Emergency Aerial Catch with Cooperative Drones: Decentralized Force-Consensus Adaptive Control for Human Rescue

Luca Vignola, Alessandro Wiget, Riccardo Ghetti
D-MATH - ETH ZÜRICH

***Abstract*— We extend a distributed control framework for a team of quadcopters collaboratively manipulating a deformable net, modeled as a network of mass-spring-dampers, to catch a falling body. Our approach incorporates existing adaptive force-consensus strategies to estimate the unknown payload's mass in real time, allowing smooth tracking of desired force trajectories. We then state and prove stability guarantees for our framework. Finally, through numerical simulation, we address practical challenges such as drone reconfiguration after a failure, and passenger safety during the aerial catch maneuver.**

I. Introduction

In emergency scenarios such as fires in high-rise buildings, jumping from a window may become the only escape option for trapped individuals. In such life-threatening situations, rapid deployment of rescue tools becomes essential.

Inspired by this challenge, we propose a novel aerial system that could assist firefighters in catching falling people using a net carried by a team of quadcopters. Moreover, to ensure real-time responsiveness we design such a system using a distributed control architecture, reducing the computational overhead.

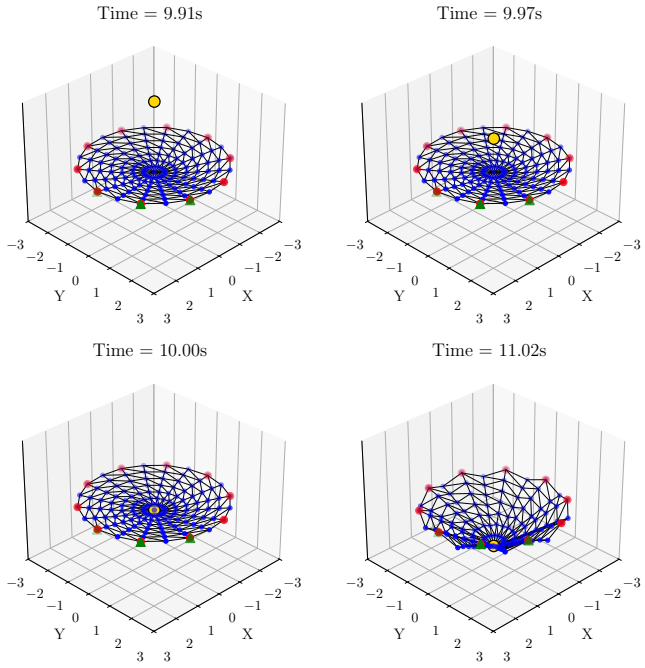


Fig. 1: Snapshots of the simulation of a falling body (yellow ball) on the net (blue dots) carried by the drones (red dots)

Work on flexible payloads has primarily focused on small deformations in structured geometries such as rings [1] and cables [2], [3], and [4]. In the first paper in particular, a centralized LQR controller was implemented, and the model relied on a zero-latency estimator. In [2], the cloth is approximated by a set of catenary curves starting at the Crazyflie drones and connecting at a vertex below the center of mass of the swarm. It should be noted that this approximation of a hanging cloth holds only under certain stable conditions described by the authors.

The approaches in [3] and [4] address towing a cable, which is either approximated as a rigid body or as a set of rigid links connected by spherical joints. It is also important to note that these two approaches do not solve the communication issues between the quadrotors. The former uses an LQR controller, while the latter defines a leader and a follower control schemes, effectively separating the control task between matching a trajectory and counteracting deviations.

On avoiding communication to tow a payload, [5] proposes an approach that introduces an SE(3) controller to follow a predetermined tracking objective. However, this setup requires the *centro-symmetry* assumption, which will not be needed for our proposed controller. We must acknowledge, however, that we will employ a variant of the trajectory optimization problem presented in [5] in Section V-B, where we will minimize the maximum jerk for passenger comfort.

Other approaches to solving the task of lifting an object with multiple drones are presented in [6], where the problem is cast as an SOCP optimization problem and solved centrally, and in [7] and [8], where the lifted object is approximated as a point mass and an LQR controller is used. The task presented in [7] is similar to the problem of catching a falling person from a building; however, the weight of the ball is negligible in that setting. The same objective of catching a heavy object was similarly tackled in [8], where tests were conducted for a varying payload.

Finally, [9] provides a framework to cooperatively manipulate a flexible payload through a consensus on the masses. However, the payload is approximated as a rigid body. The same approach has also been applied in the presence of environmental disturbances in [10]. Moreover, for the same task of estimating the payload mass, an attempt was made in [11] using a neural network instead of relying on consensus among the drones carrying the payload.

The remainder of this paper is structured in the following

way: in Section II, we formulate our model of the system, extending the one in [9]. In Section III, we start with the controller introduced in [9] and enhance it to allow smooth tracking of desired values for the lateral forces. Then, in Section IV, we prove the asymptotic stability of this extended closed-loop system. Finally, in Section V, we show the capabilities of our framework in numerical simulation.

II. Problem Formulation

We consider a flexible net, modeled by a network of M point masses interconnected via spring-dampers elements. The masses are arranged in concentric circular layers. A team of N quadcopters is connected to the net by attaching each drone to a mass located on the outermost ring.

It is worth noting that, for the purposes of our stability analysis, the specific geometric arrangement of the nodes is not critical. The only requirement is that the underlying graph representing the mass-spring-damper network remains connected.

