
Robotik Projekt

- Aufgabenblatt 3 -

Laserscanner

Schreiben Sie eine Node in C++ oder Python, sodass der Roboter einem Hindernis per Laserscanner folgt. Wenn das Hindernis zu nah oder zu weit entfernt vom Roboter ist, soll dieser anhalten. Die Werte für die Entfernung zum Hindernis, die Geschwindigkeit des Roboters wie auch den Öffnungswinkel, in dem Hindernisse gesucht werden, sollen konfigurierbar per Parameter [1] sein. Werden mehrere Hindernisse erkannt, soll das Objekt mit der geringsten Distanz verfolgt werden. Die Regelung darf so einfach wie möglich gehalten werden; vermutlich bietet es sich an, Dreh- und Fahrbewegung voneinander zu trennen.

Da die Daten des Laserscanners per WLAN übertragen werden, wird es vermutlich notwendig sein, die QoS ReliabilityPolicy auf BEST_EFFORT zu stellen. Dies ist wie folgt möglich:

Für Python:

```
qos_policy =
    rclpy.qos.QoSProfile(reliability=rclpy.qos.ReliabilityPolicy.BEST_EFFORT,
                        history=rclpy.qos.HistoryPolicy.KEEP_LAST,
                        depth=1)
self.subscription = self.create_subscription(
    LaserScan,
    'scan',
    self.listener_callback,
    qos_profile=qos_policy)
```

Für CPP:

```
rmw_qos_profile_t qos_profile = rmw_qos_profile_sensor_data;
auto qos = rclcpp::QoS(rclcpp::QoSInitialization(qos_profile.history, 1), qos_profile);

subscription_ = this->create_subscription<sensor_msgs::msg::LaserScan>(
    "scan", qos, std::bind(&MinimalSubscriber::topic_callback, this, _1));
```

Eine Inspiration zur Nutzung des Laserscanners können Sie hier finden:

https://github.com/gerolicht/robotik_projekt/

[1] <https://docs.ros.org/en/humble/Concepts/Basic/About-Parameters.html>