A Natural Interaction Interface for UAVs Using Intuitive Gesture Recognition

Meghan Chandarana[1], Anna Trujillo[2], Kenji Shimada[1] and B. Danette Allen[2]

[1] Carnegie Mellon University, Pittsburgh, Pennsylvania, United States of America
{mchandar, shimada}@cmu.edu
[2] NASA Langley Research Center, Hampton, Virginia, United States of America
{a.c.trujillo, danette.allen}@nasa.gov

Abstract. The popularity of unmanned aerial vehicles (UAVs) is increasing as technological advancements boost their favorability for a broad range of applications. One application is science data collection. In fields like earth and atmospheric science, researchers are seeking to use UAVs to augment their current portfolio of platforms and increase their accessibility to geographic areas of interest. By increasing the number of data collection platforms, UAVs will significantly improve system robustness and allow for more sophisticated studies. Scientists would like the ability to deploy an available fleet of UAVs to traverse a desired flight path and collect sensor data without needing to understand the complex low-level controls required to describe and coordinate such a mission. A natural interaction interface for a Ground Control System (GCS) using gesture recognition is developed to allow non-expert users (e.g., scientists) to define a complex flight path for a UAV using intuitive hand gesture inputs from the constructed gesture library. The GCS calculates the combined trajectory online, verifies the trajectory with the user, and sends it to the UAV controller to be flown.

Keywords: Natural Interaction · Gesture · Trajectory · Flight Path · UAV · Non-expert User

1 Introduction

Rapid technological advancements to unmanned aerial systems foster the use of these systems for a plethora of applications [1] including but not limited to search and rescue [2], package delivery [3], disaster relief [4], and reconnaissance [5]. Many of these tasks are repetitive or dangerous for a human operator [6]. Historically, unmanned aerial vehicles (UAVs) are piloted remotely using radio remotes [7], smartphones [8], or ground control systems. This is done at the control surface level and requires a deep understanding of the extensive workings of the internal controller (much like a pilot) and the various algorithms used for system autonomy. Recently, advances in autonomy have enabled UAVs to fly with preprogrammed mission objectives, required trajectories, etc.

As their manufacturing costs decrease so does the cost of entry, thereby making UAVs more accessible to the masses. UAVs have also gained traction due to more robust wireless communication networks, smaller and more powerful processors, and embedded sensors [9]. Although these systems have the potential to perform a myriad of intricate tasks whose number increases with the complexity of the technology, very few people possess the required skills and expert knowledge to control these systems precisely.

Fig. 1. Example ozone sensor used on a UAV for collecting atmospheric data [10].

1.1 Atmospheric Science Mission

Researchers are leveraging these autonomous vehicles to transition from traditional data collection methods to more autonomous collection methods in fields such as environmental science. Currently, environmental scientists use satellites, air balloons, and manned aircrafts to acquire atmospheric data. Although many studies require the analysis of correlative data, traditional collection methods typically only deploy individual, uncoordinated sensors, thereby making it difficult to gather the required sensor data. The current methods are also costly in terms of time and equipment [11]. Many systems such as remotely operated aerial vehicles require expert users with extensive training for operation. In addition, the possibility of sensors being lost or damaged is high for unpredictable methods like aerosondes and balloons.

Atmospheric scientists are interested in using UAVs to collect the desired data to fill the data gaps seen in current methods while giving scientists the capability to perform more elaborate missions with more direct control [12]. The portability of many sensors (Fig. 1) engenders each UAV to carry its own – potentially unique – sensor payload, thereby generating a larger density of accumulated in-situ data. Each enhanced – high density – data set would provide the necessary supplemental information for conducting more sophisticated environmental studies.

Ideally, scientists would like to deploy multiple UAVs to fly desired flight paths (while simultaneously taking sensor data) without being obligated to understand the underlying algorithms required to coordinate and control the UAVs. This mission management centered role requires the implementation of a Ground Control System (GCS) interface that can accurately map seemingly simplistic operator inputs into fully defined UAV maneuvers. Fig. 2 depicts an example mixed-reality environment where the operator is able to manage the mission by naturally interacting with a fleet of vehicles whose geo-containment volume has been defined by the operator.

