Beamforming in multi-agent robotic formation

July 31, 2020

1 New Ideas

- Formation control: decouple the high level control to lower level control to agents.

  yes, focus on this problem also as it will be useful when implem enting.

- Optimal Control: To choose between changing weights and rotation.
- ROS simulation of the agents

1.1 Decouple the high level control low level

To derive a map which maps this high level control strategy to a low level inputs to each agents.

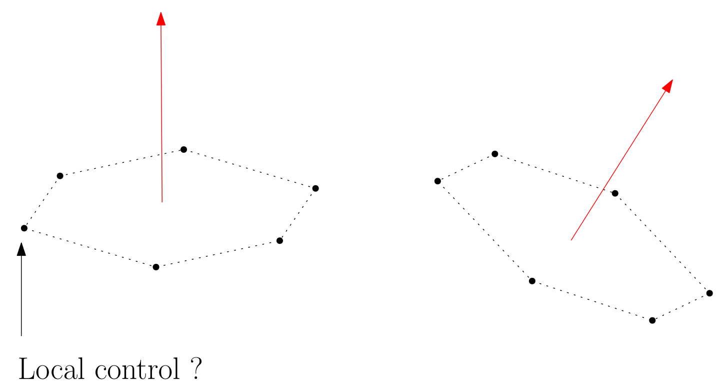


Figure 1: Rotation of the plane of formation

1.1.1 Literature survey

From [1], there are three types of the formation control,

- Position-based control: where we know absolute position of the agents in the formation.

- Displacement based control: where we know the displacement of the agents while constraining that the local frame of the agents is aligned with global frame.

- Distance based control: This needs relative distance of the agents in the formation. **Sir in our paper I mentioned this as the type of the formation**, where the formation is treated as the rigid body.

Since the formation is tracking a trajectory we need to know the absolute position of some of the agents in the formation. **In our case, absolute position of the centroid is enough to find others.**

Our goal: Tracking a trajectory with constraint on the rotation of the rigid body plane. Now we need to find a map which takes higher level objective and gives the low level control inputs to the each agents in the formation (polygonal formation). **By which I mean that, when a rotation and transition of the polygonal plane is intended along the trajectory, what should be the trajectories followed by the individual agents?**

Which I did not find in the literature as of now but I will look into it if any.

A formation control for UAVs whose centroid is following a trajectory is considered in [2, 3].

In the literature it is not been tried to achieve global orientation while following a trajectory. Now I am reading [4], [5].

1.2 Optimal control

Since the agents are following a trajectory, this will be a dynamic optimization problem, where along the trajectory the agents in the formation has to choose between changing rotation or multiplying by the weights (DTB weights) which is decided by the cost of the energy spent on the both tasks. Now the task is to formulate the cost function of the DTB weights and rotation task.

First, energy required for the motion of each agent is considered. Agent is a quad rotor. Energy required for shifting the phase of the signal is considered next. The energy model is taken from [6].

1.2.1 Actuator model

Quad rotor uses brush less DC (BLDC) motors as the actuators. The electrical part of the motor has following equation.

$$v(t) = Ri(t) + L\frac{\partial i(t)}{\partial t} + \frac{\omega(t)}{k_v} \tag{1}$$

The torque and current are related as follows,

$$\tau(t) = k_t i(t) \tag{2}$$

The mechanical part of the motor is modelled using the equation,

$$J\frac{\partial \omega(t)}{\partial t} = \tau(t) - k_\tau \omega^2(t) - D_v \omega(t) \tag{3}$$

D_v is the viscous damping coefficient of the motor. Inductance of the small DC motor can be neglected, so equation (1.2.1) reduces to

$$v(t) = Ri(t) + \frac{\omega(t)}{k_v} \tag{4}$$

1.2.2 Quad rotor dynamics

Dynamics of the quad rotor is obtained from [7].

$$\begin{aligned}
m\ddot{x} &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)T \\
m\ddot{y} &= (\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)T \\
m\ddot{z} &= (\cos\phi\cos\theta)T - mg \\
I_x\ddot{\phi} &= (I_y - I_z)\dot{\theta}\dot{\psi} + J\dot{\theta}(\omega_1 - \omega_2 + \omega_3 - \omega_4) + lu_1 \\
I_y\ddot{\theta} &= (I_z - I_x)\dot{\phi}\dot{\psi} + J\dot{\phi}(\omega_1 - \omega_2 + \omega_3 - \omega_4) + lu_2 \\
I_z\ddot{\psi} &= (I_x - I_y)\dot{\phi}\dot{\theta} + u_3
\end{aligned} \tag{5}$$

Control inputs are given as follows:

$$\begin{aligned}
T &= k_b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\
u_1 &= k_b(\omega_2^2 - \omega_4^2) \\
u_2 &= k_b(\omega_3^2 - \omega_1^2) \\
u_3 &= k_\tau(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)
\end{aligned} \tag{6}$$

Euler angle are assumed to be in acceptable range.

1.2.3 Energy model

Energy consumed by the quad rotors is given by, [8].

$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^{4} \tau_j(t)\omega_j(t)dt \tag{7}$$

Using previous equations, equation (1.2.3) becomes,

$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^{4} (\dot{\omega}_j(t) + k_\tau\omega_j^2(t) + D_v\omega_j(t))\omega_j(t)dt \tag{8}$$

1.2.4 Using motor efficiency

To have more realistic model, an efficiency function is used in equation (1.2.3). As in [8] $f_r(\tau(t), \omega(t))$ is used as the efficiency function, which is modeled using polynomial,

$$f_r(\tau(t), \omega(t)) = a(\omega(t))\tau^3(t) + b(\omega(t))\tau^2(t) + c(\omega(t))\tau(t) + d(\omega(t)) \tag{9}$$

and

$$
\begin{aligned}
a(\omega(t)) &= a_1\omega^2(t) + b_1\omega(t) + c_1 \\
b(\omega(t)) &= a_2\omega^2(t) + b_2\omega(t) + c_2 \\
c(\omega(t)) &= a_3\omega^2(t) + b_3\omega(t) + c_3 \\
d(\omega(t)) &= a_4\omega^2(t) + b_4\omega(t) + c_4
\end{aligned}
$$

These parameters are to be found for given motor. Then the consumed energy becomes,

$$
E_c = \int_{t_0}^{t_f} \sum_{j=1}^{4} \frac{(\dot{\omega}_j(t) + k_\tau \omega_j^2(t) + D_v \omega_j(t))}{f_{r,j}(\tau_j(t), \omega_j(t))} \omega_j(t) dt \tag{10}
$$

Final time t_f can be variable.

