

Virtual Laboratory of Robotics for practices of industrial robotics using ROS middleware and compatible tools.

Yeser A. Morales C.
Facultad de Electrotecnia y Computación
Universidad Nacional de Ingeniería
 Managua, Nicaragua
 myalfredo03@ieec.org

Abstract — The aim of this paper is to present a software tool that allows a student to know the basic concepts of industrial robotics, providing for this purpose a scenario with models of industrial robots and a G6UI that allows to enter Configurations by the student of robotics to know the direct and inverse kinematics of the model of robot to study. This software tool is based on the middleware widely used by robotics worldwide called ROS (Robot operating System), which provides libraries, viewers, simulators, etc. Of great importance to this work. This paper also shows the methodology of teaching and technique considered for the realization of this tool of software that aims to be placed like a data source for the verification of the basic theory of robotics.

Keywords- ROS, RBDL, RVIZ, GUI, Industrial Robot, Virtual Practice.

I. INTRODUCCION

El campo de la robótica ha tenido, principalmente en las últimas décadas, un desarrollo vertiginoso y está impactando positivamente, principalmente en países desarrollados, en áreas como la medicina, la industria, y la investigación, entre otras. Su impacto en los países en vías de desarrollo llegará más temprano que tarde y es importante que los mismos cuenten con profesionales con las competencias requeridas para llevar a efecto las diferentes tareas relacionadas con el desarrollo y uso de robots.

En países en vías de desarrollo, como Nicaragua, la robótica se encuentra en un estado incipiente [1], lo cual se debe, entre otras razones, a una limitada oferta de programas académicos y técnicos en dicho campo. Una explicación para la baja oferta es que la inversión requerida para equipar un laboratorio, con las condiciones necesarias para impulsar la investigación y la enseñanza de la robótica, es muy alta. Desde el punto de vista de la formación, la robótica es una excelente herramienta para mostrar la importancia de la interdisciplinariedad dado que en misma convergen diferentes disciplinas entre las que destacan las ciencias básicas, física y matemática, e ingenierías tales como la computación, la mecánica, la eléctrica, y la electrónica.

Se espera que los sistemas de producción automatizados incrementen sus usos, Ver Fig. 1. aunque en Nicaragua no hay muchos procesos que utilicen Sistemas totalmente automatizados con Robots, la integración de estos sistemas han de ser un bien aceptable y tendrán que ir aumentando paulatinamente y es este uno de los objetivos que mueve el

desarrollo de este trabajo de comenzar a preparar a los profesionales en desarrollo y que se cuente con una visión de los sistemas industriales con nuevas tendencias tecnológicas, en el sector industrial se encuentra una afirmación a flote la cual cita que «No hemos formado a las personas para trabajar en la industria y ahora no podemos encontrar el talento necesario» [2].

Una alternativa que impactaría positivamente en el desarrollo de la robótica en el país, es el uso de modelos virtuales de robots que permitan ser simulados en un computador y ejecutar prácticas que brinden al estudiante datos de la simulación y que permitan corroborar los conceptos básicos teóricos de la robótica industrial como Cinemática, Dinámica y modelación Matemática.

Las practicas virtuales de robótica puede ser ejecutadas por muchos estudiantes a la vez en diferentes computadores y desde cualquier sitio donde se cuente con la instalación. Si se utiliza una plataforma de software libre, la inversión requerida disminuye considerablemente.

Para dar respuesta a la problemática planteada, se desarrolló un laboratorio virtual de robótica industrial (LVR) que incorpora varios modelos virtuales de robots industriales para el estudio de estos mismos, El escenario virtual es de código abierto y escalable y cumple con ciertos estándares en el estudio de la robótica y en el caso del desarrollo de este trabajo exclusivamente de la Robótica Industrial para esto se utilizo el middleware Robot Operating System, ROS. El sistema operativo ROS, permite el acceso al uso de simuladores, visualizadores de información y modelos 3D, sensores y cálculos relacionados con el diseño de robots, particularmente de robots manipuladores de carácter industrial.

