10th ANKARA INTERNATIONAL AEROSPACE CONFERENCE **AIAC-2019-193**
18-20 September 2019 - METU, Ankara TURKEY

DECENTRALIZED FORMATION CONTROL FOR A LARGE SCALE SWARM OF QUADROTOR HELICOPTERS

Sinan Oğuz*
STM Savunma Teknolojileri Mühendislik A.S.
Ankara, Turkey

Mehmet Altan Toksöz†
Middile East Technical University
Ankara, Turkey

Mehmet Önder Efe‡
Hacettepe University
Ankara, Turkey

ABSTRACT

In this letter, we propose a decentralized hybrid control mechanism for a large scale swarm of quadrotor helicopters. First, the mechanism includes a method for finding an appropriate assignment of quadrotors to the locations in the desired formation with collision-free trajectories. Second, it utilizes a formation control approach in which a large number of quadrotors achieves the desired formation within a short period of time without any oscillation. Last, it uses a potential function which provides inter-agent collision avoidance. Our simulation and real world experiments showed that a large number of quadrotors in the random initial locations achieves the desired formation successfully without any collisions.

INTRODUCTION

UAVs in particular those with VTOL ability, such as quadrotor helicopters have the potential to perform many civilian and defense tasks. Their wide availability together with new results in artificial intelligence, control theory, etc. enables widespread applications of UAVs and groups (i.e., swarms) of UAVs which are moving and performing tasks in a coordinated fashion.

The development and accessibility of swarms of simple and cheap UAVs may allow us to provide alternative solutions to real-world problems in areas such as search and rescue, surveillance, reconnaissance, defense, etc. However, tackling such problems requires the development of robust and scalable coordination algorithms. Some approaches use inspiration from nature with the objective to develop robust, scalable and flexible coordination algorithms. As is seen in the nature, it is expected that swarms of engineered agents such as UAVs with coordination and local interactions among the agents and between the environment will be able to perform complex tasks which are beyond the capability of a single agent. Achieving complex tasks may require utilization of more than one sensor, actuator or payload. Therefore, use of the quadrotors as a swarm may allow an agent to utilize the

*Research and Development Engineer, Email: sinan.oguz@stm.com.tr
†PhD, Independent Researcher, Email: matoksoz@gmail.com
‡Prof. Dr., Email: onderefe@hacettepe.edu.tr

resources/capabilities of other agents through communication and/or coordination and provide it with extra capabilities. This has the potential to significantly increase not only the performance of the individual agent but also the overall group performance [A. Kushleyev, 2013; A. A. Bandala, 2014; B. Yu, 2013; G. Gioioso, 2014; M. Saska, 2014].

A fundamental coordination problem in swarms is the formation control problem since usually to achieve a common task the agents have to arrange in a given predefined shape. The algorithms proposed in the literature to control formation of swarm have been classified into various different approaches. For example, in [K.-K. Oh, 2015] algorithms are categorized into position-based, displacement-based, and distance-based strategies.

This paper presents a decentralized hybrid control mechanism to achieve desired formations for a large group of quadrotor helicopters while avoiding collisions. The mechanism includes a method for generation of collision-free assignment of trajectories to goal locations in the desired formation. We also present the stability analysis of the proposed formation rule using consensus based tools and Lyapunov methods. In order to avoid collisions between the agents, we extend the controller by adding a distance-based potential rule. Using the quadrotor dynamics, we convert the generated formation level forces into desired positions assuming that each swarm member has a position controller. We test the proposed control structure using both simulation and real world experiments.

METHOD

In this part, first, we present the details of the proposed decentralized hybrid control structure (Fig.1) step-by-step. Then, we provide the overall controller. Since the controller output cannot be directly applied to the swarm members, we also propose a conversion rule which generates the desired positions.

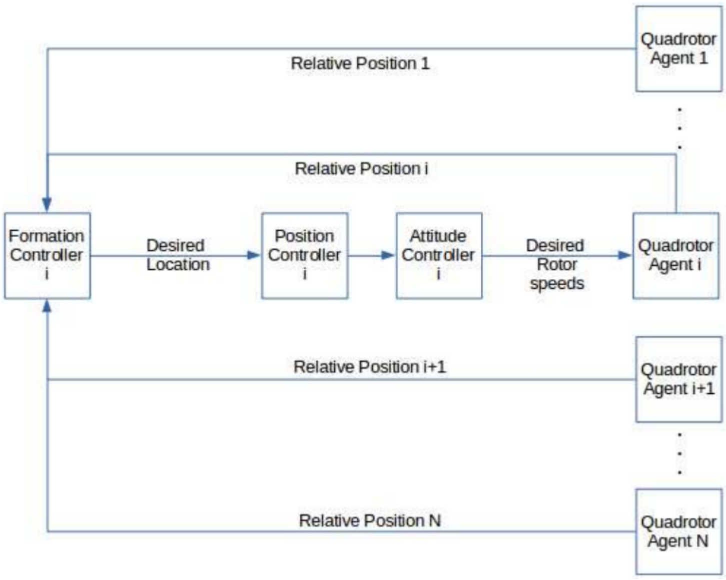


Figure 1: Decentralized formation controller.