1.2.5 Energy consumed in maneuvering the formation

Now that we have the energy for a quad rotor to follow a generic trajectory, we extend this to get the combined energy consumed for agents in the formation when the formation is tracking a given trajectory. Consider a polygonal formation, in particular a hexagonal formation, with agent at the centroid (so total of seven agents).

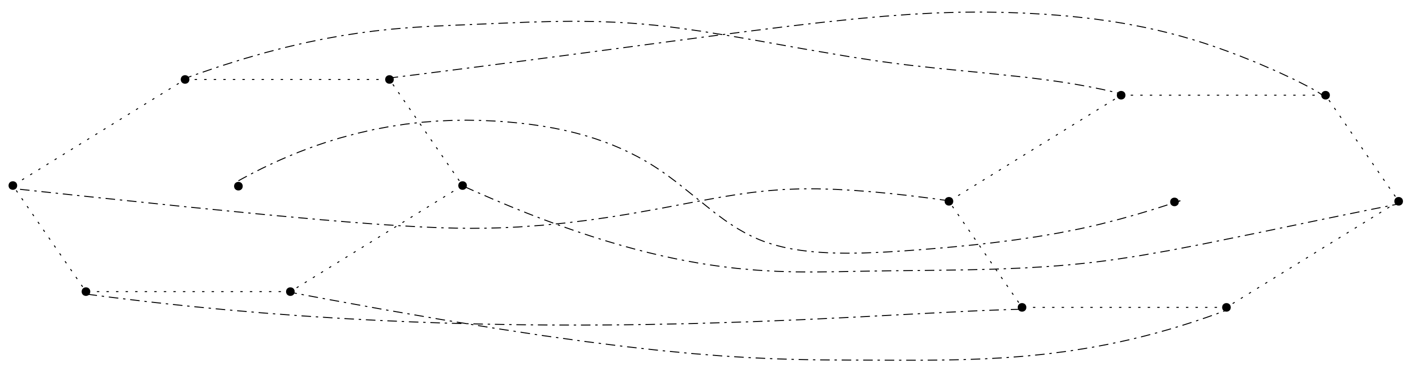


Figure 2: Trajectories followed by agents without rotation (Not accurately)

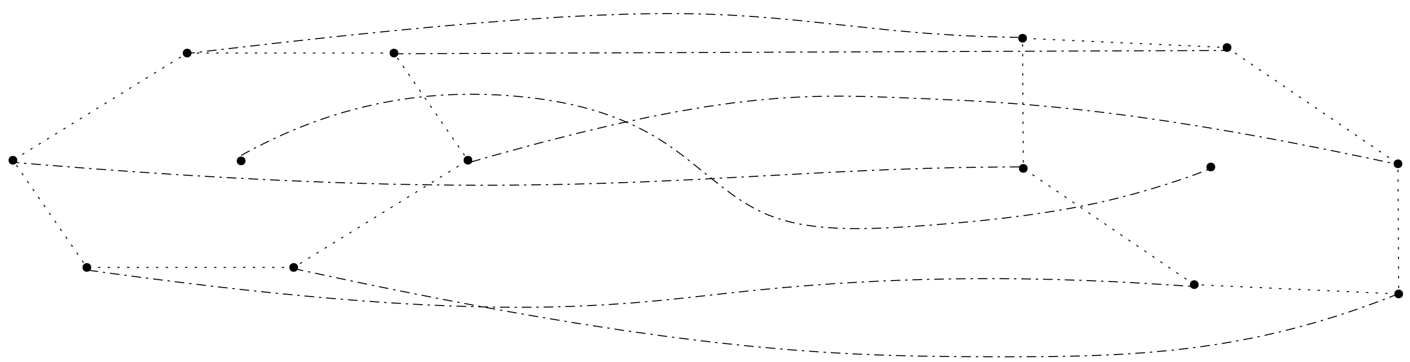


Figure 3: Trajectories followed by agents with rotation (Not accurately)

claim: Energy required for two maneuvers are different in general and supposedly energy required for maneuver without rotation of the formation plane is less than other maneuver. **(Have to prove this)**.

In general for any maneuvers the energy required for the formation to achieve that is,

$$E_{general}(t_f) = \sum_{i=0}^{n} \int_{t_0}^{t_f} \sum_{j=1}^{4} \frac{(\dot{\omega}_{i,j}(t) + k_\tau \omega_{i,j}^2(t) + D_v \omega_{i,j}(t))}{f_{r,i,j}(\tau_{i,j}(t), \omega_{i,j}(t))} \omega_{i,j}(t) dt \tag{11}$$

where n is the number of quad rotors in the formation, *general* meaning for any maneuver such as with or without rotation of the plane. **Possible extension is to minimize the energy required for the formation plane maneuver using optimal control [8]**. The constraints also include the formation to be intact along with other constraints. (it will be a nonlinear optimization).

$$\begin{gathered} \min_{\omega(t), t \in [t_0, t_f]} E_{general}(t_f) \\ Constraints \end{gathered} \tag{12}$$

1.2.6 Energy consumed for Phase shifting of the input signals

Suppose a signal $s(t)$ is transmitted from each agents, it is assumed that signal is sinusoidal. Let the signal is phase shifted by ϕ_i, $s(t) = e^{(j\omega t - \phi_i)}$. To implement this there will be some energy spent. Goal is to model that energy. For now let the energy consumed is $E_{ph}(t)$, along the trajectory. **This I am yet to get a mathematical model of the energy consumed.**

1.2.7 Control problem

Need to choose either E_{ph} or E_{fo} whichever is minimum, If energy required for phase shifting is less then signals are phase shifted otherwise the plane of the rotation is rotated to align the beam. If rotation of the formation plane is used then optimal control can be used to get any of the following - minimum energy, minimum time, *etc*, minimizing the energy is primary requirement.

Doubt: How can I know that which energy requirement is more, before solving optimal control problem.

constraints would be different for two maneuvers, that is formation related *constraints* will change and other constraints would remain same. For now, the *constraints* are abstract. Figure 4 depicts the switching control required. **This may not be required, have to discuss with sir**.

