Tunisian Republic

Ministry of Higher Education and Scientific Research University of Carthage

National Institute of Applied Sciences and Technology

Personal and Professional Project

Subject

Visual servoing of a manipulator robot

Specialty: Industrial Computing and Automation

Prepared by

Wissem MALLEH

Amine ROMDHANE

Dorra SAADALLAH

Defended on 01/06/2023, in front of the jury:

Mrs. Fatma Zahra OUESLATI	President of the jury
Mrs. Amani AYEB	Supervisor

Academic year: 2022 / 2023

Acknowledgements

We would like to express our deepest gratitude and appreciation to all those who have contributed to the successful completion of our professional and personal project.

First and foremost, we extend our sincere thanks to our supervisor**, Mrs. Ama**ni **AYEB**. Her invaluable expertise, constant motivation has played a pivotal role in shaping our project. Her insightful feedback, constructive criticism, and patient guidance have been instrumental in refining our ideas and enhancing the quality of our work.

We extend our sincere gratitude to our esteemed jury member, **Mrs. Fatma Zahra OUESLATI**, for graciously accepting the task of reviewing our project. As a distinguished professor at the National Institute of Applied Sciences and Technology (INSAT), her expertise will enrich our evaluation process.

Furthermore, we would like to express our gratitude to the **clubs of INSAT** for their immense support and collaboration. Their enthusiastic participation and active involvement have enriched our project by providing valuable resources, diverse perspectives, and meaningful contributions. The synergy created through our collaboration with the clubs has not only enhanced our project's outcomes but also fostered a vibrant and engaging environment for personal and professional growth.

Table of Contents

Acknowledgment I

List of Figures IV

General Introduction 1

1 Project Context 2

1.1 Solution Overview 3

1.2 Existing Studies 3

1.3 Comparative Study 4

1.4 Problem Statement 4

1.5 Adopted Solution 5

2 Robotic arm design and modelisation 6

2.1 Components of the Robotic Arm 7

2.2 Design 8

3 Control Method and Simulation 9

3.1 Control Method 10

3.2 Simulation with Gazebo and RViz 10

3.3 Incorporating the Camera in the URDF.. 12

3.4 Utilizing the Camera Topic.. 12

General conclusion and perspectives 14

Bibliography 15

List of Figures

2.1 Robot Arm Assembly 7

2.2 Robot Arm Joints 7

2.3 Cinematic Modelisation 7

2.4 Eye-to-Hand 8

3.1 MoveIt-poses-defining 9

3.2 Gazebo-Arm-simulation 10

3.3 RVIZ-Arm-simulation 11

3.4 Gazebo-image-view 13

3.5 OpenCV-shape-detection 13

General Introduction

The automation of tasks in the industry has undergone a tremendous evolution in recent decades. One key area of this transformation is the field of pick and place, where robotic arms are used to grasp and move objects with increased precision and efficiency.

In fact, Pick and place is a recurring task in many industries, including assembly, logistics, and manufacturing. Traditionally, robotic arms used for these operations are statically programmed, limiting their adaptability to variable situations. Moreover, traditional control methods based solely on position and motion information are not always sufficiently accurate and fast enough to meet the growing demands of the industry.

Therefore, it becomes crucial to integrate computer vision systems to enable the robotic arm to perceive its environment and adjust its movements accordingly. This approach, known as visual servoing, allows the robotic arm to guide itself using real-time visual information to ensure more precise, faster, and flexible pick and place operations.

By using image processing algorithms, the robotic arm will be able to track moving objects, adjust its trajectory in real-time, and respond quickly to changes in the environment. This approach will significantly enhance the efficiency and flexibility of the robotic arm while reducing potential errors and damages.

To design, simulate, or control a robot, it is necessary, among other things, to have models of the mechanism. Several levels of modeling are possible, depending on the specifications of the intended application: this leads to geometric, kinematic, and dynamic models from which the robot's movements can be generated, or static models that describe the mechanism's interactions with its environment. Obtaining these different models is not easy, and the difficulty varies depending on the complexity of the kinematics of the articulated chain.

1

Project context

Introduction

The field of robotics has witnessed significant advancements in recent years, enabling the development of intelligent systems capable of performing complex tasks. Visual servoing, a fundamental technique in robotics, focuses on utilizing visual feedback to control the motion and positioning of a manipulator robot. Simulations play a vital role in the initial design, development, and testing phases of such projects, allowing for cost-effective and efficient exploration of various scenarios and control strategies.

