Shared-Control Teleoperation of Omnidirectional Aerial Robots for Continuous Valve Rotation

Jinjie Li, Yicheng Chen, Haokun Liu, and Moju Zhao

***Abstract*— Omnidirectional aerial robots offer full 6-DoF independent control over position and orientation, making them popular for aerial manipulation. Although there have been advancements in robotic autonomy, human operation remains essential in complex aerial environments. Previously, we proposed an aerial teleoperation system [1] that brings the rotational flexibility of human hands into the unbounded aerial workspace, which was evaluated through a vertically mounted valve-turning task. However, due to the limited range of human forearm's motion, the robot could not continuously rotate the valve, hindering its application in industry. In this work, we propose an approach to initiate continuous rotation when the forearm angle exceeds a threshold. This switch of control authority exemplifies the concept *shared control* from human-robot interaction, marking a further step in aerial teleoperation.**

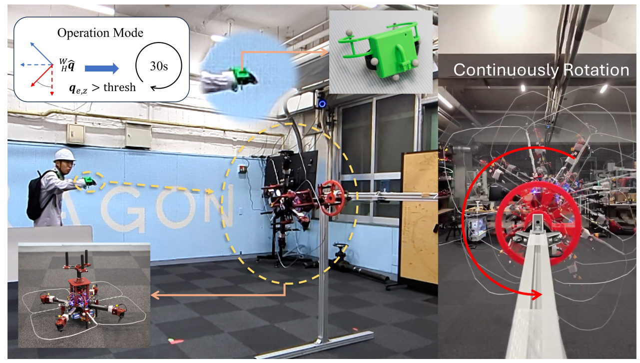

Fig. 1. **A human operator controls an omnidirectional aerial robot to continuously turn a vertically mounted valve.** The robot is in *Operation Mode*, where its target pose is in general mapped directly from the operator's hand pose. When the forearm rotation exceeds a threshold, continuous 360° rotation is triggered. The link of video: https://youtu.be/JuPAP6E4yR4.

I. Introduction

Aerial manipulation is gaining popularity due to its ability to operate freely in 3D space, particularly in high-altitude environments that are difficult for humans to access [2]–[4]. Although fully autonomous manipulation remains the ultimate goal for robotics researchers, the complexity of the aerial environment necessitates keeping a human operator in the loop [5]. In our previous work [1], we proposed a six-degree-of-freedom teleoperation framework for aerial manipulation. Specifically, we designed four interaction modes for different tasks, including *Spherical Mode* and *Cartesian Mode* for long-range movement, *Operation Mode* for precise manipulation, and *Locking Mode* for temporary pauses,where hand gestures were utilized for seamless mode switching. The proposed framework was evaluated through a valve-turning task[1], and to the best of our knowledge, this was the first time the rotational dexterity of the human wrist was intuitively demonstrated on an omnidirectional aerial robot. However, due to the limitations of both the controller and the physical structure of human forearm, we were not able to achieve continuous rotation.

In this work, we introduce the concept of *shared control* [6] from the human–machine interaction (HMI) domain into aerial operations and design a mechanism to temporarily shift control authority to the robot for continuous valve turning. The approach is briefly introduced as follows, and its performance is evaluated in a real-world experiment.

II. Omnidirectional Aerial Teleoperation

The teleoperation system for an omnidirectional aerial robot consists of four main elements, as shown in Fig. 1: (1) *a human operator*, (2) *motion-tracking marker sets* to localize the hand, (3) *a data glove* to capture hand gestures, and (4) *an omnidirectional aerial robot*.

To thoroughly exploit the omnidirectionality of the human hand, an aerial robot capable of hovering in any orientation is required. Here, we use an omnidirectional upgrade of the tiltable quadrotor presented in [7]. The control method is actuator-level NMPC, with an upgrade to handle singularities during continuous rotation.

We define the body frame $\{\mathcal{B}\}$ of the drone at its center of gravity (CoG). With the flight controller, the robot can be simplified as a first-order model:

$$\dot{\boldsymbol{p}}_B = 1/\boldsymbol{t}_p \cdot (\boldsymbol{p}_{B_c} - \boldsymbol{p}_B)\,, \tag{1a}$$

$${}^{W}_{B}\dot{\boldsymbol{q}} = 1/\boldsymbol{t}_q \cdot \left({}^{W}_{B}\boldsymbol{q}^* \circ {}^{W}_{B}\boldsymbol{q}_c\right), \tag{1b}$$

where $\boldsymbol{p}_B$, $\boldsymbol{p}_{B_c}$, ${}^{W}_{B}\boldsymbol{q}$, and ${}^{W}_{B}\boldsymbol{q}_c$ represent current position, target position, current quaternion, and target quaternion, respectively, $\circ$ refers to quaternion multiplication, * refers to conjugation, as well as $\boldsymbol{t}_p$ and $\boldsymbol{t}_q$ denote time constants determined by the control performance. The estimated position and orientation are denoted as $\hat{\boldsymbol{p}}_B$ and ${}^{W}_{B}\hat{\boldsymbol{q}}$.

