A Geometric Pulling Force Controller for Aerial Robotic Workers

David Wuthier Dariusz Kominiak Emil Fresk George Nikolakopoulos

Robotics Group at the Control Engineering Division of the Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden

Abstract: The aim of this article is to establish a geometric, pulling force control scheme in order to enable the concept of Aerial Robotic Workers (ARWs), where the capabilities of the Unmanned Aerial Vehicles (UAVs) are enhanced by aerial manipulators in order to exert known pulling forces on the environment, with characteristic applications such as levers actuation, debris removal and safety assessments. The proposed novel approach consists of interfacing a cascaded position control scheme with a manipulation framework in such a way that the UAV, together with the manipulator, are being controlled in a complete system. The validity of the proposed scheme as well as the ability of the UAV to track a desired pulling force is validated through a real-world experiment.

Keywords: flying robots, robots manipulators, field robotics, aerial manipulation, unmanned aerial vehicles

1. INTRODUCTION

Aerial Robotic Workers (ARWs) arise as a new generation of Unmanned Aerial Vehicles (UAVs), capable of physically interacting with the environment, by endowing them with manipulators (Wuthier et al., 2016). These systems appear as a leap forward in the area of mobile manipulation, as they are exceedingly expanding the available workspace. Also, ARWs offer a viable solution to the problem of performing physical tasks remotely, with a maximum range and a minimum reaching time, while establishing a 'safe for humans' working environment in critical operations, as at a nuclear power plant in case of catastrophe, urgent maintenance task on wind turbines, safety assessment of newly blast-generated voids into mines, to name a few.

Ideally, the flying platform would provide the manipulator with an aerial anchor, similarly to a fixed-to-the-ground base in term of reaction forces (Featherstone and Orin, 2008). More realistically, this platform is required to counteract a force and a torque along each of its six Degrees-of-Freedom (DOFs), which is actually not the case with common rotor configurations, these latter being under-actuated with their linear accelerations in the horizontal plane depending on their tilting angles. To overcome this drawback, holonomic rotor configurations have been designed in Jiang and Voyles (2013) and Ryll et al. (2015). However, such configurations imply an impaired energy efficiency due to competing thrust directions, and the adaptation on a larger scale to common main-tail-rotors configurations is not straightforward anymore. It is therefore convenient to be able to perform manipulation tasks from an under-actuated platform, be it for exerting forces (Fumagalli et al., 2014), or torques (Korpela et al., 2014). Recent work has shown astounding force exertion capabilities when aligning this interaction force with the Center Of Gravity (COG) of the UAV (Hoekstra et al., 2017), thus getting rid of significantly destabilizing effects. Once again, this allowance is not necessarily available on any ARW. Kondak et al. (2014) have demonstrated a pulling-out task from a main-tail-rotor helicopter by relying both on a kinematic decoupling between the end-effector’s pose and the UAV’s pose, and a dynamic decoupling compensating for the forces and torques applied on the fuselage through the main rotor torques. A 6 DOFs force/torque sensor was utilized at the interface between the manipulator and the UAV. This approach proposes to compensate for the applied forces through position control, hence allowing no position to be tracked without static error while applying any force.

* This work has received funding from the European Unions Horizon 2020 Research and Innovation Program under the Grant Agreement No.644128, AEROWORKS
Thus, the aim of this article is to advance the current state of the art by the following four novel contributions: a) exhibit holonomic behavior with an under-actuated flying platform, taking advantage of the supplemental DOFs brought by the manipulator, b) design a pulling force controller that deals with the misalignment between the applied force and the UAV’s COG, c) avoid the utilization of any force/torque sensor, and d) allow this controller to track a pulling force without suffering from any static error in positioning.

The rest of the article is structured as it follows. Initially, essential definitions and notation are provided in Section 2. The proposed control architecture, intended to demonstrate a novel concept of aerial manipulation, while relying on geometric control, is exposed in Section 3. The implementation of this control structure for the ARW illustrated on Fig. 1, with corresponding experimental results, are presented in Section 4. Finally, directions for future work are given and conclusions are drawn in Section 5.