Figure 5 shows optimal control can be employed in the structure.

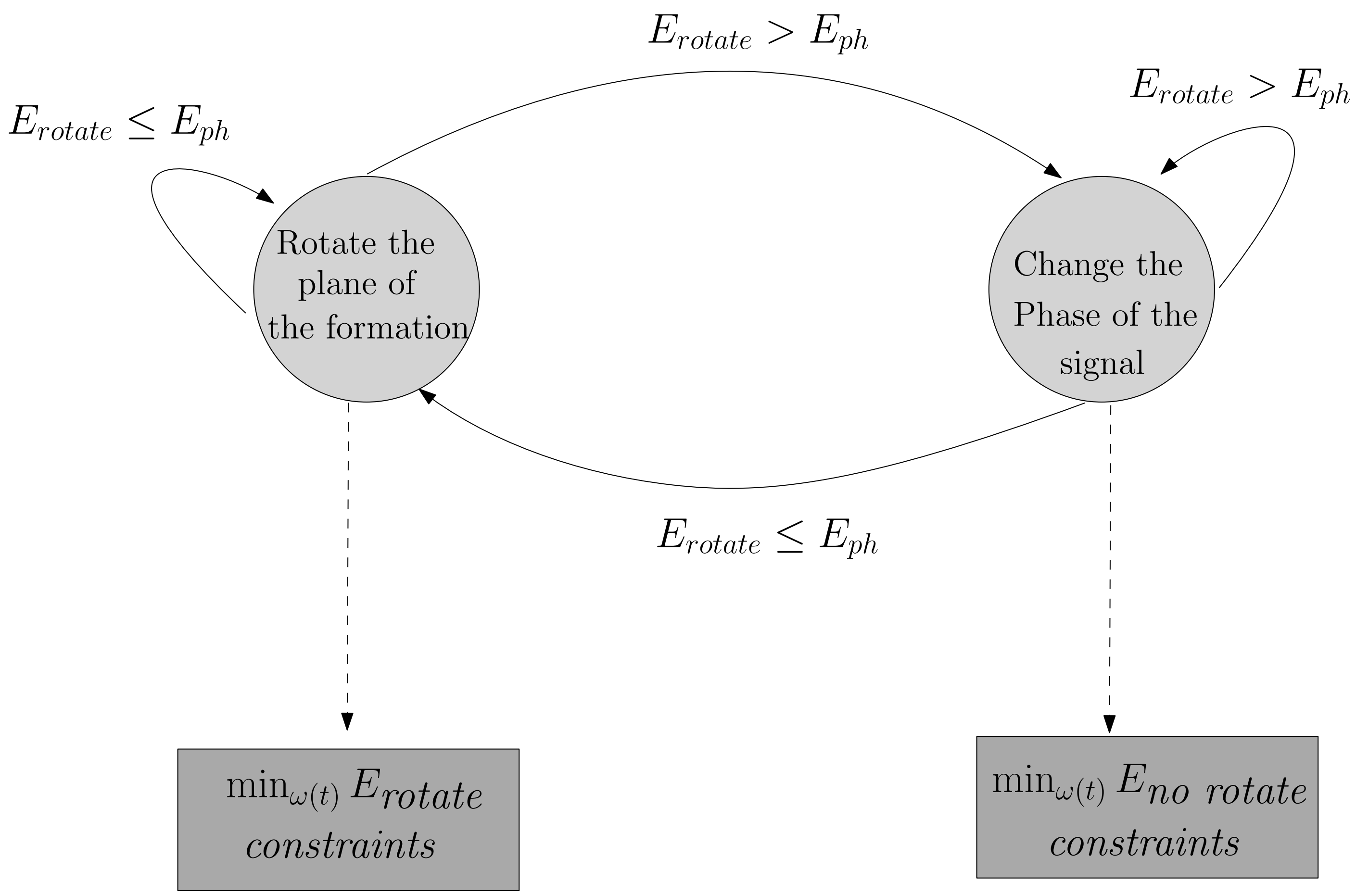


Figure 4: State Machine for hybrid control

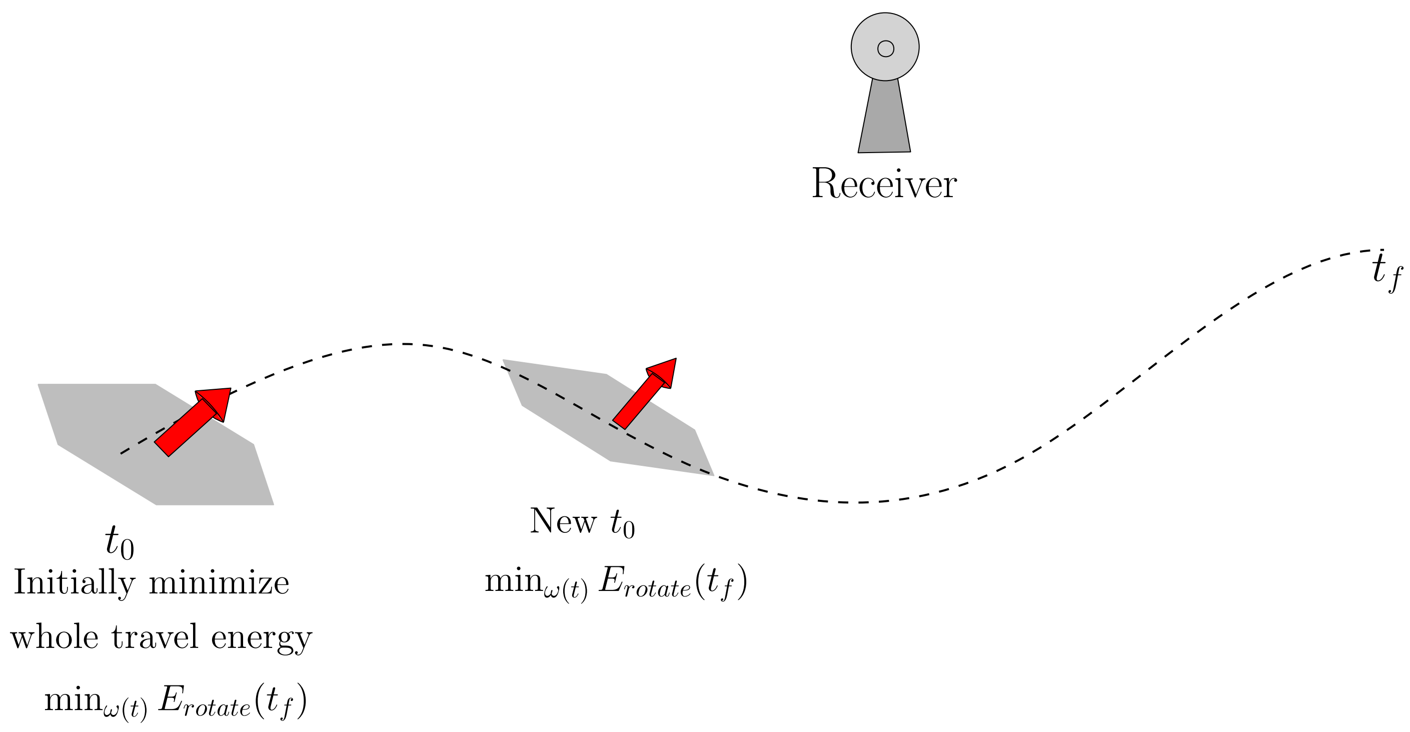


Figure 5: Trajectory followed

1.3 Preliminaries

1.3.1 Graph theory

$$G \in (V, E)$$

V and E are vertex and edge set. The graph is undirected if edges connecting the vertices do not impose a condition on the direction of the edge. Here it is assumed that graphs are undirected. Maximum number of edges possible are $\frac{n(n-1)}{2}$, if a graph has the maximum possible edges it is called complete graph.

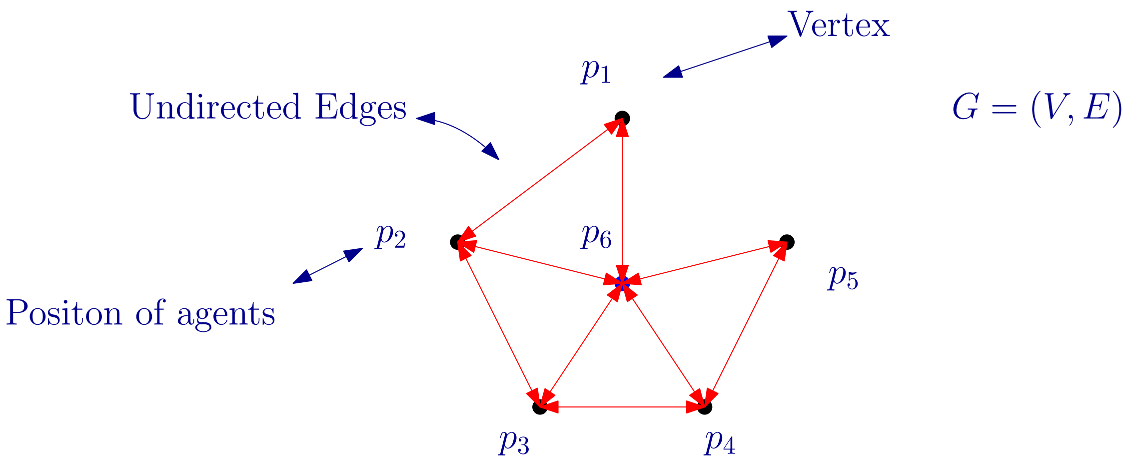


Figure 6: Pantagon

How the agents are connected in the graph are depicted by adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$. The adjacency matrix $\mathcal{A} = [a_{ij}]$ is defined as

$$\begin{aligned} a_{ij} &= 1 \text{ if } (i, j) \in E \\ &= 0 \text{ otherwise} \end{aligned} \tag{13}$$

So the adjacency matrix for the Graph in Figure 6 is ,

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{14}$$

Then we define Laplacian $\mathcal{L} = [l_{ij}]$ matrix as,

$$\begin{aligned} l_{ii} &= \sum_{j=1}^{n} a_{ij} \\ l_{ij} &= -a_{ij} \text{ , } i \neq j \end{aligned} \tag{15}$$

Note that the Laplacian matrix is symmetric and satisfies

$$\sum_{j=1}^{n} l_{ij} = 0, \ \ , i = 1, \ldots, n \tag{16}$$

The Laplacian matrix has some other interesting and useful properties. If $\lambda_1 \leq \cdots \leq \lambda_n$ denote the n eigenvalues of $\mathcal{L}$, then $\lambda_1 = 0$ and $\lambda_2 \geq 0$.