Trajectory Assignment

Assume that there are N quadrotors randomly initialized at the locations specified by $p_i \in \mathbb{R}^n$ where i is the identity of a quadrotor. Our goal is to transfer the large number of quadrotors from the initial locations to final locations in the desired formation successfully without any collisions and oscillations within a short time interval. For this, let us define the final location for the i^{th} quadrotor in the formation as g_i. To minimize the sum of distances traveled by all quadrotors, we need to assign an appropriate goal location for each quadrotor. This is called the combinatorial transportation assignment problem which can be stated by the following optimization expression

$$\min_{\phi} \sum_{j=1}^{N} \sum_{i=1}^{N} \phi_{i,j} ||p_i - g_j||_2 \tag{1}$$

where $\phi \in \mathbb{R}^{N \times M}$ is the *assignment matrix* with N quadrotors navigating from initial locations to M desired goal locations in an n-dimensional Euclidean space

$$\phi_{i,j} = \left\{ \begin{array}{ll} 1, & if\ quadrotor\ i\ is\ assigned\ to\ goal\ j \\ 0, & otherwise \end{array} \right\} \tag{2}$$

We require either all goals to be assigned or all quadrotors to be assigned which results in:

$$\begin{array}{l} \phi^T \phi = I_M, \quad if\ M \leq N \\ \phi \phi^T = I_N, \quad if\ N \leq M \end{array} \tag{3}$$

The optimal assignment ϕ using the minimum sum of distance optimization in equation 1 results in non-intersecting paths. This is the well-known linear transportation assignment problem and therefore the solution of the problem could be found using an optimal assignment solving algorithm such as the Hungarian Algorithm [M.Turpin, 2014].

Formation Control

After assigning the optimum goal locations, we may define the formation control rule for each quadrotor. In the proposed framework, we assume that each quadrotor in the system sense the positions of the other swarm members with respect to a reference coordinate system using a simple communication system. In a quadrotor, while the inner control loops stabilizes the platform, the outer ones are used to track a reference path. Because the time constants of these two type of loops are quite different, we may decouple them into low and high level dynamics [X.Dong, 2015].

We assume that each quadrotor can be modeled in n-dimensional space with the following single-integrator dynamics

$$\dot{p_i} = u_i, i = 1, 2, ..., N \tag{4}$$

where $p_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^n$ are the position and the overall control input of i^{th} quadrotor, respectively. Let $\delta_{ji} \in \mathbb{R}^n$ denote the desired position vector between i^{th} and j^{th} members. Note that for consistency and feasibility of the formation $\delta_{ij} = -\delta_{ji}$ must be satisfied for all pairs (i, j), $j \neq i$ Note also that we have $\delta_{ii} = 0$ for all i. Then, the formation control portion of u_i for the i_{th} quadrotor could be designed such that for $t \geq 0$ the quadrotor positions $p_i(t)$ satisfy

$$\lim_{t \to \infty} p_j(t) - p_i(t) = \delta_{ji} \tag{5}$$

for all i and j, $1 \leq i$, $j \leq N$, $i \neq j$.

The above problem can be solved by a relative position-based controller of the form

$$u_i^f = k_p \sum_{j=1}^{N} (p_j - p_i - \delta_{ji}) \tag{6}$$

where $k_p > 0$ formation control gain. The equivalence can be shown by the equation

$$k_p \sum_{j=1}^{N} (p_j - p_i - \delta_{ji}) = k_p \sum_{j=1}^{N} (p_j - p_i) + b_i \tag{7}$$

where the input bias is $b_i = -k_p \sum_{j=1}^{N} \delta_{ji}$. As pointed out in [J.A.Fax, 2004] the dynamics along each axis are decoupled which indicates that one dimensional analysis is valid for the $n-$ dimentional case. The same approach is utilized in this section. In other words, for the rest of this section we perform the analysis assuming the $n = 1$, although the results hold for a general finite $n \geq 1$. In order to use the properties of the Laplacian matrix, we express the dynamics of the swarm system with the control input in a more compact matrix form as

$$\dot{\mathbf{p}} = -k_p L\mathbf{p} + \mathbf{b} \tag{8}$$

where $\mathbf{p} = [p_1, p_2, ..., p_N]^T$, $\mathbf{b} = [b_1, b_2, ..., b_N]^T$ and $L \in \mathbb{R}^{N\times N}$ is the Laplacian matrix of the interaction topology graph of the swarm. It is defined as $L = D - A$, where $D \in \mathbb{R}^{N\times N}$ is a diagonal matrix whose i^{th} diagonal entry is the in degree of the i^{th} vertex and $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix of the topology graph. Because the network structure of the control rule in equation 6 is fully connected (i.e., it is a complete graph), the entries of L are given by

$$l_{ij} = \left\{ \begin{array}{ll} -1, & i \neq j \\ (N-1), & i = j \end{array} \right\} \tag{9}$$

For these values of its elements the Laplacian matrix has only one zero eigenvalue and the rest of its eigenvalues are positive and the same (i.e., it is positive semi-definite). Note also that the vector $\mathbf{b}$ is an eigenvector of L with the eigenvalue of N and we have

$$L\mathbf{b} = N\mathbf{b} \tag{10}$$

Let us define the Lyapunov-like function for the swarm system as

$$V(\mathbf{p}) = \frac{1}{2}(\mathbf{p} - \frac{1}{k_p N}\mathbf{b})^T(\mathbf{p} - \frac{1}{k_p N}\mathbf{b}) \tag{11}$$

