IAC–21–C1.9

Multi-Agent Attitude Task Allocation and Control in a Swarm of Magnetically Controlled CubeSats

Ahmed Mahfouz
University of Luxembourg, Luxembourg, Ahmed.Mahfouz@uni.lu

Salman Ali Thepdawala
Skolkovo Institute of Science and Technology, Russian Federation, SalmanAli.Thepdawala@skoltech.ru

Nourhan Abdelrahman
Skolkovo Institute of Science and Technology, Russian Federation, Nourhan.Abdelrahman@skoltech.ru

Dmitry Pritykin
Moscow Institute of Physics and Technology, Russian Federation, pritykin.da@mipt.ru

Abstract
The collective behavior of four magnetically controlled CubeSats in LEO is considered. The spacecraft fly as a group and when commanded, use intersatellite link to negotiate and determine the most favorable arrangement to provide maximum sky coverage around a specified reference direction by onboard sensors. The most favorable arrangement is attained through attitude task allocation consensus and arrived at by a fully magnetic controller implemented onboard of each spacecraft.
keywords: swarm, CubeSat, magnetic attitude control, task allocation, multi-agent optimization

1. Introduction

The study has been conducted as a part of the Skoltech University project to deploy a swarm of 3U CubeSats in LEO. The mission has to be a technology demonstration for inter-satellite link (ISL) capabilities, hence a number of experiments involving ISL usage have been considered for the mission. The primary mission experiment is collective gamma-ray bursts detection, which requires the satellites' coordinated attitude control [1]. Prior to this work, we designed the attitude determination and control subsystem with magnetic actuation [2] and studied collective measurements processing algorithms to enhance the onboard magnetic field model [3]. The experiment this study is about is to showcase the ISL usage for negotiation-based consensus algorithm implemented in the CubeSats to make them act as a multi-agent system, which exhibits swarm behaviour by optimal attitude task allocation on receipt of a command from the mission control center.

The proposed experiment is in line with the recent developments aimed at demonstrating Distributed Spacecraft Autonomy (DSA) [4]. The need for intelligent distributed spacecraft infrastructure has long been emphasized [5], but it is only now that NASA is planning a CubeSat swarm mission to implement the distributed autonomy by using the on-board GPS receiver to perform in-situ, swarm-level reconfiguration in response to observed features in the Topside Ionosphere [4]. The promise of DSA is greater flexibility and lower cost of mission planning and scheduling systems [6]. The approaches considered for usage in autonomous architectures are multi-agent paradigm [7] or bio-inspired self-organization paradigm (i.e ant colony optimization [8] or stigmetry-based approach [9]). The experiment considered in this study goes along with the multi-agent optimization approach for distributed attitude task allocation in a swarm of four CubeSats actuated by magnetic attitude control system. It should be noted that the experiment does not imply full autonomy as it is triggered by a mission control center command, however the attitude task allocation problem is to be solved by the swarm autonomously.

Prior research shows that PD-based magnetic three-axis attitude control is very sensitive to the choice of the controller gains [10, 11]. Moreover, the magnetic controller gains are dependent on the required attitude [12]. Gain tuning usually relies on periodicity of the required trajectories in simplified models and employs different techniques to optimize the degree of stability for the required motion. The first part of this study shows how to obtain the controller gains as functions defined for all possible re-

quired attitudes with respect to the orbital frame. This is done in two steps. The first step is linearization of the spacecraft rotational dynamics in the vicinity of the required attitude regime and subsequent numerical optimization (carried out in terms of Floquet theory). The procedure allows obtaining optimal gains for all possible required attitudes, however, depending on the spacecraft inertia tensor and estimated environmental effects there can be zones of unfeasible attitudes, to which the controller cannot converge.

The second part of this study shows the collective attitude control scenario. The task of the swarm is to ensure maximum sky coverage around the principal direction uplinked to the spacecraft by the mission control center. This is measured by the minimum angular distance between the principal direction and any point in the sky that does not fall into the field of view of any of the spacecraft's instruments. Thus the swarm spacecraft act as four agents to solve a multi-objective optimization problem maximizing the coverage and negotiating to allocate their individual attitudes so as to maximize the resulting degrees of stability given the attitude feasibility constraint. The implemented optimization algorithm employs a negotiation-based approach and can be scaled to a greater number of spacecraft.

This paper is divided into seven different sections. Section 2 states the orbital parameters of the swarm and describes the mathematical model of their rotational dynamics. The Floquet analysis and its utility in the gain tuning procedure for magnetic control are then explained in Section 3. The following section (4) formulates the maximum sky coverage problem for a swarm of CubeSats. Then, Section 5, under consensus algorithm, details the 'EXTRA' algorithm employed in this work for resource allocation in a swarm of magnetically controlled CubeSats. Section 6 investigates and evaluates the implementation of the algorithm for the given test cases. Lastly, Section 7 reiterates the significance of the solution and its scalability and adaptiveness to other swarm networks.

