

# Collaborative Object Transportation Using MAVs via Passive Force Control

Andrea Tagliabue<sup>1</sup>, Mina Kamel<sup>1</sup>, Sebastian Verling<sup>1</sup>, Roland Siegwart<sup>1</sup>, Juan Nieto<sup>1</sup>

**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 navigation system. Experimental results 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 fantasy of researchers and entrepreneurs [1][2] as a new tool for good delivery. Quadcopters and hexacopters have shown to be especially suited for this task, thanks to their ability to navigate in cluttered environments [3] and being able to deliver a payload with extreme accuracy [4]. However, their inherent limited size constrain the range of applications to the transportation of small and light-weight objects.

Collaborative strategies can significantly enhance MAV transportation capabilities and can provide a cost effective solution with respect to the deployment of a single, more capable MAV [5]. 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 [6]. A second possible approach is based on distributed control algorithm. In this case, the agents only share a common goal and the individual control laws are obtained with respect to the grasping point on the payload [7].

Both of the described collaborative 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 approach, instead, relies 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 [8] collaborative approach based on the master-slave(s) paradigm. We chose one of the available MAVs to be the leader - the master - while the remaining agents

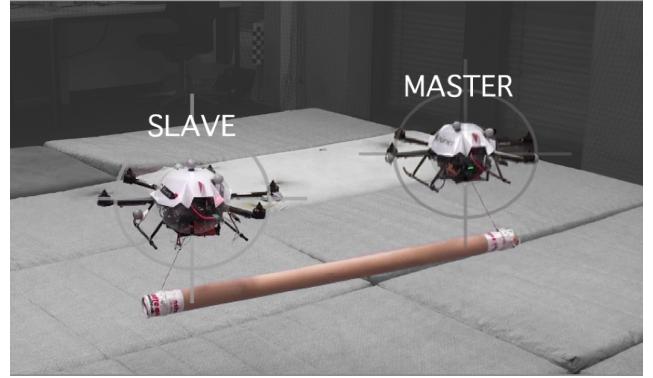


Fig. 1: Experimental setup for collaborative object transportation. The master vehicle is controlled remotely, while the slave provides compliance to the movement of the master using an admittance controller.

act as slaves. The task of the master is simply to lift the payload and pull it in the desired direction. The slaves - which are also attached to the payload - actively guarantee compliance to the master's actions by sensing the force that master applies on the payload and changing their 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. To obtain the pose of the MAV, as well as its velocity and angular velocity, we use a visual-inertial navigation system mounted on the slave agent.

In our setup we assume the following: (a) the payload is big enough to guarantee that at least two MAVs can autonomously grasp it, (b) all the agents are already attached to the payload and (c) the attitude of the vehicle 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 [9].

### A. Contributions

The main contribution of this work is the novel approach for collaborative object 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, (c) the position of grasping point, and is potentially scalable to multiple agents. To effectively demonstrate our approach, we describe the implementation of a force estimator based on the UKF which handles quaternions, is computationally light, and fast. It is

<sup>1</sup> Autonomous Systems Lab., Department of Mechanical and Process Engineering, ETH Zurich, Zurich, Switzerland.

able to detect a collision with a wall in less than 20 ms and provides an accurate enough force estimate for human-vehicle interaction without relying on an external motion capture system, but using a VI (Visual Inertial) navigation system.

### B. Structure of this work

We start by presenting existing approaches and technologies for cooperative transportation via MAV in Section II. We then proceed by providing an overview of the control strategy and architecture in Section III and we detail the implementation of the force estimator and admittance controller in Sections IV and V, respectively. We conclude by showing experimental results in Section VI and conclusion 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 [10] [11], but do not provide experimental results. Collaborative transportation using UAVs are extensively discussed by [5] and [12]. [6] makes 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 [7].

In terms of passive force control and admittance control, a comprehensive description is provided in [13]. [14] describes an example of admittance control for quadcopters and human-machine interaction, but heavily relies on motion capture systems for the force estimation.

An external forces and torque estimator for multirotor vehicles based on the UKF and similar to the one we propose is addressed on [15], which relies on the work of [16] for attitude quaternion estimation.