A. Dynamics

Let $x_i \in \mathbb{R}^3$ and $a_i \in \mathbb{R}^3$ be the position of drone i and its connection node respectively, in an inertial frame. Exactly as in [9], we assume that the reaction force between each drone and its corresponding node in the net is the gradient of a positive definite function of their relative displacement, i.e.

$$z_i{=}x_i - a_i \quad i = 1, \dots, N \tag{1}$$
$$f_i{=}\nabla P_i(z_i) \tag{2}$$

where $P_i(z_i)$ is assumed to satisfy:

$$P_i(z_i) = 0 \;\Leftrightarrow\; z_i = 0 \tag{3}$$
$$\nabla P_i(z_i) = 0 \;\Leftrightarrow\; z_i = 0 \tag{4}$$

Let c_j be the 3D positions of the nodes of the net with $j = 1, \dots, M$; We note that $\{a_i\}_{i=1}^N \subseteq \{c_j\}_{j=1}^M$.
Let m_i be the mass of drone i and m_j be the mass of a node of the net. We define M_c as the total mass of the net, that is, $M_c = \sum_{j=1}^M m_j$.
The translational dynamics of each node j can be written in a compact way as follows:

$$m_j\ddot{c}_j{=}f_{\text{int}_j} + \sum_{i=1}^{N} 1_{\{c_j=a_i\}} f_i - m_j g e_3 \tag{5}$$

where g is the gravitational acceleration, $1_{\{*\}}$ is the indicator function, used to take into account the reaction force f_i in case point j is one of the connection nodes, and e_3 is the unit vector $[0, 0, 1]^T$. Additionally:

$$f_{\text{int}_j} = \sum_{k \in \mathcal{N}_j} f_{jk} \tag{6}$$

where $\mathcal{N}_j$ is the set of nodes connected to j via the spring-damper elements and f_{jk} represents the interaction force transmitted through them between node j and k.

The dynamics of drone i, can instead be written as:

$$m_i\ddot{x}_i = F_i - f_i - m_i g e_3 \tag{7}$$

where F_i represents the force applied by agent i, which we assume to be our control input.

B. Control Objective

We consider the objective of controlling the drones in a distributed fashion, in order to track a constant velocity reference v_d and a set of desired contact forces f_i^d.
From (5), a constant velocity implies:

$$\sum_{i=1}^{N} f_i^d\, 1_{\{a_i=c_j\}} + f_{\text{int}_j}^d{=}m_j g e_3, \quad \forall j = 1, \dots, M \tag{8}$$

Then, we can take the sum over all the nodes of the net and note that for connected mass-spring-damper systems we have, by the action-reaction principle:

$$\sum_j f_{\text{int}\,j} = \sum_j \sum_{k \in \mathcal{N}_j} f_{jk} = 0 \tag{9}$$

As a result, the f_i^d forces have to satisfy:

$$\sum_{i=1}^{N} f_{i,x}^d = 0, \quad \sum_{i=1}^{N} f_{i,y}^d = 0 \quad \text{and} \quad \sum_{i=1}^{N} f_{i,z}^d = M_c g \tag{10}$$

Exactly as in [9], we additionally assume that for any f_i^d there exists a z_i^d such that:

$$f_i^d = \nabla P_i(z_i^d), \quad \text{and} \quad \nabla^2 P_i(z_i^d) \succ 0 \tag{11}$$

Finally, for efficiency reasons, we want each drone to share the unknown weight of the net equally, i.e.:

$$f_{i,z}^d = \frac{M_c}{N} \tag{12}$$

Importantly, the generality of this framework allows us not only to manipulate the net, but also any mass that could be added to it. Indeed, we can model the falling body as a point of unknown mass m_b and assume its impact with the net to behave as an anelastic collision with a single node j^* of the net. Therefore, while the feedback law is running, we can simply update:

$$\dot{c}_{j^*} \leftarrow \frac{\dot{c}_{j^*} m_{j^*} + v_b m_b}{m_{j^*} + m_b} \tag{13}$$
$$m_{j^*} \leftarrow m_{j^*} + m_b \tag{14}$$

This way, by only changing the total mass to be estimated, we can solve the problem of catching a falling body within the same framework, which was the main motive behind our paper.

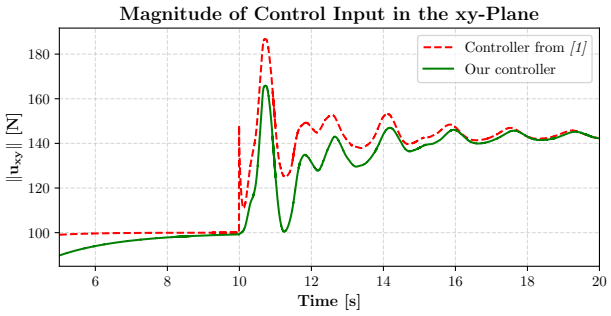


Fig. 2: Control input norms in the xy-plane, during a change of reference in the desired lateral force f^d_{xy} at time $t = 10s$. We compare our proposed enhanced controller to the one presented in [9], to show the achieved smoother tracking. The data refers to *Agent3* of the numerical experiment in section V-A

III. Method

A. Consensus Law

As in [9], we use consensus to update the estimate $\hat{M}_{c,i}$ of each drone i of the payload's unknown mass:

$$\dot{\hat{M}}_{c,i} = -\lambda_i(\dot{x}_i - v_d)^\top e_3 + \sum_{j\in\mathcal{N}_i} \lambda_i(\hat{M}_{c,j} - \hat{M}_{c,i}) \quad (15)$$