We also require the hand's position $\hat{\boldsymbol{p}}_H$ and orientation ${}^{W}_{H}\hat{\boldsymbol{q}}$ to determine the robot's target pose, which is obtained by Motion Capture System. The interaction framework functions as a planning module, aiming to calculate $\boldsymbol{p}_{B_c}$ and ${}^{W}_{B}\boldsymbol{q}_c$ based on the operator's action.

III. Hand-Based Interaction

We define an *Operation Mode* in [1], in which we aim to establish an intuitive connection between the drone and the operator's hand. In summary, we map the pose of the hand

All members come from DRAGON Lab, Department of Mechanical Engineering, The University of Tokyo, Tokyo 113-8656, Japan (email: `jinjie-li@dragon.t.u-tokyo.ac.jp`)

[1]The video of previous work: https://youtu.be/n0IQEnjPzrw

Algorithm 1 Our Shared Control Approach

Input: 1) ${}^{W}_{H}\hat{\boldsymbol{q}}$, hand quaternion; 2) ${}^{W}_{B}\hat{\boldsymbol{q}}$, robot quaternion; 3) ${}^{H}\boldsymbol{\omega}$, hand angular velocity; 4) ${}^{W}_{H}\boldsymbol{q}_{\text{start}}$, initial hand quat to start the rotation; 5) T_{rot}, rotation period; 6) $\varepsilon_{\text{entry}}{=}0.2$, $\varepsilon_{\text{exit}}{=}0.1$, $\theta_{\text{trig}}{=}0.4$, $T_{\text{guard}}{=}5\,\text{s}$, parameters for continuous rotation.

Output: ${}^{W}_{B}\boldsymbol{q}_c$ and $\boldsymbol{\omega}_c$, the target orientation and angular velocity of the aerial robot, respectively.

while in *Operation Mode* **do**
  $\boldsymbol{q}_e \leftarrow$ QUATERROR$({}^{W}_{H}\hat{\boldsymbol{q}}, {}^{W}_{H}\boldsymbol{q}_{\text{start}})$;
  $\mathbf{v}_e \leftarrow \text{sign}(q_{e,w})\,[q_{e,x},\, q_{e,y},\, q_{e,z}]^T$;
  if not ISROTATING() **then**
    if ALIGN$(\mathbf{v}_{e,xy}, \mathbf{0}, \varepsilon_{\text{entry}})$ **then**
      if $|\mathbf{v}_{e,z}| > \theta_{\text{trig}}$ **then**
        $t_0 \leftarrow$ NOW();
        $\psi_0 \leftarrow$ CONVERTTOYAW$(\mathbf{v}_{e,z})$;
        $d_r \leftarrow \text{sign}(\mathbf{v}_{e,z}) \in \{+1, -1\}$;
      end if
    end if
    return $\{{}^{W}_{B}\boldsymbol{q}_c \leftarrow {}^{W}_{H}\hat{\boldsymbol{q}}, \boldsymbol{\omega}_c \leftarrow {}^{H}\boldsymbol{\omega}\}$;
  end if
  $\tau \leftarrow$ NOW() $- t_0$;
  $\boldsymbol{q}_{e,r2h} \leftarrow$ QUATERROR$({}^{W}_{B}\hat{\boldsymbol{q}}, {}^{W}_{H}\hat{\boldsymbol{q}})$;
  $\mathbf{v}_{e,r2h} \leftarrow \text{sign}(q_{e,r2h,w})\,[q_{e,r2h,x},\, q_{e,r2h,y},\, q_{e,r2h,z}]^T$;
  $c_1 \leftarrow$ ALIGN$(\mathbf{v}_{e,r2h}, \mathbf{0}, \varepsilon_{\text{exit}})$;
  $c_2 \leftarrow (\tau > T)$;
  if $(c_1 \vee c_2) \;\wedge\; (\tau > T_{\text{guard}})$ **then**
    RESET();
    return $\{{}^{W}_{B}\boldsymbol{q}_c \leftarrow {}^{W}_{H}\hat{\boldsymbol{q}}, \boldsymbol{\omega}_c \leftarrow {}^{H}\boldsymbol{\omega}\}$;
  end if
  $\omega \leftarrow 2\pi/T \cdot d_r$;
  ${}^{W}_{B}\boldsymbol{q}_c \leftarrow$ EULERTOQUAT$(0,\; \pi/2,\; \omega\tau + \psi_0)$;
  $\boldsymbol{\omega}_c \leftarrow (0, 0, \omega)$;
  return $\{{}^{W}_{B}\boldsymbol{q}_c, \boldsymbol{\omega}_c\}$
end while

to the robot through the following equations:

$$\boldsymbol{p}_{B_c} = \hat{\boldsymbol{p}}_B(t_0) + \boldsymbol{k} \cdot (\hat{\boldsymbol{p}}_H(t) - \hat{\boldsymbol{p}}_H(t_0))\,, \quad {}^{W}_{B}\boldsymbol{q}_c = {}^{W}_{H}\hat{\boldsymbol{q}}. \quad (2)$$

