**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**

PROTOCOLO DE TRABAJO MONOGRÁFICO

SISTEMA OPERATIVO PARA ROBOTS(ROS): APLICACIÓN EN EL DESARROLLO DE UN LABORATORIO VIRTUAL PARA EL ESTUDIO DE LOS FUNDAMENTOS DE LA ROBÓTICA INDUSTRIAL.

Presentado por: Br. Yeser Alfredo Morales Calero

Tutor: MSc. Alejandro Alberto Méndez Talavera

XXXXX 00, 2018

**Índice de Contenido\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

[INTRODUCCIÓN………………………………………………….………….….…………4](#Introduccion)

[ANTECEDENTES………………………………………………………….….…………...6](#Antecedentes)

[JUSTIFICACIÓN…………………………………………………….………….….……..10](#Justificacion)

1. [OBJETIVOS…………………………………………………….…………….……….12](#Objetivos)
   1. [Objetivo general…………………………………….………………….…..…….12](#ObjetivoGeneral)
   2. [Objetivos específicos……………………………………….….…….….………12](#ObjetivosEsp)
2. [MARCO TEÓRICO……………………………………..……….……………...…….13](#MarcoTeor)
   1. [Tipos de Laboratorios……………………………………...…………….…..….13](#TiposDeLAB)
      1. [Laboratorios Físico…………………..…………………………….…..….13](#LABFisico)
      2. [Laboratorios Virtual………………………………….…………….…..….14](#LABVirtual)
      3. [Laboratorio Remoto online………………………………….…….…..….14](#LABRemoto)
      4. [Realidad Virtual 3D………………………………………….…….…..….14](#LABVirtual3D)
   2. [Modelo de utilización del laboratorio virtual para robótica industrial………16](#ModeloLAB)
   3. [Sistema operativo para robot (ROS)……………….…………………………..17](#ROS)
      1. [Conceptos básicos de ROS…………………………...…………………18](#ConceptosROS)
      2. [Lenguajes de programación soportados…………………………….….20](#LanguageC)
      3. [Aplicaciones y librerías……………………………………………….…..21](#AppROS)
      4. [Herramientas de ROS…………………………………………………....21](#HerramROS)
   4. [Interfaz gráfica de usuario (GUI) ……………………………………………...21](#GUI)
      1. [Interacción dentro la GUI…………………………………………………22](#GUIInteraccion)
      2. [Visualización de datos……………………………………………………22](#GUIVisualizacion)
         1. [Visualización 3D de robots……………………………………….22](#GUIVisualizacion3D)
         2. [Gráficos de dispersión XY………………………………………..22](#GUIVisualizacionXY)
      3. [Programación de Rutinas de robots…………………………………….23](#ProgrammingRobots)
      4. [Información de los fundamentos de Robótica Industrial………………23](#InfoGUI)
   5. [Modelos de robots en el LVR…………………………………………………...24](#ModelosLVR)
      1. [Modelado de brazos robóticos de diferentes DOF……………..…..…24](#ModelosbrazosDOF)
   6. [Entornos de simulación de robots……......…..........................………………25](#EntornosSimu)
      1. [Software de compañías basadas en CAD ………………….………….26](#EntornosCAD)
      2. [Software proveído por los fabricantes de robots…………….….……..26](#EntornosFab)
      3. [Softwares de programación de varios fabricantes…………………….26](#EntornosvariousFab)
      4. [Software Simuladores de Robots de propósito general………………27](#EntornosGNRL)
         1. [Simuladores de propósito general comerciales………………..27](#EntornosGNRLComer)
         2. [Simuladores de propósito general de uso libre………………...28](#EntornosGNRLOpen)
   7. [Modelación Cinemática y Dinámica…….…………………………………......29](#KinemDyn)
      1. [Cinemática Directa e Inversa ………………………...………..………..29](#FKIK)
      2. [Espacio de trabajo del robot……………………………………………..29](#Workspace)
   8. [Control de movimiento en el entorno…………………………….…………….30](#CTRLMOV)
      1. [PID loop……………………………...……………………….……………30](#PIDLoop)
   9. [Hardware externo………………………………………………..…………….…30](#HARDext)
      1. [Joystick control…………………………………………………………....31](#Joystick)
   10. [Manual de uso del laboratorio virtual……………………….…………..31](#Manual)
3. [HIPÓTESIS Y VARIABLES……………………………………….…………………32](#Hipot)
4. [DISEÑO METODOLÓGICO…………………………………………………………33](#DesignMet)
5. [CRONOGRAMA DE EJECUCIÓN………………………………….………………36](#Cronograma)
6. [BIBLIOGRAFÍA…………………………………………………………..……………37](#Biblio)

[ANEXOS……………………………………………………………………..…………….42](#anexos)

**Introducción**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

La robótica es uno de los campos de la tecnología que se desarrolla velozmente y encuentra cada día nuevas áreas de aplicación de tal forma que la clasificación de los robots debe ser ajustada constantemente. En la actualidad encontramos aplicaciones de la robótica en el campo personal y doméstico, servicios profesionales, investigación y desarrollo. Unos de los primeros campos donde los robots entraron en acción fue la industria en la cual destaca el uso de los mismos en el sector automotriz (soldadura y/o manejo de materiales). A pesar de que el uso de los robots en la industria data de la década de los 60s, en Nicaragua el uso de los mismos es relativamente bajo. El hecho mencionado tiene un impacto negativo en el desarrollo de la industria nicaragüense. Son pocas las industrias que poseen robots industriales y, de igual forma, son pocas las instituciones de educación superior que cuentan con la infraestructura adecuada, y el recurso humano, para la enseñanza e investigación de la robótica. Una de las razones es que un laboratorio físico, para experimentar en los diferentes aspectos asociados con los robots industriales, requiere de una inversión relativamente alta dado que es necesario garantizar el espacio físico, adquirir los robots y su mantenimiento.