That is, $\mathcal{L}$ is a positive semi-definite matrix. If G is connected, then $\lambda_2 > 0$ (i.e., $\mathcal{L}$ has a single zero eigenvalue). In this case, the eigen vector associated with λ_1 is $1_n \in \mathbf{R}^n$ such that,

$$\mathcal{L}1_n = 0 \tag{17}$$

This implies that

$$\mathcal{L}x = 0, \text{ if and only if } x \in \mathbb{R}^n \text{ with } x_i = x_j, \forall i, j. \tag{18}$$

1.3.2 Framework

Framework is the mathematical notion to capture the formation, it is a realization of the graph in euclidean space. Hence we define,

$$F = (G, p) \tag{19}$$

as the framework, where $p = [p_1 \dots p_n] \in \mathbb{R}^{nm}$. Framework is the graph along with the positions of the agents in the formation.

we define $\phi : \mathbb{R}^{nm} \to \mathbb{R}^l$ for a framework as the **edge function**, which is given by,

$$\phi(p) = [\dots, \|p_i - p_j\|^2, \dots] \ \forall \ (i, j) \in E \tag{20}$$

So for the Figure 6 the edge function, is

$$\begin{aligned} \phi(p) = [&\|p_1 - p_2\|^2, \|p_1 - p_6\|^2, \|p_2 - p_3\|^2, \|p_2 - p_6\|^2, \|p_3 - p_4\|^2, \\ &\|p_3 - p_6\|^2, \|p_4 - p_5\|^2, \|p_4 - p_6\|^2, \|p_5 - p_6\|^2] \end{aligned}$$

1.3.3 Rigid graphs

Rigid is in the usual sense that the rigid body is one which can not be deformed. (and other equivalent definitions are possible).

Fact: For a rigid body translation and rotation can be decoupled.