2. Equations of Motion

We consider a group of four 3U CubeSats deployed into a circular orbit at an altitude of 400 km and an inclination of 52 degrees. Each spacecraft is assumed to have a mass of 3 kg with dimensions 100x100x340 mm in X, Y and Z directions respectively, and the inertia tensor of $\mathrm{diag}\,[0.031, 0.031, 0.005]\,\mathrm{kg}\cdot\mathrm{m}^2$.

Following are the reference frames used in the paper:

- Orbital reference frame $\mathcal{F}^o$: its origin is at the center of mass of the satellite, the z-axis points away from the center of the Earth, the y-axis is directed along the cross product of the satellite's center of mass position and velocity vectors, and the x-axis completes the frame according to the right-hand rule.
- Body-fixed reference frame $\mathcal{F}^b$: its origin is at the satellite's center of mass. Its three axes coincide with the three principal axes of inertia of the satellite.

All vector transformations between reference frames are described by unit quaternions.

The orbital motion of each CubeSat is assumed to correspond to a purely Keplerian circular orbit (four CubeSats fly in a train formation). Thus the position of ith spacecraft in its orbit is determined by its linearly changing argument of latitude u_i . The characteristic time of the considered experiments does not exceed a few orbital periods, hence Earth oblateness does not have any significant effects on the qualitative results.

Let us denote by $\boldsymbol{\Omega}^b$ the satellite's absolute angular velocity and by $\boldsymbol{\omega}^b$ its angular velocity relative to the orbital frame $\mathcal{F}^o$. The two angular velocities are then related as follows:

$$\boldsymbol{\Omega}^b = \boldsymbol{\omega}^b + \widetilde{\mathfrak{q}}^{ob} \circ \boldsymbol{\omega}_0^o \circ \mathfrak{q}^{ob} \tag{1}$$

where $\boldsymbol{\omega}_0^o = \begin{bmatrix} 0 & \omega_0 & 0 \end{bmatrix}^T$ is the angular velocity of $\mathcal{F}^o$ frame, ω_0 is the mean motion of the satellite in the orbit. Designating the unit quaternion that transforms from the $\mathcal{F}^b$ frame to the $\mathcal{F}^o$ frame by $\mathfrak{q}^{ob} = \begin{bmatrix} q_0^{ob} & \mathbf{q}^{ob} \end{bmatrix}$, we can describe the kinematics of the spacecraft as

$$\dot{\mathfrak{q}}^{ob} = \frac{1}{2}\mathfrak{q}^{ob} \circ \boldsymbol{\omega}^b. \tag{2}$$

The satellite is considered to be a rigid body, and all the environmental disturbances except for the gravity-gradient torque are ignored. This yields the satellite's dynamical equations of motion as follows:

$$\mathbb{J}^b\dot{\boldsymbol{\omega}}^b = -\boldsymbol{\omega}^b \times \mathbb{J}^b\boldsymbol{\omega}^b + \mathbf{M}_{ctrl}^b + \mathbf{M}_{gg}^b, \tag{3}$$

where $\mathbb{J}^b$ denotes the inertia tensor of the spacecraft, $\mathbf{M}_{ctrl}^b$ is the control torque provided by the actuators, and $\mathbf{M}_{gg}^b$ is the gravity gradient torque. The gravity gradient torque is given by

$$\mathbf{M}_{gg}^b = 3\omega_0^2\mathbf{e}_3^b \times \mathbb{J}^b\mathbf{e}_3^b. \tag{4}$$

The control torque is given by the following equation:

$$\mathbf{M}^b_{ctrl} = \mathbf{m}^b \times \mathbf{B}^b, \tag{5}$$

where $\mathbf{B}^b$ is the geomagnetic induction vector and $\mathbf{m}^b$ is the control dipole moment generated by the magnetorquers.

The control dipole moment is derived from Lyapunov based PD-controller [12]:

$$\mathbf{m}^b = -4q_{e,0}K_p\mathbf{B}^b \times \mathbf{q}_e - K_d\mathbf{B}^b \times \boldsymbol{\omega}^b, \tag{6}$$

where $K_p, K_d > 0$ are the controller gains and $\mathfrak{q}_e = \begin{bmatrix} q_{e,0} & \mathbf{q}_e \end{bmatrix}^T$ is the error quaternion defined in the following equation:

$$\mathfrak{q}_e = \tilde{\mathfrak{q}}_d \circ \mathfrak{q}^{ob}, \tag{7}$$

where $\mathfrak{q}_d$ is the desired quaternion (desired $\mathfrak{q}^{ob}$ at steady state).

