A Study on an Interaction Between a Drone and a Bystander.

Faizaan Mohammed Mustafa (20871464)

***Abstract*—Drones are commercializing in many ways including them being used for delivering packages to consumers, surveying landscapes, and in Agriculture for many tasks. This paper focuses on the aspects of the human-robot interaction that can occur in a situation where a bystander is receiving a package from a UAV. A system for the human-robot interaction is discussed and a pilot study is conducted on an interaction between a drone and a bystander.**

***Index Terms*—Social Robotics, Drones, Logic, Human-Drone Interaction**

I. Introduction

The demand for drones is rising as it is expected that about 122 thousand shipments are expected for IoT (Internet of Things) enterprise drones by 2023[1]. From military use to quick deliveries, drones are becoming beneficial for businesses and are being widely adopted. Such devices have the capability to reach areas where there is no need for human intervention without taking too much energy, effort, or time. They also have been used for aerial photography, safety inspections, crop monitoring, border surveillance, also storm tracking [1]. This makes drones an asset with many useful use cases. With the many applications mentioned above, it is inevitable that the drone may interact with human beings from time to time in different applications. Since drones are being prepared to be used for applications where they must interact with people in settings that are residential, or urban; It is important that the safety of the people is always considered if such devices are used. There may be a risk of a collision that can occur between the human and the drone. According to an 18-month study performed by the Alliance for Safety of UAS through Research Excellence (ASSURE) to see what type of injuries resulted from a collision between a drone and a human; cuts, bruises, and lacerations were found to be the most common injuries that could occur on a human[2] Serious injuries such as eye damage caused by the rotating blades of the drone were also reported that this required for the drone to have a perfect collision with the human. Such risks lead to the need of having rules for safety and regulation in the construction and use of drone technologies. For avoiding any harm to humans who may end up interacting with the drone, we propose an HRI system that makes use of sound alerts, and LEDs for notifying the user when it is near the human. This system will be tested at various distances between a bystander and the drone in a mixed-factorial human-robot interaction (HRI) experiment that will be presented later in this paper.

II. Related Works

A lot of research is presented on the interaction between a drone and a human being.[3] stated that adding an emotional component to personal drones can make those drones more acceptable to people. They define the emotional and personality traits that can be added to the drone to make it more acceptable. A user study was conducted with 20 participants to show how well three emotional states are recognized by the participants. The results showed that their social drones can become acceptable as most participants compared the drone to a pet.

[4] talks about how the blades of drones when exposed can raise safety concerns and how to address this, designs of drones that are “safe-to-touch” are being proposed for better safety of the people who interact with the drone. They proposed a study where there was a drone that was safe to touch and a drone that was unsafe to touch to find out if people would prefer interacting with a drone that was safe to touch by instinctively touching the drone. They had a set of tasks that included the safe-to-touch drone and the not-safe-to-touch drone that were to be performed by the participants of the study. They found out that 58 percent of participants were touching the drone while 39 percent of the interactions over all the tasks were touch-based [4]. The participants reported that they were mentally less concerned about their safety when interacting with the safe-to-touch drone compared to the unsafe drone.

[5] present a social drone that is human-centered and that is being used in a crowded environment. The drone consisted of a social shape, social face, and social voice for interacting with humans. They did a proxemic study to find out what distance would the humans comfortably accept between them and the drone. They found out that the social shape of the drone with an added greeting voice decreased the accepted distance between them and the drone. They also discovered the lateral distance and the height of 1.8 m compared to distance and height of 1.2m decreased the comfortable distance between the drone and the human as the drone approached the human[5].

[6] explores a study where there is a human-drone interaction between participants and a drone with the inclusion of safety glass in the middle. The study revealed the drone’s size, altitude, and added gender factor did not affect the proxemics of how close the participants got to the hovering drone with the limitation caused by the safety wall in the middle. But it was revealed that the participants were attracted to drones that were present at the eye level of the participants compared to

being over-head. Also, females were more attracted to drones that were more closely resembled males. These related works inspired us to design and construct our pilot study on a non-verbal interaction between a drone and a bystander.

III. Research Questions

For our research questions, we primarily focused on different aspects of the non-verbal interaction that occurs between the drone and the bystander. This includes the acceptance of people if they were to be exposed to signals when interacting with the drone, the perception of the people based on safety if the drone gives them signals indicating if the drone is landing or if it is at a safe altitude, and the perception of people on the usability of such a drone.

