Revista Romana de Interactiune Om-Calculator 11(2) 2018, 109-126 © MatrixRom

Creating enhanced interfaces for cyber-physical-social systems: the remote drone control experiment

Alexandru Matei[1], Alexandru Butean[2]
[1] University "Lucian Blaga" of Sibiu; Connected Intelligence Research Center;
10, Victoriei Bd., Sibiu, 550024, România
E-mail: alex.matei@ulbsibiu.ro
[2] University "Lucian Blaga" of Sibiu; Department of Computer Science and Electrical Engineering; 4, Emil Cioran, Sibiu, 550024, România
E-mail: alexandru.butean@ulbsibiu.ro

Abstract. In the area of cyber-physical-social systems there is always a huge demand for integration and adaptation; most often, these two tasks go together. There are very few practical cases when every module fits perfectly into a system. The concept of puzzle pieces can be applied only on a high-level logic, but in the experimental development stages things tend to be more complex. The purpose of our work is to be able to integrate various modern interfaces into existing systems in order to ease the interaction process or experiment unexplored areas. Our current contribution presents the case of changing the classic remote control of a drone towards using enhanced interfaces, like using a leap-motion controller and / or a mobile processing device. The main purpose of this work is to understand and observe the proper ways to deal with communication and protocol delays, human adaptation to enhanced controls and identify the integration challenges. Once all this can be achieved at a decent level we plan to extrapolate and conduct more work for trying to apply similar logic for controlling other devices like: collaborative robots, industrial transportation machines and other complex cyber-physical-social devices.

Keywords: cyber-physical-social systems; Leap Motion; human-computer interaction; drone control

1. Introduction

Along with the technology revolution, the human kind must also evolve into understanding and finding new ways to interact with novel devices for easing classical tasks or opening new perspectives.

The main purpose of this study is to present the logic behind a cyber-physical system that uses Leap Motion to control a simple drone without an SDK or other wireless communication out of the box. The hardware components and the software logic are part of an extensible modular architecture meant to adapt also to other use-cases.

This study represents only one of the many stages that will follow in order to better understand how new devices can be integrated into the control loop. The remote drone control experiment is part of a series of similar experiments that aim to pursue 2 important parameters:

- the communication delay as described by Cardoso et al. (2011): how close to real-time can we get when the system involves a chain of devices
- human acceptance as briefly described in (Weiss et al, 2008): how fast can humans adapt to the new controls and how are these controls better than the classic ones

Preceding this experiment, the effort included several simpler use-cases involving single step motors and a remote-controlled toy car. Once we understand how these 2 major parameters are changing as we increase the level of complexity to a 3D spatial environment (drone), we plan to extrapolate the logic towards controlling an industrial collaborative robot.

2. Related work

Since its first release in July 2013, Leap Motion has been used in multiple research studies and interesting experiments. Some of the fields that pioneered these experiments include engineering, robotics (Bassily et al, 2014), medicine (Iosa et al, 2015) and others. Since our primary focus for the exploratory research was to use this device to replace the control function from a remote control, this section will describe the most relevant state of the art research conducted with the help of the Leap Motion controller.

2.1 A Markerless Human–Robot Interface

In a variety of environments, during the teleoperation process for robot manipulators it is often required to use multiple human operators. An interesting experiment done by Guanglong & Zhang (2015) aims to achieve control of a dual robot manipulator by analyzing the double hand–arm movements performed in the natural process of an object manipulation task. The position of the hands is obtained using a Leap Motion controller. Since the measurement errors can sometimes go over a desired limit, a particle filter and a Kalman filter are used to estimate the position and the orientation of the hands. For precision, instead of using the measured data from the Leap Motion, the system uses the estimated data from a Microsoft Kinect device, since humans have motor and perceptive limitation. For testing the accuracy of the solution, the task was to screw a bolt into a nut. The gap between the nut and the bolt was of 0.5 mm and the success of screwing the bolt into the nut confirmed the efficiency of the method. This solution improved the time to finish the operation and also reduced the number of faults, compared to previous approaches.

2.2 Gesture recognition approach for a robotic wheelchair

For people who suffer from severe mobility impairments, technology has an even greater purpose than for the healthy ones. The user interfaces for people with special needs were always a challenge for researchers. While an electric wheelchair seems to need a straightforward control interface, for some people who suffer from more severe diseases (quadriplegics, handicapped children or progressive Parkinson's disease) and have a limited muscular mobility, there is a need for different types of interfaces. One approach proposed by Boyali & Hashimoto (2014) employs the BS-SRC (Block Sparse, Sparse Representation based classification) algorithm described in (Wright et al. 2008). This approach was used to recognize different custom gestures or palm positions for controlling the robotic wheelchair. Using the Leap Motion controller, five palm positions were detected and used: forward, reverse, stop, left and right. The accuracy of palm positions recognition using this algorithm is over 99%. A filter was also used to eliminate some rarely false positive recognition patterns. The data used from the sensor consists only of a few parameters: palm orientation, velocity and normal vector. The simplicity of the method favored high adaptability for various illness levels and allowed the study to focus towards achieving a real-time control system.