Let us note that the model of rotational dynamics given by equations (2)-(3) is intentionally simplified so as not to include all disturbances and uncertainties. The full model where the control loop design was properly investigated and disturbance identification and rejection algorithms derived, tested and simulated has been presented in [2]. The results obtained in [2] allow us carrying out the present study with the simplified model described in this section assuming that all major disturbances are dealt with by the control system.

3. Controller Gain Tuning

The gain tuning routine is based on the Floquet theory and was first proposed in [12]. The idea is to reformulate the equations of motion as a system of equations with periodic coefficients and for the given required steady state of the system find a pair of gains that ensures the greatest degree of stability for the particular trajectory the controller is set to converge to. The detailed description of the routine can be found in[12], where it was first described, and in [13], where it is reformulated in terms of quaternions instead of Euler angles and applied to all possible spacecraft end-attitudes. Let us note here that it takes rewriting the equations of motion in the non-inertial orbital frame and using the direct dipole model for geomagnetic field representation to reduce the equations of motion to a system with periodic coefficients. The equations are then linearized in the vicinity of the required attitude, the monodromy matrix is constructed, and a pair of gains is obtained as a solution of the following optimization problem:

$$\begin{aligned} f(K_p, K_d) &= \max_i \left[\Re\left[\log\left[\lambda_i(K_p, K_d)\right]\right]\right], \\ (K_p, K_d) &= arg \min_{K_p, K_d > 0} f\left(K_p, K_d\right), \end{aligned} \tag{8}$$

where λ_i are the monodromy matrix eigenvalues and the Floquet theory states that having the real parts of the logarithm of the eigenvalues negative corresponds to the stability o the obtained periodic trajectory. Provided that all real parts are negative, the distance between the greatest of them and zero determines the degree of stability of the solution in consideration. Optimization problem (8) yields a pair of coefficients (K_p, K_d) providing the maximum degree of stability for the trajectory corresponding to the required attitude.

The choice of gains depends on the orbit of the satellite as well as the attitude around which it needs to be stabilized ($\mathfrak{q}_d$). It is due to the physical limitation of the magnetic control system that the satellite might not be able to stabilize around a certain orientation. It has been concluded that for a fixed orbit, there is no set of gains that can stabilize the spacecraft around a number of desired attitudes (the unstable attitudes pool). However, there is a stable attitudes pool for which the gains need to be optimized in order to attain optimal performance. Clearly, within the stable attitudes pool and after finding the optimal gains, the satellite's performance around some desired orientations is better than that around some others.

As an example, the contour plot of the objective function (8) for a desired attitude $\mathfrak{q}_d = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$ (gravity-gradient orientation for this case) is shown in Figure 1. It is found after the optimization that this attitude belongs to the stable attitudes pool. The red dot indicates the solutions of the optimization problem (8). The performance of the magnetic control system using the optimized gains for $\mathfrak{q}_d = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$ is presented in Figure 2.

The same gain-tuning routine is further used to generate the controller gains for a representative set of the spacecraft possible orientations. This data is used in the next section to provide an optimized attitude set points for a swarm of CubeSats seeking maximum sky coverage. Although the attitudes are parameterized by unit quaternions, it has been concluded that it would be more intuitive to express the desired attitude of the spacecraft as a set of Euler angles with sequence of intrinsic rotations of 123, $\boldsymbol{\alpha}_{d,123} = \begin{bmatrix}\alpha_x & \alpha_y & \alpha_z\end{bmatrix}^T$. The desired Euler angles

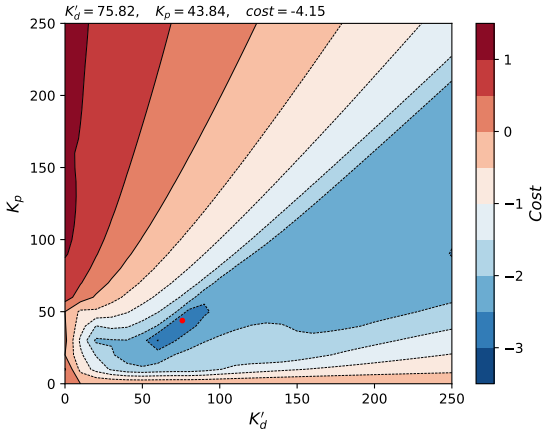


Fig. 1: Contour plot of the cost function for $\mathfrak{q}_\mathfrak{d} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$