Systems which simulate human-human communication increase their accessibility to non-expert users [13]. These systems utilize natural interaction-based interfaces.

Previous researchers explored the operators' informational needs [14] and how human operators can collaborate with autonomous agents using speech and gestures [15].

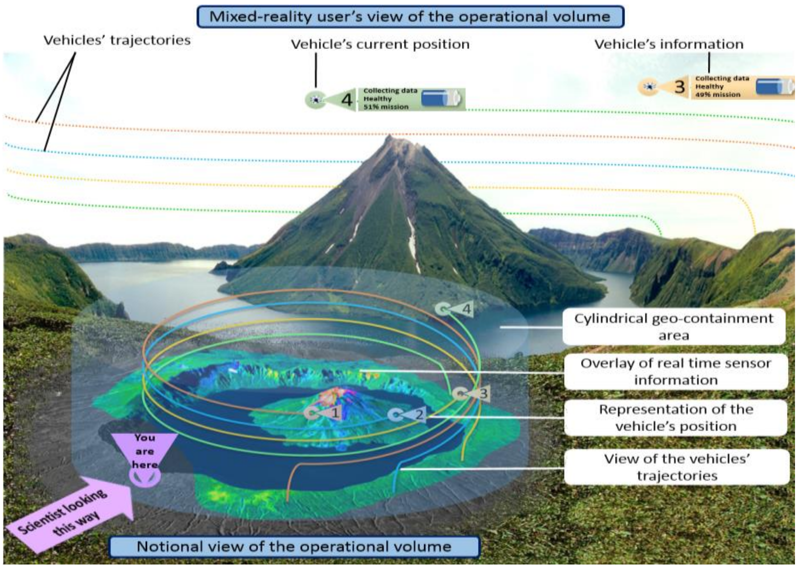

Fig. 2. Artist's representation of the future vision for an outdoor, mixed-reality environment in the context of an atmospheric science mission with multiple UAVs.

The remainder of this paper will outline a natural interaction interface for a GCS in the context of an atmospheric science mission and is outlined as follows. Section 2 describes gesture-based human-robot interaction. Section 3 explains the developed natural interaction interface and the system modules. Section 4 examines the results. Section 5 draws some conclusions and discusses future work.

2 Gesture-Based Human-Robot Interaction

Using a more "natural" input is intuitive and increases system efficiency [16], [17]. However, natural interaction systems are often difficult to develop as some interpretation is required to computationally recognize their meaning. In the context of generating trajectory segments for an atmospheric science mission, gestures are an effective input method as humans naturally use gestures to communicate shapes and patterns amongst themselves [18], [19].

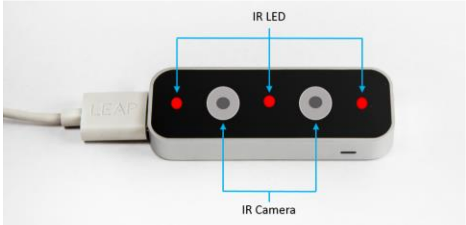

Fig. 3. Leap Motion^{TM} ccontroller.

In the past, researchers have used a variety of gesture input methods in order to simulate interaction between an operator and a computer. Many of these input methods required the operator to hold or wear the sensor [20], [21], [22]. These input devices tended to restrict the operator's hand movements, thereby reducing the natural feel of the interface. Over time, researchers developed systems where sensors were no longer mounted or attached to the operator. Some of these systems focused on using full body gestures to control mobile robots [23], while others used static gestures to program a robot's pose by demonstration [24], [25] or even encode complex movements [26]. Neither type of system focused on using unmounted sensors to map simple, dynamic hand gestures to complex robot movements.

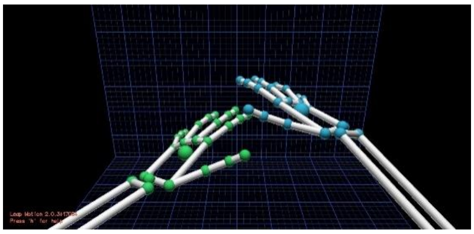

Fig. 4. Visual representation of the Leap Motion^{TM} controller's estimate of the operator's finger locations.