2.3 Multi – Leap Motion sensor based demonstration

Finger occlusion is a known problem when dealing with gestures and hand movements. The problem applies only when using a single camera perspective, that is why, a demonstration was conducted by Jin et al. (2016) to show how this particular issue can be solved when the data capturing is done from a multi Leap Motion system. Since the data interpretation has to be done from multiple aggregated inputs, the main point of interest is to identify the correct physical scenario for sensor placement. Multiple arrangements of two Leap Motion controllers were tested: face to face setup, orthogonal setup and 120-degrees setup. Through various testing sessions involving different gestures, the authors concluded that the best setup is the one with the two Leap Motion sensors that are positioned in the 120-degrees setup. In this scenario, when combining the data from the two sensors, the accuracy of the recognition improved with 8% in average. From a more practical point of view, this research is very important for a specific set of operations: tabletop object manipulation tasks focused on precise movements that do not require real-time operations.

2.4 Gesture control of drone using a motion controller

There has been previous work conducted for controlling a drone with a Leap Motion controller described by Sarkar et al. (2016). The authors implemented a drone controller for a Parrot AR drone using the Leap Motion. The main computer runs Linux and uses Robot Operating System (ROS) framework to interact with the drone over Wi-Fi. This is possible because this Parrot AR drone has onboard Wi-Fi and an SDK for easy development and control. Compared to this research, our approach is to use a drone that is not provided with an SDK and adapt it to a different control method than the classical remote control.

2.5 Natural User Interfaces for Human-Drone Multi-Modal Interaction

Another research where a Leap Motion controller is used to control a drone was done by Fernandez et al. (2016). Their focus is on comparing different methods of interaction such as a Graphical User Interface (GUI) and some Natural User Interfaces (NUIs). These interfaces are implemented in a framework for aerial robotics called Aerostack developed by Sanchez-Lopez et al. (2016) and include speech, body position, hand gesture, visual marker interaction and GUI using windows layout, buttons, graphics and images. Similar to the article presented in section 2.4, a Parrot AR drone is used and Aerostack framework is based on ROS.

3. Overall Architecture

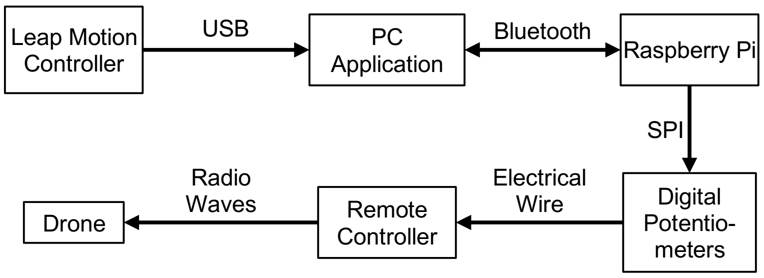


Figure 1. Overall architecture

Figure 1 describes the overall system architecture that was designed and implemented for the purpose of the experiment. The architecture consists of a Leap Motion Controller that is connected to a PC through a USB cable, a software application that runs on that PC, a Raspberry Pi that communicates wireless with the PC application, digital potentiometers that are controlled by the Raspberry Pi, a drone and a classic remote controller. In the following two sections we will describe each component and why we made these choices, explaining the problems we encountered.

By using a modular architecture and the Bluetooth communication protocol that is available on a multitude of modern devices we kept the architecture open into working with other input devices.

4. Hardware and Electronic Components

For the experiment it was decided to use a series of hardware products and components detailed in this section.

4.1 The drone

The chosen drone for this experiment is a mini-drone, model Pocket Drone Quadcopter 124 as seen in Figure 2. It is an inexpensive drone that comes with a remote control and offers a lot of interesting features like: 6-axis stabilization, headless mode and one key return. Due to its size (70mm) and material (plastic and rubber), the experiment needed something that would not break after a long series of attempts. The protection frame and the propellers were severely damaged multiple times and they were replaced at least 10 times during the development stage.

