OpenCV & BIM Integration in ROS2 For Panel Positioning Assisting

By

Yasmin Ragab

Radwa Abdelhafez

Noureldeen Nagm

CR _ Prototyping Project 2022 WINTER

construction robotics | international M.Sc.programme

Content

- Introduction
- Solution Proposal
- Prototyping Target
- Development Steps
  - ROS2
  - Robot URDF Model
  - Aruco Pose Estimation
  - Pose from BIM
  - RVIZ
  - Alignment Calculation
  - BASH File
- Prototyping Result
- Outlook

Introduction

With the rise of competition and the continuous emergence of design trends, facade complexity is increasing by the day and complicating the construction process accordingly. One of the main processes of façade construction is the installment of exterior wall panels such as glass panels which requires the positioning of the panel in place. One of the emerging procedures is the use of mini cranes and glass cranes for the placement of panels from inside the buildings.

Fig.01
Panel installation using tower crane

Fig.02
Personnel assisting mini crane

Despite the fact that the use of mini cranes is an improvement when compared to the use of tower cranes, it is still a complex procedure that requires physical assistance and guidance from a team of personnel in order to guide each panel in place.

Problem Statement:

Manually orienting the glass panels in place can be a tedious operation that consumes time and exposes both fixer's personnel and the surrounding environment to hazardous situations.

Solution Proposal

Goal:

Increase the efficiency of panel positioning and reduce the risk of damage to the load, personnel, and surroundings.

Concept:

A robust self-assisted protocol that is able to determine the current orientation state of the lifted panel and the needed alignment to achieve the target state.

Idea:

Computer vision and BIM integration for panel alignment assistance.