## III. SYSTEM DESIGN

### A. The Master-Slave Paradigm

The master-slave paradigm is a common communication approach for cooperative tasks and is used in many different fields, from industries, informatics to the railway sector. In our context, we defined it as follows. We divide the agents (the MAVs) which take part in the collaborative transportation maneuver in two categories, (a) master and (b) slave. The master, which is unique, follows a reference trajectory given by an external operator or a path planning algorithm while grasping, lifting and pulling the payload. It executes its standard on-board reference tracking and state estimation algorithms and it behaves as if it was carrying the payload alone. The slave agent - which can be one or more - 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.

Master and slave do not share information via a conventional communication link, but rather via the force applied

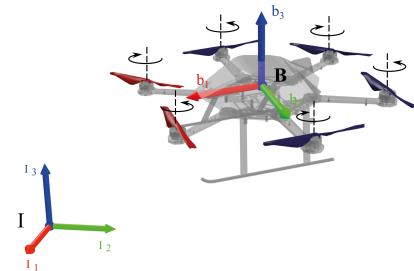


Fig. 3: Reference frames defined for this work.  $I$  represents the inertial reference frame, while  $B$  the MAV reference frame.

to the payload, which is sensed by the slave through a force estimator.

### B. 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 pose tracking feedback loop. Slave's control architecture is composed by four main building blocks.

- **State estimator:** estimates the slave's (a) position, (b) velocity, (c) attitude, and (d) angular velocity w.r.t the  $I$  frame. It can either be an external motion capture system such as [17] or a VI navigation system. The attitude is expressed as a quaternion.
- **Force and torque estimator:** estimates the external force acting on the slave expressed in  $I$  frame and external torque around  $z_B$  axis in  $B$  frame. Additionally to the full output of the state estimator - which is used in the update step - the force and torque estimator needs the measurement of the speed of the rotors to propagate the model dynamics in the state prediction step.
- **Admittance controller:** provides a reference pose (position and attitude) for the MAV position and attitude controller, given the estimate of the external force and torque and a desired trajectory. According to the choice of its parameters, it can better track the desired trajectory or give full compliance to the estimated external disturbances. For example, during the cooperative transportation, it complies fully to the force components on  $x_I$  and  $y_I$  axis while tracking an altitude reference on  $z_i$ . More details can be found in Section V.
- **Position and attitude controller:** MPC based controller, that tracks the trajectory generated by the admittance controller by providing a rotor speed command.

### C. Coordinate System

The relevant coordinate frames for this work are two: an inertial reference frame attached to the ground  $I$  and a non-inertial reference frame attached to the Center of Gravity (CoG) of the vehicle  $B$ . They are represented in Figure 3.

### D. Hardware and Software

a) **Hardware:** The MAV used for our experiments is the **AscTec Firefly hexacopter** equipped with an on board computer based on a **quad-core 2.1 GHz Intel i7** processor

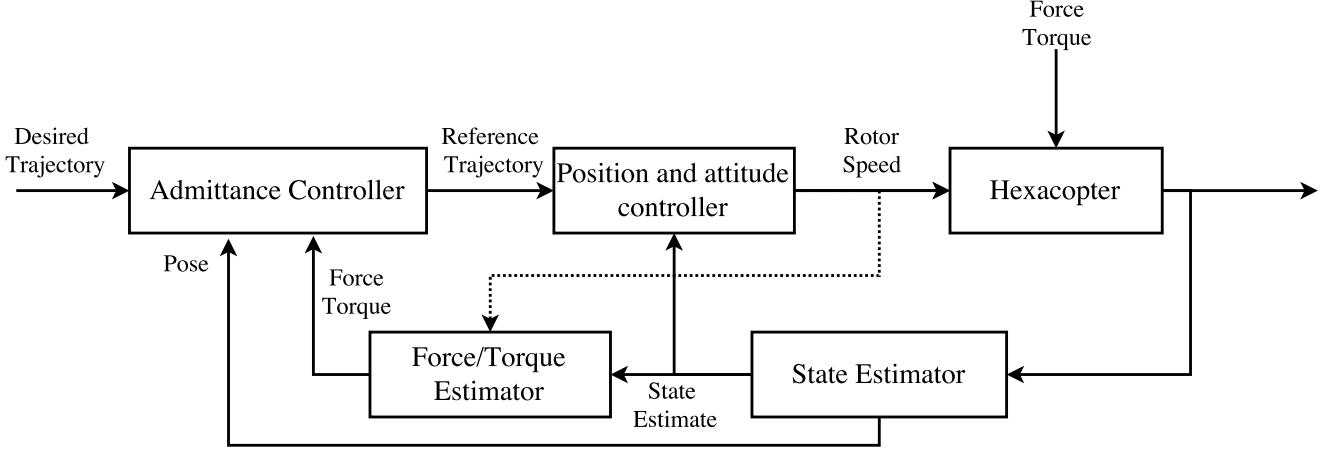


Fig. 2: Active interaction control scheme for the slave agent. The user-provided desired trajectory is modified by the admittance controller according to the external force and torque estimated by the UKF filter based estimator. The state estimator can be either based on a visual-inertial navigation system or an external motion capture system.