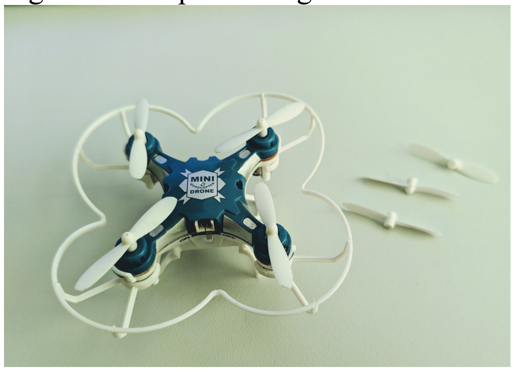


Figure 2. The drone with protection frame on and spare propellers

4.2 The Leap Motion controller

As already stated, a Leap Motion controller was used for the control part in the user side. This device uses two infrared cameras for stereoscopic imaging as there is no depth sensor. Beside the two cameras, there are also three infrared LEDs for a better illumination of the scene and not for infrared pattern projection as in Kinect sensor from Microsoft as shown by Zhang (2012). Compared to Kinect, Leap Motion (as displayed in Figure 3) is very small and has a shorter-range observable area but with a better accuracy. A study done by Weichert et al. (2013) shows that the accuracy of this controller is 0.7 mm. Leap Motion captures a dome area that is sent to the processing desktop station where it is processed and then the data is exposed through an SDK. Leap Motion controller can be used while standing on a desk or attached to a virtual reality headset.

There are 2 different versions of SDK, one for each of the two use cases, as displayed in Figure 4. Leap Motion API exposes a lot of data about the arms, hands, fingers and bones of each finger of users in the observable area like: position, velocities, distances, normal vectors, orientation, predictions for the occluded parts and the confidence of the prediction. Beside the human parts, the API can recognize gestures and also track objects that look like a pointing tool, for example a pencil. These tools need to have a long and thin shape.

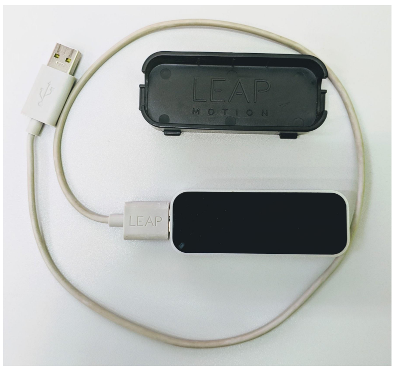


Figure 3. Leap Motion Controller

All the tracking data for a single moment in time is packed in a frame object that is published and updated with a rate of up to 200 fps. This frame rate depends on the computer specifications, the activity and number of users visible in the field of view of Leap Motion Controller. It is also possible to request a new frame at a specific time or a previous frame. Leap Motion controller stores up to the last 60 frames in the frame history.

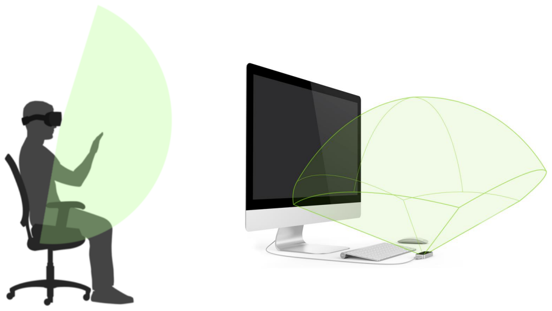


Figure 4. Leap Motion sensing area and two different setup types. Taken from (Leap Motion, 2018)

4.3 Digital potentiometers

With this drone model there are two options for controlling the device. One is to bypass the remote controller by finding out the radio protocol. For this approach we needed to use a circuit like nRF24L01 that can be connected to a computer through USB and implement that specific protocol. The problem with this approach was that these protocols are not documented and not even open to the public because they are developed by a private company. This protocol usually changes for each drone model even for the same company.

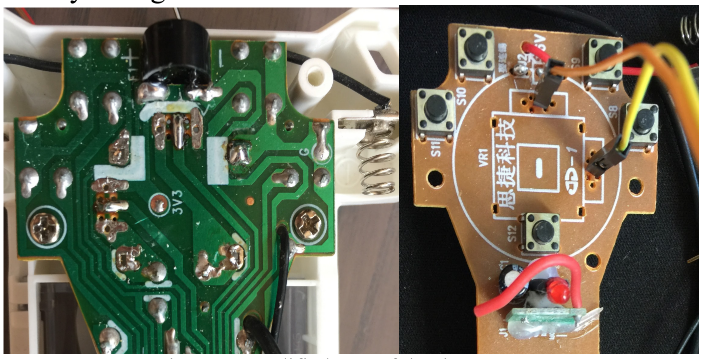


Figure 5. Modified part of the drone remote

The other approach, that we also tried was to modify the remote control and change the input type. This approach was successful. The remote control of the drone uses potentiometers and knobs to take input from the person controlling the device. We needed to replace these mechanical potentiometers with digital ones as represented in Figure 5 and in Figure 6. To do that, we had to unsolder the original potentiometers form the remote and solder wires for the signal that will be generated by the digital potentiometers.