2. DEFINITIONS AND NOTATION

Consider an UAV as depicted in Fig. 2, where the world frame \( \{x^W, y^W, z^W\} \) is fixed to the ground, with \( z^W \) pointing upward, aligned with gravity. The body frame \( \{x^B, y^B, z^B\} \) is fixed to the UAV, with its origin on the UAV’s COG \( C \). \( x^B \) points forward, \( z^B \) points upward and, in order to get a right-handed frame, \( y^B \) points to port. The UAV’s pose, consisting of its position together with its attitude, is expressed by the rigid body transformation from the body frame to the world frame. The attitude transformation from body to world frame is expressed using a rotation matrix \( R_B^W \in SO(3) \), with the special orthogonal group

\[
SO(3) = \{ R \in \mathbb{R}^{3 \times 3} | R^T R = I, \ det \ R = 1 \}. \tag{1}
\]

Otherwise, for the sake of qualitative description, Euler angles with rotation sequence \((1, 2, 3)\) as in Diebel (2006) offers another convenient representation. Accordingly, when referring to a transformation from the body to the world frame, the roll, pitch and yaw angles respectively correspond to a rotation around \( x^W \), then around \( y^W \) and finally around \( z^W \).

The rates corresponding to a pose are designated by a twist, which is therefore the combination of linear and angular velocities. An odometry consist of a pose together with a twist. Also, a wrench refers to a force together with a torque. The notation below is followed throughout the rest of this article:

- \( (\cdot)^C \) superscript, corresponding coordinate frame \( C \)
- \( (\cdot)_p \) capital subscript, corresponding point \( P \)
- \( u_i \) unit vector along axis \( i \)
- \( p \in \mathbb{R}^3 \) position
- \( r_{AB} \in \mathbb{R}^3 \) vector from point \( A \) to point \( B \)
- \( R_A^B \in SO(3) \) rotation matrix from frame \( A \) to frame \( B \)
- \( F \in \mathbb{R}^3 \) force
- \( T \in \mathbb{R}^3 \) torque

In the present framework, a robot is said to be holonomic if and only if its actuators allow it to control each component of its pose independently (according to its base link). For example, an UAV equipped with six rotors with such a configuration generating forces and torques along each of its three body frame axis, able to track a given position with a given attitude (within a specified envelope). Conversely, a robot whose actuators do not allow it to control each of its six rigid body DOFs independently is said to be under-actuated. For example, an UAV with the common quadrotor configuration, having its horizontal linear accelerations depending on its tilting angles, therefore relying on these angles for hovering over a specific spot.

3. CONTROL ARCHITECTURE

Accordingly, as depicted on Fig. 3a, an ARW consisting of an under-actuated UAV endowed with a manipulator remains under-actuated as long as its base link is considered to be the UAV frame. Redefining its base link as the first link of the manipulator (Fig. 3b), while assuming an universal joint (2 DOFs) as the base joint, is now enabling it to track every pose and to provide every reaction wrench to the rest of the manipulator. The novel control architecture represented on Fig. 4 finds its roots in this simple shift of representation. However, due to practical considerations, a planar manipulator is assumed here, while preserving the ability to demonstrate holonomicity in the \( x^B z^B \)-plane.

3.1 Navigation

When navigating, hence not interacting with the environment, a reference position is fed to the position controller,
Fig. 4. Proposed control architecture, consisting of the UAV position controller established by Lee et al. (2010), extended with a pulling force controller (dashed lines).

Fig. 5. Block diagram representing the pulling controller. which will compare it to the estimated position to run a PD control algorithm outputting a force command in the world frame. This command is translated into desired attitude and thrust commands, which are then tracked by the attitude controller running another PD control algorithm on the manifold of attitude. There comes also the reference for the yaw angle. The resulting outputs are torque and thrust commands in the body frame, that the *mixing* converts into rotor velocities for operating the UAV. The available sensors are assumed to be an Inertial Measurement Unit (IMU) together with a motion capture system (providing absolute pose measurements). These sensor data are fused by an Extended Kalman Filter (EKF) for estimating the current odometry information.