La falta de infraestructura para el desarrollo e investigación de la robótica ha sido experimentada en la mayoría de las instituciones de educación superior latinoamericana muchas de las cuales han recurrido a la utilización laboratorios virtuales, diseñados e implementados por dichas instituciones o adquiridos en el mercado internacional. MATLAB, software ampliamente utilizado en el campo de las ingenierías, cuenta con un toolbox para robótica mediante el cual se puede modelar y simular el comportamiento de un manipulador, sin embargo, el mismo tiene limitaciones. Simscape Multibody (antes llamado SimMechanics), de MATLAB y Simulink, permite la simulación de sistemas mecánicos en 3D, tales como robots. El software mencionado requiere de una licencia y el costo de la misma aumenta en dependencia del número de usuarios. También se puede utilizar el software de programación de robots, versión evaluación, de algunos fabricantes, tales como Robot Studio de ABB. Dicho software, además de limitaciones temporales, solo puede ser utilizado en los robots fabricados por la empresa. Para trabajar con diferentes robots, es necesario adquirir y aprender los lenguajes de programación de cada uno de ellos.

En la actualidad se busca estandarizar el trabajo con los robots y una de las herramientas disponibles para tal fin es el middleware “Sistema Operativo para Robot (ROS, por sus siglas en inglés)” el cual se ha convertido en el estándar de facto a nivel industrial tanto así que ya existe una versión denominada ROS Industrial. ROS es un middleware, open-source, para el desarrollo a gran escala de complejos sistemas robóticos. Se trata de una colección de herramientas, librerías y convenciones que tienen como objetivo simplificar la tarea de crear un comportamiento complejo y robusto en una amplia variedad de plataformas robóticas.

En este documento proponemos, como trabajo de monografía, dar a conocer el middleware ROS y mostrar sus posibilidades en el campo de la robótica mediante el desarrollo de un laboratorio virtual de robótica (LVR) para el estudio de los fundamentos de los manipuladores industriales tipo serie. ROS permite la programación de robots de diferentes fabricantes siempre y cuando se cuenta con las librerías apropiadas las cuales aumentan cada día dada que ROS es una comunidad y permite la reutilización de software. Para el desarrollo del laboratorio virtual, además de ROS, se utilizarán herramientas de simulación tales como GAZEBO, visualizadores como RVIZ y RQT. La integración de los diferentes nodos será realizada mediante el lenguaje de programación C++, la modelación de los robots será desarrollado utilizando el formato de descripción unificado de robots (URDF, por sus siglas en inglés), y para la creación de ambientes y escenarios utilizaremos el formato de descripción de la simulación (SDF, por sus siglas en inglés).

El laboratorio virtual contará con los recursos necesarios para que los estudiantes estudien y realicen prácticas relacionadas a la morfología de los robots, herramientas matemáticas necesarias para el estudio de los robots, cinemática de los robots y programación de manipuladores industriales.

El laboratorio virtual contará con un manual de usuario en el cual será presentada la información de los recursos disponibles en el LVR de forma tal que los docentes puedan diseñar prácticas que permitan a los estudiantes afianzar los fundamentos de la robótica industrial. Por ejemplo, el profesor podrá elaborar una práctica de laboratorio, a partir de los modelos de robots presentes en el LVR, para que los estudiantes apliquen el procedimiento de Denavit y Hartenberg para determinar la matriz de transformación homogénea.

**Antecedentes**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Con el objetivo de identificar trabajos relacionados con el propuesto en este protocolo de trabajo monográfico, se buscó información en el centro de documentación de la facultad de electrotecnia y computación, así como en el repositorio del consejo nacional de universidades, el cual presenta información sobre trabajos de investigación realizados en las universidades miembros de dicho organismo. En cuanto al Sistema Operativo para Robot (ROS), cuya primera versión fue lanzada en el 2007, dichos lugares no cuentan con información sobre trabajos en los cuales se haya utilizado dicho sistema. De igual forma, documentación de laboratorios virtuales, desarrollados en el seno de las universidades del CNU, es poca y destaca el trabajo monográfico “Laboratorio virtual para el desarrollo de prácticas en la disciplina de Control automático y Automatización industrial en la carrera de Ingeniería Electrónica” desarrollado por Pérez et al (2016). El sistema desarrollado cuenta con cuatro plantas virtuales el comportamiento de las cuales puede ser controlado mediante programas escritos por los estudiantes utilizando el TIA Portal de Siemens.

La promoción y uso de ROS muestra algunos indicios en Centroamérica, destacando el curso “Robótica,” desarrollado por la Universidad de Costa Rica, en el cual una de las unidades es dedicada al sistema operativo para robot (Ramírez, 2018). En el blog COSTARICAMAKER (2018), utilizan ROS para promover la robótica y presentan la utilización de este para el reconocimiento de patrones usando una cámara web. En un trabajo en conjunto con la Universidad de Sao Paulo, Brasil, se realizó en el Tecnológico de Costa Rica, un trabajo monográfico en el cual se utilizó ROS para la obtención de datos de sensores 3D para vehículos autónomos. El trabajo fue desarrollado por Valverde (2015).

En la Universidad Tecnológica de Panamá se elaboró el diseño e implementación de un sistema de control para un Mini carro tele operado usando ROS. El trabajo lleva por nombre MODELADO Y CONTROL DEL ROBOT MÓVIL ROBOTNIK SUMMIT XL desarrollador por Rosas et al (2013).

A diferencia de Centroamérica, el sistema operativo para robot ha tenido una amplia aceptación en américa del sur siendo el mismo utilizado en cursos robótica, así como en trabajos de tesis en las universidades de mayor prestigio. Los países de América del sur con mayor grado de adopción de ROS son Chile, Brasil, Ecuador, Argentina y Colombia.

En América del norte, México presenta un movimiento muy importante al desarrollar Robots utilizando ROS. Se organizan competencias, poseen trabajos académicos tanto de desarrollo como de investigación. Es destacable el trabajo de Estudiantes del Instituto Tecnológico y de Estudios Superiores de Monterrey que participarán en la competición para estudiantes University Rover Challenge (URC) (Eagle X 2018). Es tan importante el trabajo desarrollado en México que han ganado premios a nivel internacional tal como el 4to lugar obtenido en la competencia RoboCup del año 2017 con el robot humanoide de servicio JUSTINA el cual también está construido bajo ROS para controlar cada una de las funcionalidades de las cuales fue provista (Justina, 2017).

Es importante, dada el impacto que ha tenido en el desarrollo de la robótica, conocer el sistema operativo para robot, ROS en general y ROS industrial en particular, sus aplicaciones y tendencias.