The Leap Motion™ controller (Fig. 3) is used to track the 3D gesture inputs of the operator. This controller uses infrared-based optics to produce gesture and position tracking with sub-millimeter accuracy. Each individual bone's position within all ten fingers of the operator are estimated (Fig. 4). User's fingertips are detected at up to 200 frames per second at 0.01mm accuracy. Three infrared cameras work together to provide 8ft^3 volume of interactive space [27].

3 Ground Control System Framework

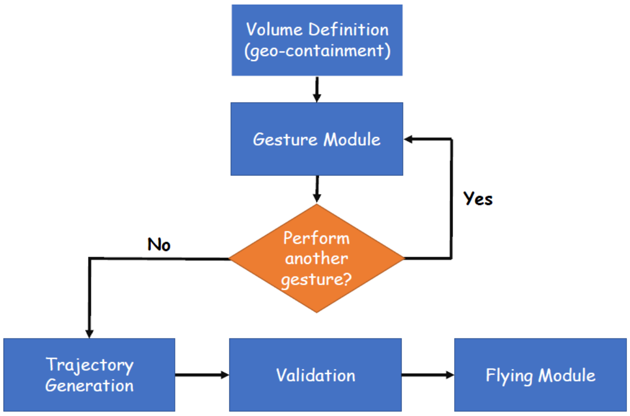

Fig. 5. Diagram of the interface's process flow between modules.

A natural interaction interface for a GCS using gesture recognition was developed to allow non-expert operators to quickly and easily build a desired flight path for an autonomous UAV by defining trajectory segments. The developed system addresses several key requirements.

i. The gesture inputs shall be intuitive and make the operator feel as if he/she were naturally interacting with another person.
ii. The interface shall be simple to walk through so as to reduce the required training time and increase the usability for non-experts.
iii. There shall be ample user feedback for decision making.
iv. The system shall abstract the operator away from the complexity of the low-level control operations, which are performed autonomously, and allow them instead to have a high-level mission management role.

The system's interface is broken down into several key modules (Fig. 5):

A. volume definition module
B. gesture module
C. trajectory generation module
D. validation module
E. flight module

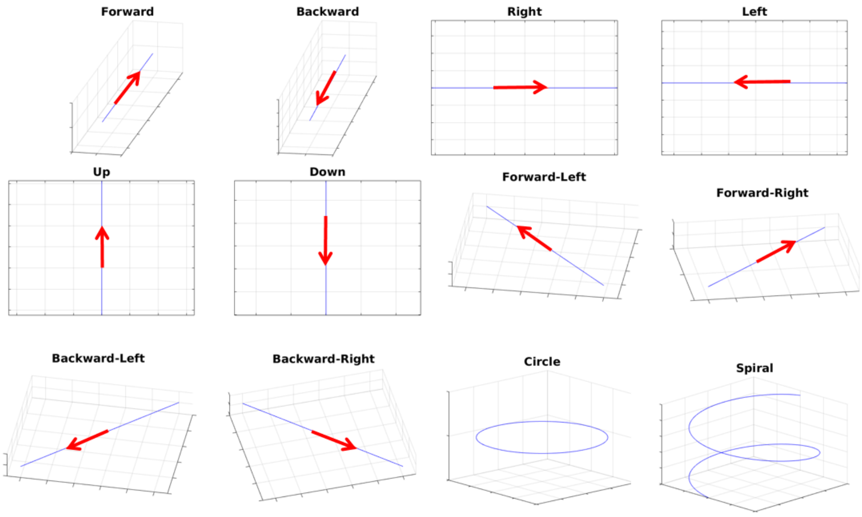

Fig. 6. Library of gesture inputs.

The first module asks the operator to define the size of the desired operational volume. The operator can then build the flight path by gesturing one of twelve trajectory segments currently defined in the intuitive gesture library (Fig. 6). The interface allows the operator to build up a path with any number of segments. Once the operator has fully defined the flight path, the system combines the trajectory segments with first order continuity before taking the operator through a validation step. This combined trajectory is then sent to the vehicle controller and the operator may then send takeoff and land commands. Throughout each step, the interface provides the operator with feedback displays so as to increase the efficiency of decision making, as well as, provide a logical progression through the system.