The first term $\hat{\beta}_i = -\lambda_i(\dot{x}_i - v_d)^\top e_3$ allows each drone to adapt its estimate based on its own vertical velocity tracking error, while the second term enforces consensus. At equilibrium, this guarantees that the estimated mass is distributed equally among the agents.
We can then stack all the equations using $\hat{\beta} = [\hat{\beta}_1, \hat{\beta}_2, ...]^T$ and $\widehat{M}_c = [\hat{M}_{c,1}, \hat{M}_{c,2}, \dots]^T$ and introduce $\tilde{M}_c = \widehat{M}_c - \frac{1}{N} M_c \mathbf{1}_N$. Finally, we can rewrite the update in (15) as:

$$\dot{\widehat{M}}_c = \hat{\beta} - \Lambda L \tilde{M}_c \quad (16)$$

where $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\}$, and $L \in \mathbb{R}^{N\times N}$ is the Laplacian matrix of the communication graph connecting the drones, defined as:

$$\ell_{ij} = \begin{cases} |\mathcal{N}_i| & \text{if } i = j \\ -1 & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$$

We will refer to this form of the update in the proof in section IV.

B. Force Control Law

In this work, we assume that the drones can directly control the total force F_i they exert. This abstraction allows us to bypass the complexity of low-level attitude and thrust control typically required to coordinate individual rotors. Under this assumption, we can directly build upon the following high-level control law, originally introduced in [9]:

$$F_i = -\Gamma_i(\dot{x}_i - v_d) + \hat{f}^d_i + m_i g e_3 \quad (17)$$

where $\Gamma_i \succ 0$ and $\hat{f}^d_i$ is the current estimate of the contact force f_i. The $\hat{f}^d_i$ z-component is constructed as

$$\hat{f}^d_{i,z} = \hat{M}_{c,i} g e_3 \quad (18)$$

Now, note that in [9] the lateral components of the force estimates were set to predefined constant values (the desired ones), i.e. $\hat{f}^d_{i,x} = f^d_{i,x}$ and $\hat{f}^d_{i,y} = f^d_{i,y}$.
Instead, we allow $\hat{f}^d_{i,x}, \hat{f}^d_{i,y}$ to be part of the state space, following their own dynamics (see below). This enables us to achieve smooth tracking of the references of desired lateral forces, preventing sudden input spikes that might damage the drones in the case of dynamic maneuvers (see Figure 2 and the reconfiguration maneuver described in section V-A). For this, we propose the following update law:

$$\begin{aligned} \dot{\hat{f}}^d_{i,x} &= -\nu_i(\hat{f}^d_{i,x} - f^d_{i,x}) - \xi_i^T e_1 \\ \dot{\hat{f}}^d_{i,y} &= -\nu_i(\hat{f}^d_{i,y} - f^d_{i,y}) - \xi_i^T e_2 \end{aligned} \quad (19)$$

where $\nu_i > 0\ \forall i$, ξ_i is defined as below and e_1, e_2 have similar definitions to e_3. The first term enforces tracking of the desired reference, while the second term will be needed to prove the convergence and stability of the system.
We are finally ready to state and prove our main result.

IV. Results

Theorem 1 (Stability): The decentralized adaptive control law in (17) asymptotically stabilizes the equilibrium of (5),(7), (15) and (19), given by

$$\begin{aligned} \mathcal{E}^* = \Big\{ \Big(\dot{x}_i, \dot{c}_j, f_i, \hat{f}^d_i, \hat{M}_{c,i}\Big) \ \Big|\ \dot{x}_i = v_d, \\ \dot{c}_j = v_d,\ f_i = \hat{f}^d_i = f^d_i,\ \hat{M}_{c,i} = \frac{M_c}{N}, \Big\} \end{aligned} \quad (20)$$

Proof:
Let ξ_i, ξ_j, ξ_{a_i} be defined as:

$$\begin{aligned} \xi_i &:= \dot{x}_i - v_d, \quad \forall i = 1, \dots, N \\ \xi_j &:= \dot{c}_j - v_d, \quad \forall j = 1, \dots, M \\ \xi_{a_i} &:= \dot{c}_j - v_d, \quad \forall c_j = a_i \quad \text{for some} \quad i \end{aligned}$$

Where v_d refers to the constant velocity we want to track, as a 3-dimensional vector.

We now start from the Lyapunov function proposed in [9], with the addition of a term representing the kinetic energy of the nodes of the net, to account for our newly introduced states.

$$\begin{aligned} V_1 = &\sum_{i=1}^{N} \left[P_i(z_i) - P_i(z_i^d) - (f_i^d)^T(z_i - z_i^d)\right] \\ &+ \frac{1}{2}\sum_{i=1}^{N} \xi_i^T m_i \xi_i + \frac{1}{2}\sum_{j=1}^{M} \xi_j^T m_j \xi_j \end{aligned}$$