and **8 GB of RAM**. The hexacopter is additionally equipped with a Visual-Inertial navigation system, developed by the Autonomous Systems Lab at ETHZ and Skybotix AG [18]. This sensor measurement, fused with vehicle's IMU provide a 100 Hz estimate of: (a) position, (b) velocity, (c) attitude, (d) angular velocity of the MAV.

*b) Software:* We implemented the force estimator and the admittance controller algorithm using **C++ and ROS**. All the algorithms run on board the hexacopter 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 position and attitude controller is based on the model predictive controller developed in [19], [20].

#### IV. EXTERNAL FORCE AND TORQUE ESTIMATOR

In this section, we present the UKF that was implemented to estimate the external force and torque acting on the MAV. 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.

##### A. Hexacopter Model with External Force and Torque

*c) Model Assumptions:* 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 (Center of Gravity) 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.

In addition, we define the rotation matrix from frame  $B$  to frame  $I$  as  $\mathbf{R}(q)_B^I$ , where  $q$  is the normalized quaternion representing the attitude of the vehicle. Finally, we simplify the 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 expressed in  $B$  frame. We compute  $U_i, i = 1 \dots 4$  from the rotor speed  $n_i, i = 1 \dots 6$ , using the allocation matrix, as defined in [21] for the AscTec Firefly hexacopter [22]:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \mathbf{K} \begin{bmatrix} s & 1 & s & -s & -1 & s \\ -c & 0 & c & c & 0 & -c \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix} \quad (1)$$

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

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

*d) 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:

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

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:

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

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

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

where  $\tau^{ext}$  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  $\boldsymbol{\tau}^{rotor}$  is the torque caused by propeller's gyroscopic effects defined according to [23] as

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

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

f) *Discretized Model*: Given the sampling time  $T_s = 0.01s$ , we discretize the analytic models (3) and (5) using the forward Euler method, while the normalized attitude quaternion  $\mathbf{q}_k$  is integrated using the approach proposed in [16]:

$$\mathbf{q}_{k+1} = \Omega(\omega_k) \mathbf{q}_k \quad (8)$$

with

$$\Omega(\omega_k) = \begin{bmatrix} \cos(\frac{1}{2}||\omega_k||T_s) \mathbf{I}_{3 \times 3} & \Psi_k \\ -\Psi_k^T & \cos(\frac{1}{2}||\omega_k||T_s) \end{bmatrix} \quad (9)$$

and

$$\Psi_k = \sin(\frac{1}{2}||\omega_k||T_s) \omega_k / ||\omega_k|| \quad (10)$$

The discretized dynamic equations can then be written in compact form as:

$$\mathbf{s}_{k+1} = f(\mathbf{s}_k, \mathbf{F}_k^{ext}, \tau_k^{ext}, \mathbf{n}_k) \quad (11)$$

with

$$\mathbf{s}_k = [\mathbf{p}_k \ \mathbf{v}_k \ \mathbf{q}_k \ \omega_k] \quad (12)$$

As the last step we introduce the notation:

$$\mathbf{s}_{k+1} = f_k(\mathbf{s}_k, \mathbf{F}_k^{ext}, \tau_k^{ext}) \quad (13)$$

where the dependency of  $f(\cdot)$  from  $\mathbf{n}_k$  is taken into account by making  $f(\cdot)$  time-variant.