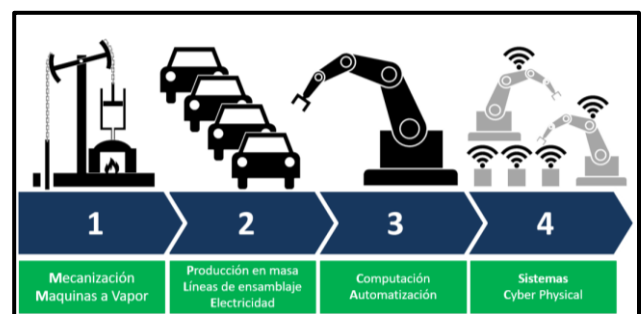


Fig. 1 Evolución de los sistemas de Producción

II. METODOLOGÍA

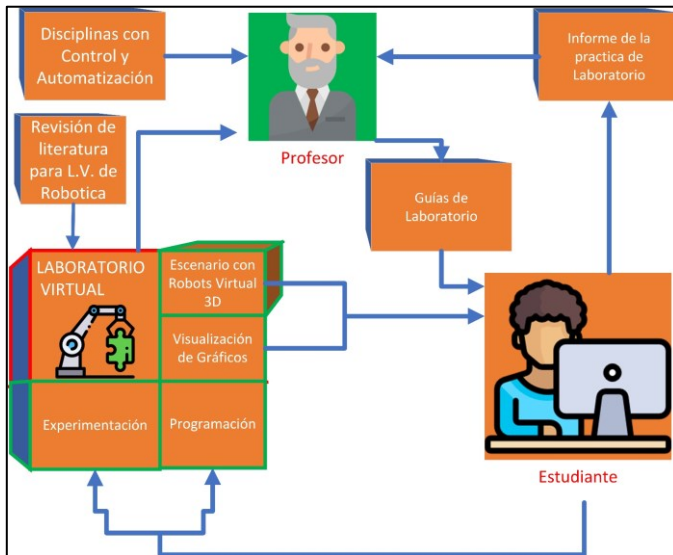


Fig. 2 Modelo de uso del Laboratorio Virtual de Robótica

A. Generalidades

Como se aprecia en la Fig 2, el docente podrá, partiendo de los temas relacionados con los fundamentos de la robótica industrial que desee desarrollar y de los recursos disponibles en el laboratorio, elaborar las guías que utilizarán los estudiantes para realizar las actividades de estudio o las prácticas de laboratorio correspondientes. El Estudiante usará el LVR para realizar los experimentos planteados, realizar sus observaciones y generar un informe en base a lo experimentado.

Las herramientas de software utilizadas en este trabajo son:



Fig. 3 Herramientas de software usadas en el LVR

1) **ROS**: El middleware ROS funciona bajo comunicación peer-to-peer para los procesos que pueden ser soportados tales como nodos, mensajes, tópicos, servicios y base de datos (bags) propias de ROS. Tales procesos de comunicación son aprovechados para permitir al nodo del modelo del robot virtual ser lanzado y comunicado con el nodo del escenario virtual y este a su vez aceptar las configuraciones del usuario por medio de la GUI, Ver Fig. 4.

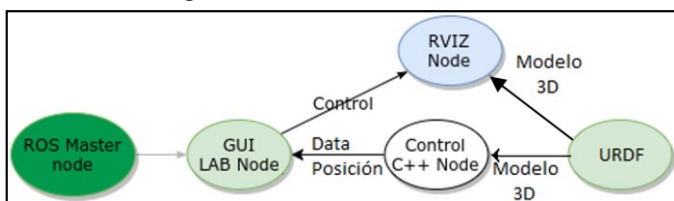


Fig. 4 Nodos de ROS

2) **RVIZ**: ROS posee una herramienta de software propia llamada RVIZ para visualización de datos tridimensionales como objetos 3D, datos de cámaras, sensores tridimensionales etc. Todos estos datos usados normalmente para la localización y orientación del robot virtual y ejecutar algoritmos para controlar el robot físico, también siendo esta herramienta de mucha ayuda al programador/desarrollador de robots, en este trabajo se utilizó esta herramienta para poder representar en un panel 3D [8] dentro de la GUI los diferentes robots virtuales de interés de estudio de la robótica industrial, se utilizaron las librerías bases de la herramienta RVIZ y se incorporaron dentro de la GUI desarrollada, para facilitar al usuario/estudiante un escenario y una GUI dentro de la misma ventana, donde este pueda hacer las configuraciones necesarias para aprender los conceptos básicos de Robótica Industrial.

