Komunikacja i sterowanie dronem za pośrednictwem MATLABa

Jakub Szczygieł

Sterowanie pozycyjne

```
function position_callback(app, ~, message)
    % pozycja
    pos = [0 0 0];
    pos(1) = message.Pose.Pose.Position.X;
    pos(2) = message.Pose.Pose.Position.Y;
    pos(3) = message.Pose.Pose.Position.Z;
    % orientacja
    w = message.Pose.Pose.Orientation.W;
    x = message.Pose.Pose.Orientation.X;
    y = message.Pose.Pose.Orientation.Y;
    z = message.Pose.Pose.Orientation.Z;
    % roll pitch yaw
    rot = quat2eul([w x y z]);
    rot = rot(end:-1:1);

    position = [pos, rot];
    if strcmp(app.OdomPozycjaSwitch.Value, 'Faktyczna pozycja')
        app.OdomUITable.Data = position;
    end
```

```
    % sterowanie pozycyjne
    if app.reached == false
        position = position([1 2 3 6]); % x y z theta(yaw)
        diff = app.goal_position-position;

        cmd_vel = app.Kp*diff+app.Ki*app.diff_sum+app.Kd*(diff-app.diff_prev);
        cmd_vel(cmd_vel>1) = 1;
        cmd_vel(cmd_vel<-1) = -1;

        % obrót od układu współrzędnych drona
        theta = position(4);
        rot_mat = eye(4);
        rot_mat(1, 1) = cos(theta);
        rot_mat(2, 2) = cos(theta);
        rot_mat(1, 2) = -sin(theta);
        rot_mat(2, 1) = sin(theta);
        cmd_vel = cmd_vel*rot_mat;

        set_cmd_vel(app.cmd_vel_pub, cmd_vel)

        app.diff_prev = diff;
        app.diff_sum = app.diff_sum+diff;

        % pozycja docelowa osiągnięta
        if sum(diff(1:3).^2) <= 0.1 % ~31,6cm
            app.reached = true;
            app.diff_prev = [0 0 0 0];
            app.diff_sum = [0 0 0 0];
            set_cmd_vel([0 0 0 0])
        end
    end
end
```

Stan lotu

```
function flying_state_callback(app, ~, message)
    flying_state = message.State;

    switch flying_state
        case 0 % landed
            flying_state = 'na ziemi';
        case 1 % taking off
            flying_state = 'startowanie';
        case 2 % hovering
            flying_state = 'unoszenie';
        case 3 % flying
            flying_state = 'lot';
        case 4 %landing
            flying_state = 'lądowanie';
        case 5 % emergency
            flying_state = 'awaria';
        case 6 % user take off
            flying_state = 'startowanie użytkownika';
        case 7 % motor ramping
            flying_state = 'rozruch silników';
        case 8 % emergency landing
            flying_state = 'lądowanie awaryjne';
    end

    app.flying_state_string = strcat('Stan lotu drona: ', flying_state);

    % zaktualizuj pole tekstowe
    app.StanDronaTextArea.Value = {app.battery_state_string; app.wifi_state_string; app.flying_state_string};
end
```