### B. Process and Measurement models

g) *Process Model*: First, we augment the state vector of the hexacopter model (12) and the system dynamic equation (13) so that they take into account the external force and torque. We obtain:

$$\mathbf{s}_{k+1} = f_k(\mathbf{s}_k) + \bar{\mathbf{w}}_k \quad (14)$$

$$\mathbf{s}_k = [\mathbf{p}_k \ \mathbf{v}_k \ \mathbf{q}_k \ \omega_k \ \mathbf{F}_k^{ext} \ \tau_k^{ext}] \quad (15)$$

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

$$\begin{aligned} \mathbf{F}_{k+1}^{ext} &= \mathbf{F}_k^{ext} \\ \tau_{k+1}^{ext} &= \tau_k^{ext} \end{aligned} \quad (16)$$

or by assuming an exponential force decaying model

$$\begin{aligned} \mathbf{F}_{k+1}^{ext} &= \mathbf{F}_k^{ext} (1 - T_s / \bar{\tau}) \\ \tau_{k+1}^{ext} &= \tau_k^{ext} (1 - T_s / \bar{\tau}) \end{aligned} \quad (17)$$

where  $\bar{\tau}$  is a constant set by the user and defines the convergence speed of the estimates. In order to define the state covariance and the process noise covariance, 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  $3 \times 1$  attitude error vector  $\mathbf{e}_k$ . More details about this substitution are explained in IV-D.

$$\bar{\mathbf{s}}_k = [\mathbf{p}_k \ \mathbf{v}_k \ \mathbf{e}_k \ \omega_k \ \mathbf{F}_k^{ext} \ \tau_k^{ext}] \quad (18)$$

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

h) *Measurement Model*: We define the measurement vector  $\mathbf{z}_k$ , the associated  $12 \times 12$  measurement noise covariance matrix  $\mathbf{R}$  and the linear measurement function  $\mathbf{z}_k = \mathbf{H} \bar{\mathbf{s}}_k + \mathbf{v}_k$ , under the assumption that the measurements are subject to zero-mean additive noise.

$$\mathbf{z}_k = [\mathbf{p}_k^m \ \mathbf{v}_k^m \ \mathbf{e}_k^m \ \omega_k^m] \quad (19)$$

$$\begin{aligned} \mathbf{z}_k &= \mathbf{H} \bar{\mathbf{s}}_k + \mathbf{v}_k \\ \mathbf{H} &= [\mathbf{I}_{12 \times 12} \ \mathbf{0}_{12 \times 4}] \end{aligned} \quad (20)$$

$$\mathbf{R} = \text{diag}([\mathbf{1}_3^T \sigma_{m,p}^2 \ \mathbf{1}_3^T \sigma_{m,v}^2 \ \mathbf{1}_3^T \sigma_{m,e}^2 \ \mathbf{1}_3^T \sigma_{m,\omega}^2]) \quad (21)$$

As a remark, we observe that the measured attitude quaternion is converted in a three states parametrization  $\mathbf{e}_k$  for the measurement vector. In addition, the measurement update step of the filter is performed using the state vector  $\bar{\mathbf{s}}_k$ . (IV-C.0.j).

### 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  $\mathbf{P}$ .

i) *Prediction Step*: The predicted state for all the states with the exception of the attitude is computed using the standard UKF prediction step, as explained in [24] and [25]. The matrix square root is computed using the Cholesky decomposition [24].

j) *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  $\bar{s}_k$ .

First, compute the Kalman gain matrix as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (22)$$

Then, the updated state covariance is obtained from

$$\mathbf{P}_k^+ = (\mathbf{I}_{16 \times 16} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- (\mathbf{I}_{16 \times 16} - \mathbf{K}_k \mathbf{H})^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (23)$$

Finally, the updated state is obtained from

$$\bar{s}_k^+ = \bar{s}_k^- + \mathbf{K}_k (z_k - \mathbf{H} \bar{s}_k^-) \quad (24)$$

#### 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  $4 \times 4$  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 [16]. 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 Rodriguez Parameters (MRP).

**Modified Rodriguez Parameters** Let  $\mathbf{q} = [q_v^T, q_s]$  be a quaternion, where  $q_v$  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

$$\mathbf{p} = f \frac{\mathbf{q}_v}{a + q_s} \quad (25)$$

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

$$q_s = \frac{-a \|\mathbf{p}\|^2 + f \sqrt{f^2 + (1 - a^2)} \|\mathbf{p}\|^2}{f^2 + \|\mathbf{p}\|^2} \quad (26)$$

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

The implemented algorithm for attitude estimation is discussed in the following steps. Its prediction step follows the USQUE approach explained in [16]. 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 [16] and [26], we only report the implemented measurement update step.