There are three research questions that are related to our study when it comes to the interaction between a drone and a bystander. The research questions for the experiment include:

A.Which distance with the use of an HRI System for a UAV delivery system results in a safer delivery while keeping the safety of the bystander in mind?

B. Which distance with the use of an HRI System for a UAV delivery system results in greater usability in delivery?

C. Can the use of signals that combine sounds and LED lighting offer a non-verbal interaction that is informative to the user and yet a convenient way to ensure a safe and seamless package delivery in various proxemics?

Thus, we focus on the ideal distance between a drone and a bystander when the drone is delivering a package while keeping the safety of the bystander in mind, the ideal distance between a drone and a bystander when the drone is delivering a package while keeping the usability of the drone by the bystander in mind, and finally, if the use of signals by the drone results in safe and seamless package deliveries at varying proxemics. These research questions form the basis for a pilot study which will be discussed in the next section.

IV. Methodology

The pilot study involved a 3*2 Mixed Factorial Design to answer our research questions. Details on the methodology were specified in the following questions.

A. **Overall Design**: where two variables were manipulated including the distance between the drone and the bystander in meters, and the presence of the signal as shown in figure 1.

The distances included 1m, 3.2m, and 6.7m which were tested in two different scenarios wherein in one situation the drone was landing, and in another situation, the drone was taking off. In both situations, a bystander was separated from the drone at the distances that are specified in this paper. When it comes to the signals, there were two signals present in the study where the first signal is used when the drone lands. For the signal variable, the signal is either present or not present. There are two signals that are present in the experiment. The first signal involves the drone forming a red LED signal and making an alert signal when the drone reaches its landing spot. The second signal involves the drone forming a green LED signal with a notification sound informing the bystander that the bystander can pass by or take his/her package. When the signal is not present, the drone landing and take off are simulated but without the signals.

Distance(m)	Signal
1m	Signals Present(1)
3.2m	No Signal Present(0)
6.7m	

Fig. 1. Manipulated Variables

Distance*(m)	No Signal Present(0)	Signal Present(1)
1	(1,0)	(1,1)
3.2	(3.2,0)	(3.2,1)
6.7	(6.7,0)	(6.7,1)
	Group 1	**Group 2**

*The distance between the drone and the bystander.

Fig. 2. 2*3 Mixed Factorial Design

Different combinations of the two variables were made to form the study as indicated in figure 2. 6 different cases were formed, and simulations related to each case were created in WeBots. As indicated in figure 2, the cases where the signal was present were included in one survey and the cases where the signal was not present were included in another survey. Both surveys are distributed to different groups with questions. The responses to the questions are recorded and are used to find answers to the research questions.

B. **Participants**: For the pilot study that was conducted, 12 random students from the University of Waterloo were divided into two groups of 6 to fill two Google forms that consisted of simulations of the various scenarios presented as part of the study. Questions were presented after each simulation. These questions were related to the research questions presented earlier in the paper.

C. **Artifacts**:

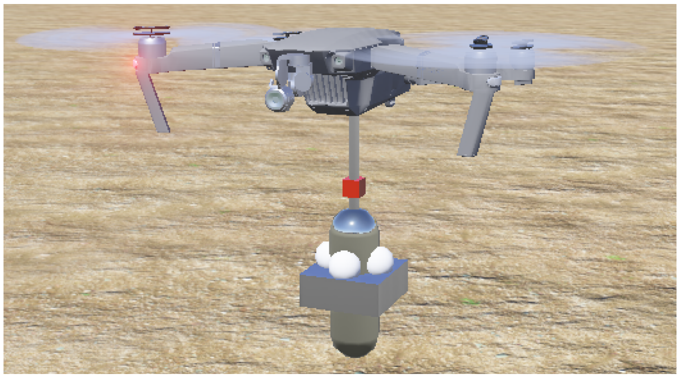

Fig. 3. HRI Communication Module

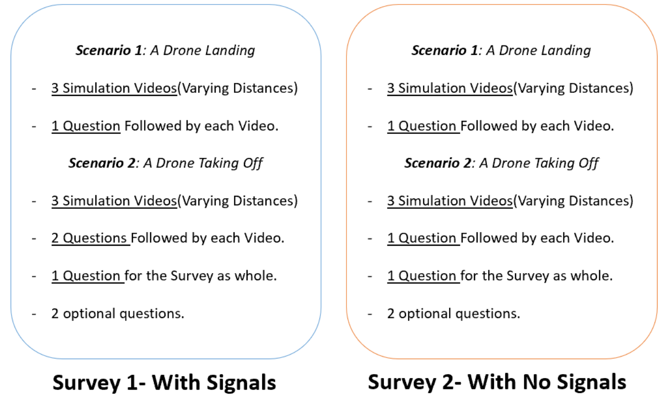

Fig. 4. Survey Format for Survey 1 and 2

The robot which was selected was a DJI Mavic 2 pro which was from the demo program present in the DJI section within the sample programs for robots present in WeBots. The world included trees, a drone base, and a vehicle, It already had a pre-programmed controller that consisted of functions for it being controlled manually. The robot base was used as the landing and take-off spot for the drone.