To ensure positive definiteness of V_1, the assumption on P_i to be strictly convex around z_i^d is enough. This can be seen

by using the Taylor-Lagrange theorem centered on z_i^d. Differentiating V_1 with respect to time we get:

$$\begin{aligned}\dot{V}_1=&\sum_{i=1}^{N}(f_i-f_i^d)^T\dot{z}_i+\sum_{i=1}^{N}\xi_i^T m_i\ddot{x}_i+\sum_{j=1}^{M}\xi_j^T m_j\ddot{c}_j\\ =&\sum_{i=1}^{N}(f_i-f_i^d)^T\xi_i-\sum_{i=1}^{N}(f_i-f_i^d)^T\xi_{a_i}\\ &-\sum_{i=1}^{N}\xi_i^T\Gamma_i\xi_i+\sum_{i=1}^{N}\xi_i^T(\hat{f}_i^d-f_i)+\sum_{j=1}^{M}\xi_j^T m_j\ddot{c}_j\\ =&\sum_{i=1}^{N}(\hat{f}_i^d-f_i^d)^T\xi_i-\sum_{i=1}^{N}\xi_i^T\Gamma_i\xi_i\\ &-\sum_{i=1}^{N}(f_i-f_i^d)^T\xi_{a_i}+\sum_{j=1}^{M}\xi_j^T m_j\ddot{c}_j \end{aligned}\tag{21}$$

where in the second equality, we substitued the closed-loop drone dynamics $m_i\ddot{x}_i=F_i-f_i-m_ige_3=-\Gamma_i\xi_i+\hat{f}_i^d-f_i$ and $\dot{z}_i=\xi_i-\xi_{a_i}$. The first two terms are similar to the ones in [9]. However, as mentioned before, we do not consider the lateral force estimates $\hat{f}_i^d$ to be set a priori to the desired values, but rather we allow them to evolve according to (19). Thus, the first two components of the first term will be non-zero in our case:

$$\begin{aligned}\xi_i^T\tilde{f}_i^d:=\xi_i^T(\hat{f}_i^d-f_i^d)=&\xi_i^T\begin{pmatrix}\hat{f}_{i,x}^d-f_{i,x}^d\\ \hat{f}_{i,y}^d-f_{i,y}^d\\ \tilde{M}_{c,i}g\end{pmatrix}\\ =&\begin{pmatrix}\xi_{i,xy} & \xi_{i,z}\end{pmatrix}\begin{pmatrix}\hat{f}_{i,xy}^d-f_{i,xy}^d\\ \tilde{M}_{c,i}g\end{pmatrix}\end{aligned}$$

To cancel the first components, we introduce the following positive definite term in the Lyapunov function:

$$V_3=\sum_{i=1}^{N}\frac{1}{2}\|\hat{f}_{i,xy}^d-f_{i,xy}^d\|^2$$

Differentiating and using (19) we get:

$$\begin{aligned}\dot{V}_3=&\sum_{i=1}^{N}(\dot{\hat{f}}_{i,xy}^d)^\top(\hat{f}_{i,xy}^d-f_{i,xy}^d)\\ =&\sum_{i=1}^{N}[-\nu_i\|\hat{f}_{i,xy}^d-f_{i,xy}^d\|^2-\xi_{i,xy}^\top(\hat{f}_{i,xy}^d-f_{i,xy}^d)]\end{aligned}$$

Indeed, adding this to the derivative of V_1 cancels out the x and y components of the first term and introduces a negative definite term. The third component will be dealt with later, by introducing a second Lyapunov function V_2 and proceeding analogously to [9].
Resuming from (21), let us analyze the last two summations. By (5), we can substitute $m_j\ddot{c}_j$ to obtain:

$$\begin{aligned}&-\sum_{i=1}^{N}(f_i-f_i^d)^T\xi_{a_i}+\sum_{j=1}^{M}\xi_j^T m_j\ddot{c}_j=\\ =&-\sum_{i=1}^{N}(f_i-f_i^d)^T\xi_{a_i}+\sum_{j=1}^{M}\xi_j^T f_{\text{int}_j}\\ &+\sum_{i=1}^{N}\xi_{a_i}^T f_i-\sum_{j=1}^{M}\xi_j^T m_j ge_3\\ =&\sum_{i=1}^{N}\xi_{a_i}^T f_i^d+\sum_{j=1}^{M}\xi_j^T(f_{\text{int}_j}-m_jge_3)\end{aligned}$$

Separating the addends associated with the a_i nodes from the rest of the net we obtain:

$$\begin{aligned}=&\sum_{i=1}^{N}\xi_{a_i}^T(f_i^d+f_{\text{int}_i}-m_jge_3)+\sum_{c_j\neq a_i}\xi_{c_j}^T(f_{\text{int}_j}-m_jge_3)\\ =&\sum_{i=1}^{N}\xi_{a_i}^T(f_i^d\pm f_{\text{int}_i}^d+f_{\text{int}_i}-m_jge_3)\\ &+\sum_{c_j\neq a_i}\xi_{c_j}^T(f_{\text{int}_j}\pm f_{\text{int}_j}^d-m_jge_3)\\ =&\sum_{i=1}^{N}\xi_{a_i}^T(f_{\text{int}_i}-f_{\text{int}_i}^d)+\sum_{i=1}^{N}\xi_{a_i}^T(f_i^d+f_{\text{int}_i}^d-m_jge_3)\\ &+\sum_{c_j\neq a_i}\xi_{c_j}^T(f_{\text{int}_j}-f_{\text{int}_j}^d)+\sum_{c_j\neq a_i}\xi_{c_j}^T(f_{\text{int}_j}^d-m_jge_3)\end{aligned}$$