En esta propuesta de trabajo monográfico se plantea mostrar la importancia de ROS elaborando los medios requeridos para tal fin, así como el desarrollo de un laboratorio virtual, teniendo a ROS como soporte, para el estudio de los fundamentos de la robótica industrial. Considerando lo limitado de los recursos disponibles para el estudio de la robótica en nuestro país, el laboratorio virtual sería de mucha ayuda en el proceso de formación de los estudiantes.

El desarrollo de laboratorios virtuales es una alternativa que ha sido asumida a nivel mundial y es así que encontramos instituciones académicas que los diseñan e implementan para uso propio o que los adquieren de empresas que se dedican al desarrollo de estos con fines comerciales como es el caso de la empresa TheConstructSim (Téllez, 2016), que proveen servicios para simular robots. Su estrategia de inmersión en el campo de la educación es garantizar todos los recursos vía web (Simulaciones de Robots en la Nube (Cloud)) para enseñar robótica y aunque esta empresa usa librerías nativas de software libre de ROS, el uso de sus plataformas tiene un costo para el paquete principiante de €15 y una cuota mensual de €39.97.

En la Universidad Nacional de Ingeniera, en Nicaragua, Pérez et al (2016) desarrollaron el “Laboratorio virtual para el desarrollo de prácticas en la disciplina de Control automático y Automatización industrial en la carrera de Ingeniería Electrónica” cuya finalidad es verificar la efectividad de programas para PLC escritos por los estudiantes. El sistema desarrollado cuenta con cuatro plantas virtuales las cuales contienen elementos comúnmente encontrados en una planta real tales como sensores, bombas, calentadores, etc., teniendo el docente la posibilidad de decidir el comportamiento deseado de la planta y solicitar a los estudiantes el programa para lograrlo. Al correr el programa, se puede apreciar la animación de las partes correspondientes tales como activaciones de sensores, cambios de nivel en un tanque, activación de un calentador, etc., lo cual brinda al estudiante información visual que le permite verificar la efectividad de su programa.

En Latinoamérica se ha realizado un trabajo importante para desarrollar laboratorios virtuales en diferentes campos de la ingeniería. En cuba, se propuso un laboratorio para procesos de control e instrumentación (Ayala, et al, 2016). También destaca el trabajo desarrollado en Guatemala para ejecución de prácticas de circuitos eléctricos por Juárez (2008) quien establece que “se considera que un laboratorio virtual será de gran utilidad para los estudiantes y los docentes de la universidad, porque el objetivo es facilitar la enseñanza y la comprensión de los conceptos teóricos vistos en clase, mediante las tecnologías de la información y la comunicación”, Diseño de un compilador para programación en MOODLE (Merino, 2015) desarrollado en Ecuador, un laboratorio virtual llamado SISMILAB para ingeniería sísmica en Colombia (Guerrero, et al, 2014). Todas estas propuestas de laboratorios virtuales poseen un propósito compartido de acercar al estudiante a la experimentación virtual.

En el campo de la robótica, asunto que nos ocupa en la presente propuesta de trabajo monográfico, México ha incursionado en la enseñanza de la robótica utilizando laboratorios virtuales. (Prieto, et al, 2010; Ortega, et al, 2016a), plantean que la enseñanza de la robótica a menudo requiere de costosos laboratorios que están más allá del alcance de la mayoría de las universidades públicas. Son pocos los trabajos de laboratorios virtuales de robótica que tengan en consideración la importancia de modelar un ambiente de trabajo virtual robótico lo más real posible involucrando gravedad, interacción entre cuerpos, velocidad de movimiento, dinámica de los cuerpos etc. Ortega, et al (2016b) analizan estas situaciones y proponen en su trabajo la integración de motores de física virtual tales como PhysX y el entorno de desarrollo Unity 3D para el desarrollo de una solución enfocada a un laboratorio virtual completo con tres niveles de desarrollo, Nivel de Ambientes 3D usando herramientas de CAD y modelado 3D, Nivel de instrumentación y control con el análisis matemático del modelo del sistema para introducir etapas de control PID en cada eslabón del Robot con configuración de operación paralela (Robot Delta) y conseguir una aproximación de respuesta del diseño virtual a un diseño real, además del Nivel de HMI que involucra al estudiante con la práctica permitiendo ajustar parámetros físicos y dinámicos del modelo virtual tales como la masa de cada elemento del robot, características mecánicas y eléctricas de los motores virtuales, parámetros del controlador PID, etc.

Laboratorios Virtuales remotos, tales como el Laboratorio Remoto usando el robot PR2 propuesto por Pitzer et al (2012), permiten a un usuario observar, desde una página WEB con GUI’s y modelos virtuales, el comportamiento bajo cámaras WEB de un robot físico al experimentar con este. Otro trabajo bajo el entorno ROS es el desarrollado por Casañ, et al (2015) quienes proponen el uso de las funcionalidades de ROS, pero agregando una plusvalía de carácter open-source como lo es la plataforma Moodle, para centralizar el acceso de los estudiantes del laboratorio remoto, todo esto bajo un mismo servidor donde se ejecutan los nodos de ROS y la base de datos de Moodle que son accedidos desde una página WEB.

Los resultados de la revisión de literatura muestran la importancia que tienen tanto ROS en el fomento del desarrollo de la robótica como los laboratorios virtuales para mejorar la formación de los estudiantes relacionados con dicho campo. Es importante que en Nicaragua utilicemos herramientas como ROS para el estudio de la robótica y es por eso que la propuesta de trabajo monográfico aquí presentada, con pocos trabajos relacionados similares en la región, está enfocada en dar a conocer las principales características de ROS, sus alcances y tendencias así como mostrar su utilidad en el mundo académico mediante el desarrollo de un laboratorio virtual que contará con modelos tridimensionales de robots cuyo comportamiento podrá ser manipulado desde una interface gráfica (GUI) diseñada para tal fin o mediante programas escritos por los estudiantes. El middleware ROS servirá como elemento centralizador de los diferentes nodos del sistema. A diferencia de otros laboratorios virtuales que están destinados a práctica específicas, en el laboratorio virtual que se propone el docente encontrará los elementos que generalmente se encuentran en un laboratorio físico y podrá decidir que experimentos realizar, elaborando las guías de laboratorio necesarias para garantizar los objetivos establecidos.