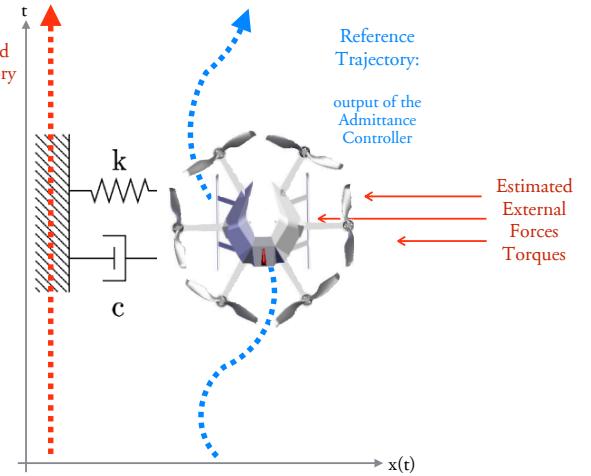


Fig. 4: Schematic explanation of admittance control. Given a desired trajectory, the admittance controller generates a new reference trajectory according to the estimated external forces by simulating a second order *spring-mass-damper* dynamic system.

#### k) *Update Step*:

- Given an attitude measurement  $\mathbf{q}_k^m$ , compute the rotation error between  $\hat{\mathbf{q}}_{k-1}^+$  and  $\mathbf{q}_k^m$  (27).

$$\delta \mathbf{q}_k^m = \hat{\mathbf{q}}_{k-1}^+ \otimes (\mathbf{q}_k^m)^{-1} \quad (27)$$

- Transform the measured rotation error quaternion  $\delta \mathbf{q}_k^m$  in the vector  $e_k^m$  using MRP (25).
- Store  $e_k^m$  in  $\hat{\mathbf{s}}_k^-$  and compute the updated attitude error mean and covariance as in IV-C.0.j.
- Retrieve  $\hat{\mathbf{e}}_k^+$  from  $\hat{\mathbf{s}}_k^+$  and compute the updated error attitude quaternion  $\delta \mathbf{q}_k^+$  from  $\hat{\mathbf{e}}_k^+$  using MRP (26).
- Rotate the predicted attitude quaternion  $\hat{\mathbf{q}}_k$  of  $\delta \mathbf{q}_k^+$  to obtain the current quaternion estimate of the attitude

$$\hat{\mathbf{q}}_k^+ = \hat{\mathbf{q}}_k^- \otimes \delta \mathbf{q}_k^+ \quad (28)$$

## 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 is generated as if the MAV, with virtual mass  $m_v$ , was connected to the desired trajectory via a spring with elastic constant  $K$  and via a damper with damping coefficient  $c$ . The deviation from the desired trajectory is caused by the estimated external force/torque, which excites the virtual mass.

Compliance with external force/torque can be tuned by changing  $K$ . Increasing values of  $K$  simulate a stiffer spring, which in turn yields to better desired trajectory tracking. Conversely,  $K = 0$  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 the differential equation 29 for the generation of the trajectory on  $x_I$ ,  $y_I$ ,  $z_I$  and 30 for the reference value of  $\psi$ .

$$m_v(\ddot{\Lambda}_{d,i} - \ddot{\Lambda}_{r,i}) + c(\dot{\Lambda}_{d,i} - \dot{\Lambda}_{r,i}) + K(\Lambda_{d,i} - \Lambda_{r,i}) = F_i^{ext} \quad (29)$$

$$J_v(\ddot{\Lambda}_d - \ddot{\Lambda}_r) + c(\dot{\Lambda}_d - \dot{\Lambda}_r) + K(\Lambda_d - \Lambda_r) = \tau_z^{ext} \quad (30)$$

The index  $i$  denotes one of the axes  $x_I$ ,  $y_I$ ,  $z_I$ , while  $\Lambda_{d,i}$  represents the desired trajectory on the selected axis and  $\Lambda_{r,i}$  the generated reference trajectory on the selected axis. The parameters  $m_v$ ,  $c$  and  $K$  can be set independently for every axis and represent, respectively, the virtual spring, the virtual damping and the virtual mass of the controller.  $J_v$  represents virtual inertia around the body  $z$  axis. In addition, the controller only takes into account the desired pose by setting  $\ddot{\Lambda}_d = \dot{\Lambda}_d = 0$ , which is equivalent to assuming that the reference is only given in discrete steps. Additionally, in the output trajectory, we set  $\ddot{\Lambda}_r = \dot{\Lambda}_r = 0$  so that the controller only generates a reference for the pose of the MAV.

