Physical Human-UAV Interaction with Commercial Drones using Admittance Control

Christopher Banks * **Antonio Bono** ** **Samuel Coogan** ***

* *School of Interactive Computing, Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332, USA (e-mail: cjbanks@gatech.edu).*
** *Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, Via P. Bucci, 42-C, 87036, Rende (CS), Italy (e-mail: antonio.bono@unical.it)*
*** *School of Electrical and Computer Engineering and School of Civil and Environmental Engineering, Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332, USA (e-mail: sam.coogan@gatech.edu)*

Abstract: In this paper, we propose a control scheme that allows generic commercial multirotor Unmanned Aerial Vehicles (UAVs) to infer human intent from physical interaction. Humans apply forces by touching it or pulling a nylon string attached to the body frame of the vehicle. The estimations of these forces, obtained via a square root unscented Kalman filter, are used by an admittance controller to generate a reference trajectory such that the vehicle reacts as a desired mass-damper system. To track this trajectory, the differential flatness property of the multirotor dynamic model is exploited. This, in fact, allows the generation of any possible kind of input required by the on-board controller supplied by the manufacturer. We validate our results experimentally on a Parrot Bebop 2 quadrotor, a commercially available UAV.

Keywords: Physical Human-Robot Interaction; Aerial Robots; Force Estimation; Admittance Control; Differentially Flat Systems.

1. INTRODUCTION

Physical interaction between UAVs and humans is traditionally considered to be dangerous and undesirable. However, due in part to the new mechanical design for safe-to-touch multirotor UAVs (Capunay et al., 2019; Yamada et al., 2019), such interactions are increasingly encouraged and exploited in different applications including search and rescue (Fotokite, 2020), aerial transport (Rastgoftar and Atkins, 2018), and haptic feedback in virtual reality (Knierim et al., 2017; Abdullah et al., 2018). A crucial aspect for deploying any application based on the physical human-UAV interaction is maintaining safety of the operator by ensuring that the UAVs respond as expected to external forces while subject to various constraints in their dynamics and controller design. Existing approaches propose novel, and usually expensive, hardware designs capable of, *e.g.*, directly measuring the applied forces (Rajappa et al., 2017). In this paper, we propose a control scheme for interaction between a human and a multirotor UAV suitable for deployment on commercially available multirotor models. One approach to human multirotor physical interaction is through an admittance control framework. This approach provides a means to modify the effect of an external force applied to a multirotor in order to move it in a safe and effective way. Admittance control is common for manipulation of robotic arms (Ott et al., 2010) and humanoid robots (Caron et al., 2019), and has also been used for multirotor control to generate compliant trajectories. In (Augugliaro and D'Andrea, 2013) physical interaction with a custom built quadrotor is provided by an admittance controller and a Kalman filter that estimates the external forces and the state of a quadrotor in the near-hover configuration. In (Tagliabue et al., 2017) the same approach is exploited for collaborative carrying among appropriately equipped hexarotors. In (Rajappa et al., 2017), a custom quadrotor equipped with a ring of force sensors, is able to distinguish between applied forces and rejected disturbances which are used by an admittance controller through a modified geometric tracking controller. These approaches for physical human-UAV interaction are expensive, require expertise in multirotor design, sensor calibration and system parameter identification techniques, or rely on linearizing the dynamics, which restricts the applicability of the method. This paper presents an approach that addresses the limitations of these previous works and extends the possibility of physical interaction between humans and UAVs to all those commercial models for which the manufacturer provides a Software Development Kit (SDK) that allows the use of the on-board controller (*e.g.*, (DJI, 2020) and (Parrot,

⋆ The first two authors equally contributed to the work.
⋆⋆This work was supported in part by the National Science Foundation under grant #1924978.

2020a)). The proposed control scheme, in fact, does not rely on specialized hardware or additional sensors on-board the UAV. Differential flatness theory, in particular, allows us to use the reference trajectory generated by the admittance controller and its successive derivatives to compute any kind of input required by the on-board controller provided by the manufacturer. As shown later, this is possible because, according to this theory, the *flat outputs* of the multirotor dynamic model coincide with the trajectory generated by the admittance controller. Through a laboratory experiment we demonstrate the use of numerical approximations for the calculation of derivatives of the reference trajectory. Moreover, depending on whether the UAV is safe to touch or not, the interaction may be through direct physical contact or via the attachment of a negligible mass string to the body frame of the UAV. The main contribution of this work is the development of a control scheme, based on mechanical admittance control and differential flatness theory, that allows for the first time a human to physically interact with commercially available multirotors. This is achieved by not using any additional force sensors, motor encoders or specialized hardware. In addition to this, we verify the applicability of the proposed scheme by laboratory experiments. This paper proceeds in the following manner. In Section 2, we introduce the quadrotor model. In Section 3 we define the admittance controller dynamics, discrete derivative filter and derivations of control inputs from the generated flat outputs. Finally, in Sections 4 and 5 we validate our control scheme experimentally on a motion capture testbed and present concluding remarks, respectively.

2. MULTIROTOR MODEL

The mathematical model used in this work is based on the following assumptions:

- The center of gravity (CoG) of the multirotor and the origin of the body-fixed frame coincide.
- The multirotor body is rigid and symmetric with respect to the three planes generated by the body-fixed frame.
- Aerodynamics effects due to blade flapping and induced drag together with the gyroscopic effect are neglected; for commercial multirotors, in fact, such related coefficients are not available (Chovancová et al., 2014).
- Only an external torque on the z-axis of the body-fixed frame is considered.
- No disturbance force, such as wind, is considered.