**Justificación**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

El campo de la robótica ha tenido, principalmente en las últimas décadas, un desarrollo vertiginoso y está impactando de forma positiva, principalmente en países desarrollados, en áreas como la medicina, la industria, y la investigación, entre otras. En Nicaragua, país en vías de desarrollo, el uso de la robótica es relativamente bajo, sin embargo, dicha tecnología tendrá mayor presencia en el futuro cercano y por consiguiente es necesario contar con profesionales con las competencias requeridas para realizar efectiva y eficientemente las diferentes tareas asociadas con la aplicación, el desarrollo y la investigación de la robótica.

La formación de profesionales en el campo de la robótica requiere de recursos humanos e infraestructura apropiada para tal fin. En Nicaragua pocas instituciones de educación superior cuentan con recursos básicos mínimos para la formación de los estudiantes en el campo de la robótica, siendo uno de los principales obstáculos para tener mejores condiciones la imposibilidad de garantizar la alta inversión requerida para la adquisición de los robots, software, accesorios asociados, instalación y mantenimiento. Compañías como FESTO ofrecen Kits denominados modulares, para prácticas de un proceso automatizado utilizando un robot Mitsubishi, que cuesta alrededor de 41,500 euros.

En el campo industrial, muchos fabricantes ofrecen manipuladores industriales los cuales tienen precios altos y, además, cada uno ofrece su propio software para la programación de los robots (Owen, 2016). Lo anterior significa que los estudiantes o especialistas en robótica industrial tendrían que aprender a utilizar el software correspondiente al manipulador en turno lo cual es imposible lograr si no se cuenta con la infraestructura adecuada. Se han realizado intentos para estandarizar lo relacionado a la programación de los robots, hacerla independiente del fabricante, y uno de los resultados más destacado en la última década es el middleware “Sistema Operativo para Robot” (ROS, por sus siglas en inglés). En la actualidad es considerado el estándar de facto para aplicación, desarrollo e investigación de la robótica y en el campo industrial se ha creado ROS-Industrial.

El desarrollo de un laboratorio virtual en robótica contribuiría en la mejora de las condiciones requeridas para el estudio de los fundamentos básicos de esta ya que brindaría a los estudiantes, y docentes, una herramienta que podría ser utilizada en cualquier lugar y en cualquier momento.

El laboratorio virtual sería desarrollado tomando como elemento principal el middleware ROS lo cual, entre otras cosas, permitiría a los estudiantes experimentar con modelos de robots de diferentes fabricantes tales como ABB, KUKA, FANUC, entre otros.

La inversión en el desarrollo e implementación del laboratorio sería muy inferior respecto a la requerida para tener un laboratorio físico ya que la mayoría de las herramientas utilizadas, tales como C++, ROS, GAZEBO, Qt creator, no requieren de licencia.

Los principales beneficiaros de los resultados de este proyecto serán los docentes y los estudiantes. Los primeros dispondrán de una herramienta a partir de las cuales podrán diseñar experimentos, con su guía apropiada, relacionados con los fundamentos básicos de los robots industriales. Los estudiantes, por su parte, contarán con una herramienta flexible, podrán utilizar en cualquier momento y en cualquier lugar, que les posibilitará verificar los fundamentos teóricos relacionados con los robots industriales contenidos en programas de asignatura relacionados o programar los robots para que realicen tareas básicas.

La sociedad nicaragüense sería también beneficiada ya que, producto del trabajo desarrollado por los estudiantes en el laboratorio virtual, los mismos contarán con una formación más sólida en dicho campo y podrán contribuir efectivamente en la utilización, desarrollo e investigación de la robótica en el país.

1. **Objetivos**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. **Objetivo general**

Desarrollar un laboratorio virtual para el estudio de los fundamentos de la robótica industrial utilizando, para su implementación, el sistema operativo para robots (ROS).

* 1. **Objetivos específicos**
* Identificar las principales características/recursos/estructura de ROS enfatizando aquellas aplicables directamente a la robótica industrial.
* Elaborar un documento donde se muestre el ABC de ROS.
* Identificar las herramientas de software requeridas, además de ROS, para el desarrollo del laboratorio virtual para el estudio de los fundamentos de la robótica industrial.
* Desarrollar la interfaz gráfica de usuario (GUI) del laboratorio virtual, según los requerimientos establecidos.
* Verificar la efectividad del laboratorio virtual mediante la realización de dos prácticas de laboratorio.
* Elaborar un manual de usuario del laboratorio virtual.

1. **Marco Teórico**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Las instituciones académicas, principalmente las de educación superior (IES), buscan constantemente mejorar la calidad de la educación y una forma de hacerlo es integrando nuevas técnicas y herramientas orientadas a mejorar los resultados del proceso enseñanza-aprendizaje. En la actualidad, en muchas IES, a la tradicional clase presencial se suman otras formas de enseñanza-aprendizaje las cuales se fundamentan en las TIC, permitiendo de esta manera nuevas formas de enseñar, aprender, generar y compartir conocimiento.

Un recurso de mucho impacto en la formación de los futuros profesionales y cuya presencia crece día a día en los procesos de enseñanza aprendizaje es el laboratorio virtual (LV), el cual puede presentar altos niveles de flexibilidad y la inversión es muy inferior a la requerida para instalar un laboratorio real.

Uno de los campos en el cual el desarrollo y utilización de un laboratorio virtual reviste mucha importancia, dada la inversión alta requerida para tener un laboratorio físico, es el de la robótica en general y en particular, la robótica industrial.

En la introducción del presente documento se estableció que el trabajo de monografía propuesto pretende dar a conocer el middleware ROS, su importancia en el desarrollo de la robótica industrial, y dar muestra de sus posibilidades mediante el diseño y construcción de un laboratorio virtual que sirva de base para el estudio de los fundamentos de los manipuladores industriales tipo serie.

En los siguientes apartados se presentan los elementos teóricos-tecnológicos requeridos para el desarrollo del laboratorio virtual.

* 1. **Tipos de laboratorios**

La experimentación de un fenómeno físico, o mediante la simulación, requiere de entornos de trabajos que reúnan los equipos o herramientas necesarias para el desarrollo efectivo de una práctica. En los ambientes académicos podemos encontrar, en la actualidad, diferentes tipos de laboratorios.

