

Komunikacja i sterowanie dronem za pośrednictwem MATLABa

Jakub Szczygiał



Subscriber odometrii

```
app.odometry_sub = rossubscriber('/bebop/odom', 'nav_msgs/Odometry', @(src, msg) odometry_callback(app, src, msg));
```

```
function odometry_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;
    rot = quat2eul([w x y z]);

    odometry = [pos, rot];
    app.OdomUITable.Data = odometry;
end
```

Pozycja z Gazebo

```
<!-- plugin do publikowania pozycji drona -->
<gazebo>
  <plugin name="p3d_base_controller" filename="libgazebo_ros_p3d.so">
    <alwaysOn>true</alwaysOn>
    <updateRate>50.0</updateRate>
    <bodyName>base_link</bodyName>
    <topicName>ground_truth</topicName>
    <gaussianNoise>0.01</gaussianNoise>
    <frameName>world</frameName>
    <xyzOffsets>0 0 0</xyzOffsets>
    <rpyOffsets>0 0 0</rpyOffsets>
  </plugin>
</gazebo>
```

GUI

MATLAB App

IP mastera Port


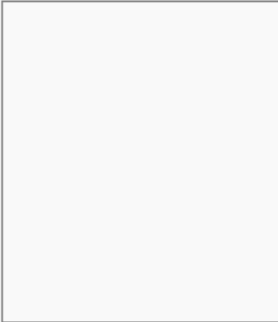
Odometria ☒ Faktyczna pozycja

x	y	z	roll	pitch	yaw
0	0	0	0	0	0

Zadana prędkość ☒ Zadana pozycja

vx	vy	vz	omega
0	0	0	0

Stan drona



Dziękuję za uwagę