Taking the derivative of V along $\mathbf{p}$, one obtains

$$\begin{aligned} \dot{V}(\mathbf{p}) &= (\mathbf{p} - \frac{1}{k_p N}\mathbf{b})^T(-k_p L\mathbf{p} + \mathbf{b}) \\ &= -k_p(\mathbf{p} - \frac{1}{k_p N}\mathbf{b})^T L(\mathbf{p} - \frac{1}{k_p N}\mathbf{b}) \leq 0 \end{aligned} \tag{12}$$

where we used the property in equation 10. Defining the set $\Omega_e = \{\mathbf{p} \mid \dot{V}(\mathbf{p}) = 0\}$ and invoking the LaSalle's invariance principle we can state that as $t \to \infty$ the state $\mathbf{p}$ will converge to the largest invariant subset of Ω_e. A closer look on Ω_e shows that $\mathbf{p} \in \Omega_e$ $\mathbf{p} = \frac{1}{k_p N}\mathbf{b} + \alpha\mathbf{1}_v$ where $\mathbf{1}_v$ is an $N-$ dimensional vector of ones and α is a real valued constant. Note from here that all the elements of Ω_e correspond to a constant translation of the equilibrium configuration corresponding to $\mathbf{p} = \frac{1}{k_p N}\mathbf{b}$. Therefore, Ω_e is itself invariant and from above we know that as $t \to \infty$ we have $\mathbf{p} \to \infty$. By evaluating $L\mathbf{p}$ one obtains

$$\begin{aligned} L(\mathbf{p}) &= \frac{1}{k_p N}L\mathbf{b} + \alpha L\mathbf{1}_v \\ &= \frac{1}{k_p N}L\mathbf{b} \\ &= \frac{1}{k_p}\mathbf{b} \end{aligned} \tag{13}$$

which is equivalent to

$$\sum_{j=1}^{N}(p_j - p_i - \delta_{ji}) = 0, \quad i = 1, ..., N \tag{14}$$

In other words, the equilibrium configurations in Ω_e correspond to the desired formation.

Lastly, to prevents the swarm from oscillations and to reduce the formation settling time following extended equation is constructed by using the derivative portion of the control rule in equation 6

$$u_i^f = k_p \sum_{j=1}^{N}(p_j - p_i - \delta_{ji}) + k_d \sum_{j=1}^{N}(v_j - v_i) \tag{15}$$

where v_i stands for the velocity of the i^{th} quadrotor, $k_p > 0$ and $k_d > 0$ are the proportional and derivative control gains, respectively.

Collision Avoidance

Both solution of the combinatorial transportation assignment problem and formation control rule described above may not totally prevent inter-quadrotor collisions. In order to guarantee collisionfree formations, we use a potential function-based method which is also expressed as repellent morse potential [D.J.Bennet, 2010; V. Gazi, 2013]. The control rule between i^{th} and j^{th} quadrotors can be formulated as

$$u_{ij}^r = \left\{ \begin{array}{ll} \alpha(e^{-\beta||r_{ji}||} - e^{-\beta r_s}), & if||r_{ji}|| \leq r_s \\ 0, & otherwise \end{array} \right\} \tag{16}$$

where $\alpha > 0$ stands for the control gain, $\beta > 0$ shows the exponent scaler, $r_{ji} = p_j - p_i$ and r_s is the safety region which activates the rule. Then, the total repellent force for the i^{th} quadrotor is expressed as

$$u_i^r = \sum_{j=1}^{N} u_{ij}^r \tag{17}$$

The Overall Controller

The overall controller for the ith quadrotor can be expressed as

$$u_i = u_i^f + u_i^r \tag{18}$$

Notice that this type of velocity control command is not directly applicable to a conventional quadrotor. Therefore, we need to convert it to a desired position which is accepted by a position controller on a quadrotor. This can be achieved by the following conversion rule

$$p_i^d(t_k + \tau) = p_i(t_k) + \int_{t_k}^{t_k+\tau} u_i(t)dt \tag{19}$$

where t_k is the current time, p_i^d is the generated reference position for quadrotor i, and τ is the integration period. The smaller the τ, the more precise the path generated. For the discrete time case, we may use the following discrete version of the rule

$$p_i^d(k) = p_i(k) + u_i(k)T \tag{20}$$

where T is the sampling period.

QUADROTOR DYNAMICS, KINEMATICS, AND CONTROL

In this section, we briefly describe the equations of motion of a quadrotor with mass m and inertia tensor J according to plus (+) configuration (Fig. 3).

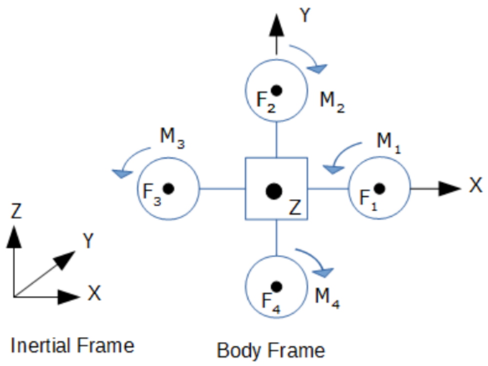


Figure 2: Forces Acting on Quadrotor

Dynamics and Kinematics

One may derive the equations of motion of a quadrotor having plus (+) configuration using Newton-Euler formalism [D. Mellinger, 2012] as

$$m\ddot{p} = -mge_3 + FR^W e_3 \tag{21}$$

$$I\dot{\Omega} = -\Omega \times I\Omega + \tau \tag{22}$$

where the parameters are described in Table 1.