3.2 Interaction

When interacting, that is to say applying a force on the environment through the end-effector, a reference force and a target location are given to the pulling controller, which then accordingly:

- Sets reference joint positions for the manipulator
- Uses these measured joint positions to update the mixing matrix (see Appendix A) with a new COG estimation
- Superimposes a force and a torque respectively upstream and downstream the attitude controller
- Inhibits position and yaw gains

An inner representation of the pulling controller block is provided on Fig. 5. When a non-zero reference force is set, position and yaw gains are zeroed by the gains inhibitor. Note the velocity gains are not affected, as they damp the system. The manipulator’s joint positions are computed by an inverse kinematics algorithm, which takes into account the target’s location together with the estimated pose to determine which base joint angle allows to bring the end-
effector as close as possible to the target, keeping the rest of the manipulator straight. An update on the current COG is obtained from a COG kinematics algorithm, which relies on the measured manipulator’s joint positions. Note that combining the inverse kinematics and the COG kinematics is assumed to make the arm equivalent to a massless rope binding the base joint to the handle.

3.3 The manipulation controller

The manipulation controller is the element providing a force command to the UAV attitude controller, based on the reference force $F_e$, and odometry information. With regard to Fig. 6, it first defines a manipulation frame $M$ as a rotation matrix;

$$R_W^M = R_v(-F_e),$$

with $R_v : \mathbb{R}^3 \rightarrow SO(3)$,

$$R_v(v) = \begin{bmatrix} v & u_z \times v & u_z \times v \\ \parallel v & \parallel u_z \times v & \parallel u_z \times v \end{bmatrix},$$

yielding the rotation matrix having its first column pointing in the same direction as the vector $v$ and zero roll angle. Note that $R_v$ requires $v$ not to be parallel with $u_z$. Then, the arm frame is defined in a similar manner:

$$R_A^W = R_v(r_{BT}).$$

$R_A^W$ and $R_W^M$ can now be considered as desired and current orientations (respectively) for computing an error vector as in Lee et al. (2010):

$$e_{AM}^W = \frac{1}{2} (R_W^M R_W^A - R_W^M R_A^W)^v,$$

with the vee map $\cdot^v : so(3) \rightarrow \mathbb{R}^3$ defined as

$$S^v = \begin{bmatrix} 0 & s_{12} & s_{13} \\ -s_{12} & 0 & s_{23} \\ -s_{13} & -s_{23} & 0 \end{bmatrix} = \begin{bmatrix} -s_{23} \\ s_{13} \\ -s_{12} \end{bmatrix},$$

identifying a skew symmetric matrix $S$ from the Lie algebra $so(3)$ to a vector in $\mathbb{R}^3$. This error vector is then turned into an angular acceleration around $T$, $\alpha_T^W$, by making use of proportional gains $K_p^M \in \mathbb{R}^3$:

$$\alpha_T^W = \text{diag}(K_p^M) e_{AM}^W.$$ (7)

Finally, the force command $F_e$ to be applied on the UAV by the attitude controller is obtained by combining the force on point $C$ corresponding to $\alpha_T^W$ (knowing the UAV’s mass $m_C$) and an arm traction force with magnitude $\|F_e\|$:

$$F_e = m_C \alpha_T^W \times r_{TC} - \|F_e\| R_A^W.$$ (8)

3.4 The torque compensator

The torque compensator is the block preventing the system to run into instability while encountering an interaction force misaligned with the UAV’s COG. It is based on the assumption of static force equilibrium. In the body frame, we know the gravitational force $F_B^g = R_W^B F_W^g$, the lateral direction by definition $e_x$, the thrust vector orientation by definition $e_z$ and the unit vector pointing in the arm’s direction $u_{BT}$, the base joint’s force magnitude $f_B \in \mathbb{R}$ can be derived from force equilibrium:

$$f_B \begin{bmatrix} u_{BTx}^B \\ u_{BTy}^B \\ u_{BTz}^B \end{bmatrix} + f_t \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + f_l \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} p_{Bx}^g \\ p_{By}^g \\ p_{Bz}^g \end{bmatrix} = 0,$$

with the thrust magnitude $f_t \in \mathbb{R}$ and a fictitious lateral force magnitude $f_l \in \mathbb{R}$. Thus, $F_B$ is obtained straightforwardly as

$$F_B = f_B^B \begin{bmatrix} u_{BTx}^B \\ u_{BTy}^B \\ u_{BTz}^B \end{bmatrix}.$$ (9)

