

Collaborative Transportation Using MAVs via Passive Force Control

Andrea Tagliabue, Mina Kamel, Sebastian Verling, Roland Siegwart and Juan Nieto

Abstract—This paper shows a strategy based on passive force control for collaborative object transportation using Micro Aerial Vehicles (MAVs), focusing on the transportation of a bulky object by two hexacopters. The goal is to develop a robust approach which does not rely on: (a) communication links between the MAVs, (b) the knowledge of the payload shape and (c) the position of grasping point. The proposed approach is based on the master-slave paradigm, in which the slave agent guarantees compliance to the external force applied by the master to the payload via an admittance controller. The external force acting on the slave is estimated using a non-linear estimator based on the Unscented Kalman Filter (UKF) from the information provided by a Visual-Inertial (VI) navigation system. Experimental results (online video [1]) demonstrate the performance of the force estimator and show the collaborative transportation of a 1.2 m long object.

I. INTRODUCTION

MAVs have recently stimulated the creativity of researchers and entrepreneurs as a new tool for delivery of goods (e.g. [2], [3]). Quadcopters and hexacopters have shown to be especially suited for this task, thanks to their ability to navigate in cluttered environments (e.g. [4]) and being able to deliver a payload with high accuracy (e.g. [5]). However, their inherent limited size constrains the range of applications to the transportation of small and light-weight objects.

Collaborative strategies can significantly enhance MAV's transportation capabilities and can provide a cost effective solution with respect to the deployment of a single, more capable MAV [6]. A common approach for collaborative transportation is based on centralized solutions, where a coordinator computes a control action for each of the agents and it shares the command with them (e.g. [7]). A second possible approach is based on distributed control algorithm. In this case, the agents only share a common goal and the individual control actions are calculated based on the grasping point on the payload (e.g. [8]).

Both of the described approaches, however, present some limitations. Centralized approaches rely on the ability of the coordinator to effectively communicate with the agents and this can often be an issue due to the limited robustness of wireless communication networks. Distributed approaches, instead, rely on the knowledge of the relative position between each agent and the payload.

In this paper we overcame these issues by showing a bio-inspired [9] collaborative strategy based on the master-slave paradigm. The task of the master is simply to lift

All the authors are with the Autonomous Systems Lab, Department of Mechanical and Process Engineering, ETH Zurich, Switzerland. {andretag, fmina, sverling, rsiegwart, nietoj}@ethz.ch

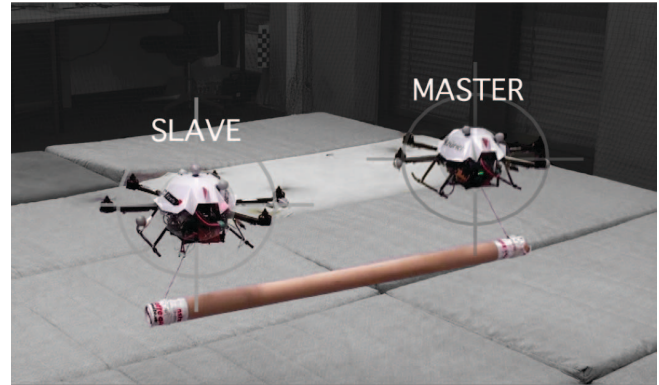


Fig. 1: Experimental setup for collaborative aerial transportation. The master agent is controlled remotely, while the slave provides compliance to the movement of the master using an admittance controller.

the payload and pull it in the desired direction. The slave - which is also attached to the payload - actively guarantees compliance to the master's actions by sensing the force that master applies on the payload and changing its position accordingly. Slave's active compliance is guaranteed through an admittance controller, while external forces are estimated via position, velocity, attitude and angular velocity information using an estimator based on the UKF. The slave agent estimates its pose, velocity and angular velocity using an on-board VI navigation system.

In our setup we assume the following: (a) the payload is big enough to guarantee that the MAVs can autonomously grasp it, (b) both the agents are already attached to the payload, and (c) the attitude of the agents is decoupled from the rotational dynamic of the payload - this is achieved, for example, by connecting vehicle and payload via a rope or a spherical joint (e.g. [10]).

A. Contributions

The main contribution of this work is the novel strategy for collaborative aerial transportation using MAVs. The approach that we propose has the advantage of not relying on: (a) communication links between the MAVs, (b) the knowledge of the payload shape, and (c) the position of grasping point. Additionally, the proposed method is theoretically scalable to more than two agents without further modifications. In this case, one agent should be elected as master, while the remaining should act as slaves.

To effectively validate our approach, we describe the implementation of a force estimator based on the UKF, which handles quaternions, is computationally light, and fast.

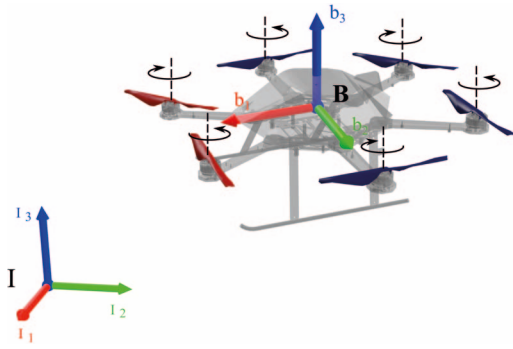


Fig. 2: Coordinate frames defined for this work. I represents the inertial reference frame, while B the MAV reference frame.

It is able to detect a collision with a wall in less than 50 ms and provides an accurate enough force estimate for human-vehicle interaction without relying on an external motion capture system, but using a VI navigation system.