We used 4 MCP4131 digital potentiometers. They are used to control throttle, roll, pitch and yaw. These digital potentiometers use Serial Peripheral Interface (SPI) with speed up to 10 MHz and they have 129 steps. Since they do not support daisy chaining, we had to use 4 separate GPIO pins for Chip Select (CS) pin, one for each digital potentiometer.

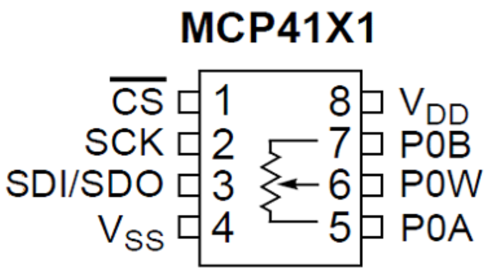


Figure 6. Digital potentiometer pinout

4.4 Raspberry Pi

Since the Leap Motion controller can only be connected to a desktop station that runs either a Linux or a Windows operating system, we needed a way to relay the data collected from the controller to the drone. The solution was to use an intermediary microcontroller that is able to communicate with the desktop using high level protocols but also to interact with the physical world. We decided to use a Raspberry Pi 3 Model B single board computer because it is small, accessible and has a lot of support from the development community. Raspberry Pi 3 (as displayed in Figure 7) packs a lot of computational power: 1.2 GHz quad core ARM 64-bit processor and 1GB of RAM. It also supports many communication and input/output interfaces like HDMI, Ethernet, Wi-Fi, Bluetooth Classic and Bluetooth Low Energy, USB, UART, SPI, I2C and 1-Wire.

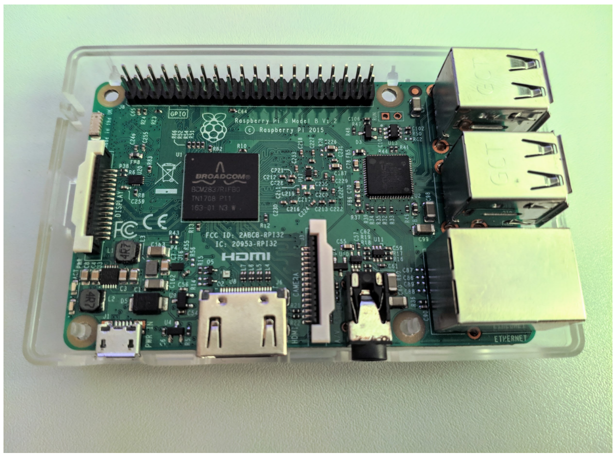


Figure 7. Raspberry Pi 3 Model B with top case open

4.5 Electrical circuit

All the electrical wiring was realized on a breadboard circuit for easy prototyping. We also used a ribbon cable to breakout the Raspberry Pi GPIO pins. The electrical schema for the digital potentiometers and Raspberry Pi is displayed in detail in Figure 8.