The compensation torque to be generated by the rotors is finally obtained by taking into account the lever arm of $F_B^B$ over the COG,

$$T_e^B = -p_{Bx}^g \begin{bmatrix} R_W^B \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} u_{BTx}^B \\ u_{BTy}^B \\ u_{BTz}^B \end{bmatrix} = f_B(u_{BTx}^B p_{Bz}^g - u_{BTz}^B p_{By}^g).$$ (10)

Note that $u_{BTx}^B = 0$ and $p_{Bx}^g = 0$ assume a planar arm evolving in the $x^B z^B$-plane. Also, the base joint is assumed to be mounted underneath the COG, so that $T_e^B$ exhibits a stabilizing effect by tending to decrease the pitch angle when tilting.

4. EXPERIMENTS AND RESULTS

As shown on Fig. 1, the proposed control architecture has been experimentally evaluated 1 within the Field Robotics Laboratory of the Luleå University of Technology (LTU-FROST). The whole scheme was implemented into the Robotic Operating System (ROS), taking advantage of its intrinsic modularity to evaluate each controller through extensive GAZEBO simulations before embedding these on the flying platform itself. The RotorS micro aerial vehicles simulator from Furrer et al. (2016) has been put into contribution, benefiting from its enhanced realism.

4.1 Experimental setup

As represented in Fig. 7, the ARW consists of an AscTec Neo4 hexrotor endowed with a Compact AeRial MANipulator (CARMA) as in Wuthier et al. (2016). Absolute pose information from a motion capture system (consisting of 20 Vicon 5 cameras) are fetched by the onboard computer (an Intel 6 NUC with Core i7-5557U) through a 5G WiFi network, while the Neo’s low-level controller is connected to the on-board computer via a serial link. A ROS network has been deployed between the on-board computer and a laptop operating an ATI 7 Capacitive Force/Torque sensor, in order to forward the gathered data to ROS. This sensor is interfaced at the base of a rod, carrying a handle at its tip. The EKF of Fig. 4 actually consists of the time-delay compensated EKF-based modular sensor fusion framework from Lynen et al. (2013).

The parameters given on Tab. 1 were used for implementation. Additionally, the mixing matrix corresponding to the typical manipulator’s configuration of Fig. 1, namely pointing forward, is given by

1 https://youtu.be/jxmmwLlvvKs?list=PL8NeWi-qHKILEoNpKGip6cYJbw
2 http://www.ati-ia.com/
3 http://www.aectec.de/
4 http://www.ascotec.de/
5 https://www.vicon.com/
6 https://www.intel.com/
7 https://youtu.be/jxmmwLlvvKs?list=PL8NeWi-qHKILEoNpKGip6cYJbw
Table 1. Parameters for our ARW, consisting of an Astec Neo with a CARMA manipulator.

| parameter | value  | unit | description                        |
|-----------|--------|------|------------------------------------|
| $m_C$     | 3.719  | kg   | mass                               |
| $I_{C_{xx}}$ | 0.0341 | kg · m² | Inertia along $x_B$                |
| $I_{C_{yy}}$ | 0.0363 | kg · m² | Inertia along $y_B$                |
| $I_{C_{zz}}$ | 0.0627 | kg · m² | Inertia along $z_B$                |
| $K_p^W$   | $[8 \ 8 \ 17]^T$ | N/m | proportional gains on position     |
| $K_d^W$   | $[6 \ 6 \ 10]^T$ | N/m² | derivative gains on position       |
| $K_p^A$   | $[5 \ 5 \ 5]^T$ | N·m | proportional gains on arm’s attitude |
| $K_p^B$   | $[4 \ 4 \ 1]^T$ | N·m | proportional gains on UAV’s attitude |
| $K_d^B$   | $[0.7 \ 0.7 \ 0.5]^T$ | N·m/ rad² | derivative gains on UAV’s attitude |