Definitions: Two frameworks (G, p) and $(G, \hat{(p)})$ with $G = (V, E)$ are:

- Equivalent if $\|p_i - p_j\| = \|\hat{p}_i - \hat{p}_j\| \ \forall; \ \ (i, j) \in E$, i.e the edge function is same.
- Congruent if $\|p_i - p_j\| = \|\hat{p}_i - \hat{p}_j\| \ \forall; \ \ (i, j) \in V i \neq j$, i.e all the distances between vertices are same.
- An isometry of $\mathbb{R}^m$ is a bijective map $T : \mathbb{R}^m \to \mathbb{R}^m$,

$$\|T(x) - T(y)\| = \|x - y\|, \forall x, y \in \mathbb{R}^m. \tag{21}$$

- **A motion of a framework $F = (G, p)$ with $G = (V, E)$ is a continuous family of equivalent frameworks $F(t)$ for $t \in [0, 1]$ where $F(0) = F$.** .

  That is, each point $p_i, i \in V$ moves along a continuous trajectory $p_i(t)$ while preserving the distances between points connected by an edge.

Note: If there exist isomorphic map between two frameworks then they are congruent. We denote all frameworks that are isomorphic to F by $Iso(F)$

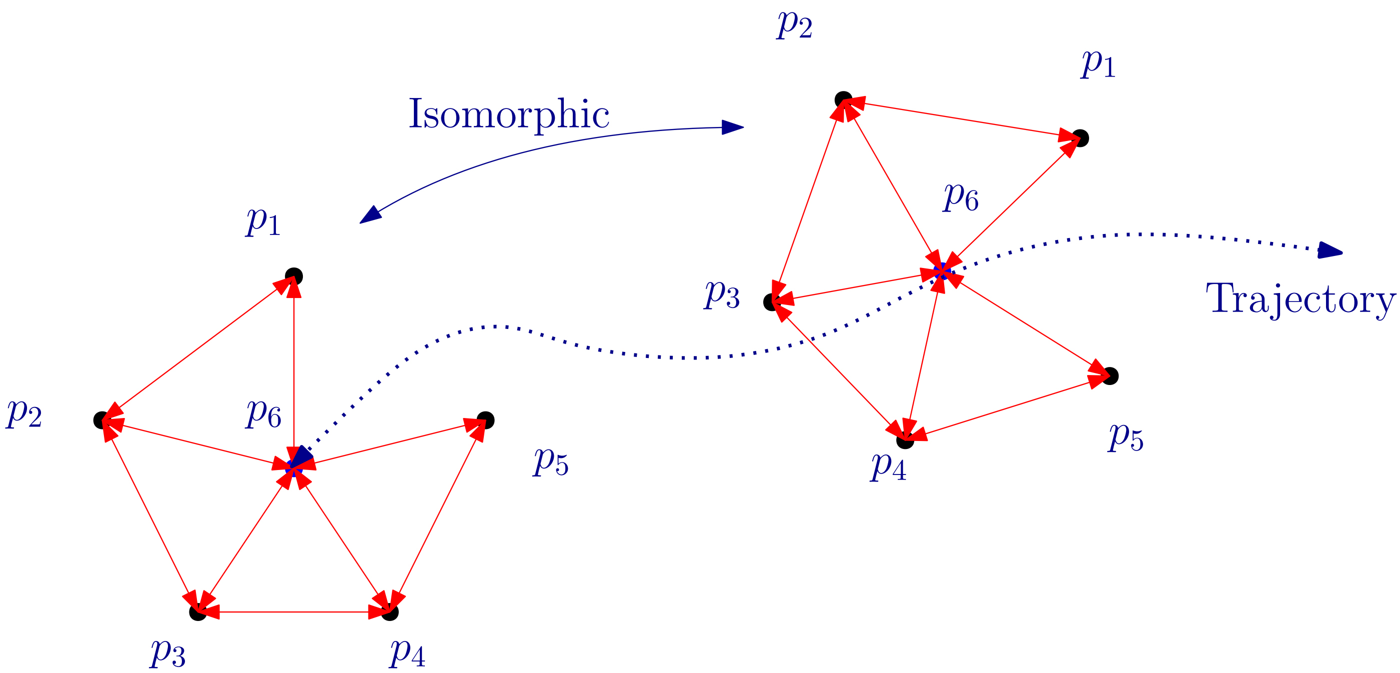


Figure 7: Isomorphic formations

A framework $F = (G, p)$ is rigid in $\mathbb{R}^m$ if all of its motions satisfy $p_i(t) = T(p_i), \forall i \in V$, and $\forall t \in [0, 1]$, i.e., the family of frameworks $F(t)$ is isomorphic. On the other hand, the framework is flexible in $\mathbb{R}^m$ if and only if it is possible to continuously move its vertices to form an equivalent but non-congruent framework.

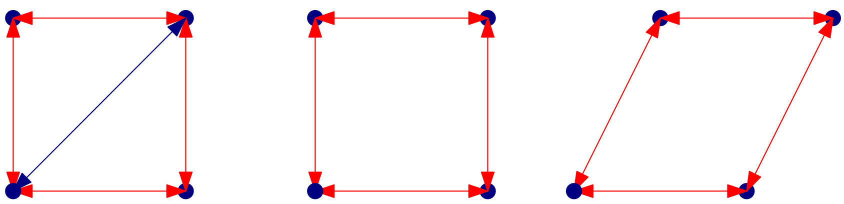


Figure 8: Flexible formations

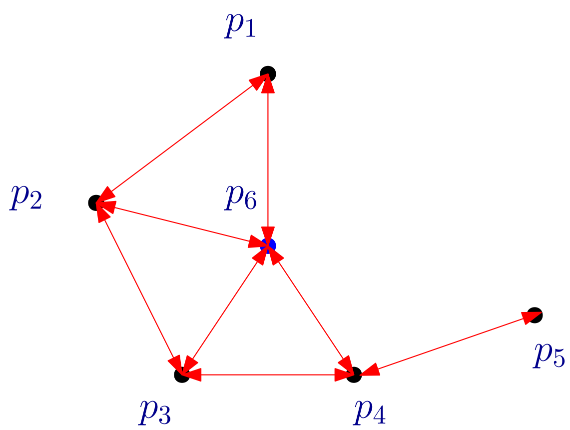


Figure 9: Flexible formations

1.3.4 Infinitesimal rigidity

Mathematically modelling rigidity of the formation is difficult, so we have to go for the infinitesimal rigidity. This acts as a sufficient condition for the rigidity. (infinitesimal rigidity implies rigidity).