* + 1. **Laboratorio Físico**

Es ampliamente utilizado en universidades con modelos clásicos de enseñanza convirtiendo estos sitios de trabajo como el único sitio concebido para elaborar un experimento el cual involucra la presencia física tanto del tutor como del alumno. Cabe destacar que la interacción directa con los equipos apropiados de un laboratorio físico aporta una experiencia dificil de igualar debido a que los alumnos perciben los resultados de carácter palpable entrando en juego los cinco sentidos (Vista, tacto, audición, olfato e incluso, a veces el gusto) Calvo, (2009a).

* + 1. **Laboratorio Virtual**

Es una alternativa al laboratorio físico y utiliza recursos computacionales haciendo uso de modelos matemáticos y recursos de visualización como modelos CAD y animaciones gráficas. Se pretende, mediante de este tipo de laboratorio, que el estudiante pueda estudiar el comportamiento de la realidad mediante el uso de modelos, sencillos o complejos, de la misma. Los recursos de un laboratorio virtual pueden ser ampliados en el tiempo permitiendo la realización de prácticas asociadas a temas de mayor complejidad.

* + 1. **Laboratorio Remoto online**

Nacen bajo la necesidad de dar acceso a los estudiantes a un espacio de trabajo, de forma online, que combina los recursos de un laboratorio físico con recursos de software. El estudiante accede al laboratorio por medio de una página web pudiendo de esta forma controlar los recursos disponibles y llevar a cabo el experimento de interés. Su aprovechamiento está basado en la accesibilidad por parte de un usuario en horarios flexibles.

* + 1. **Realidad virtual 3D**

Es la integración de recursos de hardware y software para acercar aún más al usuario a la experimentación de una teoría estimulando los sentidos tanto de la vista al exponerlos a entornos virtuales 3D y audición bajo altavoces con sonido envolvente.

En la actualidad se puede encontrar diferentes recursos para la enseñanza de la robótica Industrial tales como kits de robótica industrial, laboratorios virtuales de robótica de categoría libre o de pago para ejecutar cierta práctica. El desarrollo de un laboratorio remoto requiere contar con acceso a un laboratorio físico y enlazarlo a un servidor para que el estudiante pueda acceder vía remota garantizándole una plataforma amigable y multiplataforma como pueden ser los navegadores WEB. Candelas, F. et al. (2004) analizan las ventajas del uso de un laboratorio remoto ROBOLAB, proyecto desarrollado por ellos mismos bajo el nombre de grupo de investigación AUROVA. El proyecto utiliza herramientas gráficas para modelado y visualización de objetos 3D de un brazo robótico que coincide con el mismo modelo de un robot físico, permitiendo al estudiante realizar sus practica al observar y configurar el modelo virtual y una vez alcanzado un nivel de aprendizaje aceptable, proceder a interactuar con el robot físico utilizando la misma plataforma web. Ellos concluyen que al utilizar el laboratorio Remoto ROBOLAB “*La mayoría de alumnos prefieren disponer de un laboratorio en la universidad dónde trabajar con la ayuda de los compañeros y el apoyo didáctico del profesor, pero también hay muchos alumnos que reciben con agrado la opción de un laboratorio virtual que les ofrezca unos horarios flexibles en los que realizar los experimentos”.* (Candelas, et al, 2004b).

En resumen, un laboratorio virtual es un elemento importante en el proceso enseñanza-aprendizaje ya que destacan, entre otras, las siguientes ventajas según Calvo, et al (2009a):

1. El estudiante se familiariza con el experimento evitando acudir al aula sin conocimiento previo.
2. Comparación del comportamiento de modelos matemáticos ante una simulación permitiendo extraer sus propias conclusiones de cierta práctica.
3. Manejo de herramientas informáticas contemporáneas para la formación integral de un estudiante.
4. Repetitividad de los experimentos realizados por el estudiante que permitirá reproducir cuantas veces desee hasta consolidar el conocimiento.
5. Disminución de riesgos y accidentes que pueden ocasionar una mala práctica o configuración de un equipo físico.
6. Multiplicidad de experimentos simultáneos realizados ya que cada estudiante podrá ejecutar la práctica indicada en su computador asignado, además de esta forma se favorecen los procesos colaborativos como el de “Lluvia de ideas” al opinar cada alumno sobre su percepción adquirida al ejecutar la práctica.

Considerando lo anterior, se decidió, para dar a conocer la importancia y posibilidades de ROS, desarrollar un laboratorio virtual el cual tendrá a ROS como elemento integrador. Es importante destacar la inversión baja relativa requerida para implementar el laboratorio virtual, la posibilidad de ampliación del mismo y la posibilidad que el mismo ofrece a los docentes para diseñar nuevas prácticas a partir de los recursos disponibles.

* 1. **Sistema operativo para robot (ROS)**

El sistema operativo para robot (ROS), elemento fundamental del presente trabajo, es un middleware ampliamente utilizado en el mundo creciente de la robótica. Originalmente fue desarrollado en el 2007 en los laboratorios de Inteligencia Artificial de Stanford y en el 2008, cedieron el derecho de desarrollo al instituto de investigación de robótica Willow Garage, donde la filosofía es proporcionar bibliotecas y herramientas libres para ayudar a los desarrolladores de software a crear aplicaciones para robots

En esta propuesta de trabajo monográfico, se promueve la importancia de ROS en el campo de la robótica y se hace mediante el diseño y construcción, utilizando los recursos proporcionados por este, de un Laboratorio Virtual de Robótica, en el cual será posible realizar experimentos sobre los fundamentos de la robótica industrial específicamente de los manipuladores industriales tipo serie.

Una característica importante de ROS es su código abierto (Open Source) lo que ha llevado al desarrollo, por muchos colaboradores a nivel mundial, de una amplia colección de herramientas, librerías y convenciones. Los recursos que ofrece ROS permiten el desarrollo, a gran escala, de sistemas robóticos complejos, tanto físicos como simulados.

Los desarrolladores de ROS han establecido un conjunto de convenciones y buenas prácticas en el desarrollo de software de forma tal que se fomente la reutilización de código para robots, trabajando sobre una arquitectura robótica totalmente sólida y funcional. De esta forma se garantiza, entre otras cosas, que el proyecto no sea fallido y que sus resultados satisfagan la calidad requerida en este tipo de trabajos.