Table 1

Parameter	Description
(m)	Quadrotor mass
$p \in \mathbb{R}^3$	Position
$I \in \mathbb{R}^{3\times 3}$	Inertia tensor
$F \in \mathbb{R}$	Total input thrust
$\tau \in \mathbb{R}^3$	Input torque
$\Omega = [\overline{p}\ \overline{q}\ \overline{r}]^T$	Angular velocity
$R^W \in \mathbb{R}^{3\times 3}$	Rotation matrix
g	Gravity constant
$e_3 = [0\ 0\ 1]^T$	Unit vector

Figure 3, the four identical rotors produce the total thrust as, $F = F_1 + F_2 + F_3 + F_4$. At the same time, the rotors generate rotational moments about the body axes as

$$\tau_\phi = l(F_2 - F_4) \tag{23}$$

$$\tau_\theta = l(F_3 - F_1) \tag{24}$$

$$\tau_\psi = M_1 - M_2 + M_3 - M_4 \tag{25}$$

where τ_ϕ causes roll (ϕ), τ_θ causes pitch (θ), τ_ψ causes yaw (ψ), $M_i,\ i = 1, ..., 4$ are the moments, and l is the arm length of the quadrotor. The matrix R^W is defined using Z, Y, X Euler angles as

$$R^W = \begin{bmatrix} C\theta C\psi & S\phi S\theta C\psi - C\phi S\psi & C\phi S\theta C\psi + S\phi S\psi \\ C\theta S\psi & S\phi S\theta S\psi + C\phi C\psi & C\phi S\theta S\psi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \tag{26}$$

where S and C stands for $\sin(.)$ and $\cos(.)$, respectively. We may model the DC motors using the following first order differential equations

$$\dot{\omega}_i = \tau_m(\omega_i^{des} - \omega_i), \quad i = 1, ..., 4 \tag{27}$$

where τ_m stands for motor gain and ω_i^{des} represents the desired angular speed of the ith rotor.

Quadrotor Control

A conventional quadrotor usually has a cascaded control structure where the inner controller controls the attitude (ϕ, θ, ψ) as well as the altitude (z) and the outer loop is responsible for controlling the position (x, y). Because the outer loop has much more longer time response than that of the inner loop, the dynamics of both loops can be decoupled and studied individually. In this part, we develop control rules for each loop based on the feedback linearization technique.

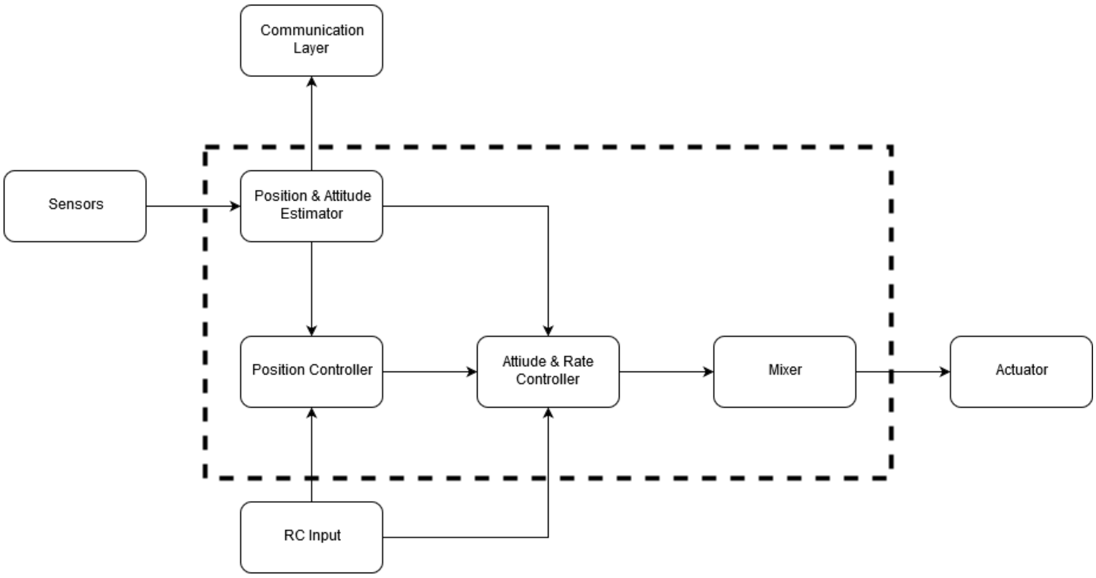


Figure 3: Typical Quadrotor Flight Controller.