By applying (8), the terms that only contain the desired forces cancel, obtaining:

$$\begin{aligned}=&\sum_{j=1}^{M}\xi_j^T(f_{\text{int}_j}-f_{\text{int}_j}^d)\\ =&\sum_{j=1}^{M}(\dot{c}_j-v_d)^T(f_{\text{int}_j}-f_{\text{int}_j}^d)\\ =&\sum_{j=1}^{M}\dot{c}_j^T(f_{\text{int}_j}-f_{\text{int}_j}^d)-v_d^T(\sum_{j=1}^{M}f_{\text{int}_j}-\sum_{j=1}^{M}f_{\text{int}_j}^d)\\ =&\sum_{j=1}^{M}\dot{c}_j^T(f_{\text{int}_j}-f_{\text{int}_j}^d)\end{aligned}$$

where we used (9) in the last step.
Now, let $G=(V,E)$ be the undirected graph describing the configuration of our net, where V is the set of nodes and E is the set of edges connecting them (representing the physical spring-dampers between them). We can then rewrite this summation of the internal forces over all nodes, as a sum over all the edges of G:

$$\begin{aligned}&=\sum_{\{j,k\}\in E}\left(\dot{c}_j^T(f_{jk}-f_{jk}^d)+\dot{c}_k^T(f_{kj}-f_{kj}^d)\right)\\ &=\sum_{\{j,k\}\in E}(\dot{c}_j-\dot{c}_k)^T(f_{jk}-f_{jk}^d)\end{aligned}\tag{22}$$

Here, f_{jk} represents the interaction between j and k, and the last equality holds by the action-reaction principle, which guarantees $f_{jk} = -f_{kj}$.
We can finally express f_{jk} to explicitly represent the spring-damper interaction, i.e.:

$$\begin{aligned} f_{jk} - f^d_{jk} =& -k_{jk}(c_j - c_k) + k_{jk}(c^d_j - c^d_k) \\ & -\mu_{jk}(\dot{c}_j - \dot{c}_k) + \mu_{jk}(\dot{c}^d_j - \dot{c}^d_k) \\ =& -k_{jk}(c_j - c_k - l^d_{jk}) - \mu_{jk}(\dot{c}_j - \dot{c}_k) \end{aligned}$$

where $l^d_{jk} = c^d_j - c^d_k$ is a constant vector representing the desired relative positions, and the desired relative velocities are $\dot{c}^d_j = \dot{c}^d_k = v_d$.
Here, $k_{jk} > 0$ and $\mu_{jk} > 0 \; \forall (j,k) \in E$ are the elastic and damping coefficients between j and k, respectively.
Substituting this back in (22):

$$\begin{aligned} =& - \sum_{\{j,k\} \in E} (\dot{c}_j - \dot{c}_k)^T k_{jk}(c_j - c_k - l^d_{jk}) \\ & - \sum_{\{j,k\} \in E} (\dot{c}_j - \dot{c}_k)^T \mu_{jk}(\dot{c}_j - \dot{c}_k) \end{aligned}$$

Now, to compensate for the cross-terms between the positions and their derivatives, we introduce another positive definite term in the Lyapunov function, representing the potential energies of the springs:

$$\begin{aligned} V_4 =& \frac{1}{2} \sum_{\{j,k\} \in E} (c_j - c_k - l^d_{jk})^T k_{jk}(c_j - c_k - l^d_{jk}) \\ \dot{V}_4 =& \sum_{\{j,k\} \in E} (c_j - c_k - l^d_{jk})^T k_{jk}(\dot{c}_j - \dot{c}_k) \end{aligned}$$

and this derivative cancels out with the cross terms.
Finally, we take $V_2 = \frac{g}{2} \tilde{M}_c^T \Lambda^{-1} \tilde{M}_c$ from [9], and construct a Lyapunov function by summing all the terms introduced above:

$$\begin{aligned} W =& V_1 + V_2 + V_3 + V_4 \\ \dot{W} =& - \sum_{i=1}^{N} \xi_i^T \Gamma_i \xi_i - g \tilde{M}_c^T L \tilde{M}_c \\ & - \sum_{\{j,k\} \in E} (\dot{c}_j - \dot{c}_k)^T \mu_{jk}(\dot{c}_j - \dot{c}_k) \\ & - \sum_{i=1}^{N} \nu_i \| \hat{f}^d_{i,xy} - f^d_{i,xy} \|^2 \leq 0 \end{aligned}$$

Note that W is only negative semi-definite since the Laplacian matrix always has a non-trivial kernel. Therefore, with this we can only conclude that the equilibrium is stable.

To prove asymptotic stability we have to invoke LaSalle's invariance principle. Similarly to the proof in [9], let M be the largest invariant set where $\dot{W} = 0$. In M, $\xi_i = 0$, $L\tilde{M}_c = 0$, $\dot{c}_j = \dot{c}_k \; \forall \{j,k\} \in E$ and $\hat{f}^d_{i,xy} = f^d_{i,xy}$. By assuming the communication graph between the drones to be connected, the null space of L is spanned by the vector 1_N, so $L\tilde{M}_c = 0 \implies \tilde{M}_c = 1_N \alpha$ for some scalar α. Since $\tilde{M}_c = \hat{M}_c - \frac{1}{N} M_c 1_N = 1_N \alpha$, we conclude that $\hat{M}_{c,i} = \hat{M}_{c,j} \; \forall i,j$.