B. Structure of this work

We start by presenting existing approaches and technologies for cooperative aerial transportation in Section II. We then proceed by providing an overview of the control strategy adopted for our approach in Section III and we detail the implementation of the force estimator and admittance controller in Sections IV and V, respectively. Finally, we show the experimental results in Section VI and we summarize the conclusions in Section VII.

II. RELATED WORK

Literature for collaborative transportation via aerial vehicle is rich of examples. Earliest publications are related to collaborative transportation of a single payload using two helicopters (e.g. [11], [12]), but do not provide experimental results. Collaborative transportation using Unmanned Aerial Vehicles (UAVs) are extensively discussed by [6] and [13]. In [7] the authors make use of a centralized cooperative strategy which relies on communication between MAVs while an example of distributed strategy based on information about the payload shape and grasping points is provided by [8].

In terms of passive force control, a comprehensive description is provided in [14]. Admittance control for quadcopters is described by [15], but it heavily relies on motion capture systems for the force estimation.

A force and torque estimator for multirotors based on the UKF and similar to the one that we propose is described by [16], and it is related to the work of [17] for the estimation of attitude quaternion.

III. SYSTEM DESIGN

A. The master-slave paradigm

The master-slave paradigm is a communication approach for cooperative tasks and is used in many different fields. In our context, we defined it as follows. The master agent tracks a reference trajectory provided by an external operator or a path planning algorithm while grasping, lifting and pulling

the payload. It executes on-board reference tracking and state estimation algorithms and it behaves as if it was carrying the payload alone. The slave agent instead grasps the payload, detects the magnitude and direction of the force that the master is applying on it and generates a trajectory compliant with the force applied by the master.

The two agents do not share information via a conventional communication link, but rather via the force applied to the payload by the master, which is sensed by the slave through a force estimator.

B. Coordinate system

The relevant coordinate frames for this work are two: an inertial reference frame I and a non-inertial reference frame B attached to the Center of Gravity (CoG) of the MAV. They are represented in Figure 2.

C. Slave control architecture

From now on we will focus our attention on the control architecture of the slave agent, since the master runs a standard trajectory tracking feedback loop. Slave's control architecture is composed by four main submodules and its control scheme can be found in Figure 3.

- **State estimator:** estimates the slave's (a) position, (b) velocity, (c) attitude (expressed as a quaternion), and (d) angular velocity with respect to I frame. It can either be based on an external motion capture system or an on-board VI navigation system.
- **Force and torque estimator:** estimates the external force acting on the slave expressed in I frame and external torque around z_B . In addition to the full output of the state estimator, the force and torque estimator needs the measurement of the speed of the rotors.
- **Admittance controller:** provides a reference pose (position and attitude) for the position and attitude controller, given the estimate of the external force and torque and a desired trajectory.
- **Position and attitude controller:** Model Predictive Control (MPC) based controller, it tracks the trajectory generated by the admittance controller by providing a rotational speed command to the rotors. The controller is detailed in [18] and [19].

IV. FORCE AND TORQUE ESTIMATOR

In this section, we present the external force and torque estimator based on the UKF, with the attitude represented as a unit quaternion. First, we derive a nonlinear discrete time model of the rotational and translational dynamics of a hexacopter, to be used in the derivation of the process model of the filter. Second, we derive the measurement model to be used in the update step. Third, we describe the prediction and update step for the whole state of the filter with the exception of the attitude quaternion. In the end, we describe the singularity free prediction and update steps which take into account the attitude quaternion.

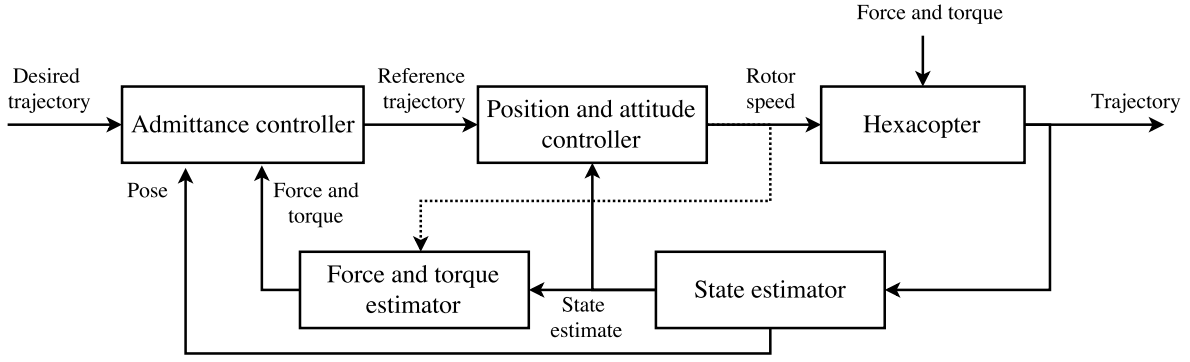


Fig. 3: Passive force control scheme for the slave agent. The user-provided desired trajectory is modified by the admittance controller according to the estimate of the force and torque estimator. The state estimator can be either based on a VI navigation system or an external motion capture system.

A. Hexacopter model with external force and torque

a) *Model assumptions and simplifications:* In order to simplify the model derivation, we assume that:

- the MAV structure is rigid, symmetric on the $x_B y_B$ and $y_B z_B$ planes, and the CoG and body frame origin B coincide;
- aerodynamic interaction with the ground or other surfaces can be neglected and we only consider aerodynamic effects dominant at low speed;
- no external torque acts around x_B and y_B axis.

