length increases (dashed line in Figure 8). Also, $a$ and $b$ in Equation (11) are constants such that the potential is differentiable and continuous around $d_s$.

$$a = \frac{C_j (d_{ij}^2 - d_s^2)^2}{2d_s}, \hspace{1cm} b = \frac{C_j (3d_{ij}^2 - d_s^2)}{2d_s}. \hspace{1cm} (12)$$

With potential (11), critical edges will not be deleted and connectivity will be maintained at the reference value.
by the following equation:

\[ u_i = \arg \min_{u \in U} f_i(x_i + u), \]

\[ f_i(x_i) = f_i^0(x_i) + f_i^{tri}(x_i). \]  

(13)

All vehicles move to improve the connectivity and the robustness. It may be less than \( k \), that is, the reference value of vertex connectivity, to perform another task such as the target search. In such a case, \( k' = k + 1 \), named \( k' \)-strategy is used for the control law, and \( k \)-connectivity is maintained by using one extra \( k \). The cost of each vertex \( C_i \) according to \( k' \)-strategy is given by the following equation:

\[ C_i' = \max \left( 0, \frac{\kappa_i' + 1 - \tilde{\kappa}_i'}{\kappa_i' + 1} \right), \]  

(14)

where \( \tilde{\kappa}_i' \) is calculated using the \( k' \) hop neighbourhood graph \( G_{k',i} = G_{k+1,i} \). Equation (13) are transformed into the following equations:

\[ u_i = \arg \min_{u \in U} f_i(x_i + u), \]

\[ f_i(x_i) = f_i^0(x_i) + f_i^{tri}(x_i), \]

\[ f_i^{tri}(x_i) = k_{df}d_i^d + k_{df}^1 + k_{df}^2; \]

\[ f_i^0(x_i) = -k_v \sum_{j \in N_i} f_{ij}^0, \]

\[
\begin{align*}
  f_{ij}^0 &= C_i x_{ij}, \\
  f_{ij}^{tri} &= \left( \frac{a}{d_i^2 - \|x_{ij}\|^2} + b \right) \frac{x_{ij}}{\|x_{ij}\|},
\end{align*}
\]

if \( \tilde{e}_c \),

(15)

4.5. Target search simulation

In this subsection, target search in an environment with obstacles is performed. The reference value of the vertex connectivity is set to \( \kappa_i' = 2 \). The target position is determined randomly within the square of \( x = 0.0 \sim 1.0, y = 0.0 \sim 1.0 \). The vehicle closest to the target point becomes the search vehicle and approaches the target through the shortest pass. Who goes to the target point is determined by vehicle consensus, such as elections. When the search vehicle is close enough to the target point (distance to the target is less than \( 1.0 \times 10^{-5} \)), the search ends and a new target is generated randomly. The other vehicles move to maintain the robust \( k \)-connected network. \( k' \)-strategy is applied with \( k' = 2 + 1 = 3 \). The cost of the search vehicle in charge of the target search is set to \( C_i' = (3 - 2)/3 \), so that the search vehicle does not move away alone from the swarm. In addition, if the critical edge of the search vehicle is likely to cut when the search vehicle moves towards the target point, the search vehicle will stop moving towards the target point and maintain the network like any other vehicle. That condition is one of the edge is the critical edge and the length of the critical edge is larger than \( d_e - 2v_{max} \) or the distance from the obstacle to the critical edge is less than \( v_{max} \). Figure 9 shows that starting from a random initial position (a), in the initial 30 steps vehicles construct the robust \( k \)-connected network without target search (b). In the initial position, there are vertices that the partial vertex connectivity is 1 (number 1 at the upper right of the vertex), and at \( \text{step} = 30 \), the minimum value of partial vertex connectivity is 2. Target search is performed after 31 steps in Figure 10. There are always three targets (asterisks), and a new target appears when the search vehicle (square vertex) reaches the target point. Vehicles execute the target search one after another while maintaining the overall connectivity with \( k \geq 2 \).

