UNIVERSIDAD NACIONAL DE COLOMBIA
DEPARTAMENTO DE INGENIERÍA MECÁNICA Y MECATRÓNICA
ROBÓTICA 2020-II
Profesor: Pedro Fabían Cárdenas.
Monitor: Jurgen Krejci Muñoz.

PROYECTO INTERMEDIO
Aplicación de Pick & Place para manufactura

1. Objetivos

- Aplicar de forma práctica los conceptos sobre manipuladores seriales adquiridos durante el curso de robótica.
- Utilizar **ROS** como una plata forma de integración, permitiéndonos la integración de manera sencillas del hardware y software.
- Implementar un entorno de simulación en **Gazebo** donde se pueda hacer el manejo de un robot PhantomX Pincher.
- Implementar un algoritmo de visión de maquina simple para la detección de piezas de manufactura.
- Desarrollar una aplicación de Pick and Place con el robot PhantomX pincher para apoyar una etapa de un proceso de manufactura.

2. Descripción

El objetivo de este proyecto es el diseño e implementación de una aplicación de **"Pick & Place"** que integre visión de máquina para apoyar una etapa de suministro de material en una línea de manufactura.

En principio, proceso de manufactura dispone de varias etapas, en particular se busca robotizar la alimentación de **2 tipos de piezas** y preparación para un posterior ensamble que tendrá lugar en la etapa siguiente. Las piezas que se organizar son respectivamente tornillos y tuercas. Las piezas son ingresadas a través de **2 áreas demarcadas** (una para cada tipo de piezas), cada zona tendrá asignada un manipulador el cual tomará las piezas que ingresan a su respectiva zona y las llevará a la zona de ensamble, en esta cada manipulador tendrá un punto específico donde ubicara y dejará el material ingresado, ver figura 1.

Las piezas llegan a la zona de alimentación de **forma aleatoria**, es decir, pueden llegar en cualquier momento y en cualquier posición/orientación dentro de la zona, pueden llegar de a una o varias simultáneamente, para poder determinar la ubicación de estos elementos en esta zona se propone usar **visión de máquina**, para detectar la presencia de elementos y determinar su posición y orientación.

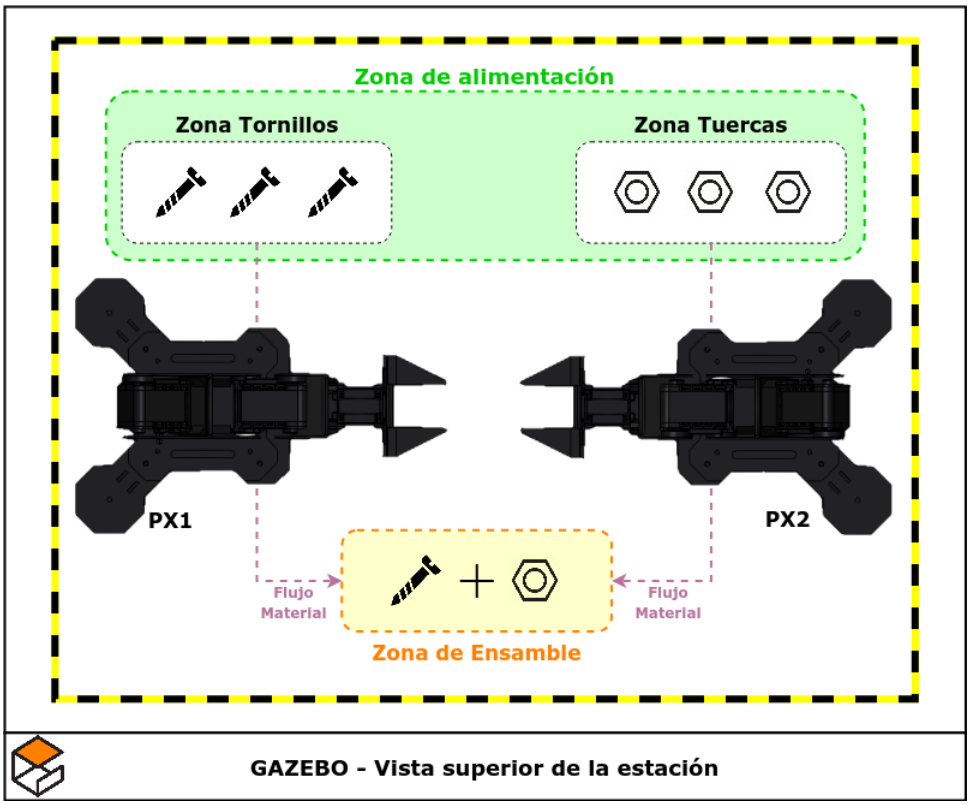

Figura 1: Diagrama de la estación

La aplicación deberá contar con un **GUI (Interfaz gráfica de usuario)** la cual permita monitorear la ejecución del proceso e implemente funciones básicas de seguridad (Inicio, parada de emergencia, pausa, etc.).