Before going to the definitions of infinitesimal rigidity let us diverge a bit to cover generically rigid graphs or generic frameworks (*Keywords are different in different textbooks please bear with me*).

I am less clear on generic rigidity, However these are not as important as other topics.

- In general for rigidity depends on the particular framework, however we can relax a bit on conditions of positions p.
- Suppose $\exists$ a framework (G, p) , let us denote graph corresponding to this framework as G_f. We define the generically rigid graph using this graph G_f as, **A graph is generically rigid if it has an infinitesimally rigid realization. These graphs are referred as rigid graphs.**
- Let G_f is a rigid graph and framework corresponding to say $F_f = (G_f, p)$ is infinitesimally rigid,then p is called generic configuration of G_f, and F_f is the generic realization.

Lemma : If G_f is a generically rigid graph, then the set of all generic configurations for G_f is a dense, open subset of R^{pn}.

This means to say that, if any configuration p' is a realization of the G_f can be well approximated by a generic configurations p such that $F_f(p) = (G_f, p)$ is infinitesimally rigid.

From the definition 1.3.3, relative distance among the agents is preserved while agents are following a trajectory.

$$\|p_i(t) - p_j(t)\| = \|p_i - p_j\|, \quad (i, j) \in E \tag{22}$$

RHS of (22) is constant, hence differentiating after squaring both sides we get,

$$\frac{d}{dt}\|p_i(t) - p_j(t)\|^2 = 2(p_i(t) - p_j(t))^\top(\dot{p}_i(t) - \dot{p}_j(t)) = 0 \quad (i, j) \in E \tag{23}$$

Since the condition also holds at $t = 0$, we have

$$2(p_i - p_j)^\top (v_i - v_j) = 0 \quad (i, j) \in E \tag{24}$$

where $v_i := \dot{p}_i(0)$ we call it as instantaneous velocity and $p_i(0) = p_i$ is initial position. (24) gives l equation with nm unknowns being velocities v_i. This can be analyzed using **rigidity matrix** $R : \mathbb{R}^{nm} \to \mathbb{R}^{l\times nm}$, which we define as,

$$R(p) = \frac{1}{2}\frac{\partial \phi(p)}{\partial p} \tag{25}$$

For pentagonal formation the rigidity matrix with respect to edge function (1.3.2).

$$R(p) = \begin{bmatrix} (p_1-p_2)^\top & (p_2-p_1)^\top & 0 & 0 & 0 & 0 \\ (p_1-p_6)^\top & 0 & 0 & 0 & 0 & (p_6-p_1)^\top \\ 0 & (p_2-p_3)^\top & 0 & (p_3-p_2)^\top & 0 & 0 \\ 0 & (p_2-p_6)^\top & 0 & 0 & 0 & (p_6-p_2)^\top \\ 0 & 0 & (p_3-p_4)^\top & (p_4-p_3)^\top & 0 & 0 \\ 0 & 0 & (p_3-p_6)^\top & 0 & 0 & (p_6-p_3)^\top \\ 0 & 0 & 0 & (p_4-p_5)^\top & (p_5-p_4)^\top & 0 \\ 0 & 0 & 0 & (p_4-p_6)^\top & 0 & (p_6-p_4)^\top \\ 0 & 0 & 0 & 0 & (p_5-p_6)^\top & (p_6-p_5)^\top \end{bmatrix} \tag{26}$$

It can be seen that using (24), $R(p)v = 0$, where $v = [v_1, \dots, v_n] \in \mathbb{R}^{nm}$.

There exist a infinitesimal motion when there a (nontrivial)nonzero solution v.

If the framework undergoes any rigid body motion, then vertex i has the velocity,

$$v_i = v^* + \omega \times p_i \tag{27}$$

where v^* is the translation velocity, and $\omega \in \mathbb{R}^3$ is the angular velocity.

It can be shown that for the $R(p)$ for the pentagon velocity corresponding to rigid body gives $R(p)v = 0$.

Now we **Define**:

A framework is infinitesimally rigid if the only solutions to $R(p)v = 0$ ***arise from rigid body motions. Otherwise, it is infinitesimally flexible.***

To check infinitesimally rigid,

A (generic) framework in R^m is infinitesimally rigid if and if,

$$\text{rank}(R(p)) = nm - \frac{m(m+1)}{2} \tag{28}$$

for 2 dimensions it is $\text{rank}(R(p)) = 2n - 3$.

Result: Consider two frameworks $F = (G, p)$ and $\overline{F} = (G, \overline{p})$ sharing the same graph. If F is infinitesimally rigid and $\text{dist}(\overline{p}, Iso(F)) \leq \epsilon$ where ϵ is a sufficiently small positive constant, then F is also infinitesimally rigid.

This comes from the generic frameworks.

1.3.5 Minimally rigid

A graph is minimally rigid if it is rigid and the removal of a single edge causes it to lose rigidity.

Pentagon formation shown earlier, is minimally rigid.

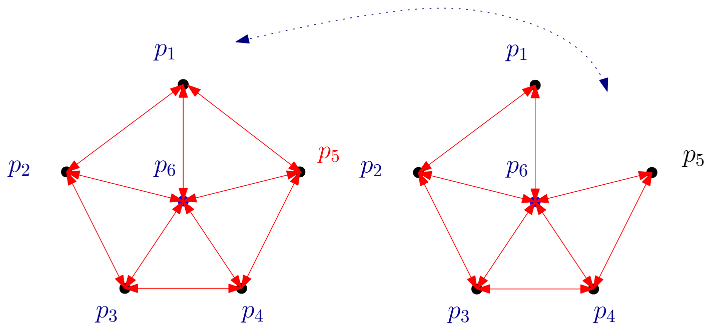


Figure 10: Minimally Rigid (right figure)

A rigid graph is minimally rigid if and only if $l = mn - \frac{m(m+1)}{2}$, where l is number of edges.

It can said as corollary of previous result. ***If a framework is infinitesimally and minimally rigid, then its rigidity matrix has full row rank and*** $R(p)R^{\top}(p)$ ***is positive definite.***

1.3.6 Framework Ambiguities

Here edges discussed so for the abstract notion for the maintaining distance, having communication link and so on.