3.1 Volume Definition Module

The operator is able to define a desired operational volume for the UAV using the volume definition module. This operational volume can demarcate the operational area, as well as, a geo-containment zone and may be set as a hard boundary for the vehicle's position in the onboard controller. As a safety measure, if the vehicle were to leave the geo-containment area, the vehicle controller can immediately command the vehicle to land. Once the user creates the geo-containment volume, all defined trajectories are restricted to this volume. Currently, this is done by manually defining the lengths and orientations of the segments such that they are guaranteed to stay within the boundary. This can be extended to automatically scale and orient the segments by using the size and shape of the boundary as parameters.

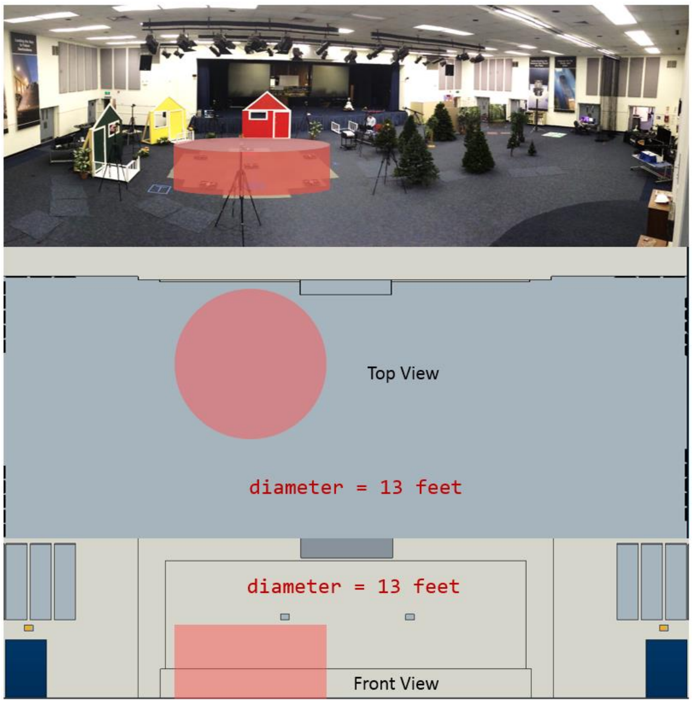

Fig. 6. Top view (*middle*) and front view (*bottom*) representations shown in the interface for use in volume definition.

In order to aid the operator in defining a volume, the interface displays multiple 2D viewpoints of the working environment so as to accurately display the full 3D shape of the volume. These viewpoints are planar diagrams of the total available working volume which, when put together give the operator a 3D understanding. A representation of the desired operational volume (to scale) in an indoor setting is overlaid on the viewpoints (Fig. 7). These viewpoints allow the operator to visualize the entire volume as shown in Fig. 7 (top). Fig. 2 shows an example volume overlay in an outdoor setting. The operator can use an intuitive pinching gesture to expand or compress the volume to the desired size. This example uses a cylindrical volume represented in Imperial units. The units used to define the operational volume are tailorable.

3.2 Gesture Module

By using the defined gesture library (Fig. 6) the operator is able to employ a variety of trajectory segments. The gesture module is responsible for characterizing the raw sensor data from the Leap Motion™ controller into desired trajectory segment shapes.

Gesture Library. Based on feedback from subject matter experts, the gesture library was developed to define flight paths that would be of interest to scientists for collecting various sensor data. The library is composed of twelve gestures ranging from simple linear gestures to more complex diagonal and curved gestures: forward, backward, right, left, up, down, forward-left, forward-right, backward-left, backward-right, circle, and spiral.

The intuitive nature of the gesture library lends itself to applying several of the gestures in other aspects of the interface, in addition to path definition. This is done by breaking the flow of the interface into several modes. After building a trajectory segment, the Message Mode gives the operator the ability to use the right and left gesture inputs to navigate a menu questioning if they have taught all desired trajectory segments. Additionally, the up and down gesture inputs are adopted for sending take-off and land commands to the UAV controller in the Flight Mode, which occurs after the operator has built all their desired segments.

3D Gesture Characterization. The operator's gesture input is characterized as one of the twelve gestures in the defined library using a Support Vector Machine (SVM) model. As opposed to a simplistic threshold separation, the SVM model provides a more robust classification by reducing biases associated with any one user.