CV ⟶ Current state

BIM ⟶ Target state

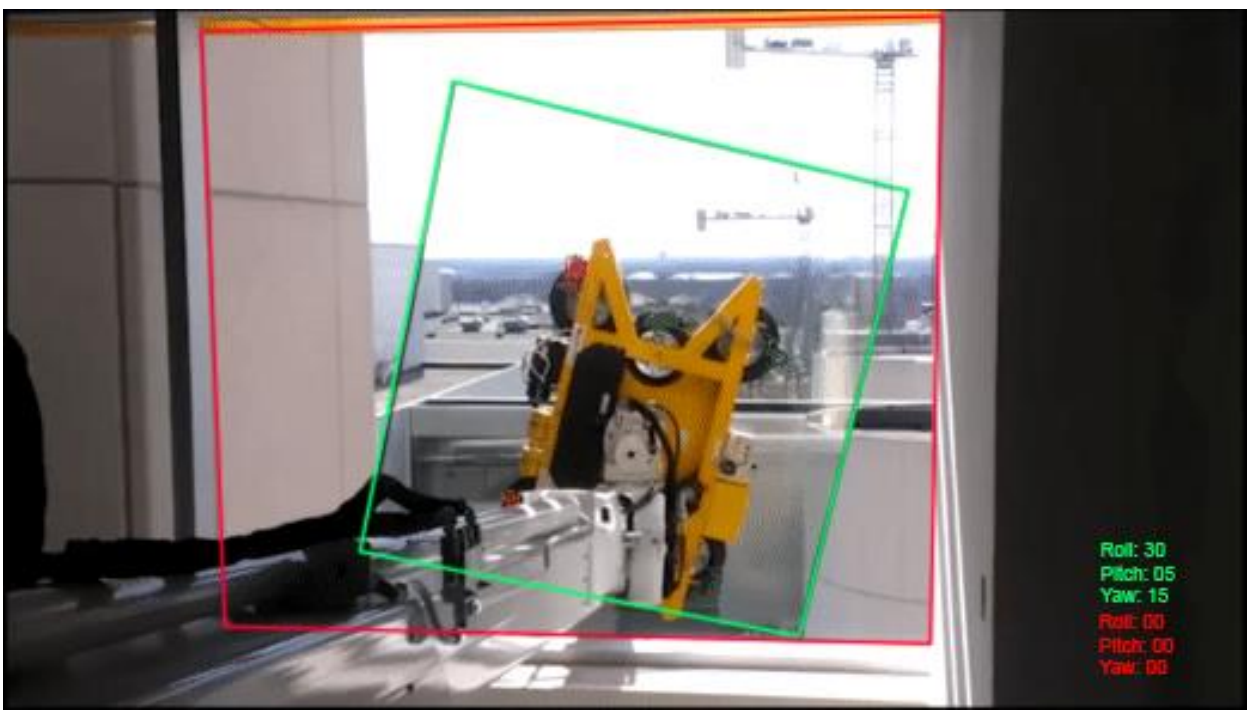


Fig.03
Proposal concept illustration

Prototyping Target

- Building a joy-stick controlled crane urdf-model and visualize it in RVIZ.
- Using OpenCV and Aruco for real-time pose detection in ROS2.
- Real-time twin posing of Aruco marker by the robot manipulating head.
- Using Dynamo to export as-designed panel pose from BIM model.
- Comparing the two poses with respect to world in Ros2 to determine the required alignment by the robot.

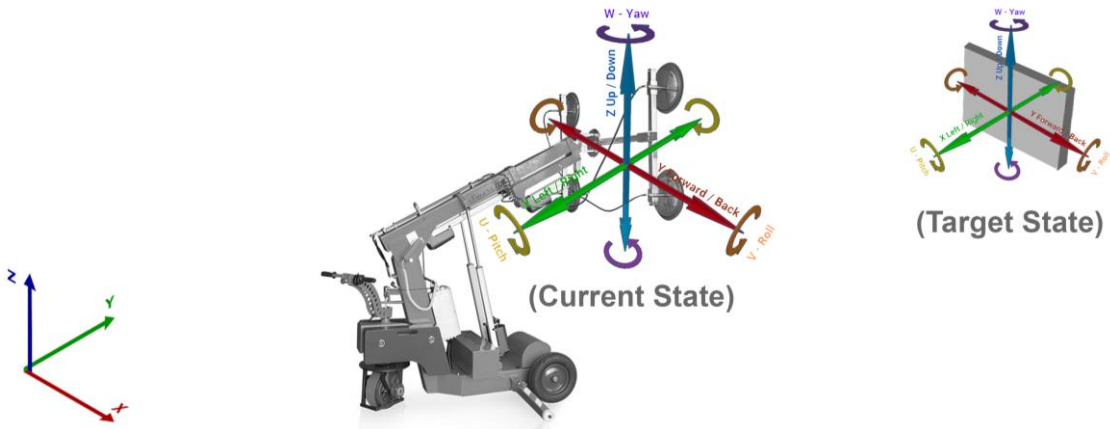


Fig.04
Prototyping target Illustration

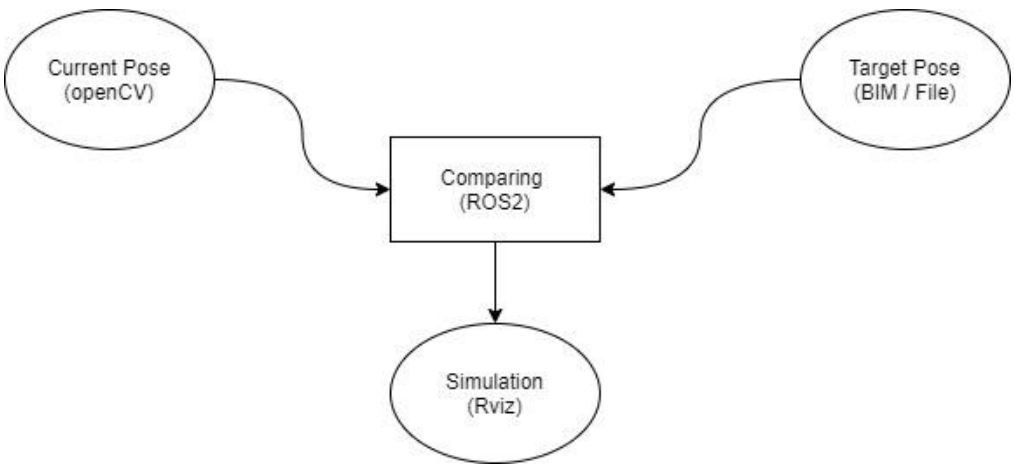


Fig.05
Prototyping target diagram

Development Steps – ROS2

ROS2 Nodes	Subscribed Topics	Published Topics
/usb_cam		/camera_info - /image_raw
/aruco_node	/camera_info - /image_raw	/aruco_poses
/joy_node		/joy
/robot	/joy – /aruco_poses	/head_transformation - /joint_states
/panel		/markers
/orientation	/head_transformation	