### C. Robustness to Noise and Offsets

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  $\Lambda_r$ . 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 [14]. Model mismatches and offset in the force/torque estimation can be taken into account by averaging the estimated external forces for a given time  $T_{avg}$  while the helicopter is hovering. The computed offset is subtracted from each force/torque estimate produced by the force/torque estimator.

### D. Landing Detection and Auto-Disengagement

The controller must be engaged 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 the output. The controller is also able to auto-disengage by detecting landing. Landing detection is achieved by checking the magnitude of the estimated external force and its angle  $\theta_L$  with respect to the normal of the plane  $x_I, y_I$ ; landing is detected when the magnitude of  $\mathbf{F}^{ext}$  is bigger than a tunable threshold  $\bar{\mathbf{F}}^{ext}$  and if  $\theta_L$  is smaller than a tunable threshold  $\theta_L$  for a given time  $T_{Landing}$ .

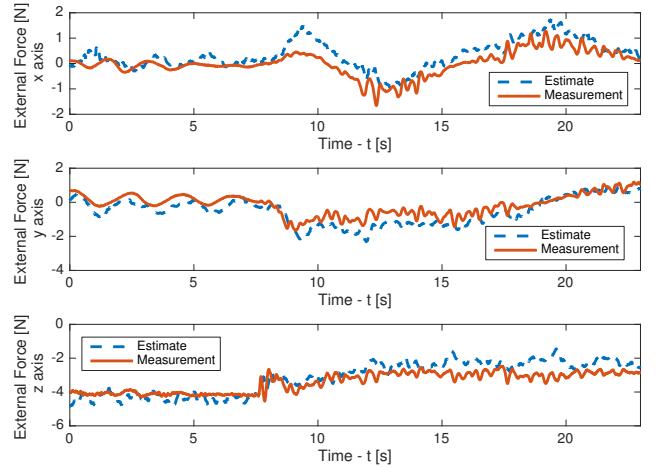


Fig. 5: Measured forces and estimated forces on  $x_I$ ,  $y_I$  and  $z_I$  axis. We performed the experiment with the hexacopter on hover and using the visual inertial navigation system as state estimator.

TABLE I: RMS estimation error for force/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

## VI. EVALUATION

We now present four main experimental results. In the first two, we benchmark the performances of the force/torque estimator by comparing its estimates with the measurements of a force sensor and by showing its reaction speed in detecting a collision with a wall, respectively. In the third experiment, we test the admittance controller by interacting with the vehicle through a rope and finally, in the fourth experiment, we show the result of a payload carrying maneuver using two hexacopters.

### A. Force and torque estimator accuracy

The first experiment was performed by connecting a Opto-force OMD-20-FE-200N force sensor at the bottom of the structure of the AscTech Firefly. 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 VI-odometry state estimator and MPC pose and attitude controller. The external forces due to the interaction measured and estimated on the  $x_I$  and  $y_I$  axis are represented in Figure 5. The RMS error between the measurements and the estimate is reported in Table I.

### B. Detecting Collision with a Wall

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

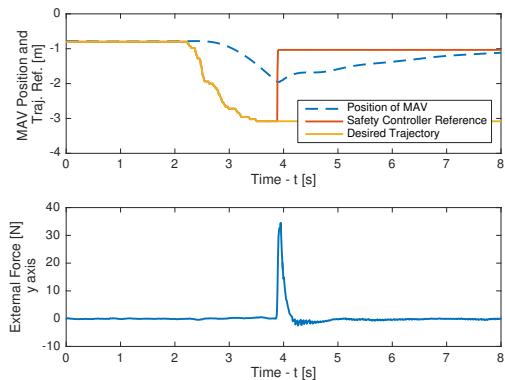


Fig. 6: Wall collision detection experimental results. The wall is placed at  $y_I = -1.95$  m. The impact generates an external force of 34.5 N and a safe trajectory reference is generated 0.84 m further than the impact point. The impact happens at about 1.2 m/s.