We label the inertial frame $\{I\}$ and the non-inertial body-fixed frame $\{B\}$, both with right-handed coordinates. As left subscripts, these labels indicate the reference system with respect to which a quantity is expressed (see Fig. 1). To express attitude, the Z-Y-X Euler angle convention is used. Therefore the rotation matrix from $\{B\}$ to $\{I\}$ is ${}^{I}R_{B} = R_z(\psi)R_y(\theta)R_x(\phi)$ with ϕ, θ and ψ representing the roll, pitch and yaw angle, respectively, and $R_z(\cdot)$, $R_y(\cdot)$, and $R_x(\cdot)$ indicating rotation about the body-fixed z, y, and x axes.

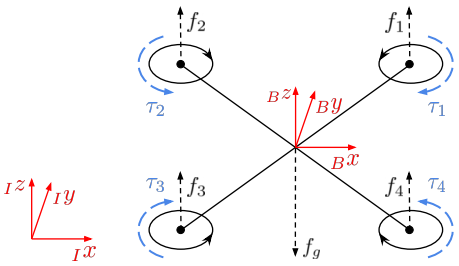


Fig. 1. Schematic representation of the Parrot Bebop 2 quadrotor. The body-fixed frame $\{B\}$ is placed in the "X" configuration with the z-axis in the opposite direction of the gravity force f_g. Rotor rotation directions are shown together with the thrust forces and induced moments.

2.1 Model equations

With the above hypotheses, we write Newton's equation of motion expressing the linear acceleration of the CoG in the $\{I\}$ frame as

$$ {}_{I}\ddot{\boldsymbol{p}} = -{}_{I}\boldsymbol{e}_z\, g + \frac{1}{m}\left({}^{I}R_{B}\, {}_{B}\boldsymbol{e}_z f_t + {}_{I}\boldsymbol{f}^{\text{ext}}\right) \tag{1} $$

where ${}_{I}\ddot{\boldsymbol{p}} = [\,\ddot{x},\ \ddot{y},\ \ddot{z}\,]^T$ is the acceleration; g is the gravity constant; m is the mass of the vehicle; f_t is the total thrust generated by the rotors; ${}_{B}\boldsymbol{e}_z \in \mathbb{R}^3$ is the unit vector along the body-fixed frame z-axis; and ${}_{I}\boldsymbol{f}^{\text{ext}} \in \mathbb{R}^3$ is the vector of the external forces acting on the multirotor.

The angular acceleration, expressed in the $\{B\}$ frame, is determined by the Euler equation

$$ {}_{B}\dot{\boldsymbol{\omega}} = {}_{B}\boldsymbol{J}^{-1}\left(-{}_{B}\boldsymbol{\omega} \times {}_{B}\boldsymbol{J}\,{}_{B}\boldsymbol{\omega} + {}_{B}\boldsymbol{\tau}^{\text{rot}} + {}_{B}\boldsymbol{e}_z\tau^{\text{ext}}\right) \tag{2} $$

where ${}_{B}\boldsymbol{\omega} = [p, q, r]^T$ is the angular velocity vector; ${}_{B}\boldsymbol{\tau}^{\text{rot}} = [\tau_x, \tau_y, \tau_z]^T$ is the vector of the torques generated by the rotors; ${}_{B}\boldsymbol{e}_z\tau^{\text{ext}}$ is the external torque applied around the body-fixed frame z-axis; and ${}_{B}\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the moment of inertia matrix. Since the multirotor is considered symmetric this is a diagonal matrix, i.e., ${}_{B}\boldsymbol{J} = \text{diag}(J_x, J_y, J_z)$.

2.2 System Input

Each rotor has an angular speed ω_i and produces along the rotation axis a thrust force, f_i, and an induced moment, τ_i, according to

$$ f_i = k_f\omega_i^2,\ \tau_i = k_\tau\omega_i^2 \quad i \in \{1, 2, \ldots, n\}, \tag{3} $$

where n is the number of rotors, k_f and k_τ are two aggregate parameters that describes the static thrust force and reaction torque respectively (McCormick, 1994). Since the motor dynamics are much faster than the body dynamics, the motor dynamics are usually ignored for controller design (Mellinger and Kumar, 2011). Then the system input, $\boldsymbol{u} \in \mathbb{R}^4$, is formed by the total thrust and the torques generated around each axis by the rotors. The relationship between the rotor velocities and the input is given by

$$ \boldsymbol{u} = [\,f_t,\ \tau_x,\ \tau_y,\ \tau_z\,]^T = \boldsymbol{\Lambda}\,\boldsymbol{\Omega} \tag{4} $$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times n}$ is known as the allocation matrix and depends on the geometrical design of the multirotor and on the k_f and k_τ parameters; $\boldsymbol{\Omega} = [\omega_1^2,\ \omega_2^2,\ \ldots,\ \omega_n^2]^T \in \mathbb{R}^n$ is the vector with the square of the rotors velocities.