Table.01
Ros2 nodes and topics communication

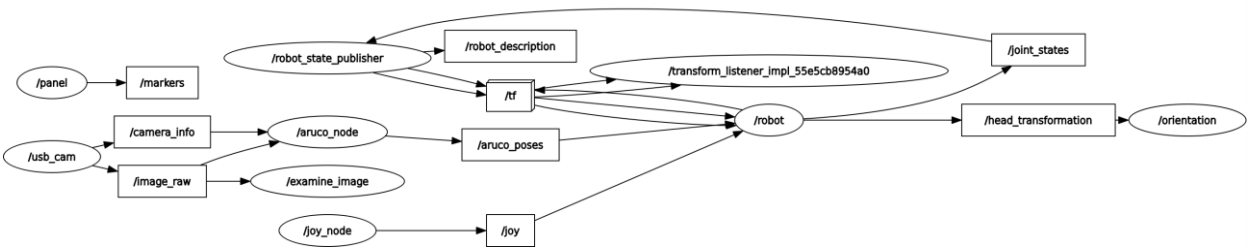


Fig.06
ROS2 GUI rqt_graph diagram

Development Steps – Robot URDF Model

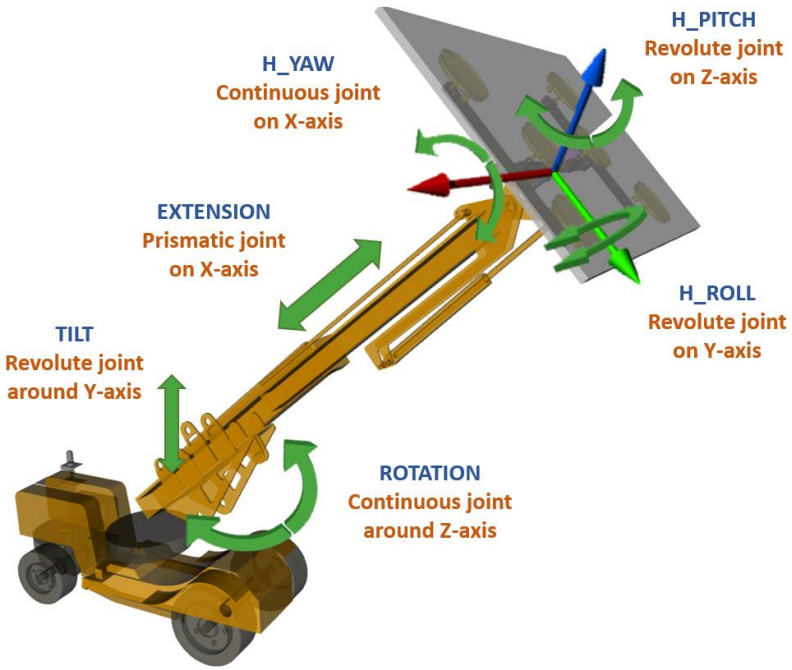


Fig.07
Crane robot URDF model kinematics

Joint Name	Parent Link	Child Link
ROTATION	BASE	H_MOTOR
TILT	H_MOTOR	ARM
EXTENSION	ARM	TRANSLATION
H_YAW	TRANSLATION	V_MOTOR
H_PITCH	V_MOTOR	R_MOTOR
H_ROLL	R_MOTOR	HEAD

Table.02
URDF model joints and links

Development Steps – Aruco Pose Estimation

Components:

Aruco marker ⟶ Represent the lifted panel in real-time.

OpenCV ⟶ Aruco marker pose estimation from usb-cam

CV bridge ⟶ OpenCV output to ROS2 message (Fig.08).