Frameworks which are flexible are trivially ambiguous since they are not even minimally rigid. However, what is more interesting is to see the Ambiguities even when the frameworks are minimally rigid.

If two infinitesimally rigid frameworks are equivalent but not congruent, then they are said to be ambiguous.

$Amb(F)$ is the set of ambiguities of the infinitesimally rigid framework. There are two types of them,

1. Flip ambiguity

2. Flex ambiguity

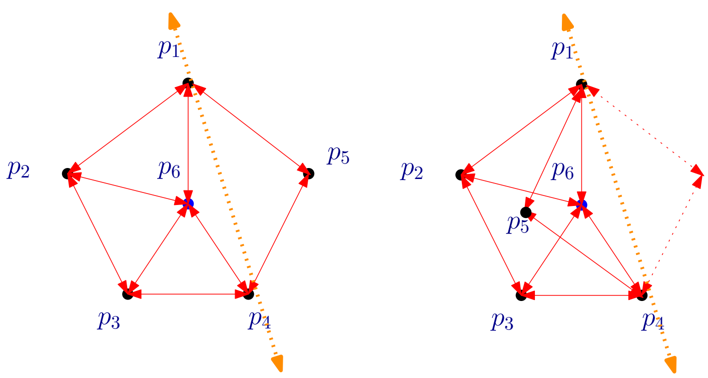


Figure 11: Flip Ambiguity

Flex ambiguity occurs mainly due to communication link break out. Suppose one of the vertex goes out of view (losses communication), if it comes back again the framework may no longer be same. **Show book for example**

To avoid converging to the ambiguous frameworks, agents should be initialized sufficiently close to desired formation.

Let $F = (G, p)$ and $\overline{F} = (G, \overline{p})$ be two frameworks sharing the same graph, and consider the function

$$\Psi(\overline{F}, F) = \sum_{(i,j)\in E} (\left\|\overline{p}_i - \overline{p}_j\right\| - \|p_i - p_j\|)^2 \tag{29}$$

If F is infinitesimally rigid and $\Psi(\overline{F}, F) \leq \delta$ where δ is a sufficiently small positive constant, then $\overline{F}$ is also infinitesimally rigid.

1.3.7 Global rigidity

To achieve global rigidity is to have underlying graph to be complete.

1.4 Formation Control Problems

The given framework is infinitesimally and minimally rigid framework. **Some Assumptions**

- The set where the agents achieve the desired formation is nonempty, i.e., there exist q^* such that $\phi(q^*) = d$ where $d = [..., d_{ij}^2, ...]$ and ϕ is the edge function.
- Along with the graph corresponding to the maintaining distance, there exist a graph for sensing G_s. We assume both these graphs are same.
- Initially the agents do not satisfy the desired inter-agent distance constraints.
- Only relative position of agent the neighbouring agents is required.

1.4.1 Formation Acquisition

Let F^* is the desired framework, G^* is the corresponding graph. First formation which is needed is to be acquired. Condition for that is

$$F(t) \to \text{Iso}(F^*) \text{ as } t \to \infty \tag{30}$$

$$\|q_i - q_j\| \to d_{ij} \text{ as } t \to \infty \;\; i, j \in V^* \tag{31}$$

$$\phi(q(t)) \to \phi(q^*) = d \text{ as } t \to \infty \tag{32}$$

Satisfying (31) is difficult.

1.4.2 Formation Maneuvering

$$\dot{q}_i(t) - v_{di}(t) \to 0 \text{ as } t \to \infty \quad i = 1, \dots n \tag{33}$$

where $v_{di}(t)$ is the velocity of the swarm (formation).

One of the agents is taken as reference for axis of rotation which is called as leader agent. But this requires,

- $q_n^* \in \operatorname{conv}\{q_1^*, \dots, q_{n-1}^*\}$.
- there is edge between each agent and the leader.

1.4.3 Single integrator agents

Let us assume each agent has single integrator model.

$$\dot{x}_i = u_i \tag{34}$$

Consider a relaxed condition for acquisition.

$$\|q_i - q_j\| \to d_{ij} \text{ as } t \to \infty \quad i, j \in E^* \tag{35}$$

However, (35) creates possibility of converging to a ambiguous framework.

- Relative distance between agents

  $$\tilde{q}_{ij} = q_i - q_j \tag{36}$$

  and $\tilde{q} = [\dots, \tilde{q}_{ij}, \dots] \in \mathbb{R}^{ml}, \ \ (i, j) \in E$.

- Error defined as,

  $$e_{ij} = \|\tilde{q}_{ij}\| - d_{ij} \tag{37}$$

  we need $e_{ij} \to 0$ as $t \to \infty, \ \ i, j \in V^*$ ($i, j \in E^*$ is enough).

  Error dynamics is,

  $$\begin{aligned} \dot{e}_{ij} &= \frac{d}{dt}(\sqrt{\tilde{q_{ij}}^\top \tilde{q_{ij}}} \\ &= \frac{\tilde{q}_{ij}(u_i - u_j}{e_{ij} + d_{ij}} \end{aligned} \tag{38}$$

- Let

  $$z_{ij} = \|\tilde{q}_{ij}\|^2 - d_{ij}^2 \tag{39}$$

  can be written as

  $$z_{ij} = e_{ij}(e_{ij} + 2d_{ij}) \tag{40}$$

  it can be shown that $z_{ij} = 0$ if and only if $e_{ij} = 0$.

- Consider Lyapunov function candidate as,

  $$W(e) = \frac{1}{4} z^\top z \tag{41}$$

- Differentiation of W is

$$\begin{aligned}\dot{W} &= \sum_{(i,j)\in E^*} e_{ij}(e_{ij}+2d_{ij})\tilde{q}_{ij}^\top(u_i-u_j) \\ &= z^\top R(\tilde{q})u\end{aligned} \tag{42}$$

where $u = [u_1, \dots, u_m] \in \mathbb{R}^{nm}$ are control inputs.

Lemma: For nonnegative constants c and $delta$, the level set $W(e) \leq c$ is equivalent to $\Psi(F, F*) \leq \delta$.