2.3 State-space representation

Considering (1) and (2), a possible state space representation is formed by the position and velocity of the CoG expressed in the $\{I\}$ frame, the attitude parametrized with the Euler angles and the angular velocity vector expressed in the $\{B\}$ frame so that the system state is

$$\begin{aligned}\boldsymbol{s} &= [{}_I\boldsymbol{p}^T, \boldsymbol{\eta}^T, {}_I\dot{\boldsymbol{p}}^T, {}_B\boldsymbol{\omega}^T]^T \\ &= [(x,y,z),(\phi,\theta,\psi),(\dot{x},\dot{y},\dot{z}),(p,q,r)]^T \in \mathbb{R}^{12}.\end{aligned}$$

To complete the state space equations, the Euler angles are obtained according to

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} {}_B\boldsymbol{\omega} \tag{5}$$

which allows large angle deviations from the hover condition. To apply the proposed strategy, a time-discretized version of (1), (2) and (5) is needed. This takes the form

$$\boldsymbol{s}_{k+1} = f\left(\boldsymbol{s}_k, \boldsymbol{u}_k, {}_I\boldsymbol{f}_k^{\text{ext}}, \tau_k^{\text{ext}}, T_s\right) \tag{6}$$

where T_s is the sampling time and we assume f is obtained using the forward Euler discretization method.

3. CONTROL STRATEGY

The design of our control system is mainly driven by its applicability on existing commercial multirotors. Its operation can be described in four steps:

(1) Using the pose of the multirotor and the rotor velocity input, a Square Root Unscented Kalman Filter (SR-UKF)(Van der Merwe and Wan, 2001) estimates the state of the multirotor together with the external force applied to the CoG and the torque on the z-axis of the body-fixed frame.
(2) An admittance controller generates a reference trajectory and its derivative for the position and the yaw angle, in order to shape the behavior of the multirotor as a mass-damper system excited by the estimated force and torque.
(3) A discrete filtered derivative system is applied to compute up to the third derivative of the trajectory.
(4) From differential flatness theory, the input for the multirotor is generated for the on-board controller in order to follow the desired trajectory.

Fig. 2 shows how the different modules are connected.

3.1 Square Root Unscented Kalman Filter for Force and Torque Estimation

Taking inspiration from (Crassidis and Markley, 2003) and (D'Alfonso et al., 2015), we propose the SR-UKF to estimate the external force and torque applied to the multirotor. This version of the unscented Kalman filter guarantees all the advantages of the unscented transformation and at the same time is computationally more efficient than the classic version reported in (Julier and Uhlmann, 1997).

Process model We introduce an augmented state $\bar{\boldsymbol{s}}_k = [\,\boldsymbol{s}_k^T,\ {}_I\boldsymbol{f}_k^{\text{ext}^T},\ \tau_k^{\text{ext}}\,]^T \in \mathbb{R}^{16}$. Considering the force vector and torque update functions

$${}_I\boldsymbol{f}_{k+1}^{\text{ext}} = {}_I\boldsymbol{f}_k^{\text{ext}} \tag{7}$$

$$\tau_{k+1}^{\text{ext}}\,{}_B\boldsymbol{e_z} = \tau_k^{\text{ext}}\,{}_B\boldsymbol{e_z} \tag{8}$$

and joining (6), (7) and (8), we write the process model update function as

$$\bar{\boldsymbol{s}}_{k+1} = \bar{f}(\bar{\boldsymbol{s}}_k, \boldsymbol{\Omega}_k, T_s) + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim (0, \boldsymbol{Q}) \tag{9}$$

with $\boldsymbol{Q} \in \mathbb{R}^{16\times16}$ as the constant process noise covariance matrix, which is a filter tuning parameter. The data of the rotor speeds squared ($\boldsymbol{\Omega}_k$) are used to calculate the input vector $\boldsymbol{u}_k$ by inverting (4) and then update (6) that is part of (9).

Measurement Model We assume the pose of the multirotor is directly measured by an external motion capture system or by a visual-inertia navigation system. Therefore, the measurement model takes the form of the linear equation

$$\boldsymbol{y}_k = \boldsymbol{H}\bar{\boldsymbol{s}}_k + \boldsymbol{v}_k = [\,{}_I\boldsymbol{p}_k^T,\, \boldsymbol{\eta}_k^T\,]^T + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim (0, \boldsymbol{R}) \tag{10}$$

where $\boldsymbol{H} = [\,\boldsymbol{I}_{6\times6}\,|\,\boldsymbol{0}_{6\times10}\,]$ and $\boldsymbol{R} = \text{diag}\left(\boldsymbol{1}_3^T\sigma_p^2, \boldsymbol{1}_3^T\sigma_\eta^2\right)$ is the measurement noise covariance matrix formed by σ_p and σ_η that are, respectively, the position and attitude measurement standard deviations.

Prediction and update steps The prediction step follows the classic equations, while the update step reduces to the standard Kalman filter equations because of the linear measurement model (see (Tagliabue et al., 2017)), further reducing the computational load.