* + 1. **Conceptos básicos de ROS**

El sistema operativo para robot (ROS) fue diseñado bajo una estructura distribuida y modular con el propósito de que los usuarios puedan usar los recursos requeridos según la aplicación a desarrollar. Es decir, el usuario puede seleccionar los recursos de software necesarios para implementar la solución.

En su operación, ROS presenta una red de ***procesos*** que se ejecutan simultáneamente en una estructura ***peer-to-peer*** donde cada nodo o elemento del sistema puede actuar al mismo tiempo como cliente y como servidor. Para una aplicación distribuida de ROS se requiere que los nodos peer to peer estén sincronizados bajo un middleware el cual genera un orden temporal para los eventos generados en el sistema, además de utilizar la comunicación XML/RPC que define la información de registro de cada nodo al ROSMaster, una vez registrado los nodos se procede a ejecutar la comunicación peer to peer entre los nodos utilizando el protocolo TCP/IP y, entre los muchos conceptos básicos relacionados en la misma, destacan los siguientes:

1. **Nodos** (nodes): Son procesos que realizan cálculos. Por ejemplo, en un sistema robótico, un nodo realiza la planificación de la trayectoria y otro puede suministrar una vista grafica del sistema.
2. **Mensajes** (messages): Los nodos se comunican entre ellos enviándose mensajes. Un mensaje es simplemente una estructura de datos, que contiene campos de un solo tipo de datos. Tipos primitivos de datos (enteros, booleanos, punto flotante, etc.) son soportados, así como lo son arreglos de tipos primitivos.
3. **Tópico** (topic): El tópico es un nombre usado para identificar el contenido de un mensaje. Un nodo envía un mensaje publicándolo en un tópico dado. Los mensajes son enrutados vía un sistema de transporte con una semántica de publicación/suscripción. es un nombre utilizado para identificar el contenido de un mensaje. Un nodo que está interesado en cierta clase de datos debe suscribirse al tópico apropiado.
4. **Maestro** (master): El master proporciona los elementos necesarios para que los nodos se encuentren los unos a los otros, intercambien mensajes o invoquen servicios.
5. **Servicios** (services): Es un modelo de comunicación cliente-servidor, es una manera en el que los nodos se pueden comunicar. Estos permiten que se envíen solicitudes y se reciban respuestas
6. **Bolsas** (bags): Formato para guardar y reproducir datos de un mensaje proveniente de un nodo de ROS. Son un importante mecanismo para almacenar datos, tales como datos de sensores, que son difíciles de obtener pero que son necesarios para el desarrollo y prueba de algoritmos.

La arquitectura básica de comunicación de ROS, usando 3 nodos como ejemplo simplificado es mostrada en la figura 2.1 y figura 2.2. Para establecer comunicación con diferentes nodos es necesario la utilización de registros, cuyos nombres son suministrados por el maestro.

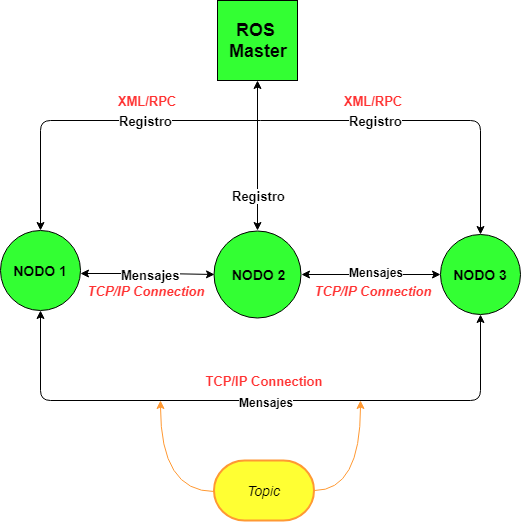


Fig.2.1.- Esquema de funcionamiento de la Arquitectura de ROS

Como se muestra en la Figura 2.1 el Nodo 1, el Nodo 2 y el Nodo 3 se registran al Master usando comunicación XML/RPC el cual, luego de conocer cada nodo registrado, permite a los diferentes nodos intercambiar la información entre ellos utilizando para esto el protocolo TCP/IP. Dado que cada nodo puede compartir múltiples mensajes, a estos se les clasifica como tópicos (topics) lo cual permite la centralización de información proveniente de varios procesos de un sistema robótico ya sea este físico o simulado.

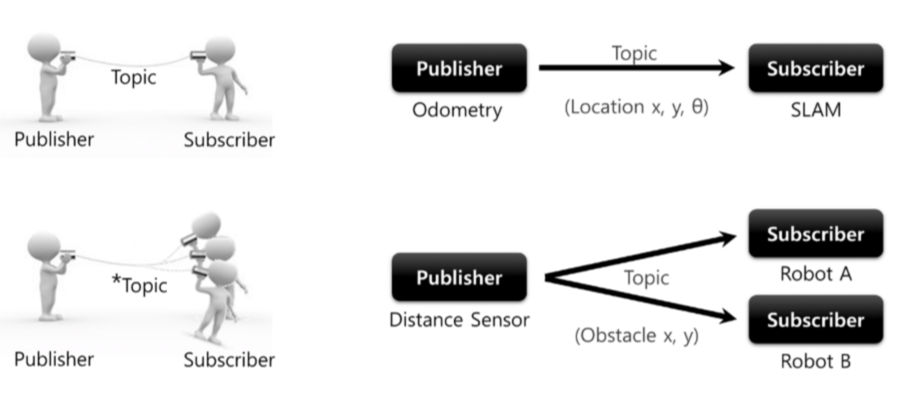


Figura 2.2. – Comunicación utilizando Topicos. Copyright 2017 por ROBOTIS Co., Ltd.

Un ejemplo del funcionamiento de la arquitectura de ROS es mostrado en la figura 2.2 en la cual se identifica el ***nodo*** “robot\_localization.” El nodo mencionado recibe mensajes desde varios nodos bajo los ***tópicos***: IMU, Camera, GPS, Detección 3D (Sensing 3D) y odometria (odometry), los datos suministrados en los ***mensajes*** son procesados por el nodo y como resultado comparte bajo un único tópico, y varios mensajes, al visualizador 3D RVIZ las coordenadas de localización del objeto físico para que este pueda ser representado de manera virtual.