3. Requerimientos de funcionamiento e implementación

- Toda la aplicación debe ser implementada usado **ROS** como plataforma de integración, la simulación de la estación se hará con **Gazebo**, la generación de trayectorias, control de los robot/estación y visión de máquina, se hará usando **Matlab** (se deja abierta la posibilidad de trabajar los algoritmos de visión usando otros frameworks como OpenCV).
- Los robots a utilizar para esta aplicación van a ser el robot **PhantomX Pincher** (PX). Si el grupo de trabajo lo prefiere puede escoger otro tipo de robot, pero debe tener en cuenta las dimensiones y fuerzas inerciales del mismo.
- Se debe implementar una interfaz gráfica que incluya como mínimo:
  1. Botones de inicio y parada del proceso
  2. Indicadores del estado del proceso (operación/parada/pausa)
  3. Posición y orientación actual del robot en el espacio operacional (TCP)
  4. Posición actual del robot en el espacio de las configuraciones (Valores articulares)

5. Visualización de la cámara de la zona física, donde vea claramente la detección de los elementos.
6. Coordenadas de las piezas a procesar dentro de la zona de alimentación

- El robot, los objetos con que interactúe y el escenario de simulación serán **virtuales** (Gazebo), pero la cámara para detección de piezas se requiere que sea **física**, para ello el grupo deberá hacer un montaje (simplificado) de la zona de alimentación de forma física, esta zona "real" será única y en ella serán ubicadas las piezas (tuercas/tornillos) mezcladas, en simulación se mapean y separan las coordenadas de las piezas reales y se asignan estas coordenadas al manipulador correspondiente dependiendo del tipo de piezas detectado, ver figura 2.

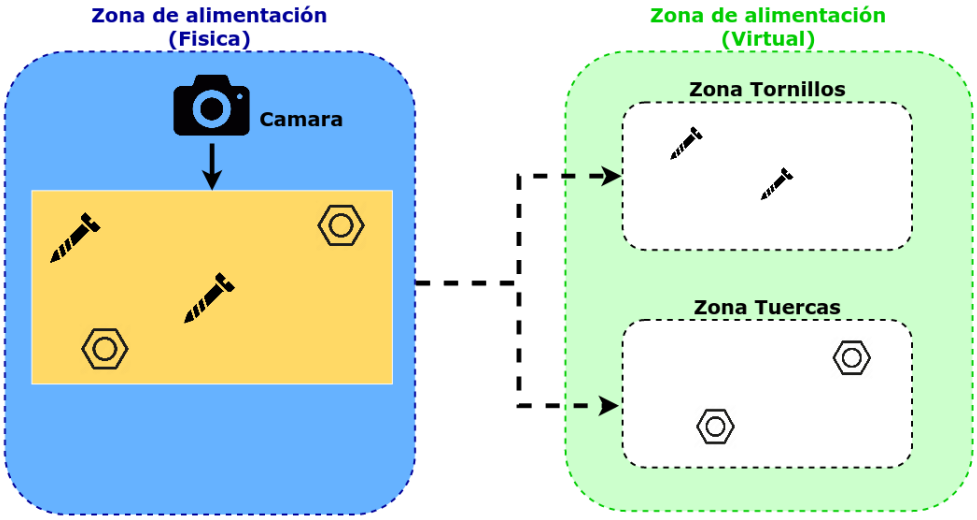

Figura 2: Integración de cámara física en la simulación

- En Gazebo no es obligatorio que existan los sólidos que representan las tuercas y tornillos, los movimientos se pueden hacer con el gripper vacío, pero será requerido que exista la demarcación en el suelo que diferencie cada zona.
- La piezas en la zona de alimentación (física) llegan a esta de **forma aleatoria**, en cualquier posición y momento, de forma combinada (tornillos y tuercas), pero para simplicidad del ejercicio se limitará a un **máximo de 6 unidades** de forma simultánea en esta zona.
- En la zona de ensamble, donde van a ser entregadas las piezas, van a existir **2 niveles** (diferenciados por su altura), el nivel inferior corresponde al punto de entrega para tornillos y el superior corresponde a tuercas. Se recomienda agregar en la simulación algún elemento o superficie que sirva para apoyar y diferencias estos niveles, ver figura 3.

Opcionales valorables: Los siguientes corresponden a una serie de ítems que **no** son de implementación obligatoria pero son positivamente valorados en el cálculo de nota si son implementados.