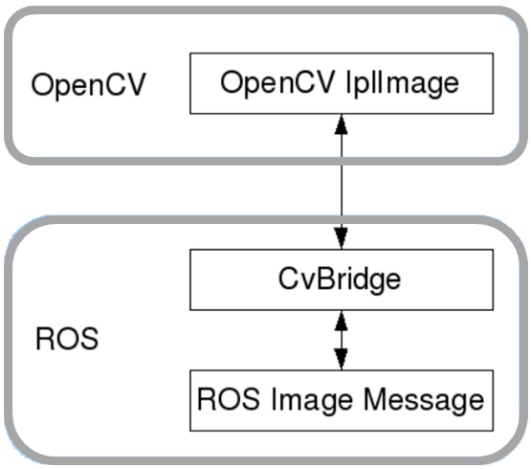


Fig.08
OpenCv to ROS

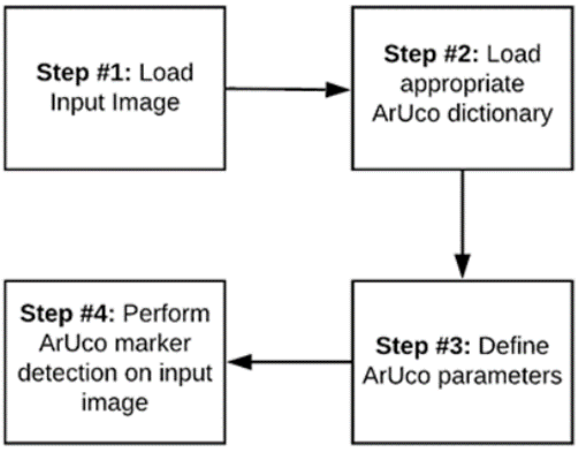


Fig.09
Aruco detection

Output Parameters:

Tvecs ⟶ Aruco marker translation vectors.

Rvecs ⟶ Aruco marker rotation vectors.

Output Conversion:

tf package was used to convert the rotation matrix into ROS2 compatible quaternions.

Tvecs ⟶ Pose position.

Rvecs ⟶ Rotation matrix ⟶ Pose orientation.