The functions included raising the altitude of the drone, lowering the altitude of the drone, making the drone go forward, making the drone go backward, turning the drone in the left direction, turning the drone in the right direction, making the robot go sideways to the left, and making the robot go sideways to the right. Each function is assigned its own key on a computer keyboard for manual control. The most important part of our study was to have the drone carry a light load while having the HRI (human-robot interaction) system attached to it.

The HRI communication system needed to have the capabilities of creating a signal that alerts the bystander or signals the bystander that it is ok for the bystander to collect his/her package or to pass by as the drone is now at a high altitude. Thus, for our study, an extended rope in solid form with an end consisting of 4 LEDs and a speaker was attached to the body slot of the drone in that drone's tree structure. Each LED was programmed to manually turn red or green, and the speaker was programmed to manually create an alert sound or create a sound that it is safe for the bystander to collect his/her package or to pass by as the drone is now at a high altitude. This formed the complete HRI communication system module and is shown in . To manually control the HRI communication system, each piece of the module was programmed and was given a manual keyboard button for initializing the signal given the situation that is presented.

The HRI communication module was designed to be implemented in the two scenarios where a drone is either landing or when the drone is at a certain height from its base with that drone having the capability to give a signal to the bystander. The module is presented in figure 3.

D. **Procedures and Measures**:

At the beginning of the first survey on the first sheet, the participant who is filling the survey is presented with a scenario where that person is asked to assume that the person is a bystander in a situation where the drone is landing. The surveys used in the pilot study included 3 simulations videos of the drone landing with the bystander at varying distances followed by a question after each video. On the second sheet, the participant is presented with the second scenario where the drone gives a signal after reaching a safe altitude away from the bystander. 3 simulation videos of the drone giving a signal with the bystander at different distances is shown followed by 2 questions after each video. Then, 1 question is presented followed by two optional questions. This format was repeated for the second survey as well where the simulation videos did not contain any signals. The format of both surveys is indicated in figure 3.

12 simulations were designed in total for the surveys with the first 6 simulations containing signals (alert and take-off signal) at the distances specified above between the drone and the bystander. The 6 simulations contain 3 landing videos and 3 take-off videos with signals from the drone to the bystander when the drone is landing or taking off. The simulations were filmed from the third person point of view. These simulations were included in the first survey with subsequent questions present after each simulation. The diagrams illustrates indicated in figure 4 and figure 5 show the signals at landing and takeoff. The code for the simulations is attached in the Appendix section.

The last 6 simulations contain 3 landing videos and 3 take-off videos at varying distances as indicated above but with no signals whatsoever for both surveys. Questions were included that inquired if there is a need for the drone to have a function to signal the bystander. The format survey The questions were mostly Yes/No questions and a Likert Scale question for determining what distance the bystander would consider ideal between the bystander and the drone. The questions are presented in figure 6.

In the first scenario of drone landing, the question asked after each simulation inquires the participant that if that participant would accept having a drone deliver them a package that

Fig. 5. Drone Landing with Alert Signal for the Bystander

Fig. 6. Drone Signal Indicating Clearance to Pass for the Bystander

Scenario	Questions	Response Type
Landing	Would you consider having a drone deliver you a package that would alert you by using alert sounds and LED lights if it detects you near it? **(Measuring Acceptance)**	Yes, No, Maybe.
After Take-off	As a bystander, would you deem having a signal informing you that it is safe to pick up your package necessary? **(Measuring Safety Perspective)**	Yes, No, Maybe.
After Take-off	Is it better to just have the alert signal go off after reaching a safe altitude away from the bystander after delivering the package? **(Measuring Safety Perspective)**	Yes, No, Maybe.
-	What range of distance would you consider to be optimal between you and the drone (in meters) for you to receive the alert signal if you were the receiver approaching the package? **(Measuring Usability Perspective)**	1m, 3.2m, 6.7m