3) **URDF**: Para la modelación del Robot en ROS El formato es URDF (Unified Robot Description Format) [4], el cual es un archivo XML en el que se describen las propiedades físicas del robot a modelar donde se especifican las dimensiones del robot, los movimientos de las articulaciones, parámetros físicos como masa e inercia, etc.

4) **RBDL**: Es un paquete de software libre de código abierto desarrollado en C++ que implementa algoritmos de dinámica y cinemática, [5]. Esta librería de software posee muchas capacidades de cómputo de importancia en el desarrollo de cualquier sistema robótico ya sea siendo un trabajo de simulación o utilizarla para los cálculos de movimiento de un robot físico, cabe destacar que para lograr explotar cada uno de los recursos de la librería es necesario la utilización de ROS y establecer la comunicación con diferentes procesos que requieran de estos cálculos en la Figura 3 se muestra el nodo de control en C++ el cual provee de cálculos de posición a la GUI de acuerdo a la posición del robot dentro del escenario siendo estos los datos de interés en el estudio de la robótica.

5) **GUI**: Para el desarrollo de la interfaz gráfica de Usuario se utilizó el IDE QT Creator [6] el cual posee un plugin para compilación de los paquetes de ROS desarrollado por la misma comunidad ROS, el cual permite dos aspectos importantes dentro del desarrollo del proyecto, los cuales son: Primero, Aprovechar la capacidad de los recursos del IDE para la creación de la GUI tales como botones, barra de texto, widgets, etc. Ver Fig. 5 Segundo disponer de todas las librerías de desarrollo de software de ROS y combinar los datos que provee ROS con los diferentes Widgets en la GUI de tal manera que la interfaz de usuario sea lo más transparente posible para el usuario/estudiante Ver Fig. 2.

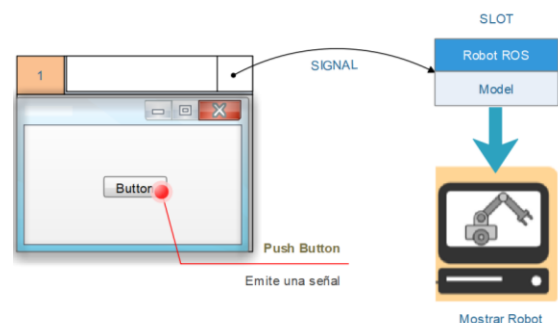


Fig. 5 GUI con Señales y Slots de Proceso de Software.

B. Diseño del laboratorio virtual de robótica

En la revisión de literatura y reconocimiento de los laboratorios en algunas instituciones de educación superior, encontramos que la principal actividad que se realiza en este tipo de laboratorios es la programación del robot, usando un lenguaje específico, para que ejecute una tarea determinada. [10]

El laboratorio virtual desarrollado, ver Figura 6, cuenta con los recursos de un laboratorio básico de robótica e incorpora otros recursos que permiten realizar otras actividades como comprobación de cinemática del robot yendo más allá de la simple programación del modelo del robot bajo consideración.

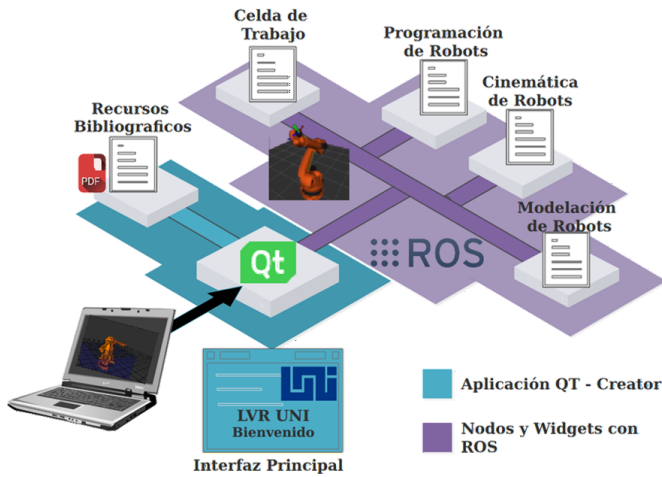


Figura 6: Modelo del laboratorio Virtual de Robótica Industrial.