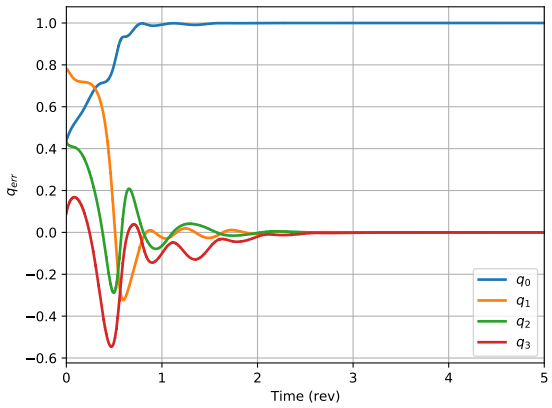


Fig. 2: Simulation example using optimized control gains

can simply be transformed to $\mathfrak{q}_d$ in order to proceed with solving the minimization problem.

It is important to note that the inertia tensor defined in Section **2** suggests that the spacecraft is symmetrical around the Z-axis of the $\mathcal{F}^b$ frame, which allows us to use only two Euler angles to fully parametrize any desired attitude, as the satellite is invariant to the rotation with respect to the Z-axis. This suggests $\boldsymbol{\alpha}_{d,123} = \begin{bmatrix} \alpha_x & \alpha_y & \alpha_z \end{bmatrix}^T = \begin{bmatrix} \alpha_x & \alpha_y & 0 \end{bmatrix}^T$. All possible orientations can now be thought of as points on a unit sphere with α_x being the elevation and α_y being the azimuth.

A differential evolution global optimization algorithm has been used to run the optimization (8) for 1700 data points on the sphere, and the optimization results are shown in Figures 3 and 4. Figure 3 shows a scatter plot of only the orientations that belong to the stable attitudes pool (with negative cost function) on a sphere, each point of which corresponds to particular values of α_x and α_y. Figure 4, however, is a projection of all the points on the sphere on the plane. The color bar in both figures represents the objective function of the optimization problem defined in (8).

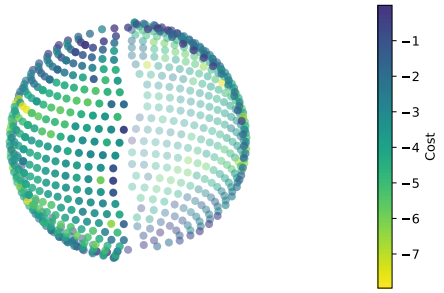


Fig. 3: Scatter plot showing area of successful convergence on a sphere

4. Maximum Sky Coverage Problem

Let us now consider a problem that requires communication and collective behavior of the four swarm CubeSats. The satellites have the same parameters as described in Section 2. In this case we might consider a mission whose aim is to carry out observations whether of astronomical events using gamma-ray detectors or space debris tracking with optical

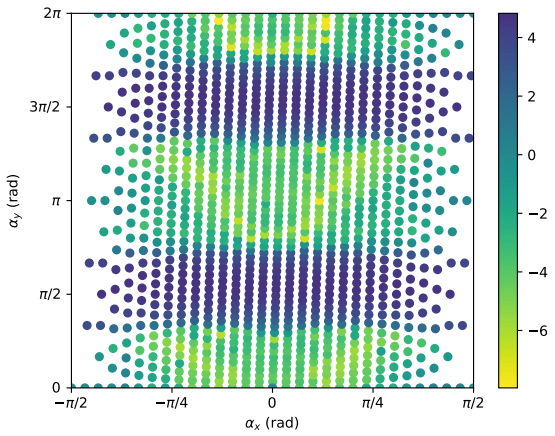


Fig. 4: A grid plot showing area of successful convergence in spherical coordinates.

instrumentation (such as developed in [14]). When a certain event of interest is expected the swarm is commanded from the mission control center to provide the maximum sky coverage around a principal reference direction $\mathbf{e}_O$ (Fig. 5). For simplicity we will assume that $\mathbf{e}_O$ is represented in the orbital frame of any of the spacecraft. The CubeSats are assumed to be operating at small distances from one another (the distances are 500-1000 m so as the swarm orbital configuration is compatible with other planned experiments described in [1]) and the events of interests are sufficiently distant to neglect the difference between the positions of the swarm spacecraft and their orbital frames.