1.1. Solution Overview:

The proposed solution is a robotic arm simulation using Gazebo and RViz, integrated with OpenCV for shape detection. The main objective is to perform precise positioning of objects based on their shape without the need for exact piece placement. Unlike existing industrial arms that rely on sensors and are fixed in position, our solution leverages computer vision capabilities to detect and manipulate objects regardless of their position errors. The solution is intelligent and programmable, enhancing flexibility and adaptability. [Fuentes]

1.2. Existing Studies:

In the existing studies related to robotic arm solutions used in industries, several inconveniences and limitations have been identified. These inconveniences primarily revolve around the reliance on sensors for precise positioning and the fixed nature of the industrial arms. The following are some key inconveniences associated with the arms used in industries nowadays:

1. **Dependency on Sensors:** The current industrial arms heavily rely on sensors to accurately detect and position objects. This dependency on sensors increases the complexity and cost of the systems, as well as the need for regular calibration and maintenance. Moreover, sensors can be sensitive to environmental factors and may not provide consistent and reliable results in all conditions.

2. **Limited Flexibility:** Industrial arms are designed for specific tasks and often have limited flexibility in adapting to new or changing requirements. They are typically programmed to perform predefined movements and lack the ability to dynamically adjust their operations based on real-time inputs.

3. **Fixed Positioning:** Industrial arms are typically fixed in position within the manufacturing setup. This fixed positioning restricts their ability to operate in different environments or perform tasks in various locations. Moving or reconfiguring the arms to accommodate changes in production setups can be time-consuming and disruptive.

4. **Exact Positioning Requirements:** Current industrial arms often require objects to be placed in precise positions for successful manipulation. This requirement increases the complexity of the workflow and can lead to errors and inefficiencies if the objects are not positioned accurately.

It also limits the flexibility to handle variations in object positions or unexpected scenarios.

5. **Lack of Adaptability:** Industrial arms are typically programmed for specific tasks and lack adaptability to handle variations in object shapes or sizes. They may struggle to adjust their movements or gripping strategies when encountering different types of objects, leading to errors or failures in the manipulation process.

1.3. Comparative Study:

The existing solutions in the industry, which heavily rely on sensors for precise positioning, will be critically evaluated. This critique will highlight their limitations in terms of flexibility, adaptability, and the requirement for precise piece placement. It will emphasize the need for alternative approaches that can overcome these limitations.

These inconveniences highlight the need for alternative solutions that can overcome these limitations and offer increased flexibility, adaptability, and efficiency in object manipulation tasks. By leveraging computer vision and simulation technologies, our proposed robotic arm solution aims to address these issues and provide a more intelligent and programmable approach to handle object positioning based on their shapes.

1.4. Problem Statement:

The problem statement for the project will be defined, focusing on the limitations of existing industrial arms that rely on sensors for precise positioning and the need for a more flexible and adaptable solution. The problem statement will address the challenges faced in manipulating objects with varying positions and the potential inefficiencies in current industrial processes. Examples of specific problem statements may include:

- How can we develop a robotic arm simulation solution that overcomes the limitations of existing sensor-based systems in terms of flexibility and adaptability?
- What are the potential drawbacks of relying solely on sensors for precise positioning in industrial arms, and how can computer vision-based solutions mitigate these drawbacks?
- How can we improve the efficiency and effectiveness of object manipulation tasks by leveraging computer vision capabilities in a robotic arm simulation?

- What are the specific advantages of our proposed solution in terms of handling position errors and programmability, and how does it compare to existing industrial arms? [Vincze]

1.5. Adopted Solution:

The adopted solution will be presented, outlining the integration of Gazebo and RViz for robotic arm simulation and the utilization of OpenCV for shape detection. The solution will be described in detail, emphasizing its intelligent and programmable nature, its ability to handle position errors, and its potential to enhance flexibility and adaptability in object manipulation tasks.

Through this project, we aim to explore the potential of simulation-based visual servoing for manipulator robots, providing insights, and contributing to the advancement of robotic control systems.