Para modelar los robots a utilizar en esta herramienta de software se utilizó el formato de ROS para modelación de robots URDF el cual es caracterizado por aceptar modelos de robots tipo árbol o tipo serie, descartando los robots paralelos que también son utilizados en la robótica industrial, cabe destacar que se contemplan dos tipos de modelos los ya desarrollados por empresas como ABB, Kuka, Fanuc etc. que normalmente poseen 6 DOF (grados de libertad), y los modelos menos complejos desarrollados para la inicialización del estudiante los cuales son de 2 y 3 DOF.

Como panorama general (The Big Picture) un modelo de robot descrito en URDF consiste en un conjunto de elementos de eslabones (links) y de un conjunto de elementos de unión (joints) que conectan los eslabones en serie.

```
<robot name="2dof">
  <link name="left_arm">
    <inertial ixx="100" ixy="0" ixz="0" iyy="100" iyz="0" izz="100"/>
  </link>
  <joint name="arm_1_to_arm_base" type="revolute">
    <origin xyz="0 0 1" rpy="0 0 3.1416"/>
  </joint>
</robot>
```

Se modelaron robots básicos describiendo las propiedades visuales, de colisión de los eslabones y propiedades cinemáticas de los joints esto como respuesta de la naturaleza de este trabajo en analizar la cinemática del robot. Los otros modelos de robots utilizados son los provistos por el repositorio de ROS-Industrial [9] en el cual los modelos descritos en URDF poseen archivos CAD y están definidos todas las propiedades necesarias de cinemática y dinámica para hacer corresponder el modelo

virtual al modelo físico del robot que se encontrara en una fábrica de producción, aunque se encuentren muchos datos dentro del archivo, el algoritmo utilizado en este trabajo es capaz de detectar las configuradas como interés en este trabajo que es la cinemática y obviar la parte dinámica.

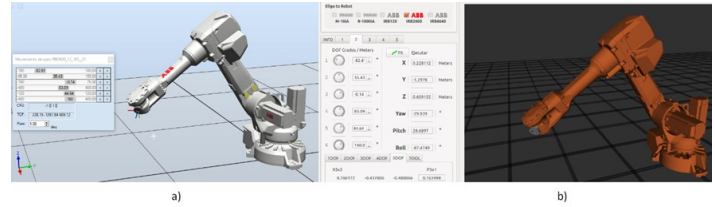


Fig. 7 Pose 1 del Robot ABB IRB2600 a) Robot Studio b) Visualizador Plugin RVIZ

Para validar los paquetes de ROS Industrial se usará el Software oficial de ABB conocido como Robot Studio y el uso de RViz para comparar los modelos de los robots, en este caso el robot a someter a prueba es el modelo IRB2600 a 2 Posiciones aleatorias y comprobar en el visualizador RViz y Robot Studio los mismos valores para el efector final (TCP) y los Joints Propios de cada Robot. Los ángulos de los Joints y su diferencia son mostrados en la Tabla 1, Así como sus poses en la Fig. 7.

Tabla 1: Validación de Modelo ABB IRB260

| Posicionde Robot en ROS Contra Robot en ABB Robot Studio | | | | | |
|--|--------------------|-------------|-------------------------|----------------------------------|---------------------------|
| Joints | ROS y Robot Studio | Coordenadas | Posición del TCP en ROS | Posición del TCP en Robot Studio | Error de Posición del TCP |
| | Angulo en Grados | | Distancia en Metros | Distancia en Metros | Error en Metros |
| Pose 1 Robot IRB2600 | | | | | |
| 1 | -82.61 | X | 0.228112 | 0.22815 | 0.000038 |
| 2 | 35.43 | | | | |
| 3 | -0.14 | Y | -1.2978 | -1.29784 | 0.00004 |
| 4 | 83.09 | | | | |
| 5 | 44.64 | Z | 0.609155 | 0.60912 | 0.000035 |
| 6 | 160.0 | | | | |

C. RBDL para Calculo de Cinemática.

La teoría clásica de robótica especifica como prioridad el cálculo y análisis cinemático del robot el cual corresponde a la cinemática directa y cinemática inversa siendo esta la de mayor complejidad debido a que dada la posición y orientación del efector final del manipulador, se calcula todos los conjuntos posibles de ángulos articulares que podrían utilizarse para obtener esta posición y orientación dadas, los cuales todos estos conjuntos posibles implican de cálculos que de acuerdo el número de Joints se vuelve más complejo y no es practico realizarlos sin utilizar algoritmos de computación.