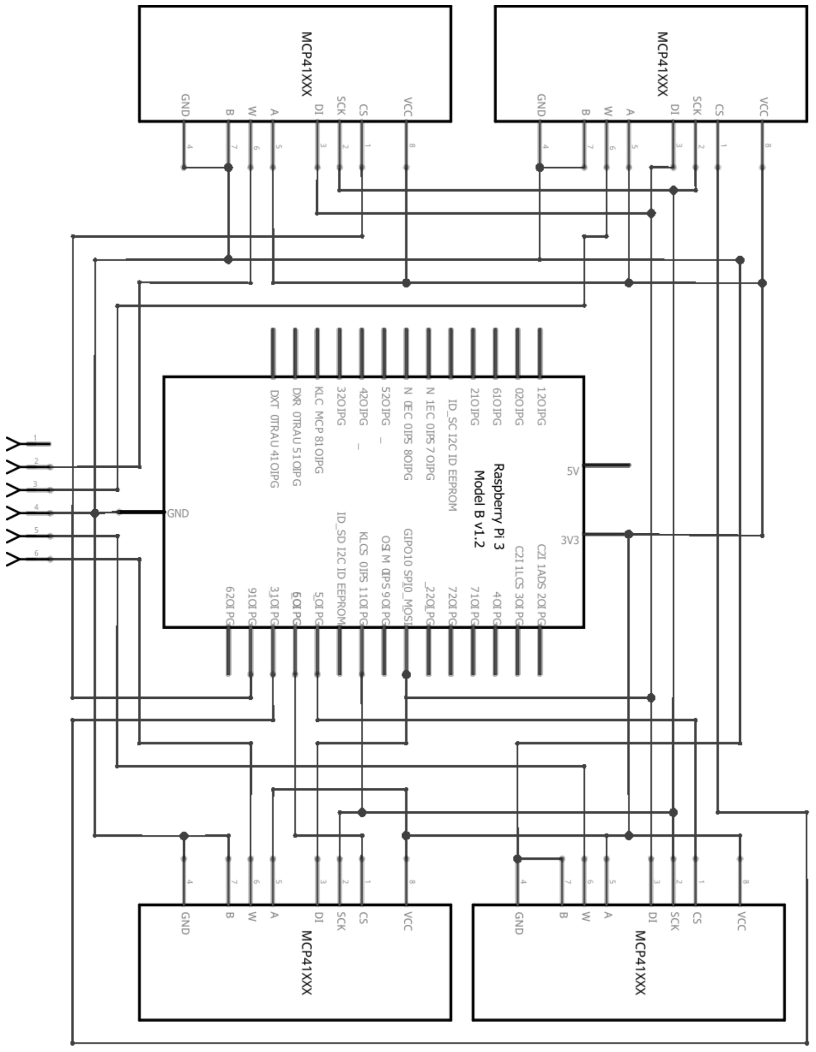


Figure 8. Electrical circuit diagram

4.6 Complete hardware representation

After putting together all the parts and logic, as seen in Figure 9, the complete hardware picture presents the components: drone, modified drone remote, Raspberry Pi, Leap Motion Controller, breadboard and the potentiometers.

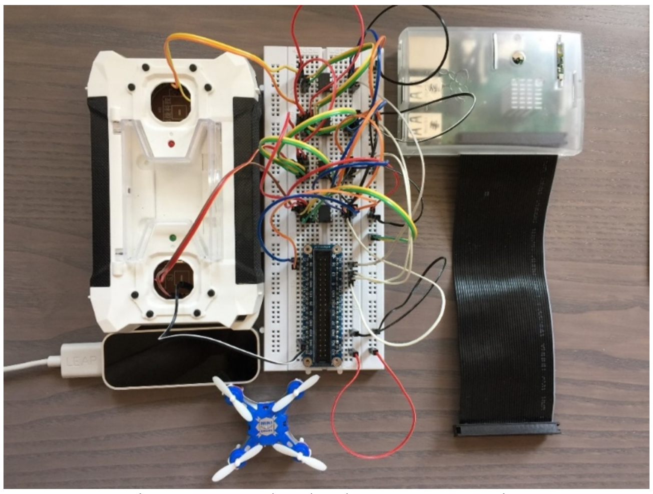


Figure 9. Complete hardware representation

5. Software logic

As for the software part, both Raspberry Pi and the desktop station are running Windows operating systems. This allowed us to write both software applications using C#, to have shared code and to easily deploy the application to the Raspberry Pi from the desktop computer using Visual Studio IDE. Raspberry Pi runs Windows 10 IoT Core and the developed application for it is a Universal Windows Platform (UWP) application. The desktop application that we developed is a Windows Presentation Foundation (WPF) application because this the only one that supports the integration of the Leap Motion SDK.

To communicate between the desktop and Raspberry Pi, we used the Bluetooth protocol as a technology that opens new perspectives for future development. The Raspberry Pi application is set to act as a Bluetooth server and the desktop as a Bluetooth client. To be able to use Bluetooth in our WPF application, we needed to rewrite a module and extend it with UWP capabilities by referencing the required Windows 10 API, which are not available by default like in the UWP application.

The logic for both applications is detailed in the following two diagrams: Raspberry Pi application logic in Figure 10 and desktop application logic in Figure 11.