The algorithm to trigger the continuous vertical rotation is listed in Alg. 1. In fact, shifting control authority to the robot is not challenging. In contrast, the most challenging part is determining *when* the control authority should be returned to the operator. Teleoperation systems should be resilient to users' incorrect actions, including tremors and involuntary reflexes. However, when something goes wrong during manipulation, the operator should always know how to move the robot out of danger. In our case, we first use quaternion error to ensure there is no singularity problem in any situation, and then maintain the operator's control over position throughout the process. We also ensure that the robot stops rotating if the orientation of the robot and hand become too close. We found that the robot's behavior after autonomous turning is critical: if the robot suddenly jumps to the pose mapped from the hand, the operator may be startled. Hence, we try to set the stop orientation as close as possible to the hand.

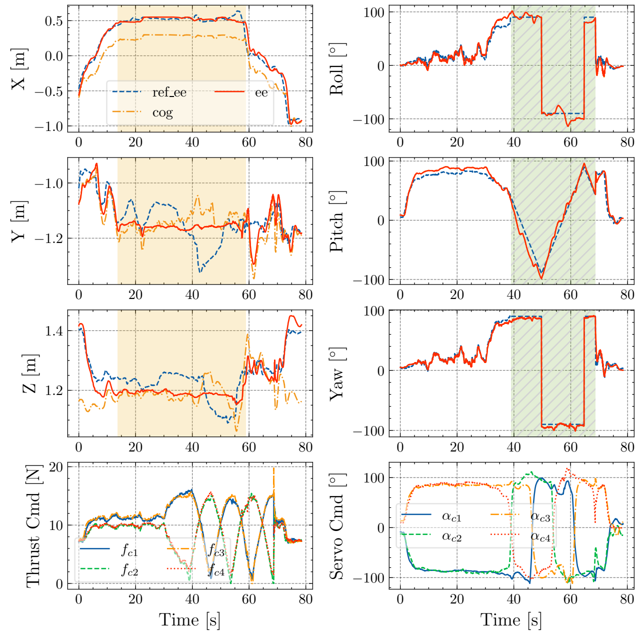

Fig. 2. **Data of valve turning.** When the rotation of the human hand exceeds a threshold, the robot begins continuous rotation, with the range plotted as a green forward-slash area. The yellow area highlights the period when the robot is attached to the valve. Wrench estimation was shut down for safety. The RMSEs are: p_x: 0.0690 m, p_y: 0.0622 m, p_z: 0.0468 m; roll: 7.8917°, pitch: 9.0527°, yaw: 4.1610°.

IV. Experiments

We used a fork as the end-effector for valve turning, and an operator used the hand to control the robot. Since continuous valve rotation exceeds the natural flexibility of the human wrist and forearm, we introduced a *shared-control* approach: when the operator's hand rotation went beyond a threshold, the robot performed continuous vertical rotation for 360°. Although the attitude was governed, the position was always controlled by the operator for safety. As our framework did not include impedance control, the wrench compensation was also shut down to avoid excessive internal forces during turning. The experimental data are shown in Fig. 2.

From Fig. 2, we can see that the robot can continuously rotate the vertically mounted valve. We observed an offset in position when the valve was engaged, indicating possible internal forces. Since the valve is rigid, internal forces accumulated through imperfect alignment may damage the end-effector, highlighting the need for compliance in the future. Additionally, the effector slipped out at the end of the yellow range, possibly due to uneven rotation caused by model error. More accurate calibration of the entire model may help ensure proper engagement.

V. Conclusion

In this work, we introduced the shared-control concept into our previous system, achieving continuous valve rotation in the real world. In the future, this philosophy could be extended to a general framework for human intent recognition and human–robot collaboration.

References

[1] J. Li, J. Li, K. Kaneko, H. Liu, L. Shu, and M. Zhao. Six-DoF Hand-Based Teleoperation for Omnidirectional Aerial Robots. [Online]. Available: http://arxiv.org/abs/2506.15009

[2] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 626–645, Feb. 2022.

[3] M. Zhao, K. Nagato, K. Okada, M. Inaba, and M. Nakao, "Forceful valve manipulation with arbitrary direction by articulated aerial robot equipped with thrust vectoring apparatus," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4893–4900, Apr. 2022.

[4] T. Nishio, M. Zhao, K. Okada, and M. Inaba, "Design, control, and motion planning for a root-perching rotor-distributed manipulator," *IEEE Trans. Robot.*, vol. 40, pp. 660–676, 2024.

[5] K. Darvish *et al.*, "Teleoperation of humanoid robots: A survey," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1706–1727, Jun. 2023.

[6] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7989–7996, Oct. 2021.

[7] J. Li, J. Sugihara, and M. Zhao, "Servo integrated nonlinear model predictive control for overactuated tiltable-quadrotors," *IEEE Robot. Automat. Lett.*, vol. 9, no. 10, pp. 8770–8777, Oct. 2024.