Anteriormente se mencionó del formato de archivo de URDF que describe las propiedades de un robot en un formato virtual, este formato es utilizado por el algoritmo de control en C++ RBDL [5], Ver Fig. 4 El cual extrae las propiedades del robot y

las organiza en una matriz de Denavith-Hartenberg para proceder generar el modelo visual del robot y ser mostrado en el nodo de RVIZ dentro de la GUI, Ver Fig. 4 además que permite mostrar los cálculos de cinemática de los vectores de rotación y posición, para la cinemática inversa se ocupa el Algoritmo Recursivo de Newton-Euler [7] el cual permite calcular la posición, velocidad y aceleración del cuerpo o modelo virtual analizado, siendo de interés el de posición para la cinemática.

Estos datos son generados por funciones ya definidas en la librería RBDL los cuales se ocupan para mostrarlos en la interfaz gráfica de usuario.

D. Diseño e implementación de GUI.

La interfaz principal está diseñada bajo el lenguaje QML, de QT Creator, creado para desarrollar aplicaciones dinámicas. Para el diseño de la interfaz se tomaron en cuenta los siguientes requerimientos, que debe garantizarla interfaz, tal como se especifica en la Fig. 8

- ✓ Centralización de todos los recursos del LVR en una interfaz principal.
- ✓ Visualización de Recursos bibliográficos y visuales (Videos).
- ✓ Manejo de Procesos (Thread) para ejecutar los nodos de ROS sin necesidad de usar línea de comandos por terminal Linux.

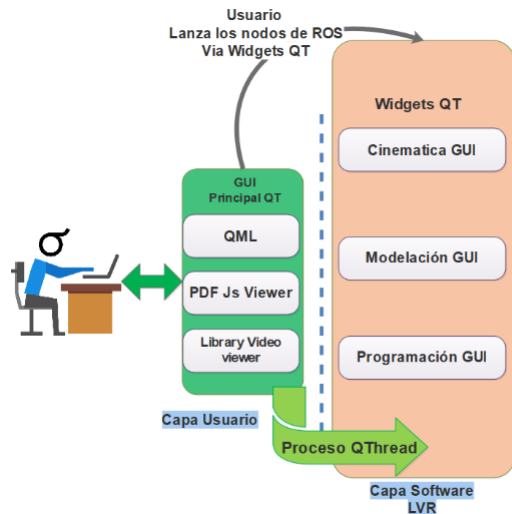


Fig. 8 Esquema de software del LVR.

La interfaz principal (Ver figura 9) fue desarrollada utilizando QML, lenguaje de programación de Qt-Creator muy parecido a un archivo HTML. Con el lenguaje QML se define un encabezado principal (header) y un cuerpo que contiene los diferentes widgets, botones, barras deslizantes, cajas de texto, etc., de la interfaz.



Fig. 9 Pantalla principal del LVR.

Esta interfaz tiene la posibilidad de lanzar los nodos de ROS a como se muestra en la Fig. 8. Usando un proceso QThread, el cual es un proceso hijo de la interfaz los cuales son definidos a continuación.

E. Nodos de ROS.

GUI de Cinemática: en este laboratorio virtual hay modelos elaborados por el desarrollador del LVR y modelos de fabricantes de robots, todos modelados en URDF, los cuales son cargados de forma transparente siguiendo diferentes algoritmos en C++ utilizados para actualizar los plugins de visualización del robot, espacio de trabajo y cálculos de la cinemática para cada robot como se observa en la Fig. 10.

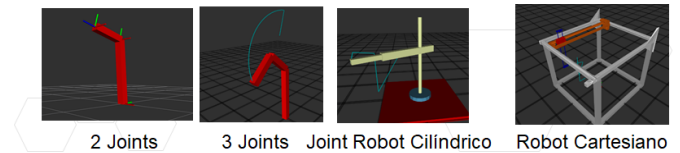


Fig. 10: Modelos de Robots.

GUI de Modelación de Robots: El resultado del desarrollo de esta interfaz es mostrado en la Fig. 11 en la cual podemos apreciar que se cuenta con una ventana de edición del URDF con autocompletado y resaltado de las palabras reservadas (highlight), un conjunto de widgets mediante los cuales el robot se mueve a la posición correspondiente de acuerdo con los ángulos de los joints ingresados por el estudiante, el panel de visualización del robot en 3D y un conjunto de herramientas para control de la ventana. Esta ventana es un nodo de ROS que es lanzado desde la aplicación principal tal y como fue definido en la sección D.