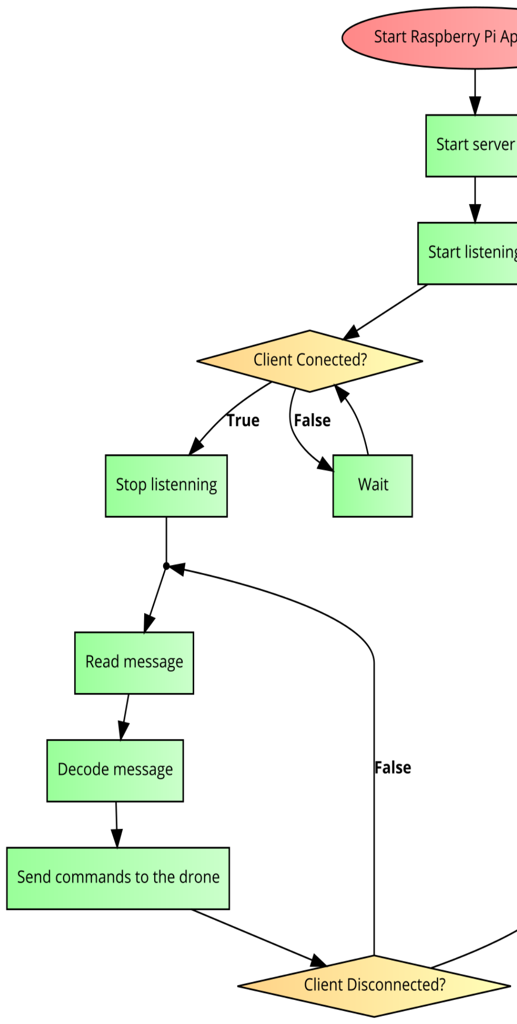


Figure 10. Raspberry Pi application logic diagram

The Raspberry Pi application is a Bluetooth server that waits for a single client to connect. After that, this server will read all the messages from the connected client, decode them and command the drone by controlling the digital potentiometers. When the client disconnects, the server will go back in the listening state.

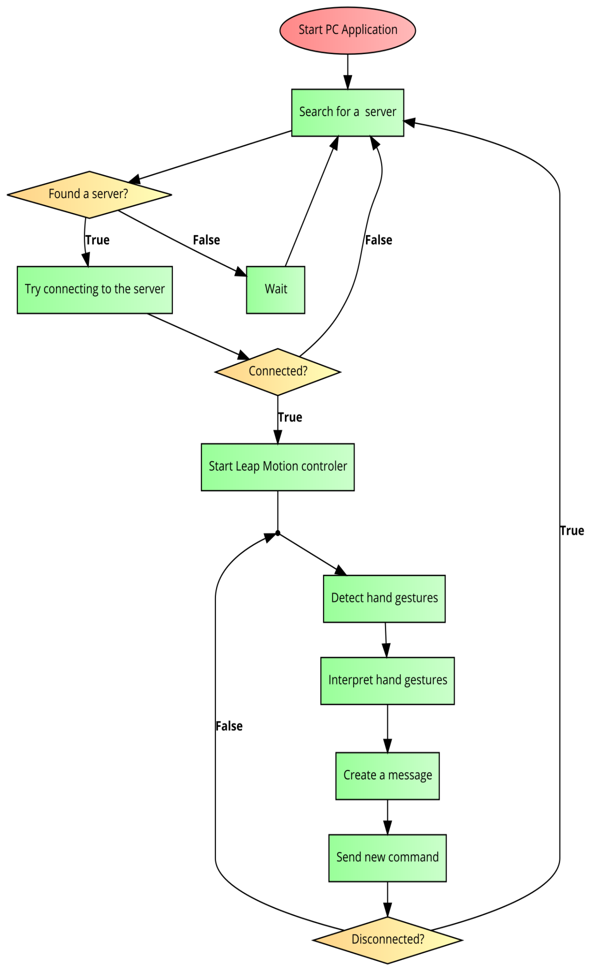
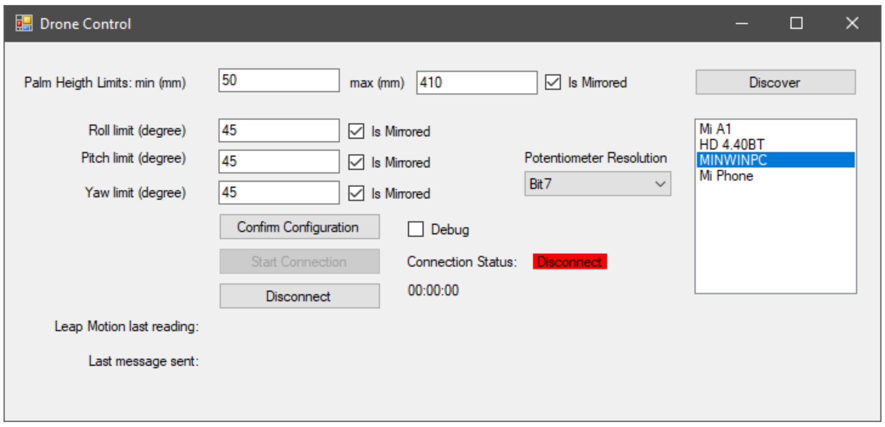


Figure 11. Desktop application logic diagram

The desktop application is a Bluetooth client that will search for a server. If it connected successfully to the server, the client application will initialize the Leap Motion controller. After that, Leap Motion API will be used to read and interpret the data from Leap Motion controller. A message with the commands for the drone will be created and then sent over Bluetooth to the server.