3.2 Generation of Desired Trajectory via Admittance Control

Admittance control is a popular technique in the field of human-robot interaction (Keemink et al., 2018). It allows a human operator to modulate the interaction behavior by controlling robot motion according to the estimated or sensed force exerted by the human. Admittance control is applicable to human-multirotor interaction where the relationship between motion and force variables is controlled by imposing a virtual inertia, damping and stiffness to the multirotor about each axis of movement and the yaw rotation axis. In our case, this leads to two equations. For the linear movements

$$\boldsymbol{M}\,{}_I\ddot{\boldsymbol{p}}_{\boldsymbol{r}} + \boldsymbol{C}\,{}_I\dot{\boldsymbol{p}}_{\boldsymbol{r}} = {}_I\hat{\boldsymbol{f}}^{\,\text{ext}} \tag{11}$$

where ${}_I\boldsymbol{p}_{\text{r}}$ is the reference motion resulting from the interaction; and $\boldsymbol{M} = \text{diag}(M_x, M_y, M_z)$ and $\boldsymbol{C} = \text{diag}(C_x, C_y, C_z)$ define, respectively, the virtual inertia and damping. Similarly for the rotation around the z axis of $\{B\}$,

$$J_\psi\ddot{\psi}_r + C_\psi\dot{\psi}_r = \hat{\tau}^{\,\text{ext}} \tag{12}$$

where ψ_{r} is the reference yaw angle and J_ψ, C_ψ are, respectively, the virtual inertia and damping. In our control scheme, however, we have a discrete time estimation of the force (${}_I\hat{\boldsymbol{f}}_k^{\,\text{ext}}$) and torque ($\hat{\tau}_k^{\,\text{ext}}$). For this reason, (11) and (12) are discretized via the forward Euler method. Finally, we can introduce

$$\boldsymbol{r} = [\,r_x,\, r_y,\, r_z,\, r_\psi\,]^T \triangleq [\,{}_I\boldsymbol{p}_{\text{r}}^T,\, \psi_{\text{r}}\,]^T \tag{13}$$

to define the desired trajectory.

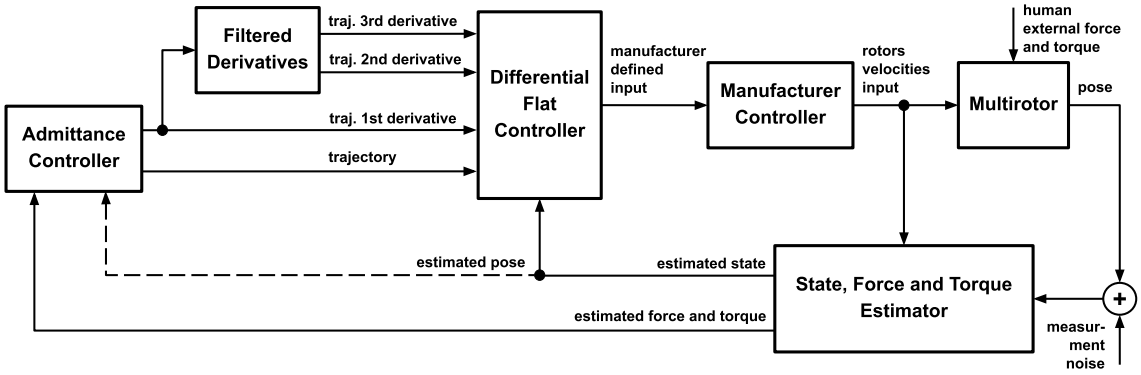


Fig. 2. The full control scheme of our proposed admittance control framework. A human operator acts on the multirotor with an external force and torque. The estimations of these quantities update the admittance controller that generates a reference trajectory such that the vehicle reacts as a desired mass-damper system. This trajectory and its derivatives calculated by discrete derivative filters are used to generate the manufacturer-defined input from the differentially flat multirotor dynamics. The dashed line, finally, represents the pose of the vehicle used to initialize the admittance controller.

3.3 Discrete filtered derivative

In order to use a controller based on the differential flatness property of the multirotor model, the derivatives of the desired trajectory need to be continuous up to the third order. In our case, however, the trajectory is generated online via the admittance controller and, thus, an analytical continuous trajectory $\boldsymbol{r} \in \mathcal{C}^3$ is not available. To overcome this issue, we numerically compute the second and third derivative of the references generated by the admittance controller. We avoided the use of the analytical second derivative given by the admittance controller because of its direct relationship with the estimated force in (11). This, in fact, makes the analytical derivative sensitive to the oscillations of the estimation thus worsening the numerical computation of the third derivative (see Fig. 3). Equations (11) and (12) describe four independent second-order dynamical systems. The state, for each of them, is formed by the physical quantity described and its first derivative. Starting from this derivative, we twice compute a filtered numerical derivative using the transfer function

$$D(s) = \frac{s}{Ts + 1} \tag{14}$$

where the time constant T is properly tuned to balance the smoothness and delay of the output. A discrete-time version of (14) is used in our control scheme.

Remark 1. This approach does not formally guarantee the continuity of the generated derivatives that is required by the differential flatness theory. However, properly choosing the time constant T and the sampling time T_s in our simulation tests like the one reported in Fig. 3, demonstrates behavior approximate to that of continuous functions.