Altitude Control:

Assuming that the attitude angles are small and ensuring $(\cos(\theta)\cos(\phi) \neq 0)$, the feedback linearization-based approach can be adopted from equation 21 as

$$F = (u_z + u_z^{pd})\frac{m}{\cos(\theta)\cos(\phi)} \tag{28}$$

where the term $u_z = g$ compensates the gravity effect and the auxiliary input u_z^{pd} is generated by the following PD controller

$$u_z^{pd} = k_{p,z}(z_d - z) + k_{d,z}(\dot{z}_d - \dot{z}) \tag{29}$$

where z_d stands for desired altitude, $k_{p,z}$, and $k_{d,z}$ represent proportional and derivative control gains, respectively.

Attitude Control:

Feedback linearization method can also be developed to stabilize the quadrotor attitude using equation 22 as

$$\tau_\phi = (u_\phi + u_\phi^{pd})\frac{I_{xx}}{l} \tag{30}$$

$$\tau_\theta = (u_\theta + u_\theta^{pd})\frac{I_{yy}}{l} \tag{31}$$

$$\tau_\psi = (u_\psi + u_\psi^{pd})I_{zz} \tag{32}$$

where I_{xx}, I_{yy}, I_{zz} are the diagonal entries of the inertia matrix, I, and u_ϕ, u_θ, u_ψ are the expressions from the system dynamics which can be calculated as

$$u_\phi := \dot{\theta}\dot{\psi}\frac{I_{yy} - I_{zz}}{I_{xx}} - \dot{\theta}\omega_r\frac{J_r}{I_{xx}} \tag{33}$$

$$u_\theta := \dot{\phi}\dot{\psi}\frac{I_{zz} - I_{xx}}{I_{yy}} + \dot{\phi}\omega_r\frac{J_r}{I_{yy}} \tag{34}$$

$$u_\psi := \dot{\phi}\dot{\theta}\frac{I_{xx} - I_{yy}}{I_{zz}} \tag{35}$$

where J_r is the rotor's inertia and $\omega_r := -\omega_1 + \omega_2 - \omega_3 + \omega_4$. The auxiliary input terms can be computed by the PD control rules

$$u_\phi^{pd} = k_{p,\phi}(\phi_d - \phi) + k_{d,\phi}(\dot{\phi}_d - \dot{\phi}) \tag{36}$$

$$u_\theta^{pd} = k_{p,\theta}(\theta_d - \theta) + k_{d,\theta}(\dot{\theta}_d - \dot{\theta}) \tag{37}$$

$$u_\psi^{pd} = k_{p,\psi}(\psi_d - \psi) + k_{d,\psi}(\dot{\psi}_d - \dot{\psi}) \tag{38}$$

where ϕ_d, θ_d, ψ_d stand for desired attitude angles, $k_{p,\phi}$, $k_{p,\theta}$, $k_{p,\psi}$, and $k_{d,\phi}$, $k_{d,\theta}$, $k_{d,\psi}$ represent proportional and derivative control gain sets, respectively.

Position Control in the Cartesian Space:

Position controllers (x, y) are considered as the outer loops in quadrotor control systems. We can design the PD control rules as

$$u_x^{pd} = k_{p,x}(x_d - x) + k_{d,x}(\dot{x}_d - \dot{x}) \tag{39}$$

$$u_y^{pd} = k_{p,y}(y_d - y) + k_{d,y}(\dot{y}_d - \dot{y}) \tag{40}$$

where x_d and y_d represent the desired references, $k_{p,x}$, $k_{p,y}$, and $k_{d,x}$, $k_{d,y}$ stand for proportional and derivative control gain sets, respectively. If we assume that the quadrotor is in hover, referring to equation 22 one can convert the outputs of the position controllers to the desired angles (ϕ_d and θ_d) using the equations

$$\phi_d = \frac{1}{g}(u_x^{pd}\sin(\psi) - u_y^{pd}\cos(\psi)) \tag{41}$$

$$\theta_d = \frac{1}{g}(u_x^{pd}\cos(\psi) + u_y^{pd}\sin(\psi)). \tag{42}$$

These equations have been derived in [D. Mellinger, 2012] and interested readers may consult that work for details. Here, we use these reference angles in the attitude controllers.

At the final stage, the output of the controllers are transferred into the squared rotor speeds using equation 43.

$$\begin{bmatrix}(\omega_1^{des})^2 \\ (\omega_2^{des})^2 \\ (\omega_3^{des})^2 \\ (\omega_4^{des})^2\end{bmatrix} = \frac{1}{k_F}\begin{bmatrix}1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ \kappa & -\kappa & \kappa & -\kappa\end{bmatrix}^{-1}\begin{bmatrix}F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi\end{bmatrix} \tag{43}$$

where $\kappa = k_M/k_F$. Here, we need to make sure that the desired rotor speed are saturated between predefined minimum and maximum values, which we choose as non-negative constants. Then, the desired motor speeds ω_i^{des} are computed using the square root operation.

RESULTS AND DISCUSSION

In our experiments, initially we present computer simulations and then we validate our algorithm on real quadrotors. The parameters of quadrotor are presented in Table 2.