Additionally, we lighten the process equations by introducing U_1, U_2 and U_3 as the total torque produced by the propellers around, respectively, x_B, y_B and z_B , and U_4 as the total thrust produced by the propellers along z_B . We compute $U_i, i = 1 \dots 4$ from the rotor speed $n_i, i = 1 \dots 6$, using the allocation matrix for the AscTec Firefly, as defined in [20]:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \mathbf{K} \begin{bmatrix} s & 1 & s & -s & -1 & s \\ -c & 0 & c & c & 0 & -c \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n_1^2 \\ \vdots \\ n_6^2 \end{bmatrix} \quad (1)$$

$$\mathbf{K} = \text{diag}([lk_n \quad lk_n \quad k_n k_m \quad k_n]) \quad (2)$$

where l is the boom length, k_n and k_m are rotor constants and $c = \cos(30^\circ)$ and $s = \sin(30^\circ)$.

b) *Translational dynamics:* We derive the linear acceleration with respect to I reference frame by taking into account the following forces: (a) gravity, (b) thrust, (c) aerodynamic effects, (d) external forces due to interaction with the vehicle. We obtain:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} \mathbf{R}_{IB}(\mathbf{q}) \left(\begin{bmatrix} 0 \\ 0 \\ U_4 \end{bmatrix} + \mathbf{F}^{aero} \right) - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} \mathbf{F}^{ext} \quad (3)$$

where \mathbf{F}^{ext} are the external forces that act on the vehicle expressed in I reference frame, m is the mass of the vehicle, and \mathbf{F}^{aero} is defined as:

$$\mathbf{F}^{aero} = k_{drag} \sum_{i=1}^6 |n_i| \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} \quad (4)$$

In addition, we define the rotation matrix from frame B to frame I as $\mathbf{R}_{IB}(\mathbf{q})$, where \mathbf{q} is the normalized quaternion representing the attitude of the vehicle.

c) *Rotational dynamics:* We derive the angular acceleration $\dot{\boldsymbol{\omega}}$ in B frame by taking into account the following torques: (a) total torque produced by propellers, (b) external torque around z_B axis, (c) inertial effects, (d) propeller's gyroscopic effects. We obtain:

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} \left(\begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \tau^{ext} \end{bmatrix} - \boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega} - \boldsymbol{\tau}^{rotors} \right) \quad (5)$$

where τ^{ext} is the external torque around the z_B axis. \mathbf{J} is the inertia tensor of the MAV with respect to B frame, defined as

$$\mathbf{J} = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (6)$$

and $\boldsymbol{\tau}^{rotors}$ is the torque caused by propeller's gyroscopic effects defined according to [21] as

$$\boldsymbol{\tau}^{rotors} = J_r \Omega_r \begin{bmatrix} \omega_y \\ \omega_x \\ 0 \end{bmatrix} \text{ with } \Omega_r = \sum_{i=1}^6 n_i - 6\omega_z \quad (7)$$

where J_r is the propeller's moment of inertia around its axis of rotation.

d) *Discretized model:* Given the sampling time $T_s = 0.01$ s, we discretize the analytic models (3) and (5), obtaining the compact form:

$$\mathbf{s}_{k+1} = f_k(\mathbf{s}_k, \mathbf{F}_k^{ext}, \tau_k^{ext}, \mathbf{n}_k) \quad (8)$$

where

$$\mathbf{s}_k = [\mathbf{p}_k \quad \mathbf{v}_k \quad \mathbf{q}_k \quad \boldsymbol{\omega}_k] \quad (9)$$

with the following discrete-time variables:

- \mathbf{p}_k is the position in I frame;
- \mathbf{v}_k is the velocity in I frame;
- \mathbf{q}_k is the unitary attitude quaternion in B frame;
- $\boldsymbol{\omega}_k$ is the angular velocity in B frame;
- \mathbf{n}_k is the discrete measurement of the rotor speed.

All the states are discretized using the forward Euler method, with the exception of the normalized attitude quaternion \mathbf{q}_k , which is integrated using the approach proposed in [17].

B. Process and measurement models

a) *Process model*: First, we augment the state vector (9) of the hexacopter model and we modify the system dynamic equation (8) so that they take into account the external force and torque. We obtain:

$$\mathbf{s}_{k+1} = \mathbf{f}_k(\mathbf{s}_k, \mathbf{n}_k) + \bar{\mathbf{w}}_k \quad (10)$$

$$\mathbf{s}_k = [\mathbf{p}_k \quad \mathbf{v}_k \quad \mathbf{q}_k \quad \boldsymbol{\omega}_k \quad \mathbf{F}_k^{ext} \quad \boldsymbol{\tau}_k^{ext}] \quad (11)$$

We assume that the change of external force and torque is purely driven by zero mean additive process noise, whose covariance is a tuning parameter of the filter. The external force and torque are updated according to

$$\begin{aligned} \mathbf{F}_{k+1}^{ext} &= \mathbf{F}_k^{ext} \\ \boldsymbol{\tau}_{k+1}^{ext} &= \boldsymbol{\tau}_k^{ext} \end{aligned} \quad (12)$$

Alternatively, we can assume an exponential force decaying model

$$\begin{aligned} \mathbf{F}_{k+1}^{ext} &= \mathbf{F}_k^{ext} (1 - T_s/\bar{\tau}) \\ \boldsymbol{\tau}_{k+1}^{ext} &= \boldsymbol{\tau}_k^{ext} (1 - T_s/\bar{\tau}) \end{aligned} \quad (13)$$

where $\bar{\tau}$ is a constant set by the user and defines the decay rate of the estimates.