- Inclusión de sólidos dentro de la simulación que representen las tuercas y tornillos detectados por la cámara (zona física), los manipuladores deberán tomar estos solidos y llevarlos a la zona de ensamble (los movimientos ya no serían con el gripper vacío)

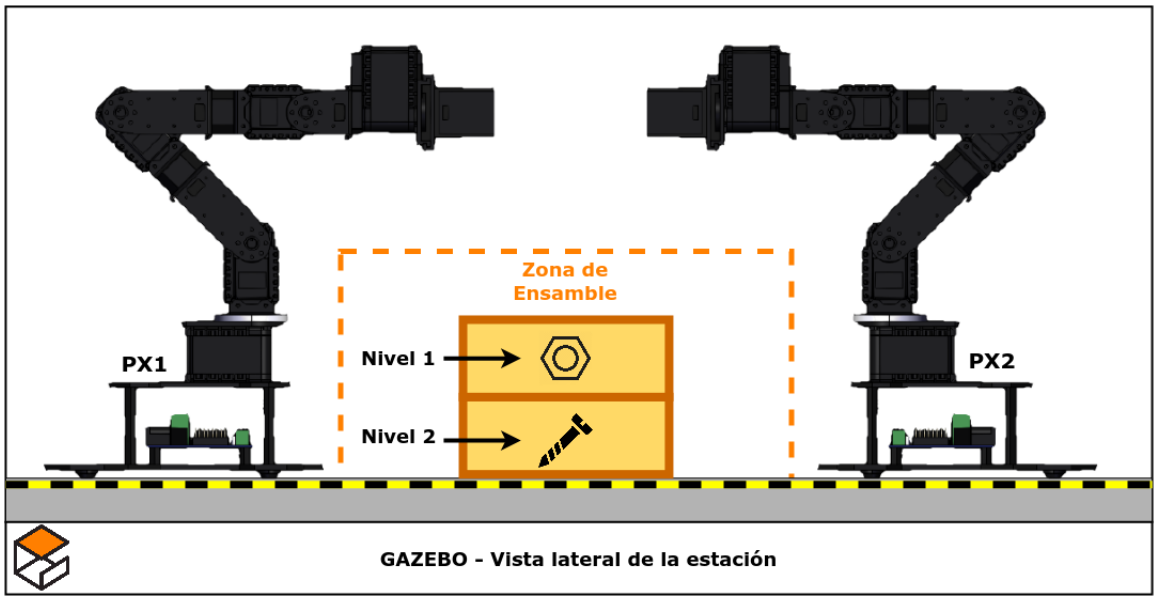

Figura 3: Zona de ensamble (entrega de material)

- Inclusión de una cámara en la simulación desde la cual el usuario pueda observar la estación en proceso a través del GUI implementado.
- Secuencia de ensamblaje colaborativa entre los manipuladores, se valorara positivamente la implementación de una secuencia de roscado, teniendo en cuenta las limitaciones (4DOF) de los manipuladores, esta seria simplificada y consistiría en que la tuerca fuese simplemente insertada en el tornillo por los manipuladores antes de ser entregados en la zona de ensamble.

4. Resultados esperados

Se espera que cada grupo de laboratorio logre:

- Integrar por medio de **ROS** herramientas de software como Gazebo y Matlab para hacer el proceso de control/operación del robot Phantom.
- Implementar un algoritmo de visión de maquina simple que permita la detección de elementos en un ambiente físico, del cual se pueda extraer información y ser levada a un entorno virtual.
- Generar trayectorias de “Pick & Place” a partir de coordenadas obtenidas por medio de visión de máquina
- Implementación de un **GUI** para el monitoreo y control de proceso
- Cooperar en el desarrollo de un proyecto entre múltiples estudiantes.
- Planear y organizar actividades para el desarrollo de un proyecto de escala mediana

La figura 4 muestra un ejemplo de arquitectura de solución, a la que se espera lleguen los grupos de trabajo.

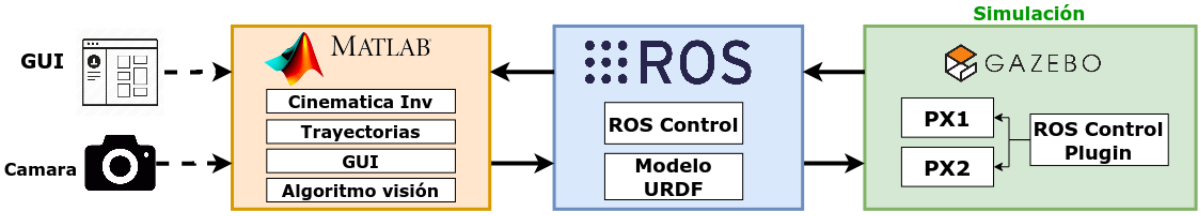

Figura 4: Ejemplo de arquitectura de implementación

5. Entregables

1. **Video de demostración de funciones:** Duración de máximo 5 min. Deberá incluir al inicio el clip de introducción del LabSiR (compartido con este informe), seguido de un front de la universidad nacional de Colombia, autores profesor, curso, año etc. En este debe incluir un breve explicación del problema y una demostración de las funciones del proceso, donde se vea claramente el proceso de visión, la trayectorias generadas en Gazebo y el funcionamiento del GUI. Se espera que el vídeo contenga narración por parte de los integrantes del equipo de trabajo. Ejemplos vídeos generales:

   - Ejemplo 1 Learning to Generate 6-DoF Grasp Poses with Reachability Awareness (ICRA 2020) o Youtube .
   - Ejemplo 2 Robot telekinesis (ICRA 2020) - 7 min. presentation.
   - Ejemplo 3 ICRA 2020 - Tactile Dexterity: Manipulation Primitives with Tactile Feedback MIT.
   - Ejemplo 4 ICRA 2020 – "Multiplexed Manipulation: Versatile Multimodal Grasping via a Hybrid Soft Gripper".
   - Ejemplo 5 Online Replanning in Belief Space for Partially Observable Task and Motion Problems

   El vídeo debe ser subido y compartido a través de Youtube.

2. **Wiki:** El grupo de trabajo deberá realizar colaborativamente un Wiki la cual servirá como forma de documentación del proyecto, para esto se va usar la herramienta Wikidot, en esta se debe incluir:

   - Proceso de diseño de la solución
   - Arquitectura de la solución
   - Explicación del código implementado
   - Tutorial para el uso y manejo del software desarrollado
   - Diagramas y planos de la estación
   - Especificación de materiales y herramientas utilizadas

   En el Wiki cada uno de los miembros debe ayudar a la construcción colectiva del mismo,para determinar la contribución de cada miembro se hace uso del karma, mas detalles ver Karma.

El nivel de Karma será valorado en la nota de cada integrante.

3. **Repositorio:** Cada grupo de proyecto debe gestionar todo su código por medio un repositorio git de su proyecto (Se sugiere usar GitHub), en este se debe subir todo el código, paquetes, escenarios y demás software desarrollado para la solución del proyecto, los miembros de cada equipo deben aparecer como colaboradores. La cantidad de aportes (commits) hechos por cada miembro sera valorado en el nota individual.
4. **Sustentación y presentación:** Cada grupo preparará una presentación de 20 minutos max. en la cual presentara su solución y hará un demo en vivo (vía vídeo llamada) de su proyecto, esta presentación debe venir acompañada de diapositivas que incluyan diagramas e información clave de la solución implementada.

Observaciones

- **Fecha de entrega: 03/12/2020** , para en la entrega subir en Moodle un unico archivo .zip que incluya: copia del video en formato .mp4, .mov o .mkv, copia de la presentación en PDF y archivo .txt que incluya los links a Wiki, Repositorio y Video en Youtube.
- **Fecha de sustentación: 04/12/2020** , Pendiente por confirmar la hora. Hora tentativa: 10am a 1pm.
- **Forma de trabajo:** El proyecto se ejecutará en grupos de 4 personas (uniendo dos parejas de laboratorio que sean del mismo grupo clase del SIA).

6. Recursos y material de apoyo

A continuación se presentan una serie de recursos que se consideran útiles para la implementación de la aplicación:

- Como incluir cámaras y otros plugins en Gazebo Link
- Creación de mundo en Gazebo Link
- Añadir texturas a un modelo en Gazebo Link
- Incluir objetos (Spawn) en un simulación de Gazebo Link
- Computer Vision Toolbox para Matlab Link
- ROS USB Camera package Link
- Creación de GUI en Matlab con GUIDE Link
- OpenCV Bridge for ROS Link

Adicionalmente se incluyen algunos vídeos de aplicaciones industriales similares para que inspiren el desarrollo de la solución:

- Easy assembly with the help of YuMi Robot at Cejn
- ABB Yumi Assembly 2

- Nut and Screw assembly with Robotiq

También se comparte el canal del LabSIR donde están publicados algunos vídeos con proyecto de semestres pasados implementados con ROS.

Referencias

[1] John J. Craig *Introduction to Robotics, Mechanics and Control*. 2005.

[2] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control*. 2004.

[3] Peter Corke. *Robotics Toolbox for Matlab, Release 9*. 2015.

[4] Peter Corke. *Robotics, Vision and Control - Fundamental Algorithms in MATLAB®*. Springer Tracts in Advanced Robotics. Springer, 2011, págs. 1-495.

[5] Richard M. Murray, Zexiang Li, S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. University of California, Berkeley.

[6] Martinez, A., Fernandez E. *Leaning ROS for robotics programming, PackT Publishing*. 2015.

[7] ROS.org Wiki, *Wiki: urdf (last edited 2019-01-11 01:15:14 by Playfish)*, 2020. Disponible en: disponible en link

[8] Arruda M., *My Robotic Manipulator 1: Basic URDF & RViz*, 2018. Disponible en link