6. User Interface

The desktop application, as displayed in Figure 12, is intended only for configuration and had an experimental development process. The most important feature is that we are allowing the user to control the sensitivity in order to match any palm size, speed and personal movement pattern. This is achieved by changing the limits for palm height and palm angles for roll, pitch and yaw. The data is taken from a single user hand. The broader the movement the finer the control will be. For example, the lowest limit will mean no throttle and the highest will mean max throttle. In case no hand is detected, the application will stop the drone. To make the drone more stable during flights we filtered the input by ignoring the small variation under a threshold, by comparing the current reading with the previous one. These variations are caused by hand tremors and changings in the lightning conditions that can influence the Leap Motion controller readings. In this setup, the Leap Motion controller is placed on the table.


Drone Control

Palm Heigth Limits: min (mm) 50 max (mm) 410 ✓ Is Mirrored

Roll limit (degree) 45 ✓ Is Mirrored

Pitch limit (degree) 45 ✓ Is Mirrored

Yaw limit (degree) 45 ✓ Is Mirrored

Potentiometer Resolution: Bit7

Debug ☐

Connection Status: Disconnect

00:00:00

Leap Motion last reading:

Last message sent:

Mi A1
HD 4.40BT
MINWINPC
Mi Phone

Figure 12. Leap Motion desktop application interface

From a user's perspective, the following steps are necessary to control the drone using the Leap Motion controller:

- Click the Discover button to search for Bluetooth devices. The Bluetooth devices will be listed in the right panel
- Configure all the parameters and select the Bluetooth device from the list on the right
- Click Confirm Configuration button

- Click Start Connection button
- Wait for the Connection Status to change to Connected
- Place the hand above the Leap Motion controller and control the drone

7. Experiments and Results

Throughout the development process and during the testing stages the most important findings were classified into the following 2 areas: human acceptance and real-time perspectives.

7.1 Human acceptance

Comparing with the classical remote control, most of the tests showed that everyone found it easier to take off the drone and fly it around the room using the Leap Motion Controller. The interesting part is that the people who were totally inexperienced with drones were able to control the drone better with the controller than with the original remote. The acceptance was high and all the testing scenarios concluded that this type of control is very intuitive and easy to adapt to different users.

We also found some side issues with our existing approach and current implementation. Since the user has to keep his eyes on the drone, sometimes the user has to rotate his whole body when the drone is moved far away on the sideways or behind him. The rotation of the body involuntarily affects the attention and focus, influencing the angles and height of the controlling hand. Since the Leap Motion controller is placed on the table, when the user rotates the whole body, all the angles are modified and the user loses control of the drone. Another issue that appeared is that sometimes the user moves his hands out of Leap Motion observation area, usually by raising the hand too high or by body rotation. A solution to solve the issue of moving the hand out of the observation area would be to play a sound when the user has its hand close to the margins of observable area.

In terms of users, the subjects that were asked to test the device, during and after the development process can be split into 2 categories:

1. technical people: 4 computer science students (age from 20 to 25) and 3 professors (age from 30 to 55)

2. non-technical people with no previous experience with drones: 1 accountant (age 30), 1 secretary (age 27).

None of them are considered drone experts. Their feedback was gathered verbally and everyone was allowed to try multiple times.

7.2 Real-time perspective

One of the main problems in any hardware-software integration task is how to deal with delays. We made a previous experiment with a toy car (unpublished) that was somehow inconclusive because of the low speed, grip and a very simple 2D physical environment. We decided to conduct the current experiment with a more complex environment (3D) and multiple control parameters (pitch, altitude, direction, etc.). While the drone is flying, even if the precision of the input is high, we diagnosed a very small delay in the reaction process. Comparing to the original remote control reaction time, it is somehow natural that by adding the extra layers (hardware and computation) a certain latency was induced. Since the delay seemed to vary a lot from a test to another, we started to make extra measurements to understand this particular case. Using a minimalistic frame analysis method, similar with Lao & Sundaramoorthi (2017), we detected a minimum (50ms) and maximum delay (250ms). We also concluded that this aspect is negligible (polarized towards the minimum value) when the drone works in a closed laboratory environment. The problem starts to become critical (polarized towards the maximum value) when the communication protocols (Bluetooth and wireless) are used for other parallel operations or there are other Bluetooth and wireless signals interfering in the radio field are other the delay starts to become crucial. We plan to study more this particular aspect in order to have a better point of view.

8. Conclusions and future work

In conclusion, the experiment achieved its purpose: the drone can be controlled with an enhanced interface. The drone remote control was adapted by adding a Raspberry Pi and 4 digital potentiometers. The communication is based on the Bluetooth protocol allowing further improvements. The configuration and input of the application is highly adaptive to the user through an intuitive interface and the control function was taken over by the

Leap Motion controller replacing the classic buttons and joysticks.

We already started to develop other applications to variate the input source and measure the results using the following interfaces: keyboard, mobile phone, modulated sound.