\[
M = \begin{bmatrix}
  992 & -172 & -616 & 275 \\
  198 & 0 & 637 & 246 \\
  992 & 172 & -658 & 217 \\
 -992 & 172 & 658 & 217 \\
 -198 & 0 & -637 & 246 \\
 -992 & -172 & 616 & 275 \\
\end{bmatrix}
\begin{bmatrix}
  (\text{rad/s})^2 \\
  \text{N·m} \\
  (\text{rad/s})^2 \\
  \text{N} \\
\end{bmatrix}
\]  
\[
(12)
\]

4.2 Experimental protocol

The present aerial manipulation experiment consists of the following phases, with reference to Fig. 8:

1. Set a position/yaw setpoint in such a way the grasper theoretically falls right at the handle’s location, rely on the safety pilot’s skill to superimpose an additional torque downstream the attitude controller in order to manually compensate for the position error between the grasper and the handle, and transition to the next phase by closing the grasper, enabling torque compensation and inhibiting position/yaw gains.
2. Vary $F_r$ according to a trapezoidal profile spanning between 0 N and 2 N.
3. transition by opening the grasper, re-enabling the position/yaw gains and disabling torque compensation, and fly back to home position.

4.3 Results

The reference and measured forces during the experiment are presented in Fig. 8. First, as the handle is being grasped at $t = 15$ s, a positive force peak is observed due to the ARW’s momentum. Then, as position control is kept for two seconds for transitioning while already interacting, a force exceeding the torque sensor’s maximum value of 5 N·m is exerted showing the instability of such a scheme. After the pulling force controller has been enabled at $t = 17$ s, the transient of Fig. 8A was observed. The reference force is then brought back to zero. At the second trial spanning from $t = 30$ s to $t = 50$ s, tracking was
achieved yielding a Root-Mean-Square Error (RMSE) of 0.48 N. As the reference force’s magnitude is abruptly brought back to zero at $t = 50$ s, a second transient has appeared. Both these transients could be explained by static friction in the base joint, preventing the controllers to make the arm behave as a massless rope. Finally, the grasper was opened at $t = 58$ s and the pulling force controller was disabled at $t = 59$ s. Note that transitioning back to free-flight mode in such order resulted in a smooth releasing maneuver rather than instabilities.

5. CONCLUSION

The proposed control architecture yielded a RMSE of 0.48 N during experimental validation, which correspond to only 1.32 % of the overall system’s weight. No severe restrictions were set on the ARW design, as the rotor configuration can be any capable of generating three linearly independent torques together with a non-zero thrust, and the manipulator can be mounted everywhere below the COG. Also, a relatively small amount of parameters were to tune when considering our specific ARW design. Basing our work on top of a well established UAV controller has been likely to bring us further than a common entire control scheme design approach, also enabling smooth transitions between free flight and aerial manipulation, which is known for being challenging. Future directions comprise efforts such as investigating the stability of such a scheme, substituting the calculations based on the static equilibrium assumption with external wrench estimation and finally bringing several ARWs to achieve tasks collaboratively.

REFERENCES

Diebel, J. (2006). [unpublished] representing attitude: Euler angles, unit quaternions, and rotation vectors.

Featherstone, R. and Orin, D.E. (2008). Dynam- ics, 35–65. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-540-30301-5_3.

URL http://dx.doi.org/10.1007/978-3-540-30301-5_3.

Fumagalli, M., Nasiri, R., Marchelli, A., Forte, F., Keemink, A.Q., Stramigioli, S., Carloni, R., and Marconi, L. (2014). Developing an aerial manipulator prototype: Physical interaction with the environment. Robotics & Automation Magazine, IEEE, 21(3), 41–50.