Fig. 7. Question Types Presented in the Survey

uses alert sounds and LED lights if it detects the participant near them. This question relates to the third research question presented earlier in section 3 where the user's acceptance of non-verbal interaction by a drone with the bystander using signals is measured. In the second scenario where the drone gives a signal after reaching a decent altitude away from the bystander, the first question asks if the participants would consider a drone that tells them that it is safe to pick up the package or to pass necessary or not. This question is placed to measure people's perception on if the drone having functionalities of signaling make the non-verbal interaction that may occur between the drone and the bystander safer. The second question asks another question that if it is ok to have the red alert signals to go off after the reaching a decent altitude away from the bystander without the use of any signal by the drone telling them that they can pass or collect their package. The second question was removed from the second survey as the question was referring to the red alert which was not shown in the second survey.

On the last sheet of the survey, a Likert based question was included asking the participant what distance would they consider to be optimal between them and the drone for them to receive the an alert signal from the drone if they were the ones receiving the package. Some optional questions that are not included in figure 6 were also included in both surveys asking for suggestions on how to improve the study, and if the students want to know the results of the study after its completion.

V. Results

The results of the survey conducted by 6 people from each group provide us some insight into what is acceptable and what needs to be changed in our pilot study.

The first research question asked how acceptable it would be for a bystander to have drone use signals to alert for safe and seamless delivery. The responses from the first survey indicate that most people would not consider having a drone alert them

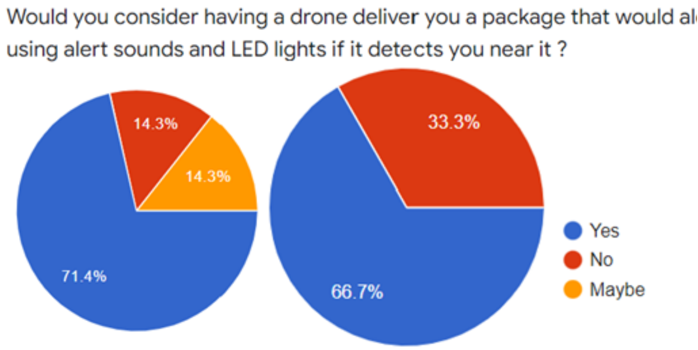

Fig. 8. Survey 1(Left) and Survey 2(Right) Responses to Question 1 “Would you consider having a drone deliver you a package that would alert you by using alert sounds and LED lights if it detects you near it ?”

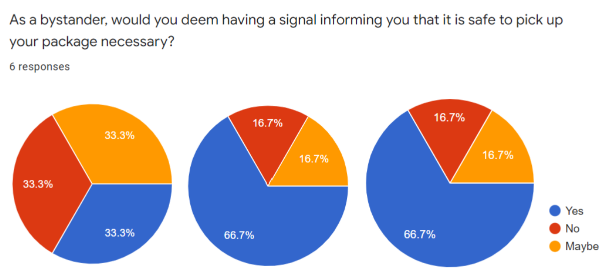

Fig. 9. Survey 1 Results for the Question on Safety

when they are distances of 6.7m and 3.2m but would always want to have these functionalities on the drone if the drone is 1 m away as shown in the pie chart in figure 7 with more almost 50 percent of the correspondents saying yes to the question that addressed the first research question. The second survey indicated similar results with 50 percent saying yes to the same question as well as shown in Figure 7.

The second research question addressed safety and distance; it was revealed from the first survey that people deem having a signal informing them that it is safe to pick up a package as necessary no matter how far they are from the drone as indicated in the responses to the question “As a bystander, would you deem having a signal informing you that it is safe to pick up your package necessary?” in Figure 8.

In survey 2, similar results were found where people still preferred that people deem having a signal informing them that it is safe to pick up a package as necessary no matter how far they are from the drone as indicated in Figure 9.

Thus, for the second research question, results were found that people still preferred that people deem having a signal informing them that it is safe to pick up a package as necessary no matter how far they are from the drone.

