

# ANRO

Michał Kwarcirski

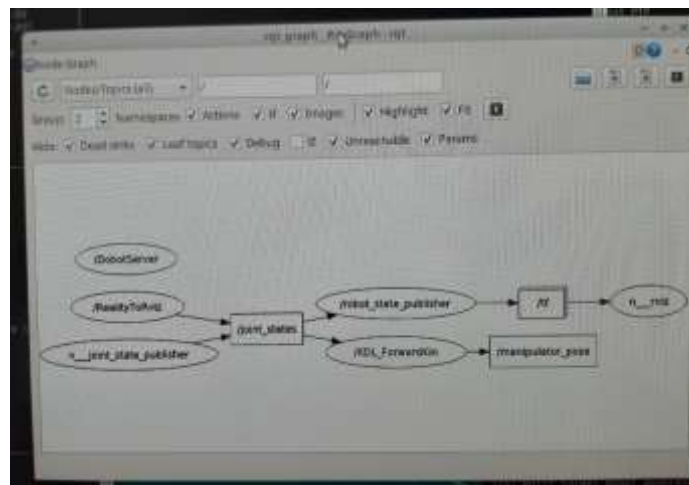
Kacper Marchlewicz

## Laboratorium 3

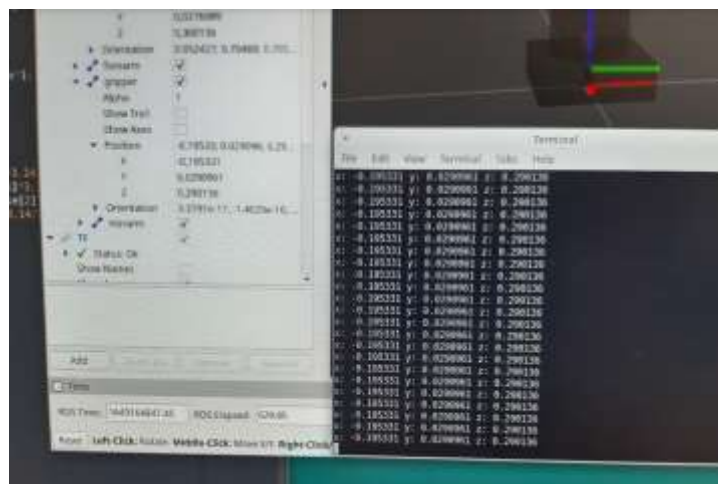
Laboratorium rozpoczęliśmy od modyfikacji modelu rdf robota. Dodaliśmy mu końcówkę która w miarę dobrze oddawała realny wygląd. Analogicznej zmiany dokonaliśmy w pliku kdl.

Stworzyliśmy zgodnie z instrukcją węzeł regularnie odpytujący serwis GetPose rzeczywistego robota. Nasz węzeł następnie publikuje wartości zmiennych złączowych na temacie 'joint\_states' z którego korzysta model RVIZ.

Schemat systemu:



Poprawność działania kdl:



Film prezentujący działanie robota i programu:

[https://1drv.ms/v/s!AkMYzt8ywVnohID7FZoXRhsoD0\\_v](https://1drv.ms/v/s!AkMYzt8ywVnohID7FZoXRhsoD0_v)

Kod programu:

```

#include "ros/ros.h"
#include "std_msgs/String.h"
#include <string>
#include "dobot/SetCmdTimeout.h"
#include "dobot/SetQueuedCmdClear.h"
#include "dobot/SetQueuedCmdStartExec.h"
#include "dobot/SetQueuedCmdForceStopExec.h"
#include "dobot/GetDeviceVersion.h"
#include "dobot/SetEndEffectorParams.h"
#include "dobot/SetPTPJointParams.h"
#include "dobot/SetPTPCoordinateParams.h"
#include "dobot/SetPTPJumpParams.h"
#include "dobot/SetPTPCommonParams.h"
#include "dobot/GetPose.h"
#include "dobot/SetPTPCmd.h"
#include "dobot/SetEndEffectorGripper.h"
#include "dobot/SetWAITCmd.h"
#include <iostream>
#include "sensor_msgs/JointState.h"

int main( int argc, char** argv)
{
    ros::init(argc, argv, "RealityToRviz");
    ros::NodeHandle n;

    ros::ServiceClient poseClient;

    ros::Publisher publisher = n.advertise<sensor_msgs::JointState>("/joint_states", 1000);
    sensor_msgs::JointState msg;
    msg.name.resize(5);
    msg.position.resize(5);

    poseClient = n.serviceClient<dobot::GetPose>("/DobotServer/GetPose");
    dobot::GetPose poseSrv;

    while(ros::ok())
    {
        if (poseClient.call(poseSrv))
        {
            msg.position[0] = (double)poseSrv.response.jointAngle[0]*3.14/180.0;
            msg.position[1] = 0-(double)poseSrv.response.jointAngle[1]*3.14/180.0;
            msg.position[2] = 1.57-(double)poseSrv.response.jointAngle[2]*3.14/180.0;
            double zmienna = (double)poseSrv.response.jointAngle[1]*3.14/180.0 +
(double)poseSrv.response.jointAngle[2]*3.14/180.0;
            msg.position[3] = zmienna - 1.57;
            msg.position[4] = 0;

            msg.header.stamp.sec = ros::Time::now().toSec();
            msg.name[0] = "joint_1";
            msg.name[1] = "joint_2";
            msg.name[2] = "joint_3";
            msg.name[3] = "joint_4";
            msg.name[4] = "joint_5";
            publisher.publish(msg);
            ros::spinOnce();
        }
    }
    return 0;
}

```

## Pytania Kontrolne

*Jakie są podstawowe rodzaje złącz występujących w manipulatorach?*

Obrotowy, przesuwany, sferyczny, cylindryczny, śrubowy

*Czy wartości zmiennych złączowych manipulatora Dobot Magician jednoznacznie determinują pozycję końcówki?*

Tak