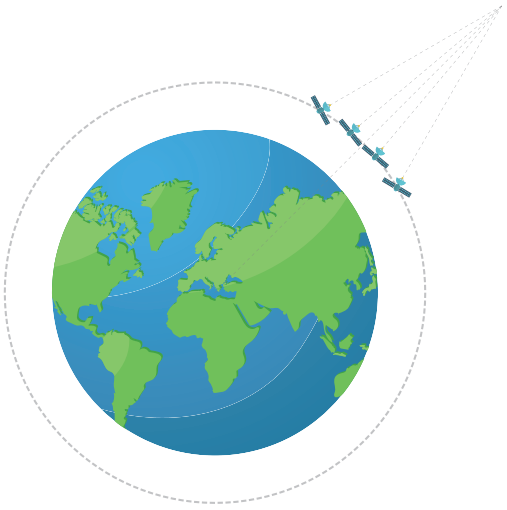


Fig. 5: sky coverage problem illustration

Let us denote by θ the field of view of the employed sensors (assuming conical field of view and that the sensors are mounted to have the field of view cone axis $\mathbf{e}_{f,i}$ coinciding with the z-axis of the ith spacecraft body frame). Also let us define the coverage maximization metric in terms of the closest direction to $\mathbf{e}_O$ which does not fall within field field of view of any of the sensors. If this direction is represented by a unit vector $\mathbf{e}_C$ our goal is to maximize the dot product $\mathbf{e}_O \cdot \mathbf{e}_C$. From symmetry the arrangements of the spacecraft providing maximum coverage is such that the lines defined by $\mathbf{e}_{f,i}$ intersect any plane orthogonal to $\mathbf{e}_O$ in points that belong to a circle (see Fig. 6). The points are distributed on this circle symmetrically, however the maximum coverage problem solution is invariant to rotation of this circle (which changes the required orientation of the four spacecraft). This invariance lets us add another objective to the optimization problem, which is expressed in minimization of the maximum cost function (as given by (8)) value associated with the four required attitudes. Thus, we can find the position of the four points on the circle which ensure the best stability of the required configuration within the maximum coverage solution. Furthermore, due to symmetry this solution can be expressed through one angle γ (as shown in Fig. 6). The case when the optimization yields a solution that requires one or more of the spacecraft to take an attitude that belongs to the unstable pool is not considered here.

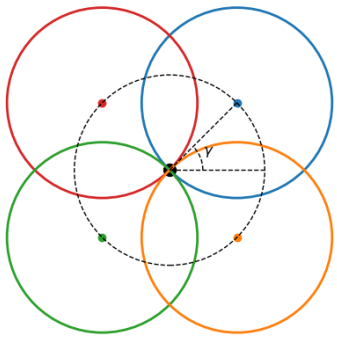


Fig. 6: Consensus problem illustration

Thus, the problem left for the swarm to solve is given the principal direction $\mathbf{e}_O$ for the sky coverage activate the swarm communication mode and by negotiation find the optimal (in terms of sky coverage) attitude arrangement of all CubeSats in the swarm that maximizes the minimum degree of stabil-

ity among the four spacecraft:

$$\min_i f_i \rightarrow \max_\gamma . \tag{9}$$

5. Consensus Algorithm

A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is defined by its vertex set $\mathcal{V}$ as well as its edge set $\mathcal{E}$. $\mathcal{V}$ is a set which contains the identifiers (i.e. names or numbers) of each vertex in the graph, while $\mathcal{E}$ is a set that contains all the pairs of vertices that are adjacent to each other (i.e. the pairs that can send or receive information from each other). A Time-invariant undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a graph in which the structure of the graph is time independent (i.e. $\mathcal{V}$ and $\mathcal{E}$ are constant). Moreover, all the pairs in $\mathcal{E}$ can exchanging information back and forth through undirected communication links.

A Laplacian matrix $\mathcal{L}$ of a graph $\mathcal{G}$ is a matrix representation of $\mathcal{G}$. In other words, the Laplacian matrix holds all the information of the graph, from the degrees of its vertices to the adjacency information of these vertices. The entries of the Laplacian matrix are defined as follows:

$$\mathcal{L}_{i,j} = \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \\ 0 & \text{Otherwise} \end{cases} \tag{10}$$

Taking the network in Fig. 7b -which will later be introduced- as an example, it is obvious that the degree of vertices 1 and 4 is 1 as both of them have only one adjacent vertex, while the degree of agents 2 and 3 is 2. The degree of each vertex is reflected on the diagonal elements of $\mathcal{L}$, while the adjacency is reflected on the off-diagonal elements. (e.g. vertices 2 and 3 are adjacent, which yields $\mathcal{L}_{2,3} = \mathcal{L}_{3,2} = -1$, while agents 1 and 4 are not adjacent, which yields $\mathcal{L}_{1,4} = \mathcal{L}_{4,1} = 0$). The Laplacian matrix of this graph can be written as,