For the last research question, the distance which is optimal for usability is determined. For both the surveys, it was indicated that people preferred 3.2 m as the optimal distance between them in both surveys and if the drone wants to use

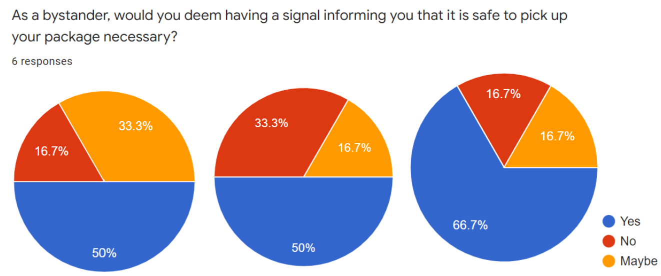

Fig. 10. Survey 2 Results for the Question on Safety

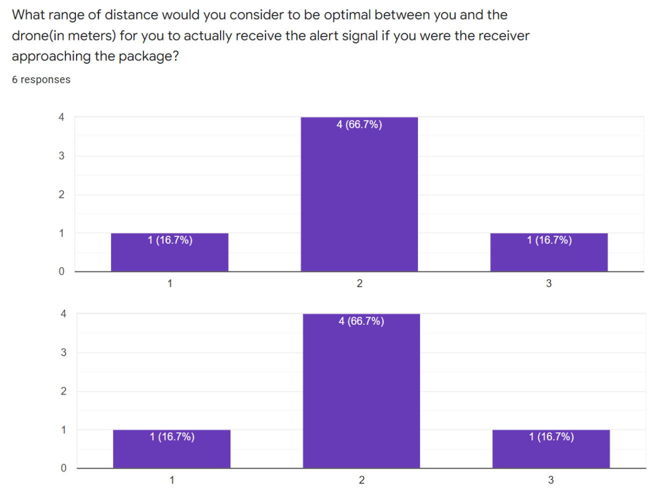

Fig. 11. Survey Results for the Question on Usability

signals to notify the bystander as indicated in Figure 6.

Thus, we can conclude that 3.2 m was the ideal distance for usability as indicated in the graphs present in figure 6.

Results that were not in contrast to the results shown previously were shown from the results of question 2 in the second scenario as presented in figure 11.

In the results shown in figure 11, no matter how far away the bystander is from the drone; most participants indicate that

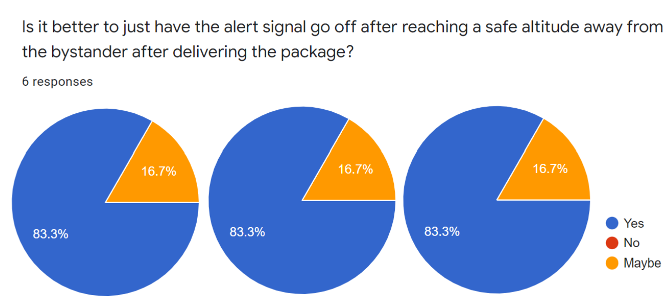

Fig. 12. Question on Using the Alert Signal Only

it is better to just have the alert signal go off after the drone reaches a safe altitude. This may have occurred due to the participants bias towards having no disturbance in the form of noise that may have occurred after the alert signal went on for most participants.

VI. Limitations and Critical Evaluations

The limitations to this experiment include firstly the limited number of participants as we were only able to get 10 participants for the pilot study. Due to our limited time frame, we could not get more participants for this study. Thus, the results of this experiment may change if a lot of participants are included in a follow-up study.

Secondly, we did not consider different contexts for this drone experiment which thus makes out experiment slightly limited to testing in an open field as shown in our simulations.

Thirdly, the distances of 1m, 3.2m, and 6.7m were found to be unrealistic as indicated in the survey feedback for drone simulations compared to real-world usage.

VII. Future Works

For our future work, we propose to use Amazon Turk for getting participants for our study. We also propose to test our experiment in different contexts with distances between the drone and the bystander that are more realistic.

VIII. Conclusion

We proposed a pilot study to study the non-verbal interaction between a drone and a bystander through the use of signals that include LED lights and sound effects. We used a 3*2 mixed-factorial design with the inclusion of variables that include the proxemics of distance and the presence or absence of a signal. Simulations were created in the WeBots simulator for each case with each case being a pair combination of the two variables discussed in this paper. Two surveys were made where in one survey, there were simulations in which the drone was giving signals to the bystander at varying distances and in the other survey, there were simulations in which the drone was not giving any signals to the bystander at all. The surveys were distributed with six responses that were received for each survey. The results of the surveys indicate that most people would consider having a drone that would use signals to notify them when the drone is as close as 1 m to the bystander. The results also indicated that people would deem having a drone indicating them that it is safe to pick up a package no matter what the distance. Lastly, it was found that people would find the need of signals from a drone useful at a distance of 3.2m. We can conclude from this pilot study that a drone having the functionality to signal people around them is very useful and can lead to a form of non-verbal interaction between the drone and the bystander that can in turn lead to ensuring the safety of the people while the drone is adopted and used for the many application that were discussed at the beginning of this paper.