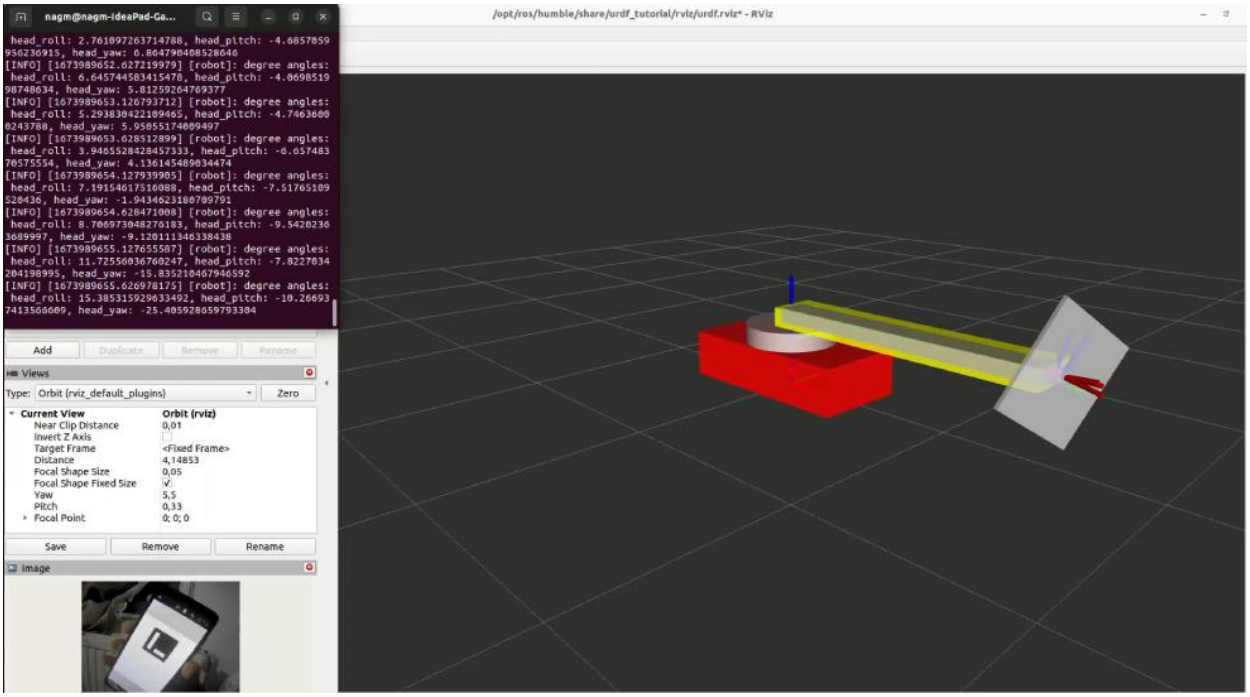


Fig.10
Aruco marker pose detection in ROS2

Development Steps – Pose from BIM

As-designed Panel orientation (target state) is extracted from a BIM model using a Python Dynamo script (Fig.11).

Panel rvecs (BIM) ⟶ txt file ⟶ Quaternion (ROS2)

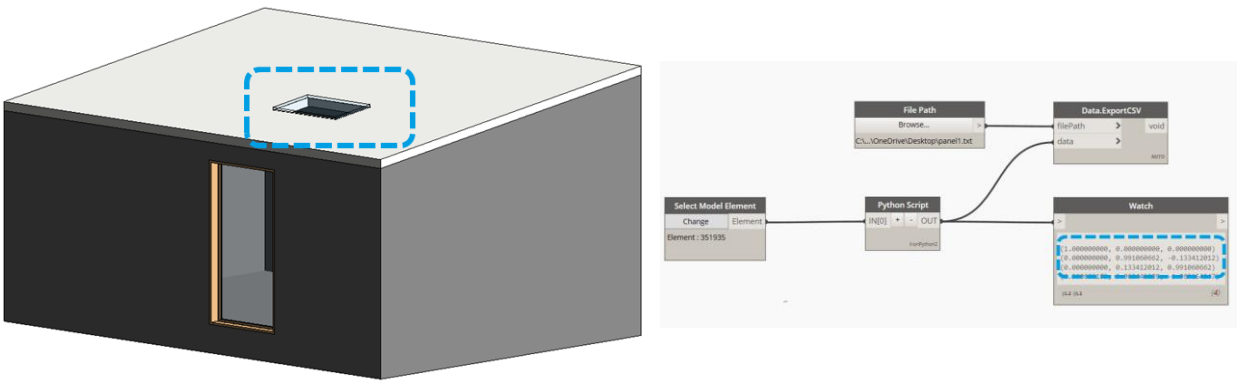


Fig.11
Exporting Panel rotation data from BIM model

Development Steps - RVIZ

URDF:

Using RVIZ for ROS2 to load the XML URDF file and visualize robot kinematics.

Markers:

Panel marker ⟶ Target state pose in BIM model.

View marker ⟶ BIM environment (surroundings)

Text marker ⟶ Current state pose of the manipulating head.

Configuration:

Using a RVIZ configuration file enabled instant launching of RVIZ in our previously set default configurations.