Furrer, F., Burri, M., Achtelik, M., and Siegwart, R. (2016). Robot Operating System (ROS): The Complete Reference (Volume 1), chapter RotorS—A Modular Gazebo MAV Simulator Framework, 595–625. Springer International Publishing, Cham. doi:10.1007/978-3-319-26054-9_23.

URL http://dx.doi.org/10.1007/978-3-319-26054-9_23.

Hoeckstra, J.J., Wopereis, H.W., Post, T.H., Folkertsmma, G.A., Stramigioli, S., and Fumagalli, M. (2017). Application of substantial and sustained force to vertical surfaces using a quadrotor. In Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE.

Jiang, G. and Voyles, R. (2013). Hexrotor uav platform enabling dextrous interaction with structures-flight test. In Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on, 1–6. doi:10.1109/SSRR.2013.6719377.

Kondak, K., Huber, F., Schwarzbach, M., Laiaacker, M., Sommer, D., Bejar, M., and Ollero, A. (2014). Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, 2107–2112. doi:10.1109/ICRA.2014.6907148.

Korpela, C., Orsag, M., and Oh, P. (2014). Towards valve turning using a dual-arm aerial manipulator. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, 3411–3416. IEEE.

Lee, T., Leok, M., and McClamroch, N.H. (2010). Control of complex maneuvers for a quadrotor uav using geometric methods on se (3). arXiv preprint arXiv:1003.2005.

Lynen, S., Achtelik, M., Weiss, S., Chili, M., and Siegwart, R. (2013). A robust and modular multi-sensor fusion approach applied to mav navigation. In Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS).

Ryll, M., Bithoff, H.H., and Giordano, P.R. (2015). A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. IEEE Transactions on Control Systems Technology, 23(2), 540–556. doi:10.1109/TCST.2014.2330999.

Wuthier, D., Kominiak, D., Kanelolakis, C., Andriakopoulos, G., Fumagalli, M., Schipper, G., and Nikolakopoulos, G. (2016). On the design, modeling and control of a novel compact aerial manipulator. In Control and Automation (MED), 2016 24th Mediterranean Conference on, 665–670. IEEE.

Appendix A. THE MIXING MATRIX

Given the rotor configuration with motors $m_i$ ($i = 1, ..., n$), the mixing matrix $M \in \mathbb{R}^{n \times 4}$ allows to convert body-frame torque/thrust commands $\tau^B = [\tau^B_x \; \tau^B_y \; \tau^B_z \; \tau^B_4]^T$ (A.1) into squared rotor velocities $\Omega^2 = [\Omega^2_x \; \cdots \; \Omega^2_n]^T$, that is to say $\Omega^2 = M \tau^B$. (A.2)

First, assume the quadratic motor model

$F_m = c_i \Omega_i^2 \omega_m^i, \quad T_m = (-1)^i c_r c_t \Omega_r^2 \omega_m^i,$ (A.3)

with $c_i$ and $c_r$, the thrust and torque constants, respectively, and $\omega_m^i$, the unit vector pointing in the direction of motor $m_i$. Then, express the lever arm of motor $m_i$ over the COG as $\tau_m^B = P_m^B - P_{COG}^B$, (A.4)

the body frame’s origin being defined arbitrarily. Let $A \in \mathbb{R}^{4 \times 4}$ be the allocation matrix, representing the contribution of each motor velocity on the torque/thrust commands:

$\tau^B = A \Omega^2$. (A.5)

This allocation matrix can be computed as

$A = [a_1 \; \cdots \; a_n], \quad a_i = \begin{bmatrix} c_t c_r \omega_m^i \times \omega_m^i \; c_r \omega_m^i \times \omega_m^i \; c_t \omega_m^i \times \omega_m^i \; c_t c_r \omega_m^i \times \omega_m^i \end{bmatrix}. \tag{A.6}$

Finally, based on energy-efficiency considerations, the mixing matrix can be obtained from the pseudo-inverse of the allocation matrix:

$M = A^T (AA^T)^{-1}. \tag{A.7}$