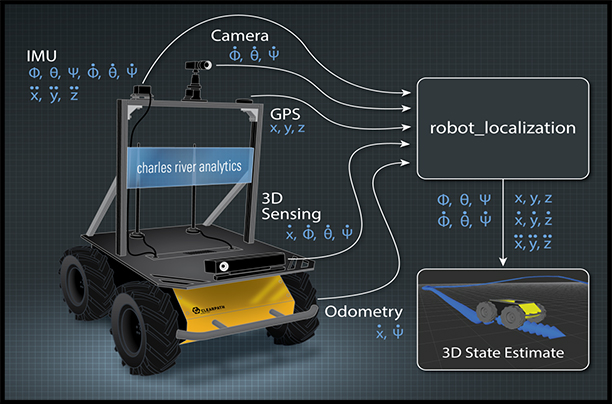


Fig.2.3 - Nodo robot\_localization recibiendo tópicos de un robot físico y mostrando su localización en el ambiente virtual.

Un nodo podemos considerarlo como un ejecutable dentro del paquete de ROS y utiliza la librería cliente, lista de nodos disponibles de ROS, para comunicarse con otros nodos, y estos a su vez se les configura para publicar o subscribirse a un tópico, o algunos procesos de ROS tales como servicios o almacenamiento de datos (Bags).

* + 1. **Librerías de ROS (¿cuáles librerías?)**

De las cualidades que hace atractivo a ROS para los propios roboticistas al diseñar y crear sistemas de robots es destacable la basta y surtida colección de librerías que puede acceder, esto de acuerdo a la distribución de ROS con el que este desee trabajar, debido a que cada distribución reciente de ROS cuenta con librerías con códigos más estables, también se eliminan librerías obsoletas y se agregan además nuevas librerías probadas y recomendadas por la comunidad de ROS. En este trabajo monográfico se trabajará con la versión Kinetic que es una de las más estable hasta hoy.

Al implementar un nodo en ROS se puede utilizar la librería roscpp para C++ o la librería rospy para Phyton, a su vez se utilizarán diferentes librerías para intercambio de mensajes, estructuración de datos para describir una trayectoria de un robot, estas librerías son propias para una aplicación de robótica.

* + 1. **Herramientas de visualización de datos en ROS**

ROS ofrece herramientas para la visualización de mensajes de ROS de sistemas robóticos, ya sean físicos o virtuales. A continuación, se describe algunos de los recursos de los cuales hace uso ROS. Entre estas herramientas destacan RVIZ, RQT, y MoveIt, las cuales son nodos en el sistema de suscripción de ROS compuestos a su vez por librerías de software basadas en ROS, con sus tópicos y servicios para permitir la entrada y salida de los datos a procesar.

1. **RVIZ**: es una herramienta de visualización 3D que permite combinar en una misma pantalla modelos de robots, datos de sensores (cámara, láser, etc.) y otros datos provenientes de mensajes 3D, esta herramienta sera importancia en el desarrollo del laboratorio para visualizar la morfología y la cinematica de los robots dentro del LVR.

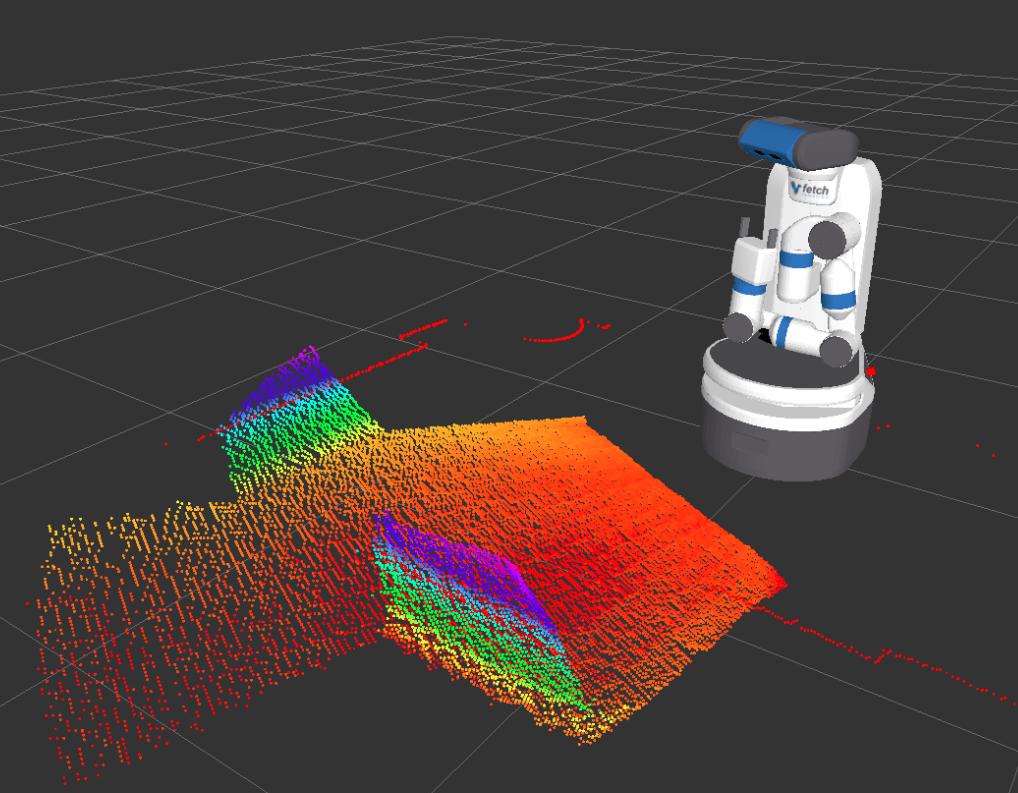


Fig.2.4. – RVIZ y visualización de modelo de robot y datos provenientes de mensajes 3D.

1. **RQT**: rqt ofrece plugins tales como ‘rqt\_image\_view’, ‘rqt\_graph’, ‘rqt\_plot’ y ‘rqt\_bag’ para la introspección y la visualización de datos provenientes de procesos de ROS, para visualizar mensajes de una cámara de un sistema robótico se utilizaría rqt\_image\_view, rqt expone los nodos y las conexiones entre ellos, lo que le permite fácilmente depurar y entender el sistema, su funcionamiento y como este se estructura. En la figura 2.3 se muestra a rqt-plot para visualizar los datos provenientes de un nodo. Dentro del LVR se utilizara RQT para graficar estructura de datos de posición del movimiento de los robots a analizar.