In order to define the state covariance and the process noise covariance matrices, we introduce a new representation $\bar{\mathbf{s}}_k$ of the state \mathbf{s}_k , where we substitute the attitude quaternion \mathbf{q}_k with a 3×1 attitude error vector \mathbf{e}_k . More details about this substitution are explained in subsection IV-D.

$$\bar{\mathbf{s}}_k = [\mathbf{p}_k \quad \mathbf{v}_k \quad \mathbf{e}_k \quad \boldsymbol{\omega}_k \quad \mathbf{F}_k^{ext} \quad \boldsymbol{\tau}_k^{ext}] \quad (14)$$

We can now introduce \mathbf{P}_k , the 16×16 covariance matrix associated to the state $\bar{\mathbf{s}}_k$, and \mathbf{Q} , the 16×16 time invariant process noise diagonal covariance matrix.

b) *Measurement model*: We define the measurement vector \mathbf{z}_k (15), the linear measurement function (16) and the associated 12×12 measurement noise covariance matrix \mathbf{R} (17), under the assumption that the measurements are subject to zero-mean additive noise $\bar{\mathbf{v}}_k$.

$$\mathbf{z}_k = [\mathbf{p}_k^m \quad \mathbf{v}_k^m \quad \mathbf{e}_k^m \quad \boldsymbol{\omega}_k^m] \quad (15)$$

$$\begin{aligned} \mathbf{z}_k &= \mathbf{H} \bar{\mathbf{s}}_k + \bar{\mathbf{v}}_k \\ \mathbf{H} &= [\mathbf{I}_{12 \times 12} \quad \mathbf{0}_{12 \times 4}] \end{aligned} \quad (16)$$

$$\mathbf{R} = \text{diag}([\mathbf{1}_3^T \sigma_{m,p}^2 \quad \mathbf{1}_3^T \sigma_{m,v}^2 \quad \mathbf{1}_3^T \sigma_{m,e}^2 \quad \mathbf{1}_3^T \sigma_{m,\omega}^2]) \quad (17)$$

As a remark, we observe that the measured attitude quaternion \mathbf{q}_k^m is taken into account in the measurement vector via the measured attitude error vector \mathbf{e}_k^m . In addition, the measurement update step (explained in subsection IV-C) is performed using the state vector $\bar{\mathbf{s}}_k$.

C. Position, velocity, angular velocity, external force and torque estimation

Notation declaration: From now on, we will use the following notation:

- $\hat{\mathbf{s}}_{k-1}^+$ denotes the estimated state before the prediction step;
- $\hat{\mathbf{s}}_k^-$ denotes the estimated state after the prediction step and before the update step;
- $\hat{\mathbf{s}}_k^+$ denotes the estimated state after the update step.

The same applies for the state covariance matrix \mathbf{P}_k .

a) *Prediction step*: The predicted value for all the states with the exception of the attitude is computed using the standard UKF prediction step, as explained in [22] and [23], with the process model derived in subsection IV-B. The matrix square root is computed using the Cholesky decomposition [22].

b) *Kalman filter based update step*: Because the measurement model corresponds to a linear function, we can use the standard KF update step for all the states in the vector state $\bar{\mathbf{s}}_k$.

First, compute the Kalman gain matrix as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (18)$$

Then, the updated state covariance is obtained from

$$\mathbf{P}_k^+ = (\mathbf{I}_{16 \times 16} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- (\mathbf{I}_{16 \times 16} - \mathbf{K}_k \mathbf{H})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (19)$$

Finally, the updated state is obtained from

$$\bar{\mathbf{s}}_k^+ = \bar{\mathbf{s}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \bar{\mathbf{s}}_k^-) \quad (20)$$

D. Quaternion based attitude estimation

Quaternions do not naturally fit in the Unscented Transformation (UT) since they are defined on a non linear manifold, while the UT is performed on a vector space. Interpreting a quaternion as a member of \mathbb{R}^4 produces a singular 4×4 covariance matrix and does not guarantee the unitary norm constraint, while considering quaternions as members of \mathbb{R}^3 - by exploiting the unitary norm constraint - always produces a singularity. For these reasons we estimate the MAV attitude by following the approach called USQUE - Unscented Quaternion Estimation - proposed by [17]. In this approach, the attitude estimate is based on the estimation of the error attitude quaternion - which is always assumed to be smaller than 180 degrees - through a three states parametrization obtained using the Modified Rodrigues Parameters (MRP).

Modified Rodriguez parameters Let $\mathbf{q} = [\mathbf{q}_v^T, q_s]$ be a quaternion, where \mathbf{q}_v is the vector part and q_s is the scalar part. A corresponding (but singular) attitude representation \mathbf{p} , with $\mathbf{p} \in \mathbb{R}^3$, can be obtained as

$$\mathbf{p} = f \frac{\mathbf{q}_v}{a + q_s} \quad (21)$$

where a is a parameter from 0 to 1 and f is a scale factor. We will choose $f = 2(a+1)$ and a to be a tunable parameter of the filter. The inverse representation is obtained as

$$q_s = \frac{-a \|\mathbf{p}\|^2 + f \sqrt{f^2 + (1 - a^2) \|\mathbf{p}\|^2}}{f^2 + \|\mathbf{p}\|^2} \quad (22)$$

$$\mathbf{q}_v = f^{-1} (a + q_s) \mathbf{p}$$

The implemented algorithm for attitude estimation is discussed in the following steps. Its prediction step follows the USQUE approach explained in [17]. The update step - where we fuse the attitude measurement at time k into the attitude error vector - assumes that the measurement for the filter corresponds to the difference between the measured attitude quaternion and the estimate of the attitude at time step $k-1$. Because the initialization and prediction step are extensively covered in [17] and [24], we only report the implemented measurement update step.

a) *Update Step:*

- Given an attitude measurement \mathbf{q}_k^m , compute the rotation error between $\hat{\mathbf{q}}_{k-1}^+$ and \mathbf{q}_k^m as in (23).

$$\delta \mathbf{q}_k^m = \hat{\mathbf{q}}_{k-1}^+ \otimes (\mathbf{q}_k^m)^{-1} \quad (23)$$