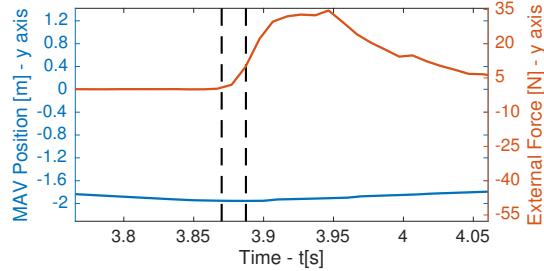


Fig. 7: Detail of the impact detection time computation. The vertical line on the left represents the impact instant  $t = 3.870$  s , while the vertical line on the right the detection instant  $t = 3.887$  s.

highlight the reaction speed of the force estimator, showing that it is able to react in less than 20 ms.

The logic for the trajectory modification is simply based on checking if the estimated external force exceeds a given threshold. If yes, than a safe reference is generated, which is computed by translating the impact point by a vector proportional to the impact force magnitude - computed at the step after the impact - and direction.

In Figure 6 we represent how the filter and the "safety" controller react in case of impact with a wall. Since the wall corresponds to the plane  $y_I = -1.95$  m expressed in  $I$  reference frame, we only show the  $y$  component of (a) desired trajectory, (b) trajectory generated by safety controller, (c) pose of the MAV, estimated using a motion capture system [17], and (d) estimated external force.

### C. Drone on a Leash

To test the capability of admittance controller we present a real-time interaction. A human operator can interact with the MAV via a string attached at the bottom of the hexacopter. The results are represented in Figure 8.



Fig. 8: Illustration of the human-hexacopter interaction. The interaction is performed through a string attached at the bottom of the MAV. It is possible to experiment with different values of virtual spring, mass and damping to understand how they affect the behavior of the slave agent in a collaborative transportation task.

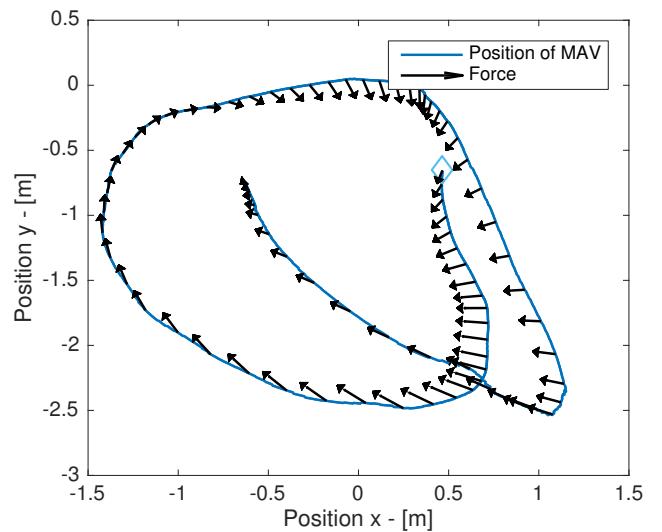


Fig. 9: Executed trajectory by the slave agent. The estimated external forces acting on the slave agent is depicted by the black arrows.

### D. Cooperative payload transportation

Finally we present the cooperative transportation of a bulky object. The payload is a 1.2 m carton tube weighing 0,37 kg. The helicopters are two identical AscTec Firefly systems and the state estimator used is the visual-intertial navigation system (VI sensor). The master helicopter is controlled by a human operator, while the trajectory for the slave vehicle is generated by the admittance controller. Both the helicopters are attached at the extremities of the payload via a nylon wire, which cancels any torque. 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  $x$  and  $y$  axis to be  $K = 0$  N/m, while on the  $z$  a stiff virtual spring of value  $K = 10$  N/m has been used.

The setup of the experiment is represented in Figure 1. The reference trajectory for the slave and its position are represented in Figure 9, while the reference trajectory and the estimated external forces for every axis are shown in Figure 10. Finally, the estimated external torque as well as