$$\mathcal{L} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{11}$$

Let us now consider the problem of distributed consensus for a multi-agent network. In the literature, one can find a handful of algorithms that can be used to solve the following consensus optimization problem:

$$\begin{aligned} & f_{coll} = \max\left(f_i(\mathbf{x})\right), \\ & \min_{\mathbf{x}} \quad f_{coll}, \end{aligned} \tag{12}$$

where $\mathbf{x}$ is the vector of decision variables and f_i is the indvidual cost function held privately by the i^{th} agent. The first algorithm that comes to mind is the Decentralised Gradient Descent (DGD) [15], however, more efficient algorithms like the decentralized exact first-order algorithm (abbreviated as EXTRA) [15] can as well be used. While DGD needs to use diminishing descent rate in order to arrive to an exact minimum point, EXTRA can use a large fixed descent rate and still consensually arrive to an exact minimizer of the collective cost function. A complete explanation of the algorithm as well as the method of construction of the mixing matrices can be found in [15].

In the context of the problem of this paper, four agents are cooperating to reach consensus on the decision variable γ while each agent knows its individual objective function $f_i(\gamma)$. The individual cost functions for each of the identical spacecraft are defined as in (8).

The consensus flow of the DGD algorithm is presented here. Although this research is adopting EXTRA as its distributed optimizer, it is much easier to present the consensus flow using DGD, especially as EXTRA works based on the same principles as DGD. The DGD algorithm for a single decision variable is implemented on-board of each of the n agents as in Algorithm 1.

It can be seen form (13) that as the update is carried out, the optimization variable is going down-hill off of the collective cost function due to the term $-\alpha \nabla f_{coll}(x_i(k))$. It can also be seen that consensus is achieved through the two terms $-\tau \mathcal{L}_{i,i} x_i(k)$ and $-\tau \sum_{\{i,j\} \in \mathcal{E}} \mathcal{L}_{i,j} x_j(k)$, as agent i is compromising its current state while trying to reach an agreement with its neighboring agents.

It is important to note that the speed at which the multi-agent system arrives to a consensus is heavily dependent on the latency in the network links as agent i collects the set of current and updated optimization variables $\{x_j(k)\}$ and $\{x_j(k+1)\}$ from all agents each iteration. It is also important to note that although not all the elements of $\{x_j(k)\}$ explicitly appear in (13), it is substantially important to collect the decision variable from all agents to calculate the gradient $\nabla f_{coll}(x_i(k))$.

It has to be acknowledged that throughout Algorithm 1, the grid in Fig. 4 was interpolated whenever an individual objective function f_i needed to be calculated.

Algorithm 1: Distributed gradient descent

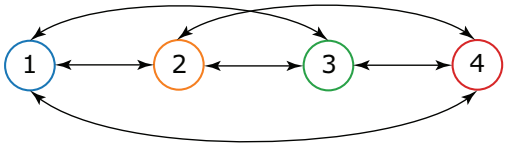
foreach $i \in \mathcal{V}$ **do**
  Choose a value for the descent rate α;
  Choose a value for the consensus rate τ;
  Define a convergence threshold ϵ;
  Initiate consensus violation $CV \leftarrow \epsilon + 1$;
  while $CV \geq \epsilon$ **do**
    Collect the set of current decision variables $\{x_j(k)\}\ \forall\, j \in \mathcal{V}$;
    Compute the gradient of the collective cost function $\nabla f_{coll}(x_i(k))$;
    $$x_i(k+1) = x_i(k) - \tau \mathcal{L}_{i,i} x_i(k) - \tau \sum_{\{i,j\} \in \mathcal{E}} \mathcal{L}_{i,j} x_j(k) - \alpha \nabla f_{coll}(x_i(k)) \quad (13)$$
    ;
    Collect the updated decision variables $\{x_j(k+1)\}\ \forall\, j \in \mathcal{V}$;
    $CV = x_i(k+1) - \frac{1}{n}\sum_{j=1}^{n} x_j(k+1)$;
  end
end

6. Results and Discussion

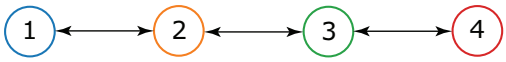
Two different time-invariant undirected communication graphs (see Fig. 7) were tested and the results of implementing the adopted algorithms for each case are presented in this section.

As seen by Fig. 7, Graph-I stipulates that all the agents lie in the neighbourhood of each other and that any agent can exchange information with any other agent in the network, while Graph-II indicates that only the neighbouring agents in the trailing formation can exchange information, which is more likely to happen in a real-life scenario.