- Transform the measured rotation error quaternion $\delta \mathbf{q}_k^m$ in the vector \mathbf{e}_k^m using MRP (21).
- Store \mathbf{e}_k^m in $\bar{\mathbf{s}}_k^-$ and compute the updated attitude error mean and covariance as in IV-C.
- Retrieve $\hat{\mathbf{e}}_k^+$ from $\bar{\mathbf{s}}_k^+$ and compute the updated error attitude quaternion $\delta \mathbf{q}_k^+$ from $\hat{\mathbf{e}}_k^+$ using MRP (22).
- Rotate the predicted attitude quaternion $\hat{\mathbf{q}}_k^-$ of $\delta \mathbf{q}_k^+$ to obtain the current quaternion estimate of the attitude as in (24).

$$\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- \otimes \delta \mathbf{q}_k^+ \quad (24)$$

V. ADMITTANCE CONTROLLER

A. An intuitive explanation

Admittance control is based on the idea that the trajectory which guarantees compliance with external force/torque is generated by simulating a *spring-mass-damper* dynamic system. As represented in Figure 4, the new trajectory along x_I is generated as if the MAV, with virtual mass $m_{v,x}$, was connected to the desired trajectory via a spring with elastic constant k_x and via a damper with damping coefficient c_x . The deviation from the desired trajectory is caused by the estimated external force/torque, which excites the virtual mass.

Compliance with external force and torque can be changed by tuning the virtual spring constant k_x . Increasing values of k_x simulate a stiffer spring, which in turn yields to better desired trajectory tracking. Conversely, $k_x = 0$ guarantees full compliance with the estimated external force.

B. Trajectory generation law

The admittance controller generates a new reference trajectory independently for every axis. The trajectory generation law has been derived by the discretization of (25) for the generation of the trajectory on x_I , y_I , z_I , and (26) for the reference value of the yaw angle ψ .

$$m_{v,i}(\ddot{\Lambda}_{d,i} - \ddot{\Lambda}_{r,i}) + c_i(\dot{\Lambda}_{d,i} - \dot{\Lambda}_{r,i}) + k_i(\Lambda_{d,i} - \Lambda_{r,i}) = F_i^{ext} \quad (25)$$

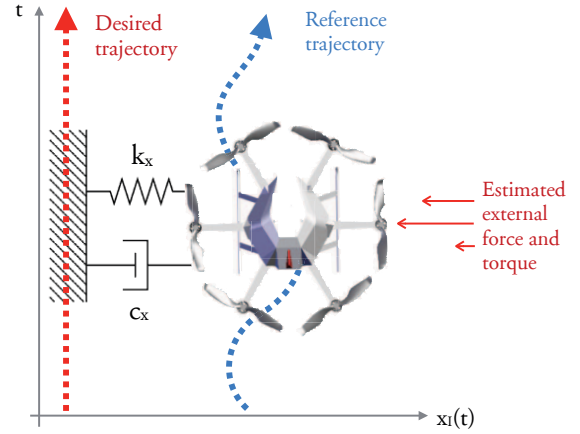


Fig. 4: Schematic explanation of admittance control. Given a desired trajectory, the admittance controller generates a new reference trajectory according to the estimated external forces by simulating a second order *spring-mass-damper* dynamic system.

$$J_{v,\psi}(\ddot{\Lambda}_{d,\psi} - \ddot{\Lambda}_{r,\psi}) + c_\psi(\dot{\Lambda}_{d,\psi} - \dot{\Lambda}_{r,\psi}) + k_\psi(\Lambda_{d,\psi} - \Lambda_{r,\psi}) = \tau_\psi^{ext} \quad (26)$$

The index i denotes one of the axes x_I , y_I , z_I , while $\Lambda_{d,i}$ represents the desired trajectory and $\Lambda_{r,i}$ the generated reference trajectory on the selected axis. The parameters $m_{v,i}$, c_i and k_i can be set independently for every axis and represent, respectively, the virtual mass, the virtual damping and the virtual elastic constant used by the controller. Similarly, $J_{v,\psi}$ represents the virtual inertia around z_B , c_ψ the virtual damping and k_ψ the elastic constant of the virtual torsion spring.

We additionally assume that the reference trajectory is given in discrete steps, which means that we set $\ddot{\Lambda}_d = \dot{\Lambda}_d = 0$. In the same way, the admittance controller only provides a pose reference by setting $\ddot{\Lambda}_r = \dot{\Lambda}_r = 0$. The underlying MPC-based position and attitude controller takes care of smoothing the pose reference and avoids jerky behaviors.

C. Robustness to noise and undesired constant external force

If no force acts on the vehicle and the desired trajectory is constant, noise and other factors not included in the model (such as propeller efficiency) may result in a non-zero force estimate, which in turn causes a drift in the reference pose Λ_r . Similarly, external factors such as constant wind or model-mismatches in the force and torque estimator may cause a constant offset in the estimated external disturbances.

In order to avoid undesired drifts in the reference trajectory, the admittance controller has been integrated into a Finite State Machine (FSM), which monitors the magnitude of the force on each axis and decides whether to reject or take into account the estimated external force to compute a new trajectory reference [15]. Model mismatches and offset in the force/torque estimation are taken into account by averaging the estimated external force for a given time T_{avg} while the helicopter is hovering. The computed offset is subtracted from the estimate produced by the force and torque estimator.

D. Landing detection and auto-disengagement

The admittance controller must be used only once the vehicle is hovering, so that it does not take into account the ground reaction force as an external force, resulting in an undesired reference trajectory as output. For this reason, the controller auto-disengages if landing is detected. Landing is detected by simply monitoring if the z_I component of the estimated external force F_z^{ext} is bigger than a tunable threshold \bar{F}^{ext} for a given time $T_{landing}$.