Next, we note that $\xi_i = 0$ implies $\dot{\hat{M}}_{c,i} = 0 \; \forall i$, $\dot{x}_i = v_d$ and $\ddot{x}_i = 0$. From the closed-loop dynamics obtained from (7) and (17), we obtain $\hat{f}^d_i = f_i$. But $\hat{f}^d_i = \hat{M}_{c,i} g e_3 + f^d_{i,xy}$, so f_i must be constant, which means that $z_i = x_i - a_i$ is constant $\forall i$. Therefore, $\dot{x}_i = \dot{a}_i$ and together with the fact that all node points have the same velocity (i.e. $\dot{c}_j = \dot{c}_k \quad \forall j,k$ since the graph of the net is connected), this means that all drones and node points have the same constant velocity v_d and $\ddot{c}_j = 0 \quad \forall j$. Using (5) and summing over all nodes:

$$\begin{aligned} 0 =& \sum_{j=1}^{M} m_j \ddot{c}_j = \sum_{j=1}^{M} [f_{\text{int}_j} + \sum_{i=1}^{N} 1_{\{c_j = a_i\}} f_i - m_j g e_3] \\ \implies & \sum_{j=1}^{M} f_{\text{int}_j} + \sum_{i=1}^{N} f_i = \sum_{j=1}^{M} m_j g e_3 \\ \implies & M_c g e_3 = \sum_{j=1}^{M} m_j g e_3 = \sum_{i=1}^{N} f_i = \sum_{i=1}^{N} \hat{f}^d_i = \sum_{i=1}^{N} \hat{M}_{c,i} g e_3 \\ \implies & \hat{M}_{c,i} = \frac{1}{N} M_c \quad \forall i \end{aligned}$$

where in the second to last step we used that the sum of the desired lateral forces is zero (see (10)).

Finally, note that by the reasoning above the set M is indeed invariant, thus applying LaSalle's invariance principle proves asymptotic stability to the desired equilibrium (20). □

V. Numerical Examples and Simulations

In this section, we showcase the performance of our theoretical framework through numerical simulation. All the code was implemented in Python. We used the *solve_ivp* function in the SciPy library to integrate the continuous-time dynamics of the system. The implementation and all the parameters used in the numerical experiments can be found at our GitHub repository .

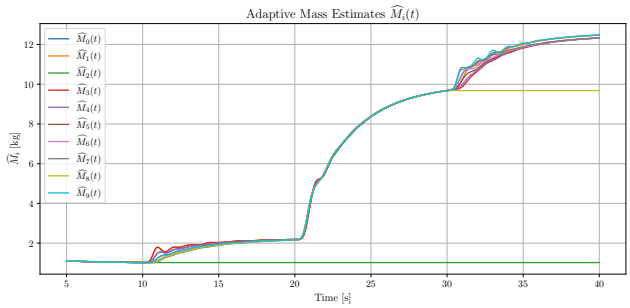
Throughout this entire section, we consider the reaction force f_i between a drone and its corresponding node to be elastic, i.e. we model the connection as a spring with known elastic constant:

$$f_i = k_i z_i$$

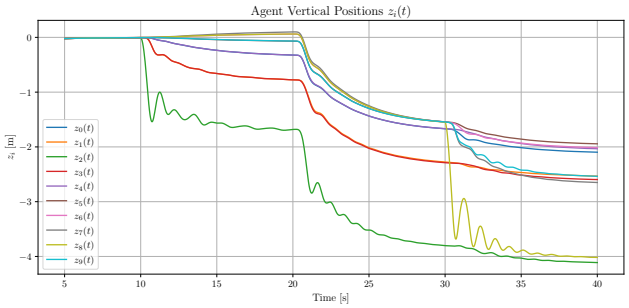
In our experiments, we set $k_i = 1000 \frac{N}{m}$

A. Reconfiguration in fault scenario

In the first scenario, we consider the fault of a drone during operation, under the assumption that the communication graph follows a ring topology. To adapt our theoretical framework to this situation, we assume that the two neighbors of the failed drone are able to detect the fault and are aware of the reference forces that the failed agent was tracking. In particular, we assume that each of them has to compensate for exactly half of the force vector that was sustained by the



(a) Estimated adaptive masses $\hat{M}_{c,i}(t)$ after reconfiguration.



(b) Vertical positions $z_i(t)$ of all agents.

Fig. 3: Evolution of the system during a *reconfiguration scenario* with $N = 10$ drones. *Agent*$_2$ dies at $t = 10s$, a person of 70 kg falls on the net at $t = 20s$, and *Agent*$_8$ faults at $t = 30s$: (a) The mass estimates still converge to consensus after the first drone fail, whereas they converge to two different values after the second fault disconnects the graph; (b) The vertical displacement of the agents shows that the system is able to adapt to the unknown load. Note that the drones reconfigure at different heights in order to share the weight of the load equally.

dead drone, in order to maintain the equilibrium of forces in the xy plane. Therefore, in the case of drone i breaking, the references of neighbors $i-1, i+1$ are updated to:

$$f^d_{x,i\pm1} \leftarrow f^d_{x,i\pm1} + \tfrac{1}{2} f^d_{x,i}$$
$$f^d_{y,i\pm1} \leftarrow f^d_{y,i\pm1} + \tfrac{1}{2} f^d_{y,i}$$

Despite one failure, the graph remains connected, ensuring that consensus is still achieved by *Theorem* 1. Moreover, our enhanced controller also guarantees asymptotic stability given any reference of lateral forces. In the case of reconfiguration, without this addition, the lateral forces would be instantly set to the desired ones, causing sudden spikes in the control input and possible damages to the hardware (see Figure 2).

We can then push our algorithm even further by considering a second faulting drone. This causes the communication graph of the drones to split into two connected components. By the proof of *Theorem* 1, asymptotic stability is still achieved for the entire state space, but the masses do not necessarily converge to the same value. In fact, the two resulting disconnected components will separately reach their own consensus such that the sum of all estimates remains equal to the total mass of the payload. This can be observed in Figure 3.a. Moreover, already after the first fault, the z coordinates of the drones will settle at different heights, as a result of the loss of symmetry.