References

[1] Intelligence, I. (2021, January 12). Drone technology uses and applications for commercial, industrial and military drones in 2021 and the future. Business Insider. Retrieved April 3, 2022, from https://www.businessinsider.com/drone-technology-uses-applications

[2] Tegler, E. (2021, November 2). What happens when a drone crashes into your face? Popular Mechanics. Retrieved April 3, 2022, from https://www.popularmechanics.com/flight/drones/a28774546/drone-head-collision/

[3] Cauchard, J. R., Zhai, K. Y., Spadafora, M., amp; Landay, J. A. (2016, April 14). Emotion encoding in human-drone interaction. IEEE Xplore. Retrieved April 5, 2022, from https://ieeexplore.ieee.org/abstract/document/7451761

[4] University, P. A. S., Abtahi, P., University, S., Profile, S. U. V., University, D. Y. Z. S., Zhao, D. Y., University, J. L. E. S., E., J. L., University, J. A. L. S., Landay, J. A., Contributor MetricsExpand All Parastoo Abtahi Stanford University Publication Years2017 - 2021Publication c, amp; Parastoo Abtahi Stanford University Publication Years2017 - 2021Publication counts9Available for Download9Citation count157Downloads (cumulative)7. (2017, September 1). Drone near me: Exploring touch-based human-drone interaction: Proceedings of the ACM ON Interactive, Mobile, wearable and Ubiquitous Technologies: Vol 1, no 3. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. Retrieved April 3, 2022, from https://dl.acm.org/doi/10.1145/3130899

[5] Alexander Yeh Chalmers University of Technology, Yeh, A., Technology, C. U. of, University, P. R. O., Ratsamee, P., University, O., Kiyoshi Kiyokawa Nara Institute of Science and Technology (NAIST), Kiyokawa, K., Nara Institute of Science and Technology (NAIST), University, Y. U. O., Uranishi, Y., University, T. M. O., Mashita, T., University, H. T. O., Takemura, H., Morten Fjeld Chalmers University of Technology, Fjeld, M., University, M. O. U., Obaid, M., ... Alexander Yeh Chalmers University of Technology Publication Years2017 - 2017Publication counts1Available for Download1Citation count26Downloads (cumulative)618Downloads (6 weeks)18Downloads (12 months)144Average Ci. (2017, October 1). Exploring proxemics for human-drone interaction: Proceedings of the 5th International Conference on Human Agent Interaction. ACM Conferences. Retrieved April 5, 2022, from https://dl.acm.org/doi/10.1145/3125739.3125773

[6] Han, J. M. (2019, June 30). Exploring the social proxemics of human-drone interaction. International journal of advanced smart convergence. Retrieved April 5, 2022, from https://www.koreascience.or.kr/article/JAKO201918266939327.page