VI. EVALUATION

We now present four main experimental results. In the first two experiments we benchmark the performances of the force and torque estimator by (a) comparing its estimates with the measurements of a force sensor and (b) by showing its reaction speed in detecting a collision with a wall. In the last two experiment we additionally make use of the admittance controller by (a) interacting with the vehicle through a rope and (b) showing the result of a payload carrying maneuver using two hexacopters.

A. Hardware and software

a) Hardware: The MAV used for our experiments is the hexacopter **AscTec Firefly** [25] equipped with an on-board computer based on a **quad-core 2.1 GHz Intel i7** processor and **8 GB of RAM**. The external motion capture system used is [26], while the VI navigation system is developed by the Autonomous Systems Lab at ETHZ and Skybotix AG [27].

b) Software: We implemented the force estimator and the admittance controller using **C++** and **ROS** [28]. All the algorithms run on-board to avoid issues related to wireless communication. Both the force estimator and the admittance controller run at **100 Hz** using about 10% of the available computational power together. The multi-sensor fusion and state estimation algorithm used for the VI navigation system is based on ROVIO [29].

B. Force and torque estimator accuracy

The accuracy of the force and torque estimator is validated by comparing its estimates with the measurements of a force sensor. The experiment was performed by using a **Optoforce OMD-20-FE-200N** force sensor connected to the bottom of the **AscTech Firefly** hexacopter. An in-extensible nylon wire was attached to the sensor and interaction was performed by pulling the wire, while the MAV was maintained steady at about 1.6 m from the ground by the MPC-based position and attitude controller. The admittance controller was not active and the state estimator used was the VI navigation system. The external force due to the interaction measured and estimated on x_I , y_I and z_I axis are represented in Figure 5. The RMS error between the measurements and the estimate is reported in Table I.

C. Detecting collision with a wall

In this experiment we use the force estimator to detect a collision with a wall and to send a new "safe" position reference to the position and attitude controller. We use

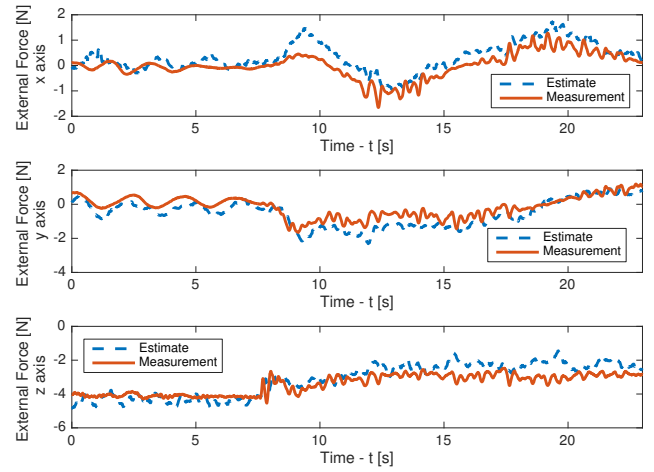


Fig. 5: Measured force and estimated force on x_I , y_I and z_I axis. We performed the experiment with the hexacopter on hover and using the VI navigation system as state estimator.

TABLE I: RMS estimation error for the force and torque estimator validation

Disturbance	RMS error	Unit
F_x^{ext}	0.4298	N
F_y^{ext}	0.4941	N
F_z^{ext}	0.5729	N

this experiment to highlight the reaction speed of the force estimator, showing that it is able to react to an impact at about 1.2 m/s in less than 50 ms.

A collision is detected by checking if the magnitude of the estimated external force exceeds a given threshold. The "safe" position reference is pre-stored in the controller. The experimental results are shown in Figure 6, where we represent the reaction of the force and torque estimator and of the "safety" controller in case of impact at 1.2 m/s. The state estimator used for the experiment is an external motion capture system [26]. Since the wall corresponds to the plane $y_I = -1.94$ m expressed in I reference frame, we only show the y_I components of (a) desired trajectory, (b) trajectory generated by safety controller, (c) pose of the MAV and (d) estimated external force.

Figure 7 explains how the reaction speed (time between detected impact and actual impact) has been derived.

D. Drone on a leash

To test the capability of admittance controller we present a real-time human-machine interaction. A human operator carries around the MAV via a string attached at the bottom of the hexacopter. The admittance controller is tuned so that it generates a trajectory fully compliant to the estimated external force and torque ($k_x = k_y = k_z = k_{\psi} = 0$). The state estimator used is based on the VI navigation system. The results can be found in the first part of the online video [1] and a preview is provided in Figure 8.

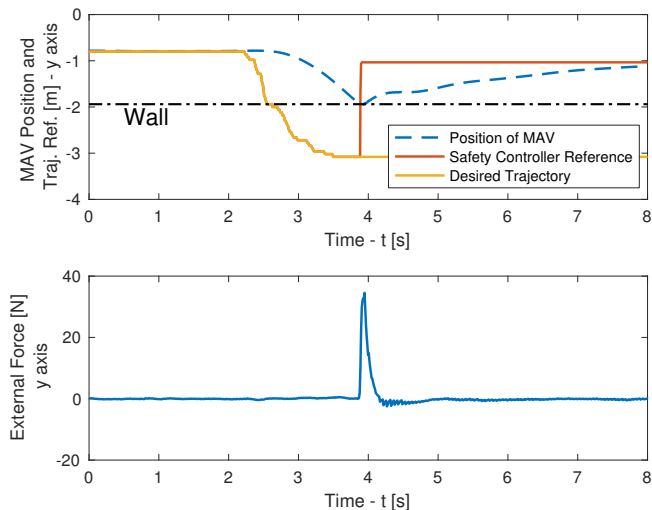


Fig. 6: Experimental results of the wall collision detection. The wall is placed at $y_I = -1.94$ m. The wall generates a reaction force of 34.5 N and a safe trajectory reference is generated 0.84 m further from the impact point. The impact happens at a speed of about 1.2 m/s.