2

Robotic arm design and modelisation

Introduction

This chapter focuses on the design and modeling process of the robotic arm equipped with a camera for pick and place tasks. It provides a detailed exploration of the various components involved in the design, the use of SolidWorks 2022 for creating the 3D model, the implementation of the design using URDF format within Gazebo for simulation, and the utilization of ROS for control, communication, and computer vision capabilities.

2.1 Components of the Robotic Arm:

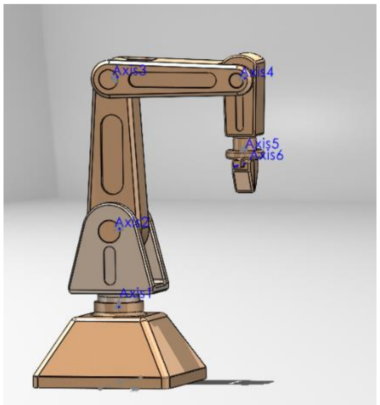


FIG 2.1: Robot Arm Assembly

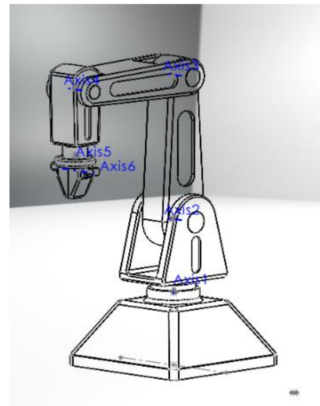


FIG 2.2: Robot Arm Joints

2.1.1-Base: The base provides stability and acts as the foundation for the entire robotic arm. It is usually a stationary component that supports the arm's movement.

2.1.2-Joints:

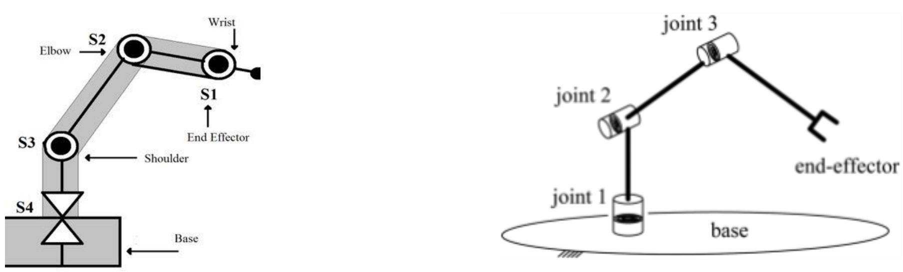


FIG 2.3: Cinematic Modelisation

Joints are the crucial parts that enable the robotic arm to move. There are various types of joints, such as revolute, prismatic, and spherical, each allowing specific degrees of freedom for movement.

2.1.3-Links: Links connect the joints and form the structure of the robotic arm. They can be rigid or flexible and determine the arm's reach, length, and overall geometry.

2.1.4-Actuators: Actuators are responsible for generating the necessary forces and torques to move the robotic arm. Electric motors, hydraulic systems, or pneumatic systems are commonly used as actuators. We are using electric motors in our case. [Tsai]

2.1.5-Effector: The effector is the tool or gripper attached to the robotic arm that interacts with the objects during the pick and place operation. It can be customized based on the specific requirements of the task.

2.1.6-Camera:

FIG 2.4: Eye-to-Hand Camera

Enables the arm to capture visual information about the task to be performed. By integrating computer vision techniques, the arm can identify and recognize objects, locate their positions, and determine the appropriate actions for pick and place operations.

2.2-Design: using SolidWorks 2022

To design the robotic arm, we employed SolidWorks 2022, a powerful computer-aided design (CAD) software. SolidWorks provides a user-friendly interface and a wide range of tools for creating 3D models, assembling components, and simulating movements. We utilized these features to design the individual components of the robotic arm, including the base, joints, links, end effector, and the camera. [SolidWorks]

3

Control Method and Simulation

Introduction

After finalizing the design in SolidWorks, we converted it to the URDF (Unified Robot Description Format) file to enable simulation using Gazebo. URDF is a markup language used to describe the structure, kinematics, and dynamics of a robot. By converting the design to URDF, we were able to import it into Gazebo, a popular robot simulation environment. Gazebo allowed us to validate the design, test the motion capabilities, and simulate pick and place operations in a virtual environment.