3.4 Dynamic Control Input Generation via Differentially Flat Outputs

Differential flatness is a well known property of many dynamical systems (Sira-Ramirez and Agrawal, 2018; Murray et al., 1995). This property allows the expression of the full state and the input of the system as functions of the flat output and its time derivatives. In other words, $\boldsymbol{h} = g_h(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(j)})$ is considered a flat output for

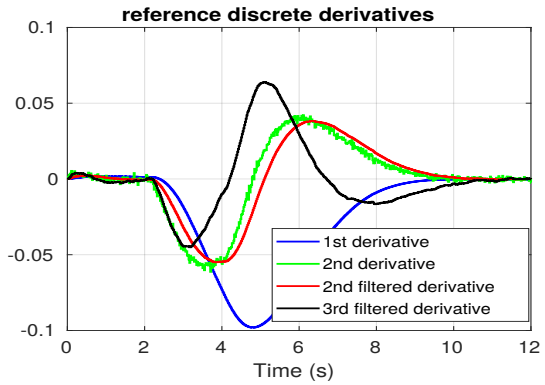


Fig. 3. A simulation of the discrete-time version of the proposed derivative filter using $T_s = 0.01\,\mathrm{s}$ and $T = 0.3$. The admittance controller reference is not shown to better visualize the derivatives.

j finite derivatives of input $\boldsymbol{u}$ if there exist two functions $\boldsymbol{s} = g_s(\boldsymbol{h}, \boldsymbol{h}^{(1)}, , \ldots, \boldsymbol{h}^{(l)})$ and $\boldsymbol{u} = g_u(\boldsymbol{h}, \boldsymbol{h}^{(1)}, \ldots, \boldsymbol{h}^{(m)})$ such that both the state ($\boldsymbol{s}$) and the input ($\boldsymbol{u}$) can be uniquely expressed as functions of the specified output and its l and m finite derivatives, respectively.

As shown in (Mellinger and Kumar, 2011) and (Ferrin et al., 2011), for the multirotor model described in Section 2, a possible choice of the flat output is $\boldsymbol{h} = [x, y, z, \psi]^T$ that allows the inference of the complete state ($\boldsymbol{s}$) and input ($\boldsymbol{u}$) with the appropriately defined functions. This means that by using the trajectory generated by the admittance controller $\boldsymbol{r} = [r_x, r_y, r_z, r_\psi]^T$ and its derivatives, it is possible to deduce the full state corresponding to the particular reference.

In the next section, we show how we leverage differential flatness theory to control a particular type of commercially available quadrotor. However, we note that this technique can be applied to any multirotor provided the desired inputs can be generated from our choice of flat outputs.

4. EXPERIMENTAL RESULTS

Experiments are conducted on a Parrot Bebop 2 quadrotor UAV. Considering the rotors rotation directions and the

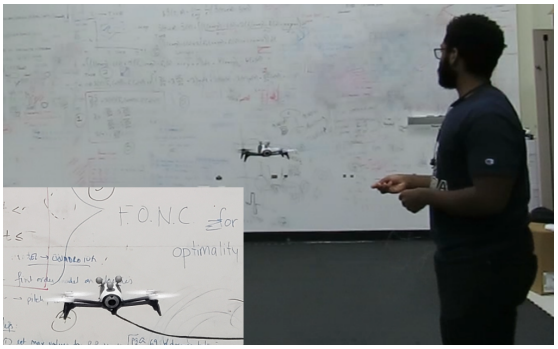


Fig. 4. An image captured of the experimental setup. A user interacts with the Bebop 2 quadrotor via a string attached to the body frame.

"X" configuration shown in Fig. 1, the allocation matrix is

$$\mathbf{\Lambda} = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2}\,k_f & \sqrt{2}\,k_f & \sqrt{2}\,k_f & \sqrt{2}\,k_f \\ -d\,k_f & -d\,k_f & d\,k_f & d\,k_f \\ -d\,k_f & d\,k_f & d\,k_f & -d\,k_f \\ -\sqrt{2}\,k_\tau & \sqrt{2}\,k_\tau & -\sqrt{2}\,k_\tau & \sqrt{2}\,k_\tau \end{bmatrix} \tag{15}$$

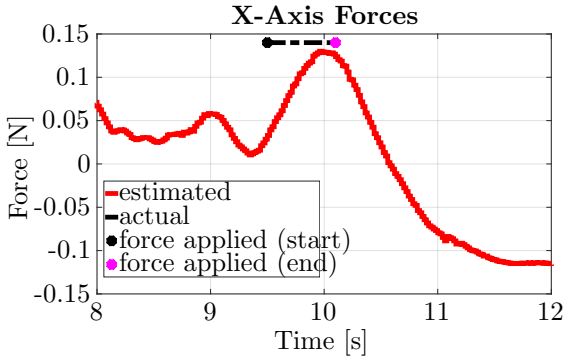
where d is the distance from the CoG to each rotor. All the physical parameters are retrieved from the code of the Parrot-Sphinx simulation tool (Parrot, 2020b) and can be viewed in (Silano et al., 2019). As defined above, standard control inputs for multirotor control are the thrust and torque vector. However, our system relies on the the ROS `bebop_autonomy` package that is based on the Parrot official `ARDroneSDK3` (Monajjemi, 2020; ARDrone, 2020) and receives as input the vector $\boldsymbol{\nu} \triangleq [\phi_d, \theta_d, \dot{\psi}_d, {}_I\dot{z}_d]$ whose components are the desired roll angle, pitch angle, yaw rate and vertical velocity in the inertial frame. Using differential flatness, these quantities are derived as the following: $\phi_d = \text{atan2}\left(\beta_2, \sqrt{\beta_1^2 + \beta_3^2}\right)$, $\theta_d = \text{atan2}\left(\beta_1, \beta_3\right)$, $\dot{\psi}_d = \dot{r}_\psi$ and ${}_I\dot{z}_d = \dot{r}_z$ where we define $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta}(\boldsymbol{r}, \dot{\boldsymbol{r}}, \ddot{\boldsymbol{r}}) = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \triangleq \begin{bmatrix} -\cos(r_\psi)\,\ddot{r}_x - \sin(r_\psi)\,\ddot{r}_y \\ -\sin(r_\psi)\,\ddot{r}_x + \cos(r_\psi)\,\ddot{r}_y \\ -\ddot{r}_z + g \end{bmatrix}. \tag{16}$$