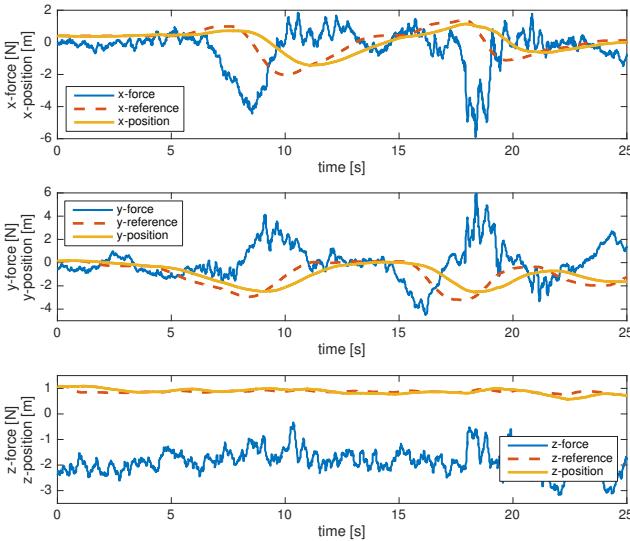


Fig. 10: Estimated external force, admittance controller reference trajectory and estimated position of the slave agent during the collaborative transportation.

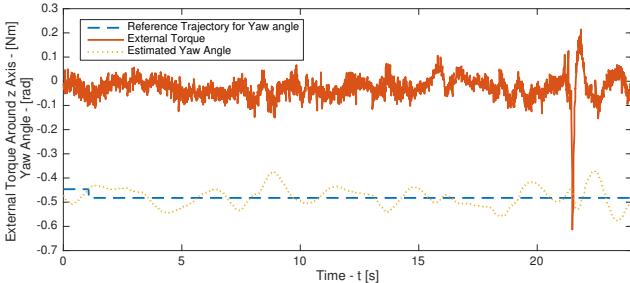


Fig. 11: Estimated external torque around  $z$  axis, reference trajectory for yaw attitude angle, and estimated yaw attitude angle (VI-sensor). The peak in torque estimation at  $t = 22$  s corresponds to the accidental collision of one of the propellers of the MAV with the payload.

the yaw angle are displayed in Figure 11.

## VII. CONCLUSION

In this work we showed that collaborative transportation of 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 estimator based on the Unscented Kalman Filter. The force estimates are obtained using the state information provided by a visual inertial navigation system. We include demonstration of the filter performance by detecting wall collision and real time interaction with a human.

## REFERENCES

- [1] 2016. [Online]. Available: <http://www.io.tudelft.nl/onderzoek/delft-design-labs/applied-labs/ambulance-drone/>
- [2] “Zipline, the future of healthcare is out for delivery, website,” <http://flyzipline.com>, 2016.
- [3] 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.
- [4] F. A. Architecture, “Architecture Installation,” no. August, 2014.
- [5] 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.
- [6] N. Michael, J. Fink, and V. Kumar, “Cooperative manipulation and transportation with aerial robots,” *Autonomous Robots*, no. September 2010, pp. 1–14, 2010.
- [7] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, “Cooperative Grasping and Transport using Multiple Quadrotors-mellingerDARS10.pdf,” pp. 1–14, 2015.
- [8] 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.
- [9] H. N. Nguyen, S. Park, and D. Lee, “Aerial tool operation system using quadrotors as Rotating Thrust Generators,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 1285–1291, 2015.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] L. Villani and J. D. Schutter, *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, E. B. H. Springer-Verlag, Ed., 2008.
- [14] F. Augugliaro and R. D’Andrea, “Admittance control for physical human-quadrocopter interaction,” *Control Conference (ECC), 2013* . . . , pp. 1805–1810, 2013.
- [15] C. D. Mckinnon and A. P. Schoellig, “Unscented External Force Estimation for Quadrotors and Experiments,” *arXiv*, 2016.
- [16] 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.
- [17] “Vicon website,” <http://www.vicon.com>, 2016.
- [18] J. Nikolic, J., R. M. Burri, P. Gohl, S. Leutenegger, P. Furgale, and R. Siegwart, “A Synchronized Visual-Inertial Sensor System with {FPGA} Pre-Processing for Accurate Real-Time {SLAM},” *{IEEE} Int. Conf. on Robotics and Automation (ICRA)*, 2014.
- [19] 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).
- [20] 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.
- [21] 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.
- [22] “Ascending technologies website,” <http://www.asctec.de>, 2016.
- [23] 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.
- [24] D. Simon, *Optimal State Estimation*, 2006, vol. 2. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/19274326>
- [25] S. J. Julier and J. K. Uhlmann, “A General Method for Approximating Nonlinear Transformations of Probability Distributions,” pp. 1–27, 1996.
- [26] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, “Quaternion Averaging,” *NASA Goddard Space Flight Center*, pp. 1–10, 2007.