3.1 Control Method:

A joint-based control method is used to operate and manipulate the robotic arm. ROS offers control libraries and packages for implementing control algorithms, contributing to the ease of control implementation in this project. [Gao.Y]

To control the robotic arm, the joint_state_publisher package is utilized, enabling the publication of joint states and control commands. This package allows sending desired joint angles or positions to the robotic arm's joints, commanding it to move in a desired manner. By defining appropriate control commands, precise control over the robotic arm's movements can be achieved. [Chaumette]

3.1.1 MoveIt:

MoveIt

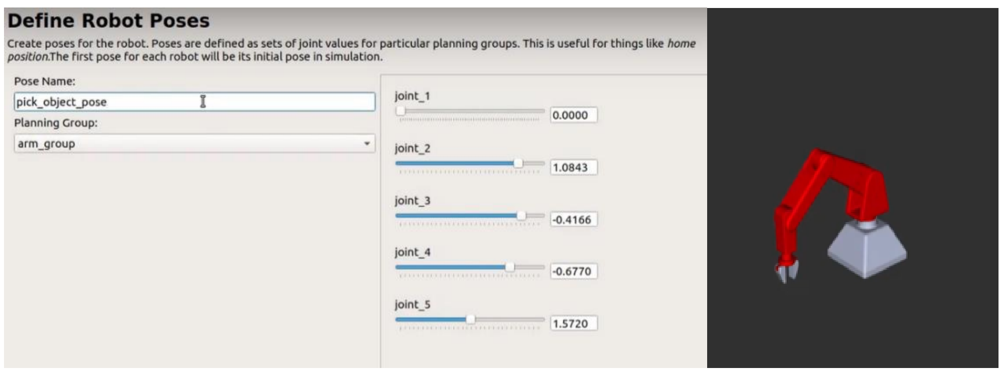


FIG 3.1: MoveIt-poses-defining

MoveIt, a powerful motion planning framework in ROS, is utilized to manipulate the robotic arm's positions and trajectories. By interfacing with MoveIt, the desired positions of the arm can be defined and controlled, enabling precise movements and accurate pick and place operations in the simulation. MoveIt provides a comprehensive set of tools for path planning, collision detection, and trajectory execution, ensuring smooth and collision-free motion of the robotic arm within the simulation environment. [MoveIt]

3.2 Simulation with Gazebo and RViz

Simulation is a crucial aspect of robotics development as it allows us to test and validate the behavior of robotic systems in a virtual environment before deploying them in the physical world. In this project, we utilize Gazebo and RViz, two widely-used simulation tools in the ROS ecosystem, to simulate the robotic arm with 3 DOF. [Kyrkjebø]

3.2.1Gazebo

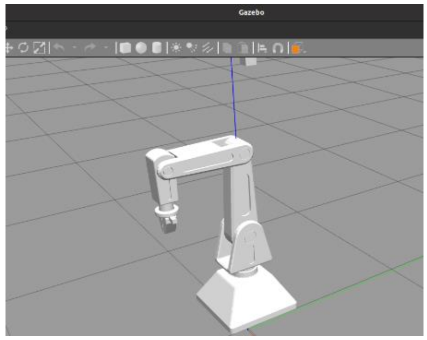


FIG 3.2: Gazebo-Arm-simulation

Gazebo is a powerful physics-based simulation platform that provides a realistic environment for simulating robotic systems. It allows us to model the physical properties of the robotic arm, such as joint dynamics, collision detection, and sensor integration. With Gazebo, we can simulate the behavior of the robotic arm in various scenarios, including object interaction, obstacle avoidance, and grasping tasks. This enables us to evaluate the arm's performance and behavior under different conditions. [Gazebo]

3.2.2RViz

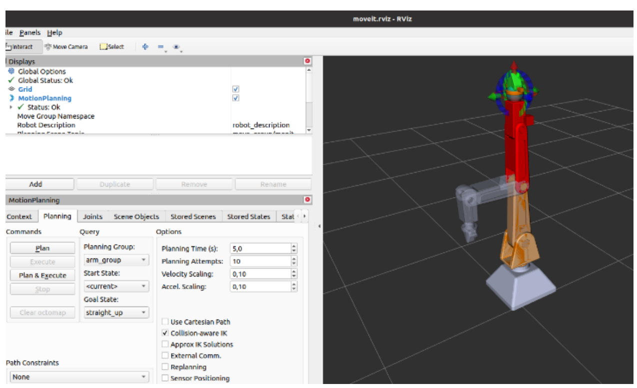


FIG 3.3: RVIZ-Arm-simulation

RVIZ is a 3D visualization tool that enables us to visualize the simulated robotic arm and its surroundings. It provides a graphical interface to display the robot's kinematic structure, joint states, and sensor data in real-time. RViz allows us to monitor the arm's movement, visualize the effects of control commands, and debug any potential issues during simulation. It enhances our understanding of the robotic arm's behavior and aids in fine-tuning the control algorithms.