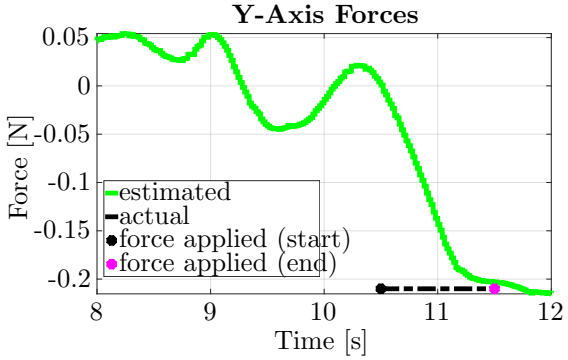
To improve the performance of the control loop, we add a feedback term governed by four PID controllers. Our entire input vector is then

$$\boldsymbol{\nu} = \begin{bmatrix} \phi_d \\ \theta_d \\ \dot{\psi}_d \\ {}_I\dot{z}_d \end{bmatrix} + \mathbf{K_P} \begin{bmatrix} e_x \\ e_y \\ e_\psi \\ -e_z \end{bmatrix} + \mathbf{K_I} \begin{bmatrix} \int e_x \\ \int e_y \\ 0 \\ -e_z \end{bmatrix} + \mathbf{K_D} \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

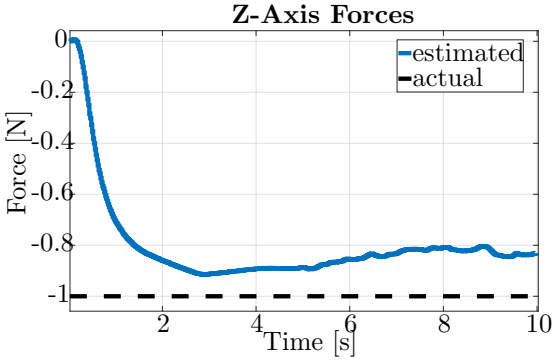
where $\mathbf{K_P}$, $\mathbf{K_I}$ and $\mathbf{K_D}$ are positive gain diagonal matrices and $e_x \triangleq r_x - x$ where x is the position of the quadrotor returned via a motion capture system, and e_y, e_ψ and e_z are defined similarly. In this experiment, the reading frequency of the rotor speed commands limited the update frequency of the filter, thus degrading its performance. For this reason, in practice, we use the motion capture system measurements. As mentioned previously in the presented scheme, we require time discretization of this subsystem. The derivatives of the reference $\boldsymbol{r}$ are substituted with the discrete time numerically computed versions. All computations are done on a PC workstation and the control input vector, $\boldsymbol{\nu}$, is transmitted to the Bebop 2 quadrotor via a ROS topic over Wi-Fi. In the following, we first report the validation of the SR-UKF and then we show the tracking performance of the proposed control scheme.



(a) Estimation of force on the x-axis exerted by a spring with spring constant($k = 7\,\text{N m}$), displaced by 20 mm. A force is applied for 0.6 seconds until the spring is displaced a desired amount.



(b) Estimation of force on the y-axis exerted by a spring with spring constant($k = 7\,\text{N m}$), displaced by 30 mm. A force is applied for 1 second until the spring is displaced a desired amount.



(c) Estimation of the force exerted by a mass of 0.1 kg attached near the CoG of to the Bebop 2 quadrotor via a nylon string.

Fig. 5. Validation of the UKF Force Estimation on the X, Y, and Z axes of the Bebop quadrotor. The actual force applied is shown via the dashed line. The start (black) and end (purple) points indicate the times where a force is applied. Estimated forces for the X, Y, Z axes are generated from the UKF and are shown via the red, green and blue curves.

4.1 Estimating Force on the ${}_Ix$, ${}_Iy$, and ${}_Iz$ axes

With the aim of making this strategy accessible to as many users as possible, we avoided the use of expensive three-