The consensus algorithm (Sec. 5) was run for both graphs for a nominal orientation of $\begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ where each satellite is equipped with a sensor whose field of view is 10 degrees. The agents could reach consensus on the value of γ after a number of iterations in both cases. The evolution of the system accordance for



(a) Graph-I



(b) Graph-II

Fig. 7: Graph scenarios

Graph-I and Graph-II is presented in Fig. 8 and Fig. 9 respectively.

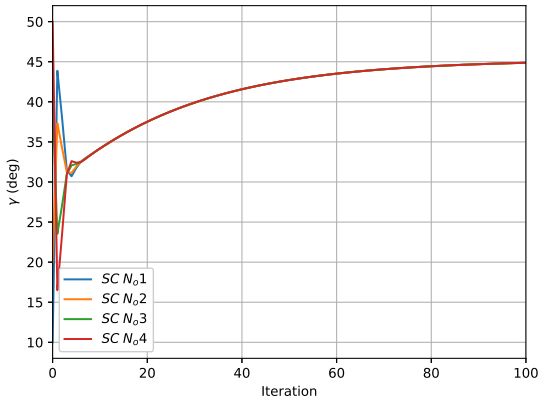


Fig. 8: Consensus evolution for Graph-I

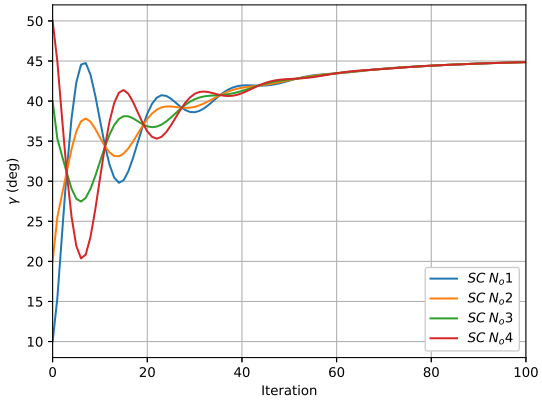


Fig. 9: Consensus evolution for Graph-II

Although the agents of Graph-I arrive to consensus much faster than those of Graph-II, the two graphs approach the optimal point after the same number of iterations (see Figs. 8 and 9). It is worth mentioning that the mean consensus violation of the agents of Graph-I at the last iteration is $5.55E-17$ degrees, compared to that of Graph-II which is $1.86E-6$ degrees.

It is important to note that the collective cost function is generally not convex, and as long as a gradient-based optimization algorithm is adopted, it is of a great importance to run the optimization multiple times with different initial conditions to make sure the optimizer does not get stuck in a local minimum. The two aforementioned optimization problems for the two graphs appeared to have arrived to the exact same solution, and the collective cost function together with the optimized solution (the red point in the plot) for both problems are shown in Fig. 10

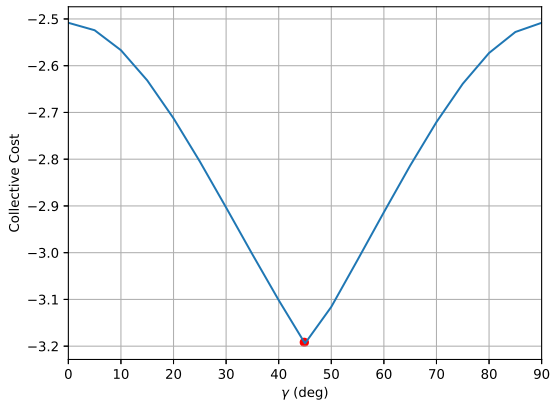


Fig. 10: Collective cost function minimization for Graph-I and Graph-II

It is clear from Figs. 8 and 9 that the two graphs eventually arrived to the same solution (γ). The value of the optimal γ is then used to define the four desired attitudes of the four spacecraft. Furthermore, the desired attitudes are in turn used to interpolate the controller gains dataset to obtain the suitable gains set for each spacecraft. The controlled dynamics of the spacecraft are simulated and the simulation results are presented in Fig. 11.

7. Conclusion

The paper outlined an experiment for Skoltech's inter-satellite communication technology demonstration mission. The mission comprises four CubeSats

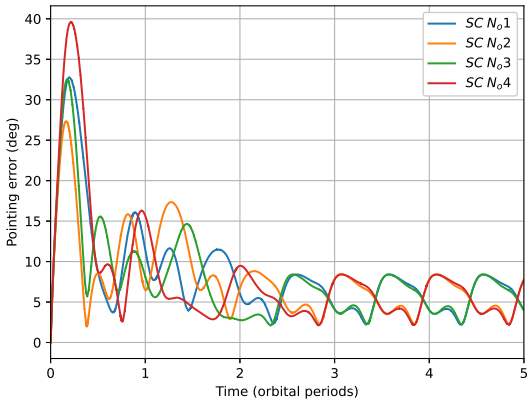


Fig. 11: Simulation of the swarm using optimized γ

in a swarm, each equipped with purely magnetic attitude actuators. This experiment, which is inline with NASA's Distributed Spacecraft Autonomy (DSA) project, is carried out mainly to investigate the ability of the multi-agent system to autonomously coordinate its actions through distributed decision making, especially when the spacecraft are unaware of each others properties.