→By combining Gazebo and RViz, we create a comprehensive simulation environment for the robotic arm. We can visualize the arm's motion, interact with objects in the simulated environment, and assess its performance without the need for physical hardware. This simulation-driven development approach accelerates the prototyping and testing phases, reducing costs and risks associated with hardware experimentation.

3.3 Incorporating the Camera in the URDF

The URDF file defines the robot's structure, joints, and visual properties, allowing us to describe the robot's kinematics and appearance. To add the camera to the robotic arm's URDF, the following steps were followed:

1. **URDF Modification:** The URDF file was edited to include a camera link and its corresponding joint. The camera link represents the physical location and orientation of the camera in relation to the robotic arm. The joint connects the camera link to the arm's existing structure, enabling it to move along with the arm's motions.
2. **Camera Properties:** Within the camera link, properties such as field of view (FOV), image resolution, and camera mount position were specified. These properties define how the camera captures the environment and provide necessary information for image processing and perception tasks.
3. **Sensor Plugin:** To simulate the camera's functionality, a sensor plugin was added to the camera link in the URDF. This plugin emulates the behavior of a camera sensor and publishes image data to a ROS topic.
4. **ROS Topic Creation:** In the ROS environment, a camera topic was created to receive and process the image data from the simulated camera. This topic allows other ROS nodes to subscribe and utilize the camera feed for various tasks, such as object detection, image processing, or visual servoing. [Quigley]

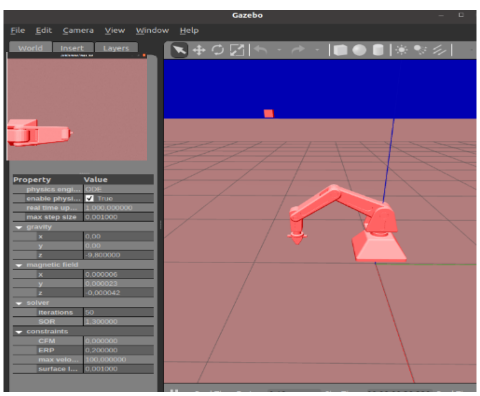
By incorporating the camera in the URDF and defining its properties, it creates a virtual representation of the camera within the robotic arm simulation. This enables to simulate the camera's viewpoint and obtain image data that reflects the arm's environment during operation.

3.4 Utilizing the Camera Topic

The camera topic serves as a communication channel between the simulated camera and other ROS nodes. By subscribing to the camera topic, we can access the image data and perform various computer vision and perception tasks. Some common use cases include:

1. **Object Detection:** The camera feed can be processed using computer vision algorithms to detect and recognize objects within the arm's surroundings. This information can be utilized for object manipulation or obstacle avoidance. [OpenCV]

2. **Visual Servoing:** The camera topic allows for visual servoing techniques, where the robot's control algorithms can utilize visual feedback to track and manipulate objects accurately. This enables precise control and manipulation tasks that require visual information. [Rosales]



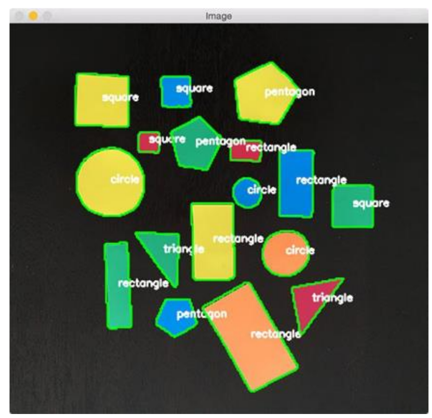


FIG 3.4: Gazebo-image-view

FIG 3.5: OpenCV-shape-detection

By integrating the camera topic into the simulation environment, we enhance the arm's perception capabilities, allowing it to interact with and understand its surroundings. This opens up possibilities for advanced control and interaction techniques, paving the way for more complex and versatile robotic arm applications. [Bradski]

