

Medical Robotics

Laboratory 1 - Introduction to Medical Robotics and Workplace

PhD Eng. Sabina Szymoniak



Faculty of Computer Science
and Artificial Intelligence

Lab Objective

- Understanding basic concepts in medical robotics
- Learning the classification of robotic systems in medicine
- Introducing GAZEBO + ROS 2 - an academic standard
- Preparing the environment for medical robot simulation
- Initial launch of the hospital environment

History and Development of Medical Robotics

- **1980s:** ROBODOC (orthopedic surgery)
- **1999:** FDA Approves the da Vinci Surgical System
- **2000-2010:** Laparoscopy, Urology, Cardiac Surgery
- **2010-2020:** Integration with 3D Imaging, AI
- **2020-2026:** Telesurgery, AMR Robots in Hospitals

2026: GAZEBO Harmonic Realistic Simulation

Applications of Robotics in Medicine

Precision Surgery

- Laparoscopy (da Vinci)
- Neurosurgery
- Ophthalmic Surgery
- Heart Surgery

Rehabilitation

- Exoskeletons (ReWalk)
- Therapeutic Robots
- Gait Analysis (OpenSim)
- Intelligent Prosthetics

Other: room disinfection, sample transport, Tele-ultrasound

Classification of Medical Robots

By Function:

1. **Assisting** (da Vinci) - support the surgeon
2. **Autonomous** (ROBODOC) - operate independently
3. **Teleoperative** - remote control
4. **Mobile AMR** - transport within hospitals

By Kinematics (GAZEBO simulation):

1. Series (UR5, KUKA LBR)
2. Parallel (Stewart Platform)
3. Anthropomorphic Exoskeletons
4. Wheeled/Mobile Base

Requirements for Medical Robots

- **Precision:** <1mm repeatability
- **Realistic physics:** tissue simulation (GAZEBO)
- **Safety:** IEC 60601-1, ISO 13482
- **Sterility:** replaceable tool tips
- **Integration:** ROS 2 + medical imaging
- **Real-time:** <10ms control loop

Work Environment - GAZEBO + ROS 2

Lab Technology Stack (2026):

- **ROS 2 Humble** - Communication (topics/services)
- **Gazebo Harmonic** - ODE/DART physics, 3D models
- **RViz2** - Sensor data visualization
- **ros_gz_bridge** - ROS ↔ Gazebo
- **Gazebo Fuel** - 10k+ models (including medical models)

Advantages: Native integration, 100k+ dev community [web:14]

GAZEBO - why is it the best simulator?

Features for medical robotics:

- **Realistic physics:** tissues, fluids, collisions
- **Medical models:** daVinci, hospital_env
- **Native ROS 2:** ros_gz_sim/ros_gz_bridge
- **Gazebo Fuel:** <https://fuel.gazebosim.org/>
- **Plugins:** endoscopic cameras, force/torque

Official docs: <https://gazebosim.org/docs>

Practice: Installation of GAZEBO + ROS 2

Ubuntu 22.04 LTS (recommended):

```
# ROS 2 Humble + Gazebo Harmonic
sudo apt update && sudo apt install
locales curl gnupg2 lsb-release
sudo apt install ros-humble-desktop-full
sudo apt install ros-humble-gz-harmonic ros-humble-ros-gz-*
```

Source environment:

```
echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Practice: Installation

Installation verification:

```
# ROS 2 test  
ros2 topic list
```

```
# Gazebo test  
gz sim -v 4 shapes.sdf
```

```
# Integration test  
ros2 launch ros_gz_sim demo.launch.py
```

Save the output as a screenshot for your report!

Docker alternative (easier): <https://hub.docker.com/r/osrf/ros2>

Practice: First Medical Simulation

Launch the hospital environment:

```
# Demo with a medical robot  
ros2 launch ros_gz_sim hospital.launch.py
```

```
# In the second terminal - RViz  
rviz2
```

In RViz2, add displays:

- RobotModel (/robot_description)
- LaserScan (/scan)
- Camera (/camera/image_raw)
- JointState (/joint_states)

Practice: First ROS 2 Node (Publisher) I

File: publisher_med.py

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import Twist

class MedicalRobotPublisher(Node):
    def __init__(self):
        super().__init__('med_robot_publisher')
        self.publisher_ = self.create_publisher(
            Twist, 'cmd_vel', 10)
        timer_period = 0.5
        self.timer = self.create_timer(timer_period, self.timer_callback)

    def timer_callback(self):
```

Practice: First ROS 2 Node (Publisher) II

```
msg = Twist()
msg.linear.x = 0.2 # Speed of the medical robot
msg.angular.z = 0.1
self.publisher_.publish(msg)
self.get_logger().info('Publishing medical robot cmd')

def main():
    rclpy.init()
    node = MedicalRobotPublisher()
    rclpy.spin(node)
```

Practice: Gazebo Fuel - medical models

Download models to the lab:

- **daVinci Surgical:**
<https://fuel.gazebosim.org/1.0/OpenRobotics/models/daVinci>
- **Hospital Environment:** search for "hospital"
- **Surgical Table + Patient:** "patient model"

Import into scene:

```
gz sim -s hospital_env.sdf patient_model.sdf
```

Tasks (1/2)

Task 1: Installation and Verification [30 min]

- Install ROS 2 + Gazebo Harmonic
- Run `ros2 topic list` → screenshot
- Run `gz sim shapes.sdf` → screenshot

Task 2: First Medical Simulation [20 min]

- Run `demo.launch.py`
- Open RViz2, add 4 displays
- Take 3 screenshots (Gazebo + RViz)

Tasks (2/2)

Task 3: Medical Publisher [25 min]

- Copy and run `publisher_med.py`
- Observe traffic in Gazebo
- Modify speeds → test 3 variants

Task 4: Exploring Gazebo Fuel [15 min]

- Find 3 medical models (links + screenshots)
- Download a daVinci model or similar
- Run in your scene

CHALLENGE Task 5: Subscriber Node [bonus]

- Read the `/joint_states` of the medical robot
- Display joint angles in the terminal

Lab 1 Report – Requirements

- **5 screenshots** from the tasks (name them descriptively)
- **Python Code** with Modifications (GitHub)
- **3 links** to Gazebo Fuel models
- **Somewhat 1 page** of theory (your notes)
- **Working time** and encountered problems

Good luck :)

Resources and Documentation

Official Tutorials:

- GAZEBO: <https://gazebo.org/docs/harmonic/>
- ROS 2: <https://docs.ros.org/en/humble/>
- ros_gz: https://gazebo.org/docs/all/ros_integration
- Gazebo Fuel: <https://fuel.gazebo.org/>

GitHub: <https://github.com/gazebo>

Thank you! Questions?

Observations and suggestions

sabina.szymoniak@icis.pcz.pl



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Project title: "All roads lead to knowledge - DOPPIO Sapienza PCz"

Project number: BPI/KAT/2024/1/00025/DEC/1