B. Passenger Safety and Comfort

In the second experiment (Figure 4), we tested the performance of our framework in a more realistic scenario, which required us to think about safety and comfort of the individuals in life-threatening situations that require the quadrotors intervention. The idea is to minimize the variations in acceleration (jerk) when slowing down to zero velocity after catching the falling object. To achieve this, we start by assuming that the system is able to estimate through sensors the initial z-coordinate $z_{b,0}$ and speed $v_{b,0}$ of the body in free fall at a certain time. To obtain accurate estimates, filtering algorithms should be used, but they are outside the scope of this paper.

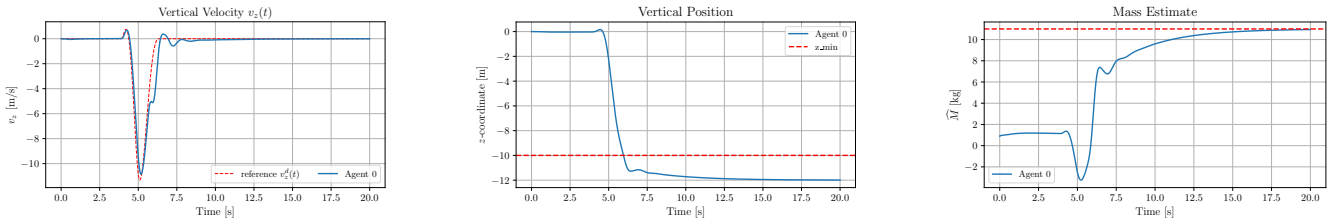
The optimization problem is presented in (24) and is structured in the following way: we restrict our search for an optimal vertical trajectory to the space of fifth-order polynomial functions for the vertical position:

$$z_j(t) = \sum_{i=0}^{5} c_{j,i} t^i \tag{23}$$

Since the main concern while decelerating is passenger safety, we construct two separate trajectory polynomials $z_1(t)$ and $z_2(t)$, representing the vertical position of the net's center before and after impact, which allows us to minimize jerk only in the second part of the trajectory. Finally, the constraints that involve time are discretized over N=50 intervals.

$$\begin{aligned}
\min_{\substack{t_1,t_2>0 \\ c_{j,i}\in\mathbb{R} \\ \forall i\in[5]\ j=1,2}} \quad & \max_{t\in[0,t_2]} |\dddot{z}_2(t)| \\
\text{subject to} \quad & z_1(0) = z_0, \quad \dot{z}_1(0) = v_0 \\
& \ddot{z}_1(0) = a_0,\ z_1(t_1) = z_{b,0} - \tfrac{1}{2} g t_1^2 + v_{b,0} t_1 \\
& \dot{z}_1(t_1) = v_{b,0} - g t_1, \quad \ddot{z}_1(t_1) = -g \\
& z_2(0) = z_1(t_1), \quad \dot{z}_2(0) = \dot{z}_1(t_1) \\
& \ddot{z}_2(0) = \ddot{z}_1(t_1), \quad \dddot{z}_2(0) = \dddot{z}_1(t_1) \\
& z_1(t) \geq z_0 - \Delta z_{max}, \quad \forall t \in [0, t_1] \\
& z_2(t) \geq z_0 - \Delta z_{max}, \quad \forall t \in [0, t_2] \\
& \ddot{z}_1(t) \in [a_{\min}, a_{\max}], \quad \forall t \in [0, t_1] \\
& \ddot{z}_2(t) \in [a_{\min}, a_{\max}], \quad \forall t \in [0, t_2] \\
& \dot{z}_2(t_2) = 0, \quad \ddot{z}_2(t_2) = 0
\end{aligned} \tag{24}$$

First and foremost, we want our system to start at its initial speed and acceleration and for it to reach the object at the point of impact at time t_1 with the same velocity and



(a) (b) (c)

Fig. 4: Simulation with $N = 10$ drones, a net of $M = 141$ node points, a total mass of 10 kg and load of 100 Kg. Fig (a) is the comparison between the optimal velocity reference and the actual speed profile of $Agent_0$. Fig (b) is the evolution of the vertical position of $Agent_0$. The red line represents the constraint imposed in the optimization problem. We see that it is not rispected in practice. Fig (c) is the evolution of the mass estimate of the same agent. In (c), we can see that while following the optimal trajectory for catching the object the estimates lose meaning during the transient. Then, when the reference settles on a constant velocity, convergence is achieved.

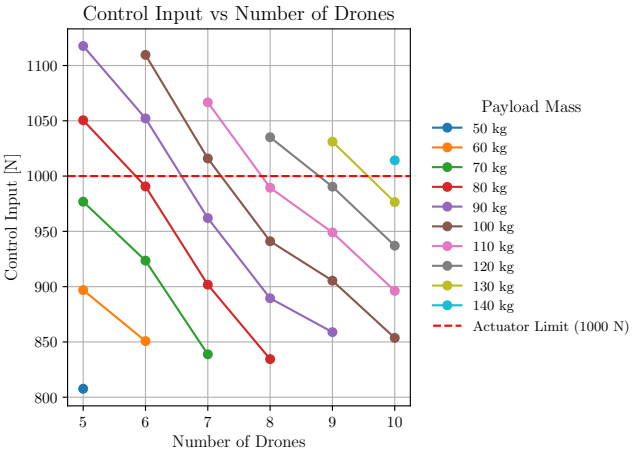


Fig. 5: Plot of the number of drones against the maximum applied control input in the simulation for different values of the body's mass. The body is here assumed to fall at the center of the net.