General conclusion and perspectives

In summary, the control method and simulation aspects are crucial components of the project. The joint-based control method enables precise control over the robotic arm's motion, while Gazebo and RViz provide a realistic simulation environment for testing and validating the arm's behavior. By leveraging these tools and techniques, you can develop and refine the control algorithms, ensuring the successful operation of the robotic arm in real-world scenarios.
By adding the camera link to the URDF and creating a camera topic in ROS, you can successfully incorporate a camera into the robotic arm simulation. This addition allows the arm to visually perceive its environment, enabling various perception and control techniques. The camera's image data can be utilized for tasks like object detection, visual servoing, environment mapping, and human-robot interaction. This enhances the arm's functionality and versatility within the simulation environment.

Image-based Visual Servoing (IBVS) and Position-based Visual Servoing (PBVS) are two main approaches to visual servoing control. IBVS directly uses image features to determine the error between desired and actual features, generating control signals for robot movement. PBVS, on the other hand, relies on the 3D position and orientation of the robot end-effector or a specific feature, comparing it to the desired pose to generate control commands. IBVS handles complex scenes but may face depth ambiguity, while PBVS provides accurate control but requires calibration and 3D models.

In addition to these traditional visual servoing methods, there is a growing possibility of integrating deep learning neural networks into the IBVS framework. Deep learning has shown remarkable success in various computer vision tasks, such as object recognition and image segmentation. By incorporating deep learning models into IBVS, it becomes possible to leverage the power of neural networks for visual perception and control.
“The derivation process of Jacobian is cumbersome and complicated due to its nonlinearity. What’s more, real-time estimation of Jacobian is needed in most actual control tasks because of its time variation, which leads to a large amount of calculation. To solve this problem, some researchers have proposed to introduce the neural network into the visual servo control.” [IWACIII2019]

Bibliography

[Bradski] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media.

[Chaumette] Chaumette, F., & Hutchinson, S. (2006). Visual servo control: Part I—Basic approaches. IEEE Robotics & Automation Magazine, 13(4), 82-90.

[IWACIII2019] International 6th Workshop on Advanced Computational Intelligence and Intelligent Informatics. University of Electronic Science and Technology of China (UESTC), Chengdu, China, November 1-5, 2019.

[Fuentes] Fuentes-Pacheco, J., & Barron-Zambrano, J. H. (2019). Survey on pick-and-place robotic systems: Toward flexible and autonomous industrial applications. Robotics and Autonomous Systems, 121, 103259.

[Gao.Y] Gao, Y., & Shibata, M. (2018). Robotic Arm Motion Planning and Control. In Robot Arm: Developments, Control and Automation (pp. 19-34). IntechOpen.

[Kyrkjebø] Kyrkjebø, E. K., & Lillevik, O. G. (2012). Developing a robotic arm simulation system. In 2012 6th European Conference on Intelligent Systems (pp. 119-126). IEEE.

[Prattichizzo] Prattichizzo, D., & Siciliano, B. (2001). Task-based control of robot manipulators: A tutorial. In IEEE Robotics & Automation Magazine, 8(4), 17-32.

[Quigley] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: An open-source Robot Operating System. In ICRA Workshop.

[Rosales] Rosales, C., & Ramírez, M. (2012). Visual servoing control: A review. IETE Technical Review, 29(2), 118-130.

[Tsai] Tsai, R. Y. (1999). Robot Analysis: The mechanics of serial and parallel manipulators. John Wiley & Sons.

[Vincze] Vincze, M., Roth, P. M., Beetz, M., Kröse, B., & Wolter, F. (2013). Robotic grasping and manipulation competition at ICRA. IEEE Robotics & Automation Magazine, 20, 98-99.

[Gazebo] Gazebo. "Gazebo: A multi-robot simulator for outdoor environments." http://gazebosim.org

[MoveIt] MoveIt. "MoveIt! Open-Source Software for Manipulation Planning." https://moveit.ros.org/

[OpenCV] OpenCV. "OpenCV: Open-Source Computer Vision Library." https://opencv.org/

[SolidWorks] SolidWorks. "SolidWorks 3D CAD Software." https://www.solidworks.com/