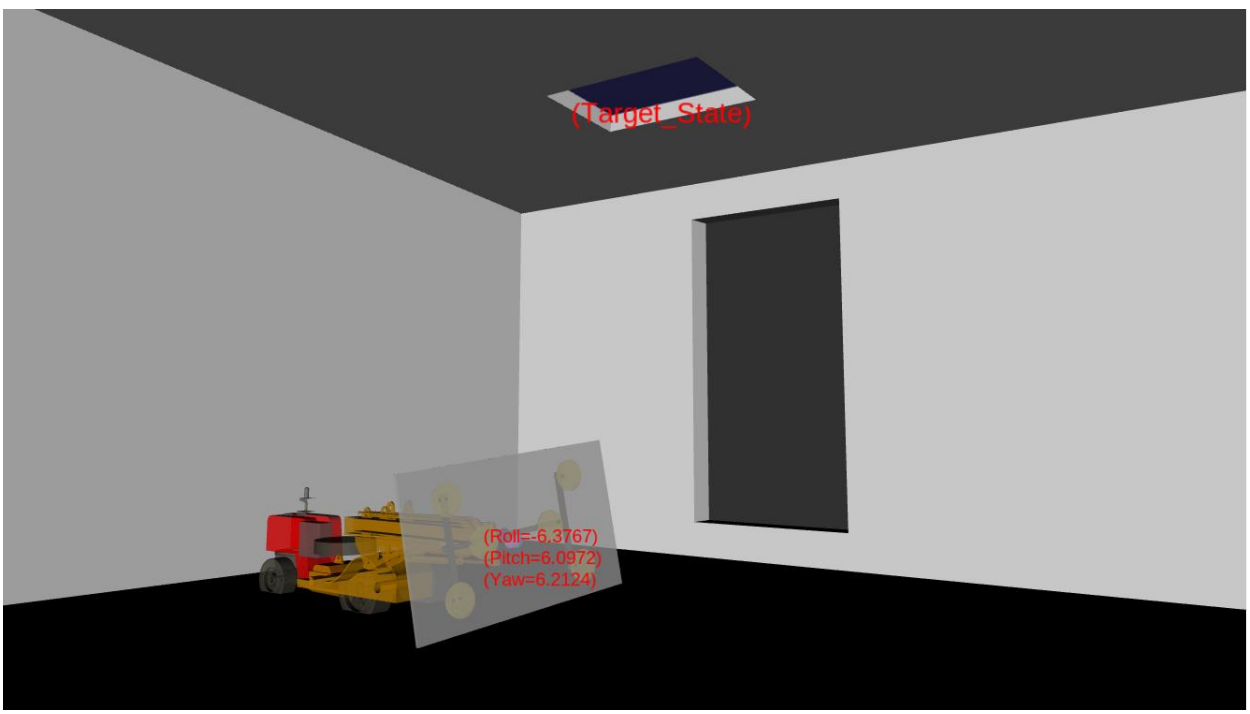


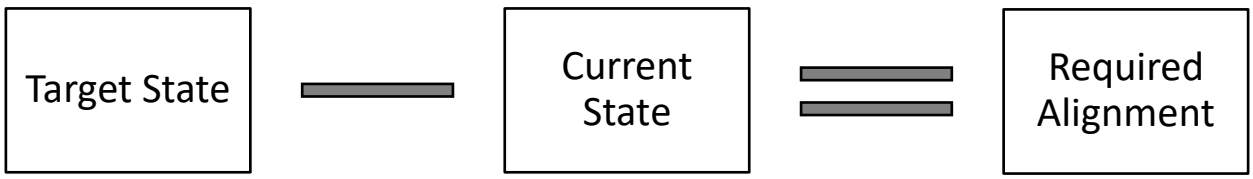
Fig.12
RVIZ VIEWPORT

Alignment Calculation

After importing the target state pose from the BIM exported text file, and calculating the current state pose of the lifted panel (manipulating head) we can compare these values to acquire the required alignment by the crane robot.

BIM txt(rvecs) ⟶ Quaternion(target state) ⟶ Euler(roll,pitch,yaw)

Quaternion(current state) ⟶ Euler(roll,pitch,yaw)