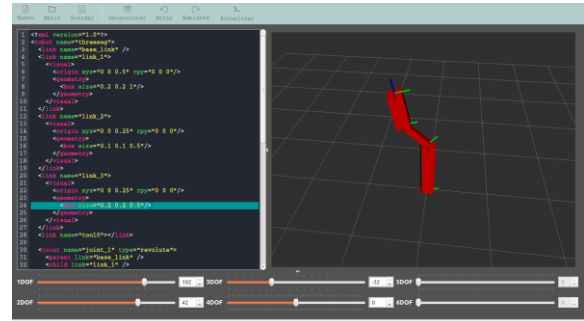


Fig. 11: Modelacion de Robots..

Programación de Robots: El módulo de programación, para verificar su efectividad, requiere de la creación de una celda de trabajo virtual, así como de una interfaz para la interpretación de las instrucciones requeridas para programar la tarea del robot.

Para la Implementación de la Celda de Trabajo con Gazebo en el laboratorio cuenta, entre otras cosas, con un robot el cual, según su estructura, define un espacio de trabajo en el cual interactuará con diferentes objetos. En general, podemos decir que el robot junto con los otros elementos, requeridos para realizar una tarea dada, conforman una celda de trabajo. Para simular la celda de trabajo se utilizó el simulador Gazebo. La descripción del mundo en Gazebo se realiza usando SDF mediante el cual se define la física, los plugins y el modelo del robot con sus parámetros físicos.

Un aspecto fundamental es el sistema para controlar el modelo del robot que será simulado. Dicho sistema de control se define utilizando el plugin **ros_control**. En el modelo del robot debe indicarse el tipo de actuador de cada joint lo cual es definido con la etiqueta **transmission** en el archivo SDF Como se observa en la Fig. 12.

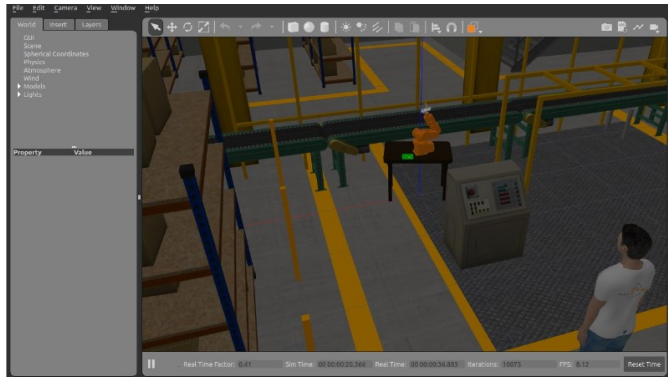


Fig. 12: Simulación de Robots en Gazebo.

La interfaz de programación fue desarrollada utilizando widgets de Qt Creator y librerías de control de texto plano mediante las cuales el usuario define las instrucciones las cuales deben ser procesadas y traducidos a comandos para el controlador PID de cada joint del robot. Por ejemplo, el comando

Move_J1 G 50 W 800

Script LVR

indica una rotación del Joint 1 (Move_J1) de 50 grados positivos (G 50) y con un tiempo de ejecución de 800ms (W 800). La interfaz de programación acepta un máximo de 6 Joints (Move_J1, Move_J2, Move_J3... Move_J6.) y cada comando en grados es un SetPoint para el PID del Plugin **ros_control**.

III. RESULTADOS

En la Fig. 13 se muestra la interfaz de cinemática donde encuentran los widgets mediante los cuales el estudiante ingresa los valores de los ángulos de cada joint y a la derecha los resultados obtenidos para las coordenadas XYZ (posición del

efector final). En el centro el modelo del robot utilizado, MH5 de Motoman, ubicado en la posición correspondiente con los ángulos ingresados por el estudiante. También se muestran los ángulos de Euler RPY, demostrando así que con la interfaz de matemática y cinemática es posible analizar diferentes tipos de robots industriales utilizando la librería RBDL.