A. Appendix

1. Code

```
/*

 * Copyright 1996-2021 Cyberbotics Ltd.

 *

 * Licensed under the Apache License, Version 2.0 (the "License");

 * you may not use this file except in compliance with the License.

 * You may obtain a copy of the License at

 *

 * http://www.apache.org/licenses/LICENSE-2.0

 *

 * Unless required by applicable law or agreed to in writing, software

 * distributed under the License is distributed on an "AS IS" BASIS,

 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

 * See the License for the specific language governing permissions and

 * limitations under the License.

 */


/*

 * Description:  Simplistic drone control:

 * - Stabilize the robot using the embedded sensors.

 * - Use PID technique to stabilize the drone roll/pitch/yaw.

 * - Use a cubic function applied on the vertical difference to stabilize the robot vertically.

 * - Stabilize the camera.

 * - Control the robot using the computer keyboard.

 */


#include <webots/speaker.h>

#include <webots/connector.h>


#include <math.h>

#include <stdio.h>

#include <stdlib.h>


#include <webots/robot.h>

#include <webots/supervisor.h>


#include <webots/camera.h>

#include <webots/compass.h>

#include <webots/gps.h>

#include <webots/gyro.h>

#include <webots/inertial_unit.h>

#include <webots/keyboard.h>

#include <webots/led.h>

#include <webots/motor.h>

#include <string.h>


#define SIGN(x) ((x) > 0) - ((x) < 0)

#define CLAMP(value, low, high) ((value) < (low) ? (low) : ((value) > (high) ? (high) : (value)))
```

```
//const WBNodeRef str= "Mv2.s1";

//const char*

double pos;

double linear =0;


//WbNodeRef *test=
wb_supervisor_node_get_from_def("rt")
;



//WbNodeRef transNode=
wb_supervisor_node_get_from_proto_d
ef("solid1");


int main(int argc, char **argv) {

  wb_robot_init();

  int timestep =
(int)wb_robot_get_basic_time_step();



  // Get and enable devices.

  WbDeviceTag rd =
wb_robot_get_device("led");

  WbDeviceTag rd2 =
wb_robot_get_device("led2");

  WbDeviceTag rd3 =
wb_robot_get_device("led3");

  WbDeviceTag rd4 =
wb_robot_get_device("led4");


  WbDeviceTag spk =
wb_robot_get_device("speaker");

  WbDeviceTag camera =
wb_robot_get_device("camera");

  wb_camera_enable(camera, timestep);

  WbDeviceTag front_left_led =
wb_robot_get_device("front left led");

  WbDeviceTag front_right_led =
wb_robot_get_device("front right led");

  WbDeviceTag imu =
wb_robot_get_device("inertial unit");

  wb_inertial_unit_enable(imu,
timestep);

  WbDeviceTag gps =
wb_robot_get_device("gps");

  wb_gps_enable(gps, timestep);

  WbDeviceTag compass =
wb_robot_get_device("compass");

  wb_compass_enable(compass,
timestep);

  WbDeviceTag gyro =
wb_robot_get_device("gyro");

  wb_gyro_enable(gyro, timestep);

  wb_keyboard_enable(timestep);

  WbDeviceTag camera_roll_motor =
wb_robot_get_device("camera roll");

  WbDeviceTag camera_pitch_motor =
wb_robot_get_device("camera pitch");

  // WbDeviceTag camera_yaw_motor =
wb_robot_get_device("camera yaw");  //
Not used in this example.


  // Get propeller motors and set them to
velocity mode.

  WbDeviceTag front_left_motor =
wb_robot_get_device("front left
propeller");
```

```
  WbDeviceTag front_right_motor = wb_robot_get_device("front right propeller");
  WbDeviceTag rear_left_motor = wb_robot_get_device("rear left propeller");
  WbDeviceTag rear_right_motor = wb_robot_get_device("rear right propeller");
  WbDeviceTag motors[4] = {front_left_motor, front_right_motor, rear_left_motor, rear_right_motor};

  int m;
  for (m = 0; m < 4; ++m) {
    wb_motor_set_position(motors[m], INFINITY);
    wb_motor_set_velocity(motors[m], 1.0);
  }

  // Display the welcome message.
  printf("Start the drone...\n");

  // Wait one second.
  while (wb_robot_step(timestep) != -1) {
    if (wb_robot_get_time() > 1.0)
      break;
  }

  // Display manual control message.
  printf("You can control the drone with your computer keyboard:\n");
  printf("- 'up': move forward.\n");
  printf("- 'down': move backward.\n");
  printf("- 'right': turn right.\n");
  printf("- 'left': turn left.\n");
  printf("- 'shift + up': increase the target altitude.\n");
  printf("- 'shift + down': decrease the target altitude.\n");
  printf("- 'shift + right': strafe right.\n");
  printf("- 'shift + left': strafe left.\n");

  // Constants, empirically found.
  const double k_vertical_thrust = 68.5;  // with this thrust, the drone lifts.
  const double k_vertical_offset = 0.6;   // Vertical offset where the robot actually targets to stabilize itself.