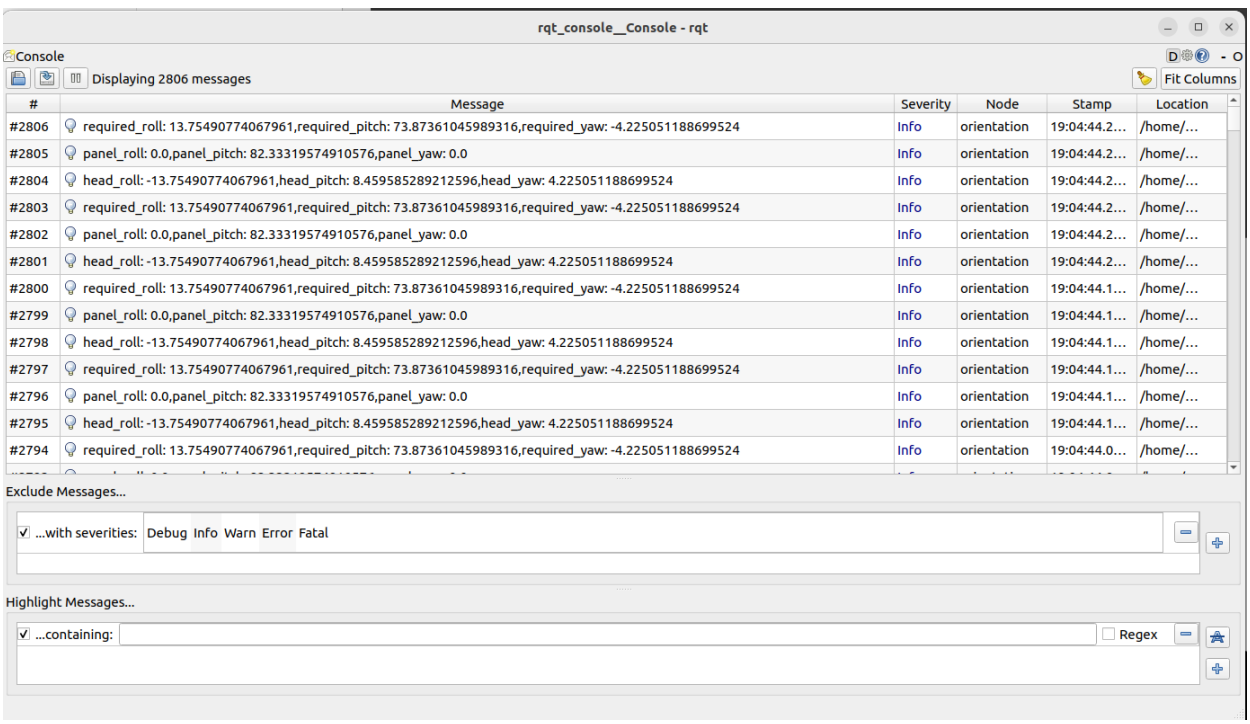
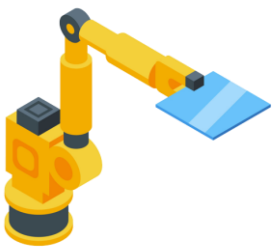


Fig.13
ROS2 rqt_console

Development Steps – BASH File

In order to launch and close the program, two bash files were created, the first bash file runs the robot launch file and the usb-camera node at the same time, while the other bash file closes the program and kills all the nodes.

Afterward, two desktop icons were created to run these bash files directly from the desktop (Fig.14).