Table 2: DJI F450 quadrotor parameters

Parameter	Value
mass (m)	0.812 kg
arm length (l)	0.22 m
moment of inertia about x-axis (I_{xx})	0.01085 $kg\ m^2$
moment of inertia about y-axis (I_{yy})	0.01092 $kg\ m^2$
moment of inertia about z-axis (I_{zz})	0.02121 $kg\ m^2$
inertia matrix (I)	$[I_{xx}\ 0\ 0; 0\ I_{yy}\ 0; 0\ 0\ I_{zz}]$
rotor's inertia (J_r)	5.225e-5 $kg\ m^2$
minimum rotor speed (ω_{min})	0 rad/s
maximum rotor speed (ω_{min})	1425 rad/s
motor constant (k_F)	1.1236e-5
moment constant (k_M)	1.4088e-7
motor gain (τ_m)	0.5721 s

Simulations

In the computer simulation, we used 20 identical DJI F450 quadrotor helicopters. For this experiment the desired formation type is triangular one. We tested how the system achieves desired formation shape and how the Hungarian algorithm generates the desired path for each quadrotor when the settling of the formation in first time from randomly generated initial locations to the square shape and when formation type changes from the square one to the triangular shape.

The proportional ($k_{p,\phi}$, $k_{p,\theta}$, $k_{p,\psi}$, $k_{p,z}$) and derivative ($k_{d,\phi}$, $k_{d,\theta}$, $k_{d,\psi}$, $k_{d,z}$) control gain sets of the inner loop controllers were set to 10 and 6, respectively. For the outer loop (position control), we set the proportional ($k_{p,x}$, $k_{p,y}$) and derivative ($k_{d,x}$, $k_{d,y}$) controller gain sets to 2 and 3, respectively. The formation controller parameters namely, (k_p, k_R) were set to 35 and 0.7, respectively.

Figure 4 shows initial locations for each quadrotor for XY cartesian coordinate system.Notice that the identity of each quadrotor is also shown in the figure.

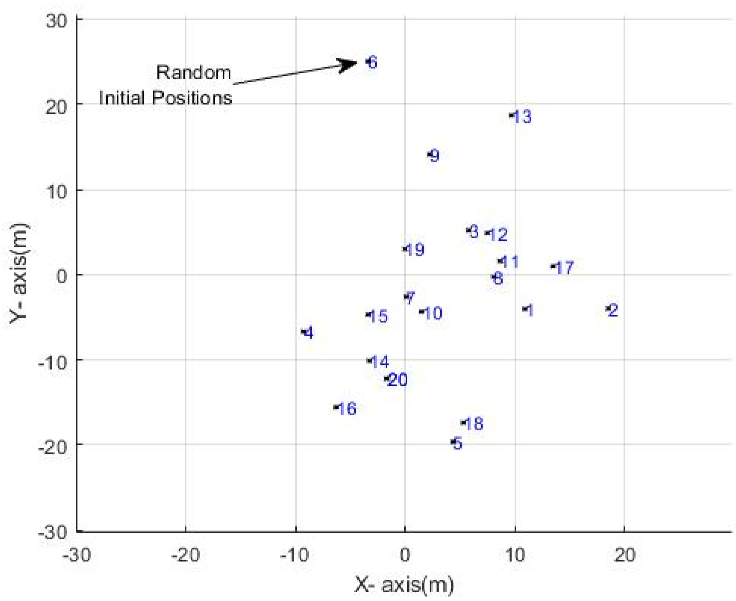


Figure 4: Initial locations for the 20 quadrotors.

In figure 5, we see how the Hungarian algorithm generates the desired path for each quadrotor from intial positions to square shape formation positions.

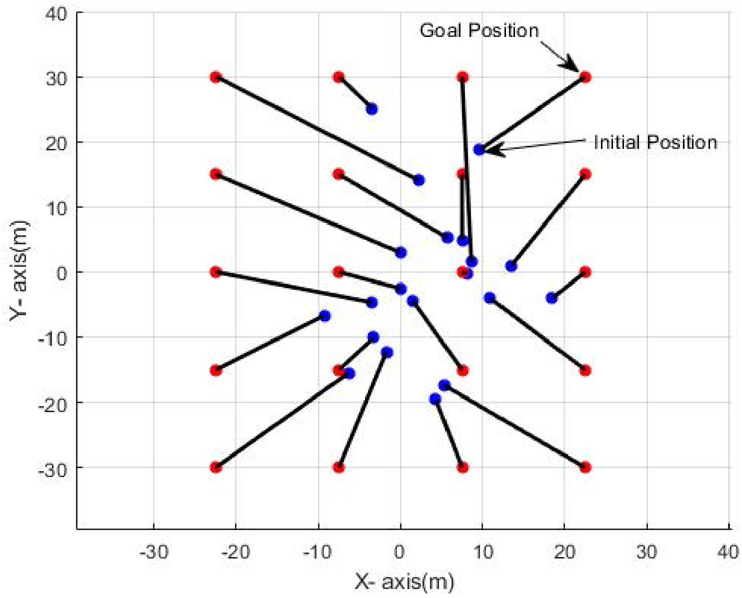


Figure 5: Generated paths from random initial positions.

Finally, in figure 6, we see that the quadrotors in the experiment achieve the desired circular formation using the proposed formation control rule without any collisions and oscillations within a short period of time.