Main Theorem Consider a the formation $F(t) = (G^*, q(t))$, and let the initial conditions of the error dynamics be such that $e(0) \in \Omega_1 \cap \Omega_2$ where,

$$\begin{aligned}\Omega_1 &= \{e \in \mathbb{R}^l \mid \Psi(F, F*) \leq \delta\} \\ \Omega_2 &= \{e \in \mathbb{R}^l \mid \mathrm{dist}(q, \mathrm{ISO}(F^*)) < \mathrm{dist}(q, \mathrm{Amb}(F^*))\}\end{aligned} \tag{43}$$

and δ is sufficiently small positive constant. The control law

$$u = u_a := -k_v R^\top(\tilde{q})z \tag{44}$$

where $k_v > 0$ is control gain, $e = 0$ is exponentially stable. The control law in terms of individual agents,

$$u_i = -k_v \sum_{j\in\mathcal{N}_i(E^*)} \tilde{q}_{ij} z_{ij} \; \forall \;\; i = 1, \dots, n \tag{45}$$

using the fact that,

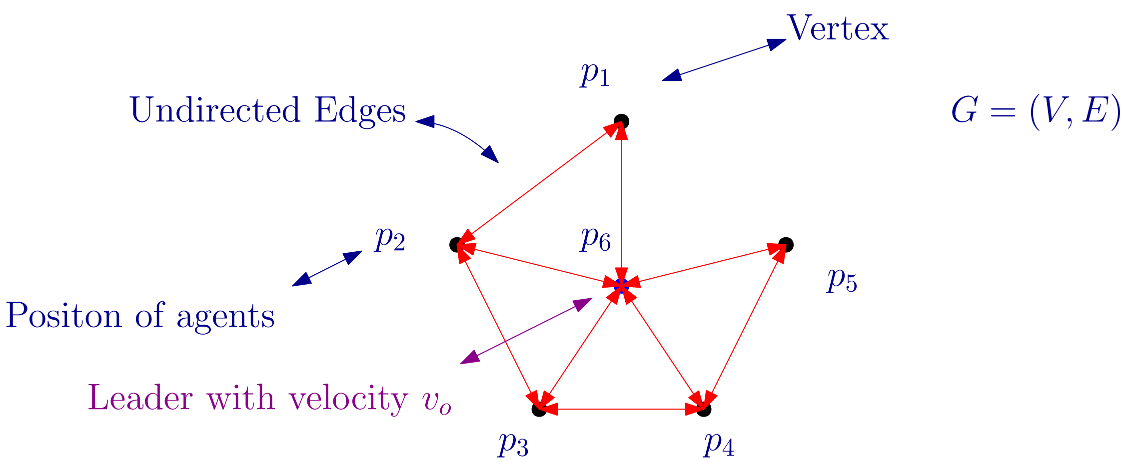
$$\begin{aligned}\tilde{q}_{ij} &:= \tilde{q}_{ij}^0 = \mathcal{R}_i^0 \tilde{q}_{ij}^i \\ u_i &:= u_i^0 = \mathcal{R}_i^0 u_i^i\end{aligned} \tag{46}$$

We can also calculate the control in its own coordinate as

$$u_i^i = -k_v \sum_{j\in\mathcal{N}_i(E^*)} \tilde{q}_{ij}^i z_{ij} \tag{47}$$

1.4.4 Formation Maneuvering

Now the formation will follow a trajectory with given linear and angular velocity.



Theorem for formation Maneuvering:

Consider the formation $F(t) = (G^*, q(t))$ with the initial conditions on e(0) given in previous theorem . Then, the control $u = u_a + v_d$ where u_a as defined earlier, $v_d = [v_d 1, \ldots, v_{dn}] \in R^{3n}$ is the desired rigid body velocity specified by $v_d i = v_0 + \omega_0 \times \tilde{q}_{in}$, $i = 1, \ldots, n$, $v_0(t) \in \mathbb{R}$ denotes the desired translation velocity for the formation, $\omega_0(t) \in R_3$ is the desired angular velocity, renders e = 0 exponentially stable.

Control in terms of individual agents,

$$u_i = -k_v \sum_{j \in \mathcal{N}_i(E^*)} \tilde{q}_{ij} + v_0 + \omega_0 \times \tilde{q}_{in}, \quad i = 1, \ldots, n \tag{48}$$

1.4.5 Double integer model

$$\begin{aligned} \dot{q}_i &= v_i \\ \dot{v}_i &= u_i, \quad i = 1, \ldots, n \end{aligned} \tag{49}$$

1.5 ROS simulations

Now I am able to fly crazyflie in gazebo (ROS simulation environment with physics engine) without any problems, I need to implement any formation control scheme on it.

Have to incorporate the OMNET++ for the realistic channel creation in simulation [9].

References

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

[2] B. D. Anderson, B. Fidan, C. Yu, and D. Walle, "Uav formation control: Theory and application," in *Recent advances in learning and control.* Springer, 2008, pp. 15–33.

[3] Y. Kuriki and T. Namerikawa, "Formation control of uavs with a fourth-order flight dynamics," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 6706–6711.

[4] D. Paley, N. E. Leonard, and R. Sepulchre, "Collective motion: Bistability and trajectory tracking," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, 2004, pp. 1932–1937.

[5] E. W. Justh and P. S. Krishnaprasad, "Equilibria and steering laws for planar formations," *Systems and Control Letters*, vol. 52, no. 1, pp. 25–38, may 2004.

[6] F. Yacef, N. Rizoug, L. Degaa, O. Bouhali, and M. Hamerlain, "Trajectory optimisation for a quadrotor helicopter considering energy consumption," in *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2017, pp. 1030–1035.

[7] L. R. G. Carrillo, A. E. D. López, R. Lozano, and C. Pégard, *Quad rotorcraft control: vision-based hovering and navigation.* Springer Science & Business Media, 2012.

[8] F. Yacef, N. Rizoug, O. Bouhali, and M. Hamerlain, "Optimization of energy consumption for quadrotor uav," in *Proceedings of the International Micro Air Vehicle Conference and Flight Competition (IMAV), Toulouse, France*, 2017, pp. 18–21.

[9] V. Matena, T. Bures, I. Gerostathopoulos, and P. Hnetynka, *Experimenting with Adaptation in Smart Cyber-Physical Systems: A Model Problem and Testbed.* Singapore: Springer Singapore, 2019, pp. 149–169. [Online]. Available: https://doi.org/10.1007/978-981-13-2185-6_7