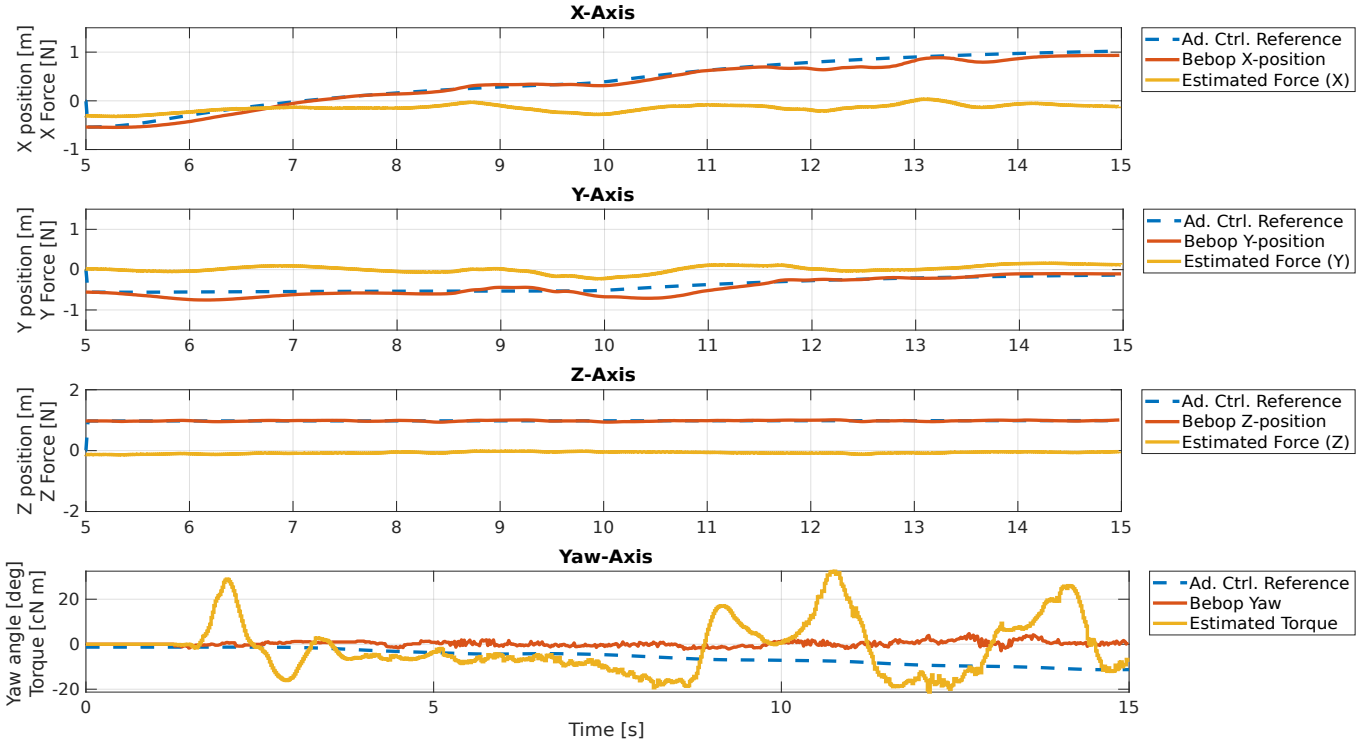


Fig. 6. Position results from human-UAV interaction. This figure shows the resulting trajectory of the Bebop 2 quadrotor from external force and torque manipulation and the estimated forces and torque that correspond to the trajectories. The rows indicate the reference positions generated from the admittance controller starting at $t = 5\,\mathrm{s}$ (blue), measured Bebop positions from the Optitrack motion capture system (orange) and estimated forces and torque (yellow). We see that the Bebop state (orange) generally tracks the admittance control reference (blue), as desired. Note that the y-axis contains both the position unit (meters) and forces/torque units (N, N/m) thus scaling the estimated quantities (forces and torque).

axis force sensors for filter validation. To compare the force estimated by the SR-UKF on the ${}_I x$ and ${}_I y$ axes, we used a spring with known spring constant $k = 7\,\mathrm{N\,m}$. We displace the spring by 20 mm and 30 mm on the ${}_I x$ and ${}_I y$ axes, respectively. After tuning, the Normalized Root Mean-Squared Error (NRMSE) between the actual and estimated force is 9.7 % on the ${}_I x$-axis and 1.6 % on the ${}_I y$-axis. To compare the estimation obtained by the SR-UKF on the ${}_I z$-axis we used as a known force a mass of 0.1 kg attached near the CoG of the Bebop 2 via a nylon string. After a tuning process, the NRMSE between the actual and estimated force is around 30%. The evolution of the estimated force is reported in Fig. 5.

4.2 Tracking Human Input

Description In this experiment, a human applies a force and torque to the Bebop 2 by pulling on the nylon string attached to the body far enough from the CoG to exert desired forces and torque on the ${}_B z$-axis. After some tests, we found that the following values of virtual inertia and damping provide adequate behavior of the quadrotor while following the human input: $(M_x, M_y, M_z, J_\psi) = (0.5, 0.5, 5, 7)$ and $(C_x, C_y, C_z, C_\psi) = (0.2, 0.2, 0.2, 0.5)$.

Force estimation noise If no force is exerted by the human operator the admittance controller does not generate new references and the multirotor remains in the same position. However, the simplifying assumptions of the model presented in Section 2 and noise result in non-zero estimated forces and torques. To overcome this issue, the estimated values are sent to the admittance controller only if they exceed their relative threshold values (in this experiment only $\|{}_I\hat{\boldsymbol{f}}^{\,\mathrm{ext}}\| \geq 0.3\,\mathrm{N}$ and $|\hat{\tau}^{\,\mathrm{ext}}| \geq 0.1\,\mathrm{N\,m}$ are considered) and otherwise a zero value is sent.

Results In Fig. 6 we show the reference trajectory generated by the admittance controller and the actual trajectory of the quadrotor. The developed admittance control scheme generates trajectories and the Bebop 2 follows these trajectories within a reasonable error for forces and torques above the designated thresholds. We provide the RMSE values for each axis in Table 1. Also, note that the behavior of the desired reference trajectories of the quadrotor can be modified by adjusting the virtual mass and dampening in the admittance controller.