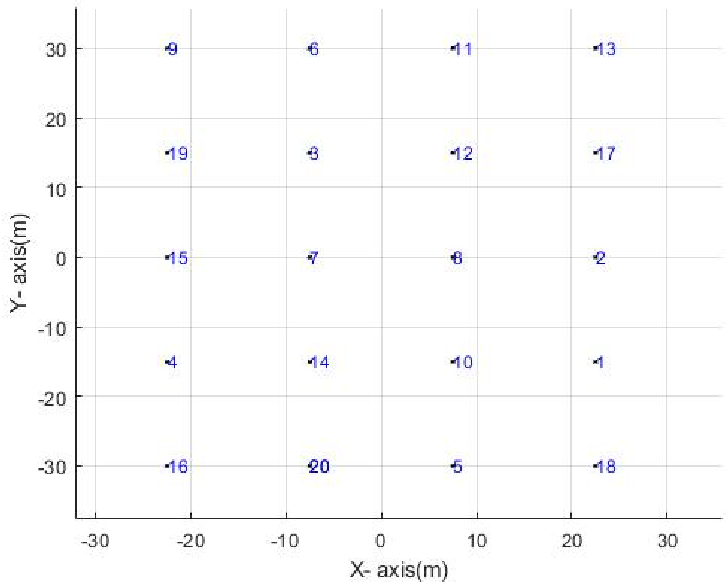
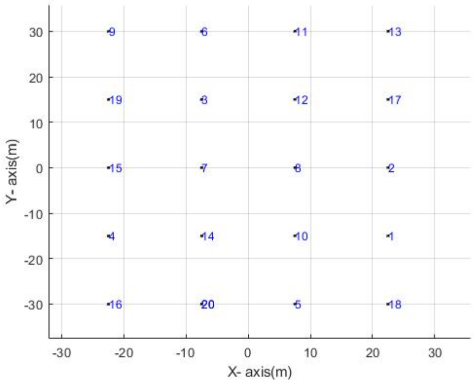
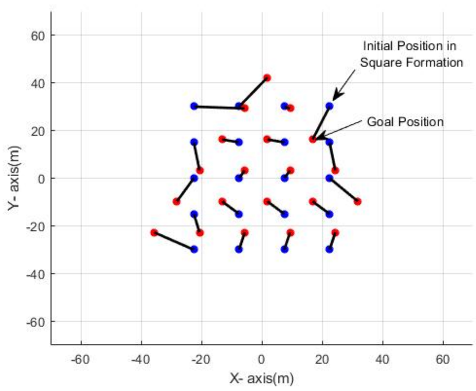


Figure 6: Achieved square shape formation for 20 quadrotors

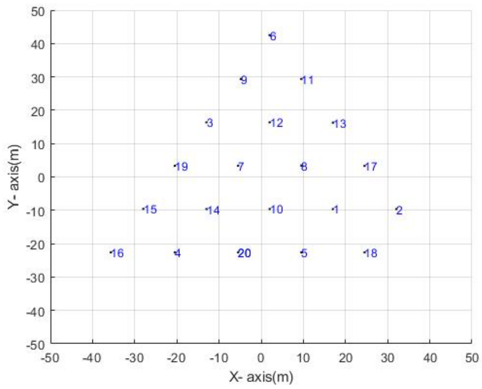
One can also observed the same path generation efficiency in changing the formation types. Following figures shows us how the Hungarian algorithm generates the desired path for each quadrotor when formation type changes from the square one to the triangular shape.



(a) Square formation



(b) Generated paths



(c) Triangular formation

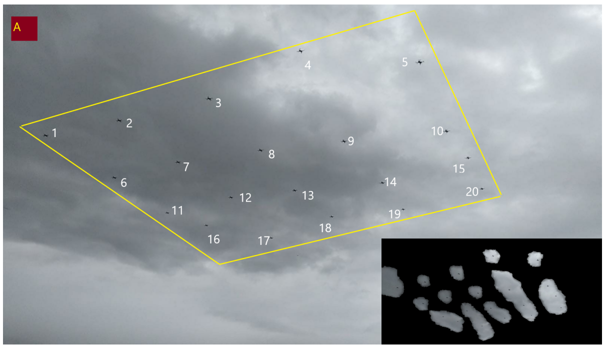
Figure 7: Simulation of the Hungarian algorithm for 20 quadrotors when changing formation type

Experiments

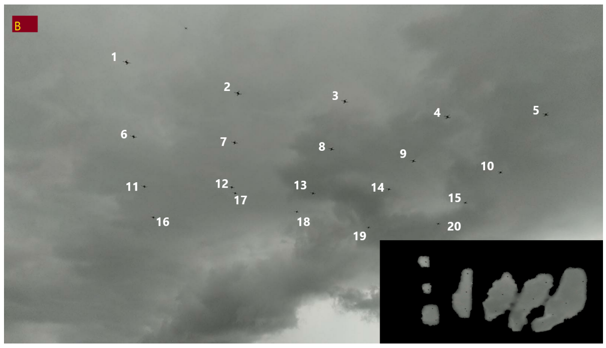
Next, we test the proposed swarm control and path generation law in hardware experiments with real quadrotors.The experiments are performed outdoor. 20 quadrotors were employed in the experiments. Similar to the simulations we considered square shape and triangular shape formation in the experiments. The algorithms for attitude, altitude, and position control of quadrotor are running on-board, the swarm algorithm is running on the computer in decentralized manner. Localization of the quadrotors was performed using COTS GPS modules.

