

# Centro E. Piaggio Amazon Picking Challenge structure

---

## Task to achieve

This project was developed to answer the problem of automated object handling in a factory-like environment using soft robotics technology. An investigation on a recent control approach for soft robots based on the exploitation of environment contain (EC) to help the robot perform the grasp, has been made.

The task that this robot wanted to achieve, was to unload a shelf full of random objects, selected from a total of 30 known objects. To achieve this task, the object must be previously recognized.

---

## Hardware solution proposed

A *pick 'n' place* manipulator composed of Variable Stiffness Actuators (VSA) and a Pisa/IIT SoftHand has been designed. In combination with these actuators a three-dimensional depth sensor to reconstruct the geometry of the actual scene, has been embedded.

A database of object's postures with a precise illumination was made. The objects are extracted from the RGB-D image collected from the camera using a dedicated software based on ROS.

Before assembling the full cartesian frame robot, a series of grasp strategies with the Pisa/IIT SoftHand were made to develop a grasp database based on the objects that needed to be grasped, in order to achieve the task proposed by Amazon.

---

## Software

To fully control the robot we implemented a custom control algorithm for sensor fusion, implementation of grasp strategies, motion planning and motor control.

The perception algorithm identifies a specific object inside the scene and sends its position and orientation to the motion planning algorithm that decides the best grasp strategies between a pre-defined set of routines.

Each of the strategies defines different movements and generates several robot states as pseudoway-points to be followed by the robot. When the routine finishes its process a bunch of robot states are stored in memory and the motion planning algorithm creates a trajectory interpolating them.

The entire process is implemented by several parallel processes that need to communicate and share information. A ROS based architecture is used to implement

inter-process communication. In particular ROS topics and ROS services, which are part of the ROS communication paradigm, are used to perform the task.

Software is divided in three communicating nodes, each of which is written in C++ language.