Launch desktop icon

Close desktop icon

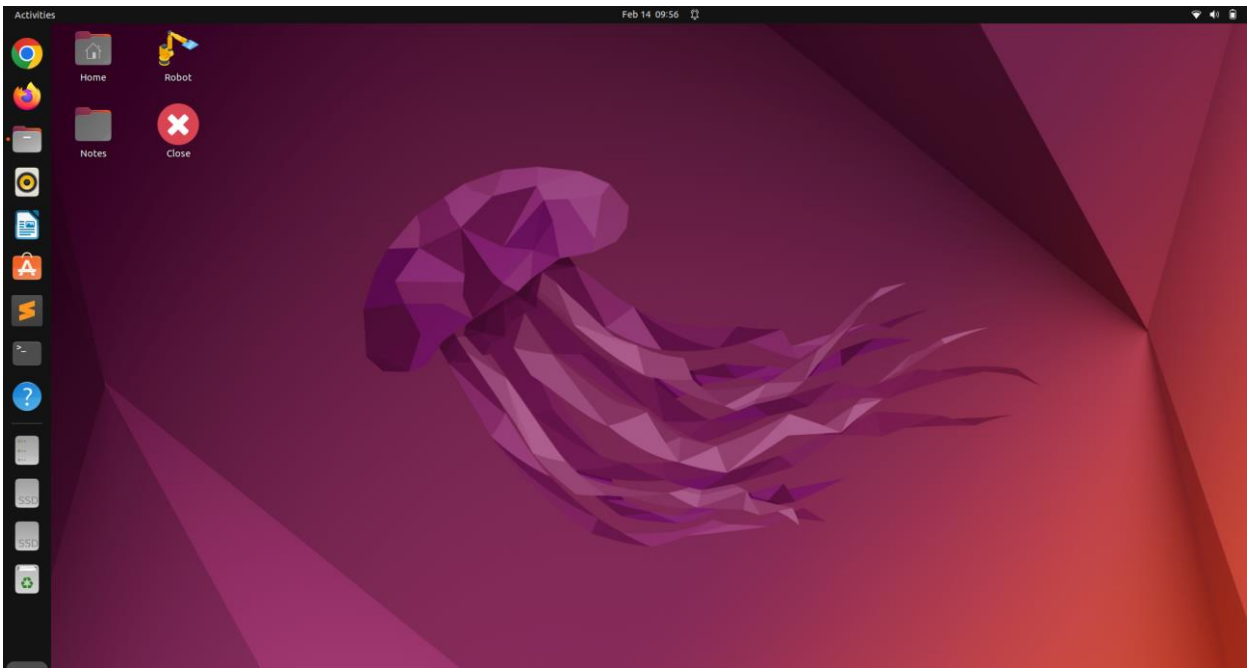


Fig.14
Launch desktop icons

Prototyping Result

Fig.15
Complete prototype

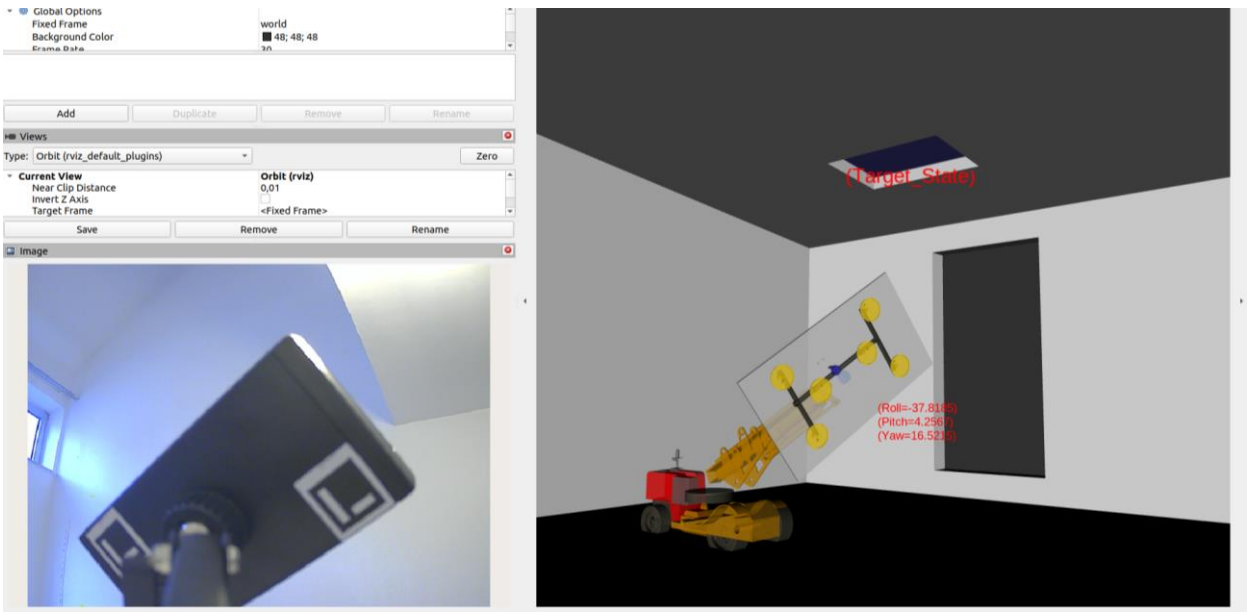


Fig.16
Prototype result RVIZ viewport

Outlook

Conclusion:

Successfully detecting the manipulating head pose and assisting the crane operator with the needed alignment for reaching the target state is in fact an improvement compared to the current adopted protocol, with that being said this prototype is yet to undertake some improvements to be ready for adoption by the construction industry.

Future Development:

OpenBIM ⟶ Programming a code that extracts the orientation of multiple panels from an IFC file instead of Dynamo from Revit.

Computer Vision ⟶ Integrating machine learning and using a depth camera for a more sophisticated 3D pose estimation instead of relying on the Aruco library.

User Interface ⟶ Building a user-friendly interface for the crane operator, using a framework such as Dash for Plotly.js a live representation of the (current state – target state – required alignment) poses in the shape of gauges charts can be achieved.

Alignment Automation ⟶ Using the calculated transformation between the current state of the manipulating head and the target state acquired from the BIM model, an attempt can be made towards automating the manipulating head pose alignment.

Digital Twin ⟶ Using sensor integration a complete virtual model for the mini crane can be created which can be used for complete robot automation.