This experiment demonstrates the ability to change the formation without any collision and ossilation. In the scenario, initially the quadrotors are randomly placed in the environment. Then, the formation is achieved 30 meter above the ground. After successful acquisition of the formation, the swarm changes its formation from square to triangular one. Because of the environmental constraints(rainy day), GPS positioning was not accurate and here one can observed the some mismatches in formation positions of some quadrotors.

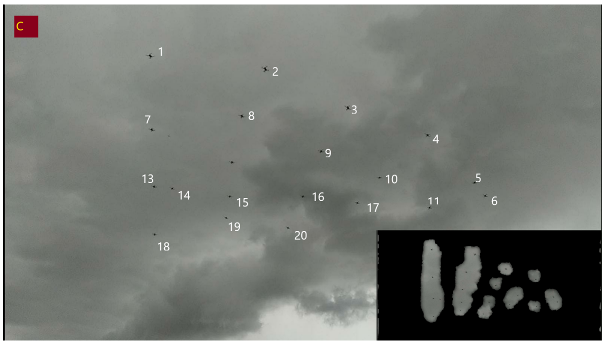
Figure 8 shows snapshots from an example conducted experiment. Numbers in the figures are used to show the number of quadrotors in the formation only.



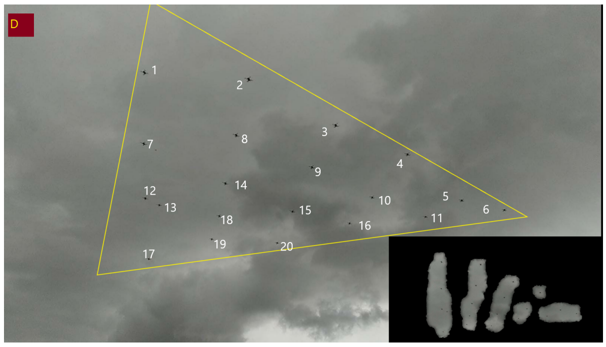
(a) Square formation



(b) Changing the formation



(c) Changing the formation



(d) Triangular formation

Figure 8: Real outdoor experiments of the Hungarian algorithm for 20 quadrotors when changing formation type

CONCLUSIONS

In this paper, we proposed a distributed swarm control mechanism with an appropriate assignment methodology to generate collision-free trajectories for a quadrotor swarm members. The method only requires sharing the positions of the swarm members using a communication system. Assuming that each quadrotor has single integrator dynamics in the formation control level, we developed a formation control rule and presented the stability analysis of it. Because the formation controller is based on point mass dynamics and the formation controller is therefore independent from the parameters of the quadrotors.This feature enables us using heterogeneous quadrotor agents in the swarm. Both simulation and real world experiments showed that the quadrotor agents employing the proposed technique successfully achieves formation without colliding with each other.

References

Bandala, A. A., Dadios, E. P., Vicerra, R. R. P. and Lim, L. A. G. (2014) *Swarming algorithm for unmanned aerial vehicle (uav) quadrotors– swarm behavior for aggregation, foraging, formation, and tracking*, Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 8, no. 9, 2014

Bennet, D.J. and McInnes, C.R. (2010) *Distributed control of multi-robot systems using bifurcating potential fields*, Robotics and Autonomous Systems, 58(3):256-264, 2010

Dong, X., Yu, B., Shi, Z. and Zhong, Y. (2015) *Time-varying formation control for unmanned aerial vehicles: Theories and applications*, IEEE Transactions on Control Systems Technology, vol. 23, no. 1, pp. 340–348, 2015.

Fax, J. A. and Murray, R. M. (2004) *Information flow and cooperative control of vehicle formations*, IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1465–1476, Sept 2004

Gazi, V. (2013) *On lagrangian dynamics based modeling of swarm behavior*, Physica D: Nonlinear Phenomena, 260:159-175, October 2013

Gioioso, G., Franchi, A., Salvietti, G., Scheggi, S. and Prattichizzo, D. (2014) *The flying hand: A formation of uavs for cooperative aerial telemanipulation*, International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 4335–4341

Kushleyev, A., Mellinger, D., Powers, C. and Kumar, V. (2013) *Towards a swarm of agile micro quadrotors*, Autonomous Robots, vol. 35, no. 4, pp. 287–300, 2013

Mellinger, D., Michael, N. and Kumar, V. (2012) *"Trajectory generation and control for precise aggressive maneuvers with quadrotors*, The International Journal of Robotics Research, vol. 31, no. 5, pp. 664–674, 2012

Oh, K.K., Park, M.C. and Ahn, H.S. (2015) *A survey of multi-agent formation control*, Automatica, vol. 53, pp. 424–440, 2015

Saska, M., Vakula, J. and Preucil, L. (2014) *Swarms of micro aerial vehicles stabilized under a visual relative localization*, International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 3570–3575

Turpin, M., Michael, N. and Kumar, V. (2014) *Capt: Concurrent assignment and planning of trajectories for multiple robot*, The International Journal of Robotics Research, 33(1):98112,2014.

Yu, B., Dong, X., Shi, Z. and Zhong, Y. (2013) *Formation control for quadrotor swarm systems: Algorithms and experiments*, 32nd Chinese Control Conference (CCC). IEEE, 2013, pp. 7099–7104