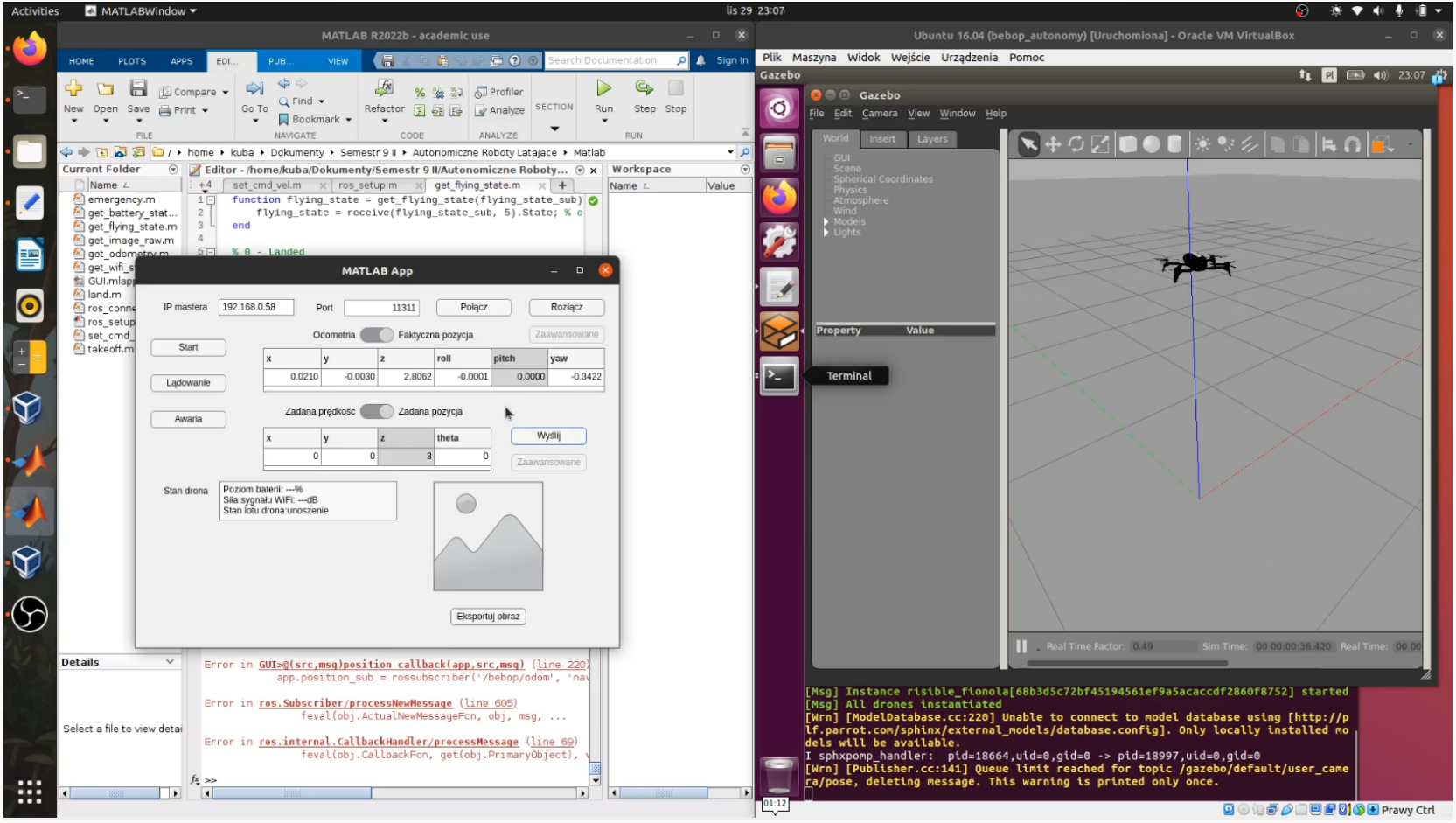
Stan c.d.

```
function image_callback(app, ~, message)
    img = readImage(message);
    app.Image.ImageSource = img;
end

function battery_state_callback(app, ~, message)
    battery_state = message.Percent;

    app.battery_state_string = strcat('Poziom baterii: ', int2str(battery_state), '%');

    % zaktualizuj pole tekstowe
    app.StanDronaTextArea.Value = {app.battery_state_string; app.wifi_state_string; app.flying_state_string};
end

function wifi_state_callback(app, ~, message)
    wifi_state = message.Rssi;

    app.wifi_state_string = strcat('Siła sygnału Wi-Fi: ', int2str(wifi_state), 'dB');

    % zaktualizuj pole tekstowe
    app.StanDronaTextArea.Value = {app.battery_state_string; app.wifi_state_string; app.flying_state_string};
end
```

Parametry w pliku launch

```
<?xml version="1.0"?>
<launch>
    <arg name="namespace" default="bebop" />
    <arg name="ip" default="10.202.0.1" />
    <arg name="drone_type" default="bebop2" /> <!-- available drone types: bebop1, bebop2 -->
    <arg name="config_file" default="$(find bebop_driver)/config/defaults.yaml" />
    <arg name="camera_info_url" default="package://bebop_driver/data/$(arg
drone_type)_camera_calib.yaml" />
    <group ns="$(arg namespace)">
        <node pkg="bebop_driver" name="bebop_driver" type="bebop_driver_node" output="screen">
            <param name="camera_info_url" value="$(arg camera_info_url)" />
            <param name="bebop_ip" value="$(arg ip)" />
            <param name="~states/enable_commonstate_batterystatechanged" value="true" />
            <param name="~states/enable_commonstate_wifisignalchanged" value="true" />
            <param name="~states/enable_pilotingstate_flyingstatechanged" value="true" />
            <rosparam command="load" file="$(arg config_file)" />
        </node>
        <include file="$(find bebop_description)/launch/description.launch" />
    </group>
</launch>
```

Działanie


Activities
MATLABWindow
lis 29 23:07
MATLAB R2022b - academic use
MATLAB App
IP mastera
192.168.0.58
Port
11311
Połącz
Rozłącz
Odometria
Faktyczna pozycja
Zaawansowane
Start
Lądowanie
Awaria
Zadana prędkość
Zadana pozycja
Wyślij
Stan drona
Poziom baterii: ---%
Siła sygnału WiFi: ---dB
Stan lotu drona:unoszenie
Eksportuj obraz
Ubuntu 16.04 (bebop_autonomy) [Uruchomiona] - Oracle VM VirtualBox
Gazebo
Terminal
[Msg] Instance risible_fionola[68b3d5c72bf45194561ef9a5acaccdf2860f8752] started
[Msg] All drones instantiated
Prawy Ctrl

Dziękuję za uwagę