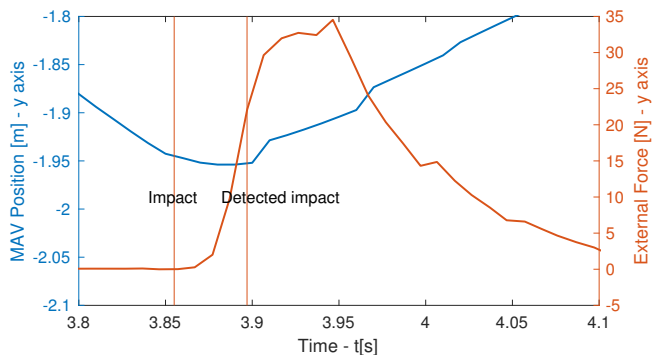


Fig. 7: Detail of the impact detection time computation. The vertical line on the left represents the impact instant $t = 3.855$ s, while the vertical line on the right the detection instant $t = 3.897$ s.

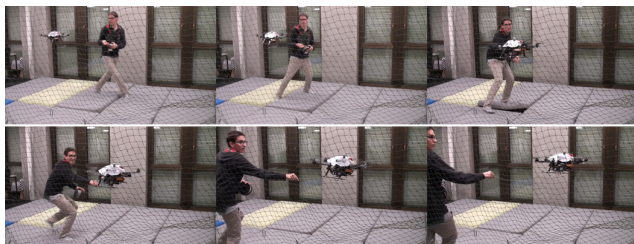


Fig. 8: Illustration of the human-hexacopter interaction from the online video [1]. The interaction is performed through a string attached at the bottom of the MAV. This experiment is especially useful to understand how different values of virtual spring, mass and damping affect the behavior of the slave agent in a collaborative transportation task.

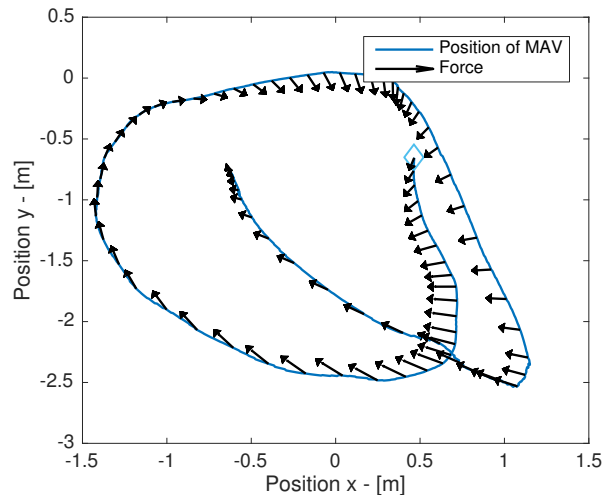


Fig. 9: Trajectory generated by the slave agent during collaborative transportation. The estimated external force acting on the slave is represented by the black arrows. The starting point is marked with a diamond.

E. Cooperative payload transportation

Finally we present the cooperative transportation of a bulky object; the setup of the experiment is represented in Figure 1. The payload is a **1.2 m carton tube weighing 370 g**, while the MAVs used are **two AscTec Firefly hexacopters**. Both the MAVs are attached at the extremities of the payload via a nylon wire, which cancels any torque. The state estimator used is the VI navigation system.

The master agent is controlled by a human operator, while the trajectory for the slave agent is generated by the admittance controller. For this experiment, the admittance controller is used to generate a compliant trajectory on the x_I and y_I axis, while tracking a given altitude reference on z_I . We achieved this by setting the virtual spring value on x_I and y_I axis to be $k_x = k_y = 0$ N/m, while on the z_I axis a stiff virtual spring of value $k_z = 10$ N/m has been used.

The result of the experiment is shown in the second part of the online video [1]. Additionally, the reference trajectory and the estimated external force for the slave agent on the $x_I y_I$ plane are represented in Figure 9, while Figure 10 and 11 show the reference trajectory and the estimated external force and torque on x_I, y_I, z_I and ψ_B . As a remark, we observe that the limited stiffness of the virtual spring on the z_I axis and the estimated force due to the weight of the payload cause, as can be observed in the online video [1], a loss of altitude of the slave agent equal to F_z^{ext}/k_z .

VII. CONCLUSION

In this work we showed that collaborative transportation of a bulky payload can be achieved without relying on a communication network between the involved agents. We achieved this by making use of an admittance controller in conjunction with a force and torque estimator based on the UKF. The force estimates are obtained using the state information provided by a visual inertial navigation system. We include demonstration of the filter performance

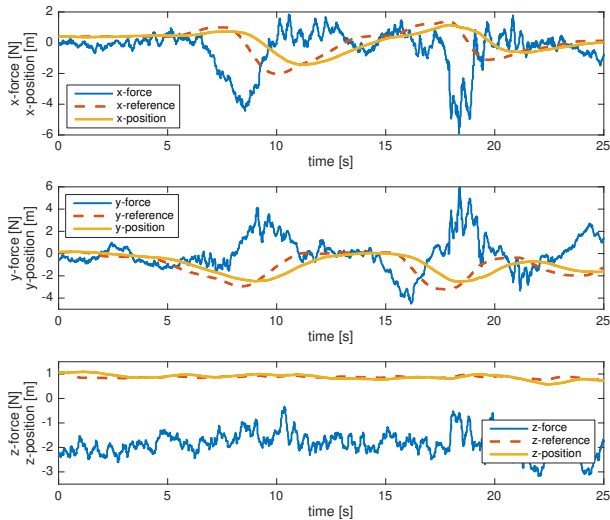


Fig. 10: Estimated external force, admittance controller reference trajectory and estimated position of the slave agent during the collaborative transportation.