Data from eleven subjects was used to train the SVM model using a linear kernel [28]. Each subject provided ten labeled samples per gesture in the library. Features were extracted from each pre-processed input sample. The features used for training included the hand direction movement throughout the gesture, as well as, the eigenvalues of the raw hand position data.

3.3 Trajectory Generation Module

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^{i} P_i \, . \qquad (1)$$

Once the operator has defined all the desired trajectory segments, the trajectory generation module translates each segment into a set of fifth order Bézier splines [29] represented by Eq. 1. The system then automatically combines the spline sets into a complete path by placing the Bézier splines end-to-end in order of their definition. Transitions between splines are smoothed with first order continuity so that any sudden direction and velocity changes are eliminated. This produces a continuous, flyable path for the UAV. Each combined trajectory includes the necessary time, position, and velocity data required by the UAV controller. Using a data distribution service (DDS) middleware, the trajectory information can be transmitted on-line to the UAV controller [30].

3.4 Validation Module

Prior to transferring the combined trajectory data taught by the operator to the UAV controller, the system displays a 3D representation of the flight path. The rendered flight path is presented to the operator so that they can validate and approve the proposed path of the UAV before flight. Fig. 8 depicts an example of a rendered flight path where the operator taught the following trajectory segments (in order): left, spiral, forward-left, and circle. This flight path simulates a UAV collecting data as it spirals up to a desired altitude, moves to a new region of interest, and circles around a target while collecting more data.

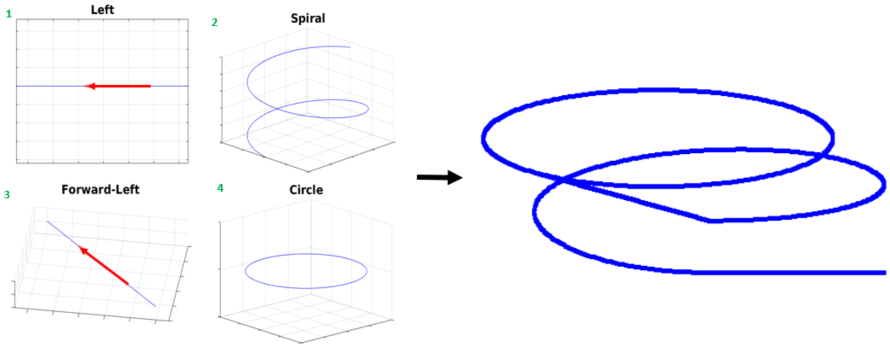

Fig. 8. Sample rendered 3D combined trajectory built by the operator (*right*) with input gestures (*left*).

3.5 Flight Module

Fig. 9. System setup for operator use (*left*) and view of UAV operational volume from operator's perspective (*right*).

After the UAV controller receives the flight data, the operator is able to initiate and end path following using the features of the flight module. In the current setup, Up and Down gestures are translated into takeoff and land commands respectively. The commands are sent directly to the UAV controller using the DDS middleware. Once a takeoff command is sent and executed, the UAV autonomously begins traversing the newly defined flight path using the transferred data. Immediately upon completing its defined path, the UAV hovers in place until the operator sends a land command.

4 Results

A straightforward system setup (Fig. 9 (left)) exudes the sense of effortless execution to the operator. The system requires just two components: (1) a flat surface to place the Leap Motion^{TM} Controller and (2) a display with the interface loaded and a USB port to connect the Leap Motion^{TM} Controller. In the context of an atmospheric science mission, the operator is situated such that they have a frame of reference for building the trajectory segments for the UAV while maintaining a safe separation (Fig. 9 (right)). The operator uses the gesture library to intuitively walk through the steps required to build desired trajectory segments. There is no limit placed on the number of segments the operator can teach.