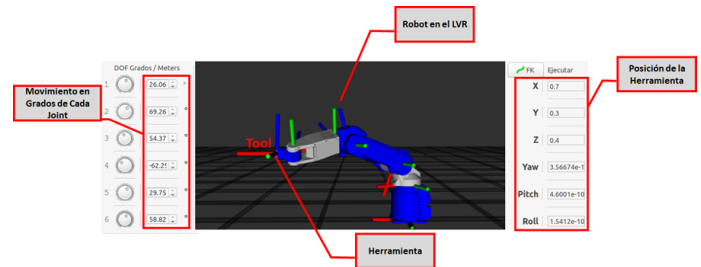


Fig. 13: Comprobación de Cinemática de robot.

Al igual que en el caso de la cinemática directa, para abordar el problema de la cinemática inversa se utiliza la librería RBDL para obtener los ángulos en que deben posicionarse los diferentes joints del robot para unas coordenadas XYZ dadas del efector final como se describe en la Fig. 14.

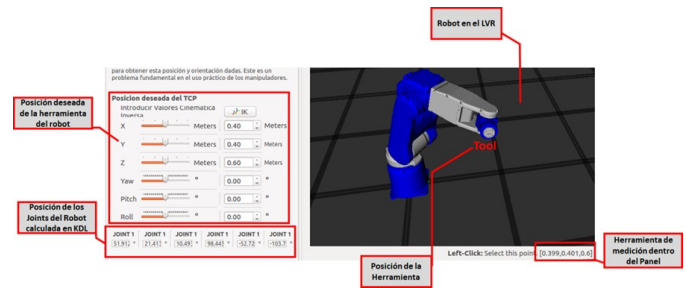


Fig. 14: Visualización del robot y su cinemática inversa respectiva.

En la Fig. 15 se puede observar el resultado cuando se alteran las dimensiones de un robot cilíndrico en el archivo URDF correspondiente el cual ha sido cargado previamente en la interfaz de visualización.

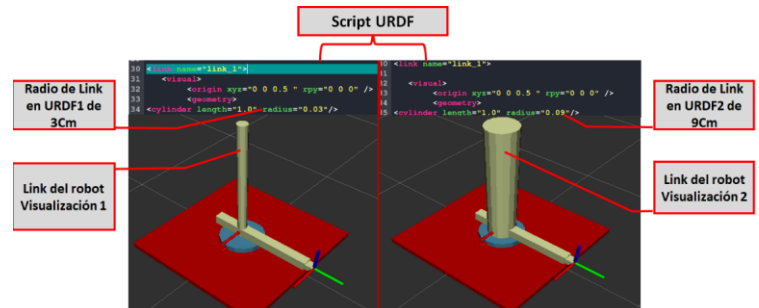


Fig.15: Ejemplo de la edición de los parámetros de un robot con un script URDF.

Cada Joint del robot posee un t pico dentro del plugin `ros_control` que describe la posici n en grados, esta informaci n es usada para graficar la respuesta del controlador PID.

En la Fig. 16 se pueden apreciar las gr ficas del movimiento de los joints en respuesta al control aplicado al robot ABB IRB120 por medio de la interfaz de programaci n. A la izquierda se aprecia las instrucciones para el movimiento de cada joint. Tambi n se pueden apreciar los widgets asociados con el controlador PID.

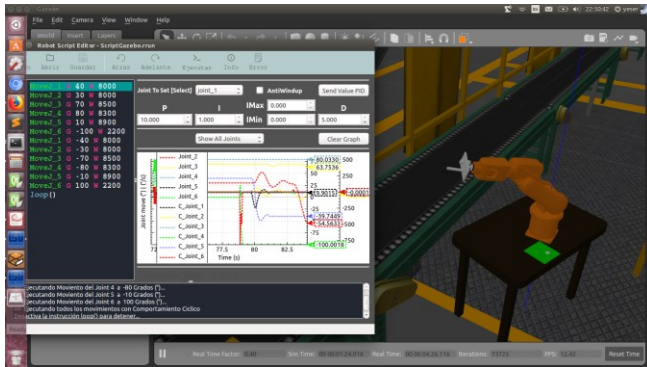


Fig.16: Robot ABB IRB120 simulado en Gazebo y controlado con Interfaz de Programaci n