Table 1: Root Mean Square Error (RMSE) values

Trajectory RMSE Values	
RMSE_x	0.0829
RMSE_y	0.1041
RMSE_z	0.0159
RMSE_ψ	0.27

5. CONCLUSIONS

This paper presents a novel approach for human-UAV physical interaction based on the differential flatness prop-

erty of the multirotor dynamics and the mechanical admittance control. This strategy, for the first time, allows users to physically manipulate commercially available multirotors without special sensors on-board the multirotor or specific aerodynamic parameters that require expensive dedicated equipment to be determined. Moreover, we leverage the full non-linear model of a multirotor to generate control inputs from the flat outputs, thus avoiding model linearization. We experimentally validated our results on a testbed with a motion capture system on the Parrot Bebop 2 quadrotor UAV.

REFERENCES

Abdullah, M., Kim, M., Hassan, W., Kuroda, Y., and Jeon, S. (2018). Hapticdrone: An encountered-type kinesthetic haptic interface with controllable force feedback: Example of stiffness and weight rendering. In *2018 IEEE Haptics Symposium (HAPTICS)*, 334–339. doi: 10.1109/HAPTICS.2018.8357197.

ARDrone (2020). ARDroneSDK3 API reference. https://developer.parrot.com/docs/SDK3/.

Augugliaro, F. and D'Andrea, R. (2013). Admittance control for physical human-quadrocopter interaction. In *2013 European Control Conference (ECC)*, 1805–1810.

Capunay, A., Valdenegro, D., Gonzalez Masso, D.A., and Garcia Carrillo, L.R. (2019). Bladeless unmanned aerial vehicle, Patent n. 20190127065, USA.

Caron, S., Kheddar, A., and Tempier, O. (2019). Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control. In *2019 International Conference on Robotics and Automation (ICRA)*, 277–283.

Chovancová, A., Fico, T., Ľuboš Chovanec, and Hubinsk, P. (2014). Mathematical modelling and parameter identification of quadrotor (a survey). *Procedia Engineering*, 96, 172–181. doi: https://doi.org/10.1016/j.proeng.2014.12.139.

Crassidis, J.L. and Markley, F.L. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 26(4), 536–542. doi: 10.2514/2.5102.

DJI (2020). DJI development platform webpage. https://developer.dji.com/onboard-sdk/.

D'Alfonso, L., Lucia, W., Muraca, P., and Pugliese, P. (2015). Mobile robot localization via ekf and ukf: A comparison based on real data. *Robotics and Autonomous Systems*, 74, 122 – 127. doi: https://doi.org/10.1016/j.robot.2015.07.007.

Ferrin, J., Leishman, R., Beard, R., and McLain, T. (2011). Differential flatness based control of a rotorcraft for aggressive maneuvers. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2688–2693.

Fotokite (2020). The fotokite sigma. a situational awareness system for first responders. https://fotokite.com/situational-awareness-system/.

Julier, S.J. and Uhlmann, J.K. (1997). New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, 182–193.

Keemink, A.Q., van der Kooij, H., and Stienen, A.H. (2018). Admittance control for physical human–robot interaction. *The International Journal of Robotics Research*, 37(11), 1421–1444. doi: 10.1177/0278364918768950.

Knierim, P., Kosch, T., Schwind, V., Funk, M., Kiss, F., Schneegass, S., and Henze, N. (2017). Tactile drones-providing immersive tactile feedback in virtual reality through quadcopters. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 433–436.

McCormick, B.W. (1994). *Aerodynamics, Aeronautics, and Flight Mechanics*. Wiley, 2 edition.

Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, 2520–2525.

Monajjemi, M. (2020). autonomy_package, a ros driver for parrot bebop drone, based on parrot's official ardronesdk3. http://wiki.ros.org/bebop_autonomy.

Murray, R.M., Rathinam, M., and Sluis, W. (1995). Differential flatness of mechanical control systems: A catalog of prototype systems. In *ASME international mechanical engineering congress and exposition*.

Ott, C., Mukherjee, R., and Nakamura, Y. (2010). Unified impedance and admittance control. In *2010 IEEE International Conference on Robotics and Automation*, 554–561.

Parrot (2020a). Parrot software developlemt kit webpage. https://developer.parrot.com/.

Parrot (2020b). Parrot-Sphinx simulation tool guide book. https://developer.parrot.com/docs/sphinx/index.html.

Rajappa, S., Bülthoff, H., and Stegagno, P. (2017). Design and implementation of a novel architecture for physical human-uav interaction. *The International Journal of Robotics Research*, 36(5-7), 800–819. doi: 10.1177/0278364917708038.

Rastgoftar, H. and Atkins, E.M. (2018). Cooperative aerial payload transport guided by an in situ human supervisor. *IEEE Transactions on Control Systems Technology*, 27(4), 1452–1467.

Silano, G., Oppido, P., and Iannelli, L. (2019). Software-in-the-loop simulation for improving flight control system design: a quadrotor case study. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 466–471.

Sira-Ramirez, H. and Agrawal, S.K. (2018). *Differentially flat systems*. Crc Press.

Tagliabue, A., Kamel, M., Verling, S., Siegwart, R., and Nieto, J. (2017). Collaborative transportation using mavs via passive force control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5766–5773.

Van der Merwe, R. and Wan, E.A. (2001). The square-root unscented kalman filter for state and parameter-estimation. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 6, 3461–3464 vol.6.

Yamada, W., Manabe, H., and Ikeda, D. (2019). Zerone: Safety drone with blade-free propulsion. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–8.