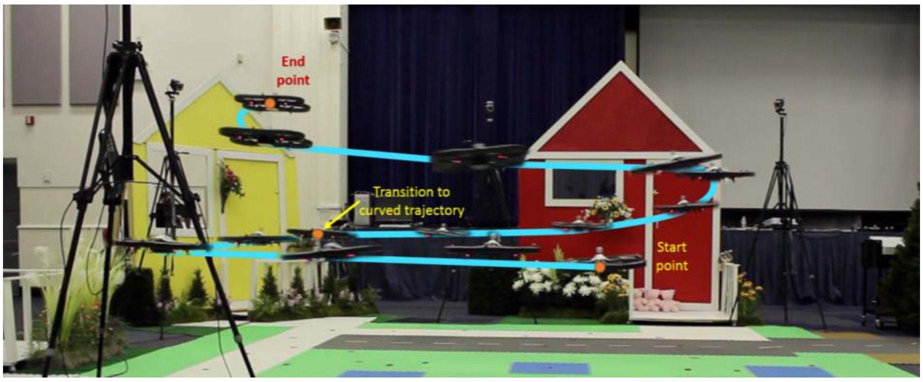

Fig. 10. Sample flight path after trajectory data was sent to the controller and a takeoff command.

Each time the operator uses the natural interaction interface, the system calculates the combined trajectory and discretizes every segment's set of Bézier splines [6] for visualization in the validation module on-line. The system is able to accurately transmit the complete trajectory data to the UAV controller such that the UAV flies a recognizable implementation of the path generated by the operator. Fig. 10 depicts a resulting sample flight path generated by employing the developed gesture-based interface. For this example, the operator used the developed interface to generate a trajectory for a single UAV with the following directions (in order): left, forward and spiral. The position of the UAV as it traverses the complete trajectory was overlaid on one image for easier understanding. A light blue line is used to trace the path of the UAV. Although not depicted, the operator was also able to perform an Up gesture to send a takeoff command to the vehicle and a Down gesture to land the vehicle.

5 Conclusion and Future Work

A GCS for autonomous UAV data collections was developed. The system utilizes a gesture-based natural interaction interface. The intuitive gesture library allows non-expert operators to quickly and easily define a complete UAV flight path without needing to understand the low-level controls. There are several planned extensions that can be made to this work to augment its efficacy and applicability.

1. The interface can be extended to include the definition of additional geometric constraints if necessary (e.g., clockwise vs counterclockwise direction on a circle).
2. The gesture library can be extended (e.g., spiral forward).

3. Gesture segmentation: A user may wish to define a complete square shape instead of teaching each segment one-by-one.
4. Extending the system to include real-time mission supervision and trajectory modification.
5. Perform user studies to fully validate the methodology.

Acknowledgments. The authors would like to thank Javier Puig-Navarro and Syed Mehdi (NASA LaRC interns, University of Illinois at Urbana-Champaign) for their help in building the trajectory module, Gil Montague (NASA LaRC intern, Baldwin Wallace University) and Ben Kelley (NASA LaRC) for their expertise in integrating the DDS middleware, Dr. Loc Tran (NASA LaRC) for his insight on feature extraction, and the entire NASA LaRC Autonomy Incubator team for their invaluable support and feedback.

References

1. Jenkins, D. and Bijan, B.: The Economic Impact of Unmanned Aircraft Systems Integration in the United States. Technical Report, AUVSI (2013)
2. Wald, M.L.: Domestic Drones Stir Imaginations and Concerns. New York Times (2013)
3. Barr, A.: Amazon Testing Delivery by Drone, CEO Bezos Says. USA Today, http://www.usatoday.com/story/tech/2013/12/01/amazon-bezos-dronedelivery/3799021/
4. Saggiani, G. and Teodorani, B.: Rotary Wing UAV Potential Applications: An Analytical Study through a Matrix Method. In: Aircraft Engineering and Aerospace Technology, vol. 76, no. 1, pp. 6-14 (2004)
5. Office of the Secretary of Defense.: Unmanned Aircraft Systems Roadmap 2005-2030. Washington, DC (2005)
6. Schoenwald, D. A.: AUVs: In Space, Air, Water, and on the Ground. In: IEEE Control Systems Magazine, vol. 20, no. 6, pp.15-18 (2000)
7. Chen, H., Wang, X. and Li, Y.: A Survey of Autonomous Control for UAV. In: IEEE International Conference on Artificial Intelligence and Computational Intelligence, pp. 267-271. Shanghai (2009)
8. Hsu, J.: MIT Researcher Develops iPhone App to Easily Control Swarms of Aerial Drones. In: Popular Science, http://www.popsci.com/technology/article/2010-04/video-usingsmart phonescontrol-aerial-drones
9. Zelinski, S., Koo, T. J. and Sastry, S.: Hybrid System Design for Formations of Autonomous Vehicles. In: IEEE Conference on Descision and Control (2003)
10. Miniature Air Quality Monitoring System, http://www.altechusa.com/ odor.html.
11. Wegener, S., Schoenung, et al.: UAV Autonomous Operations for Airborne Science Missions. In: AIAA 3rd Unmanned Unlimited Technical Conference, Chicago (2004)
12. Wegener, S. and Schoenung, S.: Lessons Learned from NASA UAV Science Demonstration Program Missions. In: AIAA 2nd Unmanned Unlimited Systems, Technologies and Operations Conference, San Diego (2003)
13. Perzanowski, D., Schultz, A. C., et al.: Building a Multimodal Human-Robot Interaction. In: IEEE Intelligent Systems, pp. 16-21 (2001)
14. Trujillo, A. C., Fan, H., et al.: Operator Informational Needs for Multiple Autonomous Small Vehicles. In: 6th International Conference on Applied Human Factors and Ergonomics and the Affiliated Conferences, Las Vegas, pp. 5556-5563 (2015)