The problem of maximum sky coverage around a reference direction for the swarm is considered, and the problem is reduced to one in which the swarm has to reach consensus on a single decision variable that defines the optimal geometry of the formation from the point of view of the degree of attitude stability of the satellite which is rendered most vulnerable in the resulting attitude task allocation arrangement.

To solve the distributed consensus optimization problem, the decentralized exact first-order algorithm (EXTRA) which is a gradient-based distributed minimization technique is implemented for two different time-invariant undirected communication graphs. The algorithm proved to be able to arrive to an exact minimum for both graphs after a number of iterations. The Distributed Gradient Descent (DGD) algorithm was also investigated for the same communication graphs, and it was found equally competent to EXTRA in the context of our problem.

The proposed experiment is especially interesting in the context of magnetically controlled satellites, as magnetic control is generally slow which gives the swarm the needed relaxation in order to arrive to consensus.

However sketchy the presented discussion of the experiment is, coupled with the results of prior re-

search of the magnetically controlled swarm, it gives us full ground to assume that the multi-agent decision making can be transferred to a more complicated or realistic environment.

Acknowledgment

This work is supported by the Luxembourg National Research Fund (FNR) – AuFoSat project, BRIDGES/19/MS/14302465.

References

[1] A. Mahfouz, D. Pritykin, et al. Coordinated attitude determination and control in a swarm of cubesats. *Proceedings of the International Astronautical Congress, IAC*, 2019-October, 2019.

[2] Anastasiia Annenkova, Nourhan Abdelrahman, et al. CubeSat Magnetic Atlas and in-Orbit Compensation of Residual Magnetic Dipole. *Proceedings of the International Astronautical Congress, The Cyberspace Edition*, 2020-October, 2020.

[3] Anton Afanasev, Mikhail Shavin, et al. Tetrahedral Satellite Formation: Geomagnetic Measurements Exchange and Interpolation. *Advances in Space Research*, 67(10):3294–3307, 2021.

[4] Daniel Cellucci, Nick B. Cramer, and Jeremy D. Frank. Distributed spacecraft autonomy. In *AIAA 2020-4232. ASCEND 2020. November 2020.*

[5] C.A. Raymond, J.O. Bristow, and M.R. Schoeberl. Needs for an intelligent distributed spacecraft infrastructure. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 1, pages 371–374 vol.1, 2002.

[6] Carles Araguz, Elisenda Bou-Balust, and Eduard Alarcón. Applying autonomy to distributed satellite systems: Trends, challenges, and future prospects. *Systems Engineering*, 21(5):401–416, 2018.

[7] Chris HomesParker. Exploiting structure and agent-centric rewards to promote coordination in large multiagent systems. 2014.

[8] Claudio Iacopino, Phil Palmer, Nicola Policella, and Alessandro Donati. *Planning the GENSO Ground Station Network via an Ant Colony-based approach.*

[9] Howard Tripp and Phil Palmer. Stigmergy based behavioural coordination for satellite clusters. *Acta Astronautica*, 66(7):1052–1071, April 2010.

[10] DS Ivanov, M Yu Ovchinnikov, et al. Advanced numerical study of the three-axis magnetic attitude control and determination with uncertainties. *Acta Astronautica*, 132:103–110, 2017.

[11] Mohammed AA Desouky and Ossama Abdelkhalik. Stability and steady state error analysis for satellite magnetic attitude regulation. *AAS/AIAA Astrodynamics Specialist Conference Proceedings*, 2020.

[12] M.Yu. Ovchinnikov, D.S. Roldugin, et al. Choosing control parameters for three axis magnetic stabilization in orbital frame. *Acta Astronautica*, 116:74–77, 2015. cited By 17.

[13] Ahmed Mahfouz, Nourhan Abdelrahman, et al. Attitude Task Allocation and Control in a Swarm of Magnetically Controlled CubeSats. *Proceedings of the 8th International Conference of Control Systems, and Robotics (CDSR'21), Virtual Conference – May 23-25*, 2021.

[14] S. Biktimirov, N. Solodovnikova, et al. A CubeSat-based Space System to Monitor Space Debris Population in LEO. *Proceedings of the International Astronautical Congress, IAC*, 2021-October, 2021.

[15] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.