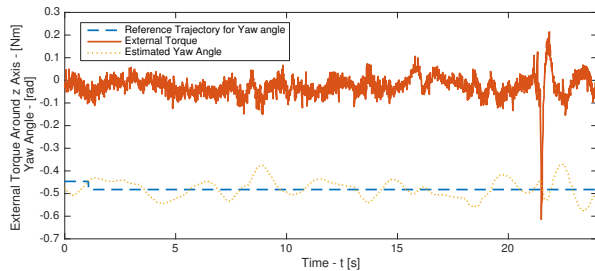


Fig. 11: Estimated external torque around z_I axis, reference trajectory for yaw attitude angle, and estimated yaw attitude angle. The peak in torque estimation at $t = 22$ s corresponds to the accidental collision of one of the propellers of the MAV with the payload.

by detecting wall collision and real time human-machine interaction.

ACKNOWLEDGMENT

This work was supported by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No.644128, AEROWORKS, and Mohamed Bin Zayed International Robotics Challenge 2017.

REFERENCES

- [1] (2017) Collaborative transportation using mavs via passive force control. [Online]. Available: <https://rebrand.ly/eth-asl-collaborative-transportation>
- [2] (2016) The ambulance drone. [Online]. Available: <http://www.io.tudelft.nl/onderzoek/delft-design-labs/applied-labs/ambulance-drone/>
- [3] (2016) Zipline, the future of healthcare is out for delivery. [Online]. Available: <http://flyzipline.com/product/>
- [4] M. Nieuwenhuisen, D. Droschel, M. Beul, and S. Behnke, "Autonomous MAV Navigation in Complex GNSS-denied 3D Environments," *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, no. October, pp. 1241–1250, 2015.
- [5] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Syst. Mag.*, vol. 34, no. 4, pp. 46–64, 2014.

- [6] I. Maza, K. Kondak, M. Bernard, and A. Ollero, "Multi-UAV cooperation and control for load transportation and deployment," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 57, no. 1-4, pp. 417–449, 2010.
- [7] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, no. September 2010, pp. 1–14, 2010.
- [8] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative Grasping and Transport using Multiple Quadrotors.pdf," in *Springer Tracts in Advanced Robotics*, 2015, vol. 83, pp. 545–558.
- [9] H. F. McCreery and M. D. Breed, "Cooperative transport in ants: A review of proximate mechanisms," *Insectes Sociaux*, vol. 61, no. 2, pp. 99–110, 2014.
- [10] H. N. Nguyen, S. Park, and D. Lee, "Aerial tool operation system using quadrotors as Rotating Thrust Generators," *IEEE International Conf. on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 1285–1291, 2015.
- [11] P. J. S. D. Mittal, M., "Nonlinear adaptive control of a twin lift helicopter system," *IEEE Control Systems Magazine* 11(3), pp. 39–45, 1991.
- [12] R. A. Reynolds, H.K., "H control of a twin lift helicopter system," *Proceedings of the 31st IEEE Conference on Decision and Control*, pp. 2442–2447, 1992.
- [13] V. Parra-Vega, A. Sanchez, C. Izaguirre, O. Garcia, and F. Ruiz-Sanchez, "Toward aerial grasping and manipulation with multiple UAVs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 70, no. 1-4, pp. 575–593, 2013.
- [14] L. Villani and J. D. Schutter, *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, E. B. H. Springer-Verlag, Ed., 2008.
- [15] F. Augugliaro and R. D'Andrea, "Admittance control for physical human-quadrocopter interaction," *European Control Conf. (ECC)*, 2013, pp. 1805–1810, 2013.
- [16] C. D. Mckinnon and A. P. Schoellig, "Unscented External Force and Torque Estimation for Quadrotors," in *ICRA*, 2016, pp. 5651–5657.
- [17] J. Crassidis and F. L. Markley, "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 536–542, 2003.
- [18] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," in *Robot Operating System (ROS) The Complete Reference*, A. Koubaa, Ed. Springer, (to appear).
- [19] M. Kamel, M. Burri, and R. Siegwart, "Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles," *ArXiv e-prints*, Nov. 2016.
- [20] M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Inversion based direct position control and trajectory following for micro aerial vehicles," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2933–2939, 2013.
- [21] S. Bouabdallah, "Design and Control of Quadrotors With Application To Autonomous Flying," *École Polytechnique Fédérale De Lausanne, À La Faculté Des Sciences Et Techniques De L'Ingénieur*, vol. 3727, no. 3727, p. 61, 2007.
- [22] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.*, J. W. . Sons, Ed., 2006.
- [23] S. J. Julier and J. K. Uhlmann, "A General Method for Approximating Nonlinear Transformations of Probability Distributions," pp. 1–27, 1996.
- [24] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Quaternion Averaging," *NASA Goddard Space Flight Center*, pp. 1–10, 2007.
- [25] (2017) Ascending technologies. [Online]. Available: <http://www.ascotec.de>
- [26] (2016) Vicon. [Online]. Available: <http://www.vicon.com>
- [27] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A Synchronized Visual-Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM," in *Robotics and Automation (ICRA), 2014 IEEE Int. Conf. on*, 2014, pp. 431–437.
- [28] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625.
- [29] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust Visual Inertial Odometry Using a Direct EKF-Based Approach," 2015. [Online]. Available: <http://e-collection.library.ethz.ch/view/eth:48374>