15. Trujillo, A. C., Cross, C., et al..: Collaborating with Autonomous Agents. In: 15th AIAA Aviation Technology, Integration and Operations Conference, Dallas (2015)
16. Reitsema, J., Chun, W., et al.: Team-Centered Virtual Interaction Presence for Adjustable Autonomy. In: AIAA Space, Long Beach (2005)
17. Wachs, J. P., K¨olsch, M., Stern, H. and Edan, Y.: Vision-Based Hand-Gesture Applications. In: Communications of the ACM, vol. 54, no. 2, pp. 60-71 (2011)
18. McCafferty, S.: Space for Cognition: Gesture and Second Language Learning. In: International Journal of Applied Linguistics, vol. 14, no. 1, pp. 148-165 (2004)
19. Torrance, M.: Natural Communication with Robots. M.S. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, (1994).
20. Iba, S., Weghe, M. V., et al.: An Architecture for GestureBased Control of Mobile Robots. In: IEEE International Conference on Intelligent Robots and Systems, pp. 851-857 (1999)
21. Neto, P. and Pires, J.N.: High-Level Programming for Industrial Robotics: Using Gestures, Speech and Force Control. In: IEEE International Conference on Robotics and Automation, Kobe (2009)
22. Yeo, Z.: GestureBots: Intuitive Control for Small Robots. In: CHI 28th Annual ACM Conference on Human Factors in Computing Systems, Atlanta (2010)
23. Waldherr, S., Romero, R., and Thrun, S.: A Gesture Based Interface for Human-Robot Interaction. In: Autonomous Robots, vol. 9, pp. 151-173 (2000)
24. Becker, M., Kefalea, E., et al.: GripSee: A Gesture- Controlled Robot for Object Perception and Manipulation. In: Autonomous Robots, vol. 6, pp. 203-221. Kluwer Academic Publishers, Boston (1999)
25. Lambrecht, J., Kleinsorge, M. and Kruger, J.: Markerless Gesture-Based Motion Control and Programming of Industrial Robots. In: 16th IEEE International Conference on Emerging Technologies and Factory Automation, Toulouse (2011)
26. Raheja, J. L., et al.: Real-Time Robotic Hand Control using Hand Gestures. In: IEEE 2nd International Conference on Machine Learning and Computing, Washington, D.C., (2010)
27. Bassily, D., et al.: Intuitive and Adaptive Robotic Arm Manipulation Using the Leap Motion Controller. In: 8th German Conference on Robotics, Munich (2014)
28. King, D. E.: Dlib-ml: A Machine Learning Toolkit. In: Journal of Machine Learning Research, vol. 10, pp. 1755-1758 (2009)
29. Choe, R., et al.: Trajectory Generation Using Spatial Pythagorean Hodograph Bézier Curves. In: AIAA Guidance, Navigation and Control Conference, Kissimmee (2015)
30. DDS Standard, https://www.rti.com/products/dds/omg-ddsstandard.html