Search vehicles move to the target point unless the critical edge is cut. Also, other vehicles move to maintain the network. In Figure 11, dashed lines represent the critical edges. Vehicles with the critical edge move so that the critical edge is not cut and the vertex connectivity does not fall below the reference value. Also, a search vehicle with the critical edge moves toward the search point unless the critical edge is likely to be cut.

Lastly, in order to investigate the effect of the triangulation potential \( f_i^{tri} \) on the connectivity maintenance,
a simulation was performed under the condition that there is no connectivity maintenance potential $f_f^v$, that is, $k_e = 0$ in Equation (15). Even in the case of only the triangle potential, a positive effect on the connectivity maintenance was confirmed as in the case with connectivity maintenance cost. However, when only a quadrangle with a neighboring vehicle was formed instead of the triangular relationship, the force for maintaining connectivity did not work and the vertex connectivity fell below the reference value. Therefore, it can be said that not only the triangulation cost but also the connectivity maintenance cost is required to maintain the connectivity.

5. Conclusion

In this paper, we have introduced the partial vertex connectivity indicating the connectivity of each vertex in the decentralized system. In addition, a robust $k$-connected control law is proposed in which input is determined according to the partial vertex connectivity of adjacent vehicles, and triangulation input is added for collision avoidance and advances in robustness. Each vehicle is moved so as not to cut the critical edge that could reduce the vertex connectivity.

In the target search simulation, it has been confirmed that a part of the vehicle searched for continuously generated targets, and other vehicles moved so as to maintain connectivity while avoiding the obstacle. In the future, we will conduct a target search in a more practical environment, for example, inside a building or urban area.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Hiroaki Kata received the BE degree from the National Defense Academy in 2013, and ME degree from Yokohama National University in 2019. He joined Japan Ground Self-Defence Force in 2013. Since 2019 he is a PhD student at Yokohama National University, where he focuses on unmanned vehicle networks.

Seiya Ueno received the BE, ME, and PhD degrees in aeronautical engineering from the University of Tokyo in 1980, 1982, and 1985, respectively. He joined the Division of Space Development, Toshiba Corporation, in 1985. In 1988, he moved to Yokohama National University, where he has been a professor since 2002. His major research fields are applications of optimal control in aerospace engineering.

References

[1] Robuffo Giordano P, Franchi A, Secchi C, et al. A passivity-based decentralized strategy for generalized connectivity maintenance. Int J Robot Res. 2013;32(3):299–323.
[2] Kim Y, Mesbahi M. On maximizing the second smallest eigenvalue of a state-depend graph Laplacian. IEEE Trans Automat Control. 2006;51(1):116–120.
[3] Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Trans Automat Control. 2006;51(3):401–420.
[4] Pantelimon G, Tape K, Carriveau R, et al. Survey of multi-agent communication strategies for information exchange and mission control of drone deployments. J Intell Robot Syst. 2019;95(3–4):779–788.
[5] Ellens W, Kooij RE. Graph measures and network robustness; 2013. Preprint, arXiv:1311.5064.
[6] Anderson T, Chang C, Martinez S. Maximizing algebraic connectivity of constrained graphs in adversarial environments. In: 2018 European Control Conference (ECC), Limassol, Cyprus; 2018. p. 125–130.
[7] Nestmeyer T, Giordano PR, Büllthoff HH, et al. Decentralized simultaneous multi-target exploration using a connected network of multiple robots. Auton Robots. 2017;41(4):989–1011.
[8] Deng Z, Xu J, Song Q, et al. Robustness of multi-agent formation based on natural connectivity. Appl Math Comput. 2020;366(4):124636.

[9] Luo W, Sycara K. Minimum $k$-connectivity maintenance for robust multi-robot systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China; 2019. p. 7370–7377.

[10] Kata H, Ueno S. Decentralized robust $k$-connected network with unmanned vehicles using vertex connectivity. In: Proceedings of the SICE Annual Conference 2020, Chiang Mai, Thailand; 2020. p. 268–281.

[11] Esfahanian A. Connectivity algorithms. Cambridge: Cambridge University Press; 2013. p. 268–281.