  const double k_vertical_p = 3.0;        // P constant of the vertical PID.
  const double k_roll_p = 50.0;           // P constant of the roll PID.
  const double k_pitch_p = 30.0;          // P constant of the pitch PID.

  // Variables.
  double target_altitude = 1.0;  // The target altitude. Can be changed by the user.

  //WbDeviceTag rp = wb_robot_get_device("linear");
  //int value;
```

```
  // Main loop
  while (wb_robot_step(timestep) != -1) {
    const double time = wb_robot_get_time();  // in seconds.

    // Retrieve robot position using the sensors.
    const double roll = wb_inertial_unit_get_roll_pitch_yaw(imu)[0];
    const double pitch = wb_inertial_unit_get_roll_pitch_yaw(imu)[1];
    const double altitude = wb_gps_get_values(gps)[2];
    const double roll_acceleration = wb_gyro_get_values(gyro)[0];
    const double pitch_acceleration = wb_gyro_get_values(gyro)[1];

    // Blink the front LEDs alternatively with a 1 second rate.
    const bool led_state = ((int)time) % 2;
    wb_led_set(front_left_led, led_state);
    wb_led_set(front_right_led, !led_state);

    // Stabilize the Camera by actuating the camera motors according to the gyro feedback.
wb_motor_set_position(camera_roll_motor, -0.115 * roll_acceleration);

wb_motor_set_position(camera_pitch_motor, -0.1 * pitch_acceleration);

    // Transform the keyboard input to disturbances on the stabilization algorithm.
    double roll_disturbance = 0.0;
    double pitch_disturbance = 0.0;
    double yaw_disturbance = 0.0;
    int key = wb_keyboard_get_key();
    while (key > 0) {
      switch (key) {
        case WB_KEYBOARD_UP:
          pitch_disturbance = -1.0;
          break;
        case WB_KEYBOARD_DOWN:
          pitch_disturbance = 2.0;
          break;
        case WB_KEYBOARD_RIGHT:
          yaw_disturbance = -1.3;
          break;
        case WB_KEYBOARD_LEFT:
          yaw_disturbance = 1.3;
          wb_led_set(rd, (int)(0xffffff * ((float)rand() / RAND_MAX)));

          break;
        case (WB_KEYBOARD_SHIFT + WB_KEYBOARD_RIGHT):
```

```
    //Turn green

    wb_speaker_play_sound(spk, spk,"c1.wav", 1, 1, 0, false);

    wb_led_set(rd, 2);
    wb_led_set(rd2, 2);
    wb_led_set(rd3, 2);
    wb_led_set(rd4, 2);

    break;

    case (WB_KEYBOARD_SHIFT + WB_KEYBOARD_LEFT):

   //Turn red

    //roll_disturbance = 1.0;
    wb_led_set(rd, 1);
    wb_led_set(rd2, 1);
    wb_led_set(rd3, 1);
    wb_led_set(rd4, 1);

    break;

   case ('P'):

    roll_disturbance = -1.0;

    break;

   case ('O'):

    roll_disturbance = 1.0;

    break;

   case (WB_KEYBOARD_SHIFT + WB_KEYBOARD_UP):
    target_altitude += 0.1;
    printf("target altitude: %f [m]\n", target_altitude);
    break;
   case (WB_KEYBOARD_SHIFT + WB_KEYBOARD_DOWN):
    target_altitude -= 0.1;
    printf("target altitude: %f [m]\n", target_altitude);
    break;

   case (WB_KEYBOARD_SHIFT + 'W'):

   // Turn LEDs off

    wb_led_set(rd, 0);
    wb_led_set(rd2, 0);
    wb_led_set(rd3, 0);
    wb_led_set(rd4, 0);

    break;
```

```
        case (WB_KEYBOARD_SHIFT +
'T'):

        // Play Sound

          wb_speaker_play_sound(spk,
spk,"t1.wav", 1, 1, 0, false);

          break;
      }
      key = wb_keyboard_get_key();
    }

        // Compute the roll, pitch, yaw and
vertical inputs.
    const double roll_input = k_roll_p *
CLAMP(roll, -1.0, 1.0) +
roll_acceleration + roll_disturbance;
    const double pitch_input = k_pitch_p
* CLAMP(pitch, -1.0, 1.0) +
pitch_acceleration + pitch_disturbance;
    const double yaw_input =
yaw_disturbance;
    const double
clamped_difference_altitude =
CLAMP(target_altitude - altitude +
k_vertical_offset, -1.0, 1.0);
    const double vertical_input =
k_vertical_p *
pow(clamped_difference_altitude, 3.0);

    // Actuate the motors taking into
consideration all the computed inputs.
    const double front_left_motor_input =
k_vertical_thrust + vertical_input -
roll_input + pitch_input - yaw_input;
    const double front_right_motor_input
= k_vertical_thrust + vertical_input +
roll_input + pitch_input + yaw_input;
    const double rear_left_motor_input =
k_vertical_thrust + vertical_input -
roll_input - pitch_input + yaw_input;
    const double rear_right_motor_input
= k_vertical_thrust + vertical_input +
roll_input - pitch_input - yaw_input;

wb_motor_set_velocity(front_left_motor
, front_left_motor_input);

wb_motor_set_velocity(front_right_mot
or, -front_right_motor_input);

wb_motor_set_velocity(rear_left_motor,
-rear_left_motor_input);

wb_motor_set_velocity(rear_right_moto
r, rear_right_motor_input);
  };

  wb_robot_cleanup();

  return EXIT_SUCCESS;
}
```