IV. CONCLUSI N

En este trabajo se ha presentado la elaboraci n de modelos de robots y el an lisis de estos mismos en una interfaz gr fica de usuario principal y tres sub interfaces de ROS para an lisis de datos para un usuario/estudiante que tenga inter s de la comprensi n de los fundamentos de rob tica industrial cabe destacar que los recursos de software cumplieron con las expectativas de 1) Cargar un modelo tridimensional de un Robot Industrial o un modelo b sico ajeno a cualquier l nea de Robots como ABB, Kuka, Motoman etc. 2) Se incorporo la librer a de computo que provey  los c lculos relacionados a la cinem tica. 3) Se incorporo a ROS como administrador de la informaci n y sus librer as como RVIZ para visualizaci n de los modelos. 4) Creaci n de la interfaz de usuario indispensable para la transparencia y f cil uso de esta herramienta de software desarrollada. Por lo tanto, en este LVR se puede programar usando la misma interfaz de programaci n diferentes modelos simulados en Gazebo, siendo digno de recalcar que al usar el plugin de `ros_control` se puede conectar a un robot real sin ning n problema.

Para probar este alcance y que en la Universidad Nacional de ingenier a siga usando ROS se conecto el LVR a un robot F sico de 5 grados de libertad y se program  por medio de la interfaz de programaci n el robot simulado y el f sico. Se le programo una rutina que alcanzara un objeto y lo depositara en un recipiente para tal actividad se le programo 10 comandos incluyendo la activaci n del comando `loop()` que permite dejar el robot en un comportamiento c clico ejecutando la rutina. (Ver Fig. 17)

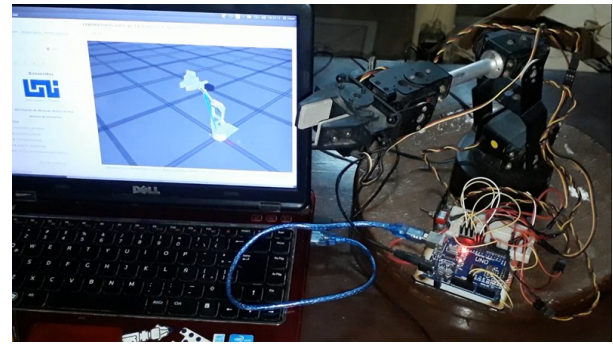


Fig. 17 Aplicaci n del LVR en un Robot F sico.

Aunque no fueron explotados totalmente los recursos de software en este primer prototipo del LVR se lograron explotar los fundamentos de rob tica industrial esto deja abierto el sendero para seguir incorpor ndole m s opciones como el an lisis de la din mica y planeaci n de movimientos repetitivos en un robot industrial

REFERENCIAS

- [1] J. Rodr guez, "Nicaragua en el radar de la Rob tica Educativa" La Prensa, [En l nea] Disponible en: <https://www.laprensa.com.ni/2017/08/23/nacionales/2284236-nicaragua-en-el-radar-de-la-robotica-educativa>
- [2] M. Climent, "As  moderniza EEUU su industria" El Mundo, [En l nea] Disponible en: <http://www.elmundo.es/economia/2015/03/13/5502ae7422601d814e8b4572.html>
- [3] Core components, ROS. [En l nea]. Disponible en: <http://www.ros.org/core-components/>
- [4] urdf, ROS Wiki. [En l nea]. Disponible en: <http://wiki.ros.org/urdf>
- [5] M. Felis. (2013, Agosto) Rigid Body Dynamics Library. [En l nea]. Disponible en: <http://rbdl.bitbucket.org/>
- [6] Qt Creator/es [En L nea] Disponible en: https://wiki.qt.io/Qt_Creator/es
- [7] Felis, Martin L. "RBDL: an efficient rigid-body dynamics library using recursive algorithms." *Autonomous Robots* 41.2 (2017): 495-511.
- [8] Timm, Alan, *inmoov_ros*, (2016), GitHub repository, [En L nea] Disponible en: https://github.com/alansrobotlab/inmoov_ros
- [9] ROS-I, [En L nea] Disponible en: <https://github.com/rosindustrial/>
- [10] Moody, J. A. O., Alonso, R. E. S., Barbosa, J. J. G., & Morales, G. R. (2016). Virtual laboratories for training in industrial robotics. *IEEE Latin America Transactions*, 14(2), 665-672.



Yaser A. Morales C. Estudiante de Ingenier a en Electr nica en Universidad Nacional de Ingenier a (UNI Nicaragua). Coordinador (Chair) del cap tulo estudiantil UNI, Robotics and Automation Society. Egresado de la carrera, con aspiraciones para una maestr a en Rob tica en el exterior.