In the near future, the plan is to study the impact of two hands operations using 2 Leap Motion controllers by assigning different functions to each hand (speed and height to the left hand and direction to the right hand).

Another important step would we be also to abstract the configuration and control towards embedding a mixed reality environment similar with (Dascalu et al, 2015), assess the learning curve and explore with a large variety of virtual models, constraints and perturbations.

Acknowledgement

This work is supported through the DiFiCIL project (contract no. 69/08.09.2016, ID P_37_771, web: http://dificil.grants.ulbsibiu.ro), co-funded by ERDF through the Competitiveness Operational Programme 2014-2020.

References

Bassily, D., Georgoulas, C., Guettler, J., Linner, T., Bock, T. (2014). *Intuitive and Adaptive Robotic Arm Manipulation using the Leap Motion Controller*. ISR/Robotik 2014; 41st International Symposium on Robotics, Munich, Germany, 2014, 1-7.

Boyali, A., Hashimoto, N. (2014). *Block-Sparse Representation Classification based gesture recognition approach for a robotic wheelchair*. IEEE Intelligent Vehicles Symposium, Proceedings. Pages 1133-38. doi: 10.1109/IVS.2014.6856392

Cardoso, J., Derler, P., Eidson, J., Lee, E. (2011) *Network Latency and Packet Delay Variation in Cyber-physical Systems*. NSW '11 Proceedings of the 2011 IEEE Network Science Workshop. Pages 51-58, ISBN: 978-1-4577-1049-0

Dascalu, M., Moldoveanu, A.D., Shudayfat, E.A. (2014). *Mixed reality to support new learning paradigms*. 2014 18th International Conference on System Theory, Control and Computing (ICSTCC). Pages 692-697. doi: 10.1109/ICSTCC.2014.6982498

Fernandez, R.A., Sánchez-López, J.L., Sampedro, C., Bavle, H., Molina, M., & Campoy, P. (2016). *Natural user interfaces for human-drone multi-modal interaction*. 2016 International Conference on Unmanned Aircraft Systems (ICUAS). Pages 1013-1022. doi: 10.1109/ICUAS.2016.7502665

Guanglong D., Zhang P. (2015) *A Markerless Human–Robot Interface Using Particle Filter* *and Kalman Filter for Dual Robots*. IEEE Transactions on Industrial Electronics, vol. 62, no. 4. 2257-2264.

Iosa, M., Morone, G., Fusco, A., Castagnoli, M., Fusco, F. R. O., Pratesi, L., and Paolucci, S. (2015). *Leap motion controlled videogame-based therapy for rehabilitation of elderly patients with subacute stroke: a feasibility pilot study*. Topics in Stroke Rehabilitation, 22(4). Pages 306-316. doi: 10.1179/1074935714Z.0000000036

Jin, H., Chen, Q., Chen, Z., Hu, Y., Zhang, J. (2016). *Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task*. CAAI Transactions on Intelligence Technology. Pages 104-113. doi: 10.1016/j.trit.2016.03.010

Lao, D., Sundaramoorthi, G. (2017). *Minimum Delay Moving Object Detection*," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, Pages: 4809-4818. doi: 10.1109/CVPR.2017.511

Leap Motion. (n.d.). Retrieved May, 2018, from https://www.leapmotion.com/

Sanchez-Lopez, J.L., Fernandez, R.A., Bavle, H., Sampedro, P.C., Molina, M., Pestana, J., Campoy, P. (2016). *AEROSTACK: An architecture and open-source software framework for aerial robotics*. 2016 International Conference on Unmanned Aircraft Systems (ICUAS). Pages 332-341. doi: 10.1109/ICUAS.2016.7502591

Sarkar, A., Patel, K., Ganesh Ram, R. K., Capoor, G. (2016). *Gesture control of drone using a motion controller*. 2016 International Conference on Industrial Informatics and Computer Systems. Pages 1-5. doi: 10.1109/ICCSII.2016.7462401

Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). *Analysis of the accuracy and robustness of the leap motion controller*. Sensors (Basel, Switzerland), 13(5). Pages 6380-93. doi:10.3390/s130506380

Weiss, A., Bernhaupt, R., Tscheligi, M., Wollherr, D., Kühnlenz, K., Buss, M. (2008). *A methodological variation for acceptance evaluation of Human-Robot Interaction in public places*. Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN, 713 - 718.

Wright J., Yang A. Y., Ganesh A., Sastry S. S. and Ma Y. (2008). *Robust Face Recognition via Sparse Representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2. Pages 210-227. doi: 10.1109/TPAMI.2008.79

Zhang, Z. (2012). *Microsoft Kinect Sensor and Its Effect*. IEEE Multimedia - IEEEMM. 19, Pages 4-10. doi: 10.1109/MMUL.2012.24.