acceleration, in order to ensure a smooth collision. Then, we impose continuity between the two polynomials and their first three derivatives in time at the moment of contact t_1, to avoid sudden and potentially dangerous variations in the forces perceived by the caught object. Most importantly, we impose a hard constraint on the maximum *vertical clearance* that the planned trajectory is allowed to optimize over, namely Δz_{max}, to avoid touching the ground while slowing down. Note that the program outputs a non-constant velocity reference, so our system will likely experience tracking errors during the catch. For this reason, when specifying this constraint, we need to consider a reasonable safety margin. Finally, we introduce a soft constraint regarding the maximum and minimum allowed acceleration, to partially mitigate extreme solutions found for the optimization problem which might be harmful to the passenger and the drones. Given the optimal trajectory that minimizes the overall jerk in the second part of the reference trajectory, we can then easily obtain the speed profile and pass it to the drones for tracking.

In our current implementation, the program only considers the z-coordinate and is run offline because of its computational complexity, hence this aspect should be further investigated for a real-time implementation.

Finally, it is important to note that the optimizer does not have access to the mass of the falling object when planning the trajectory, as it can only be estimated after the impact. Moreover, in the first moments after the collision, the drones will have a nonzero downward acceleration because of the continuity constraint, resulting in incorrect estimates of the mass. For these reasons, we decided to run the optimization problem only once and to not take into account the mass and velocity of the object during contact.

C. Required number of drones

In our setting, we did not impose any actuator limits, allowing the drones to theoretically be able to track the desired reference of the previous subsection for any value of the mass. However, this is clearly an unreasonable assumption to make in practice; in the following, we provide an experimental study on the dependency between the mass of the payload and the required magnitude of the control input. We assume each drone to have a mass of $m_{drone} = 10$ kg and an actuator limit of $1000\,N$, which are reasonable values for industrial drones. The mass of the net is also assumed to be 10 kg, and the number of point masses that are used to model the external ring of the cloth is chosen accordingly with the number of drones N, to make the configuration symmetric. Finally, we let the mass of the payload vary in the range [50 kg, 140 kg] and simulate the system to figure out the minimum number of drones that is required to track the desired trajectory while remaining within the actuator limits. Figure 5 illustrates the results. From the data obtained, one can already infer that $N_{drones} = 10$ is sufficient to guarantee safety for every mass up to 130 kg.

VI. Conclusions

This work introduces a distributed, adaptive control strategy for cooperative aerial payload transport, leveraging local

mass estimation and dynamic force tracking to achieve a smooth, stable behavior. Our controller accommodates system reconfigurations due to drone faults and ensures safe operation even under sudden changes in the payload mass. The proposed lateral force update mechanism significantly reduces control input spikes, enhancing safety and hardware longevity. Through simulation, we demonstrate the framework's capability to handle realistic scenarios, including object catching with comfort-aware trajectory planning. While the offline optimization used for trajectory generation currently limits real-time deployment, our results suggest strong potential for practical applications. Future work will focus on real-time implementation, integration with onboard sensing and filtering, and extension to more complex payload geometries and dynamic environments.

REFERENCES

[1] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Tokyo, Japan), pp. 3465–3471, 2013.

[2] R. Cotsakis, D. St-Onge, and G. Beltrame, "Decentralized collaborative transport of fabrics using micro-uavs," in *2019 International Conference on Robotics and Automation (ICRA)*, (Montreal, QC, Canada), pp. 7734–7740, 2019.

[3] P. Kotaru and K. Sreenath, "Multiple quadrotors carrying a flexible hose: dynamics, differential flatness and control, the college of engineering and berkeley deep drive.," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8832–8839, 2020. 21st IFAC World Congress.

[4] M. Gassner, T. Cieslewski, and D. Scaramuzza, "Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, (Singapore), pp. 5196–5202, 2017.

[5] Z. Wang, S. Singh, M. Pavone, and M. Schwager, "Cooperative object transport in 3d with multiple quadrotors using no peer communication," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, (Brisbane, QLD, Australia), pp. 1064–1071, 2018.

[6] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," in *Robotics Research* (C. Pradalier, R. Siegwart, and G. Hirzinger, eds.), vol. 70 of *Springer Tracts in Advanced Robotics*, pp. 1–16.

[7] R. Ritz, M. W. Müller, M. Hehn, and R. D'Andrea, "Cooperative quadrocopter ball throwing and catching," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Vilamoura-Algarve, Portugal), pp. 4972–4978, 2012.

[8] M. Doakhan, M. Kabganian, and A. Azimi, "Robust adaptive control for formation-based cooperative transportation of a payload by multi quadrotors," *European Journal of Control*, vol. 69, p. 100763, 2023.

[9] S. Thapa, H. Bai, and J. Acosta, "Cooperative aerial manipulation with decentralized adaptive force-consensus control," *Journal of Intelligent & Robotic Systems*, vol. 97, pp. 171–183, 2020.

[10] K. Klausen, C. Meissen, T. I. Fossen, M. Arcak, and T. A. Johansen, "Cooperative control for multirotors transporting an unknown suspended load under environmental disturbances," *IEEE Transactions on Control Systems Technology*, vol. 28, pp. 653–660, March 2020.

[11] J. E. Sierra-García and M. Santos, "Intelligent control of an uav with a cable-suspended load using a neural network estimator," *Expert Systems with Applications*, vol. 183, p. 115380, 2021.