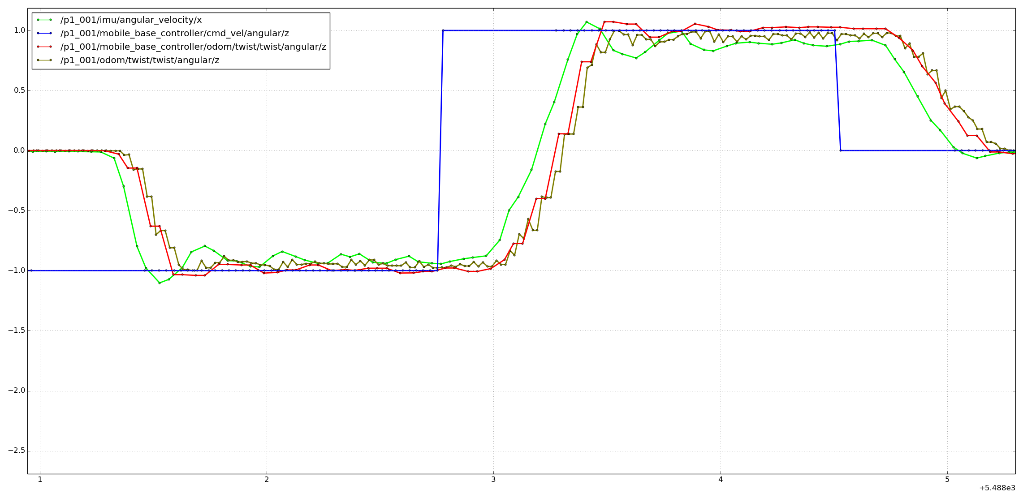


Fig.2.5. - Gráfico de línea XY de datos provenientes de nodos.

1. **MoveIt!:** Es una herramienta de software, escrita en C++, que permite la planeación de trayectorias de un robot industrial, así como la visualización de los datos y del modelo 3D valiéndose de un plugin que es incorporado a la aplicación de visualización de datos RVIZ. Actualmente, MoveIt es utilizado en diversas industrias para hacer la planeación de las trayectorias para diferentes tipos de manipuladores industriales, adoptando la visión de ROS de facilitar la programación de diferentes tipos de robots con los mismos recursos de Software.
   * 1. **Herramientas de terceros compatibles con ROS**

Al sistema en desarrollo se le puede suministrar mayor funcionalidad al utilizar ROS ya que el mismo permite la integración y utiliza la información generada por aplicaciones externas de código abierto muy populares tales como GAZEBO y OpenCV. Por ejemplo, es posible utilizar OpenCV para hacer reconocimiento de imágenes y tomar decisiones, a partir de los datos suministrados.

* + 1. **Lenguajes de programación (soportados por ROS)**

En el desarrollo del software para una aplicación de robótica, ROS permite el uso de distintos lenguajes de programación. De manera oficial soporta Python, C++ y LISP. Java podría ser soportado en el futuro, pero en la actualidad se encuentran en una fase experimental, apoyada por Google. De igual forma C# se encuentra en una fase experimental utilizando UNITY 3D, el cual es un motor de video juegos multiplataforma que permite simular robots importando el URDF como un GameObject en UNITY3D y obtener de este una simulación más realista con el redenderizado de UNITY3D.

Con ROS es posible utilizar nodos creados en diferentes lenguajes de programación tales como Phyton y C++ y garantizar el intercambio de información entre estos.

En el desarrollo de la GUI y el intercambio de información entre los nodos ROS se utilizará el lenguaje C++ y CMake el cual es una herramienta diseñada para construir, probar y empaquetar software, este compilador es el oficial soportado por la comunidad ROS desplazando al anterior compilador llamado rosbuild.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Cómo se lograrán/implementarán los recursos requeridos para prácticas relacionadas con la morfología del robot?

Se utilizara URDF para modelar los robots, RVIZ como visualizador de los modelos, los plugins de visualización del espacio de trabajo (reuleaux) y por ende el control basado en los nodos de C++ para compartir la información en la GUI.

Cómo se lograrán/implementarán los recursos requeridos para prácticas relacionadas con la cinemática del robot?

**Cinemática Directa** En la descripción del laboratorio virtual se ha mencionado la importancia de la cinemática para el análisis de un robot.

Para esto se decidió incorporar la librería KDL que permite el análisis de un robot a partir de un archivo URDF.

Conocer el valor del Efector final de acuerdo a la posición actual de cada joint, conocer el grado de apertura de cada robot y conocer la matriz de transformación homogénea característica de cada robot.

Para esto se aplican diferentes algoritmos ya propios de la librería.

**Cinemática Inversa** La cinemática inversa forma parte de lo propuesto en el trabajo monográfico pero este punto representa mayor complejidad al utilizar los algoritmos ya propuestos en la librería utilizada en la cinemática directa KDL. De tal forma que para este problema se ha elegido el uso de las librerías de MoveIt! La cuales además de utilizar las librerías nativas de KDL incorporan una mejor solución al problema de la cinemática inversa, esto es así porque MoveIt! Es el producto de incorporar diferentes métodos del estado del arte de la planeación del movimiento de un Robot (Motion Planning).

Como se lograrán/implementarán los recursos requeridos para practicas relacionadas con la programación del robot?

Para la programación se utilizara Gazebo simulador ya que permite cargar un escenario desde el archivo SDF, es diferente la forma de manipular un escenario desde Gazebo comparado con RVIZ, debido a que todas las visualizaciones de los modelos dentro de RVIZ son de carácter estático es decir que no se puede ver sólidos dinámicos (Dentro de Gazebo se puede simular una Un tornillo cayendo desde una mesa) esto sin necesidad de programación previa, desde el momento que cargue todos los modelos al escenario de Gazebo estos rebotan, balancean y se vuelven dinámicos.

Por lo tanto si se desea obtener estas capacidades de “realismo” es necesario utilizar gazebo.

Ademas de Gazebo como el principal recurso de software se utilizara programación en c++ para describir las instrucciones parecidas a RAPID y que permitan simular el robot.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* 1. **Modelo de utilización del laboratorio virtual para robótica industrial**

Los laboratorios virtuales de robótica pueden ser concebidos como un recurso de carácter multidisciplinario y los mismos pueden tener diferentes alcances, desde un laboratorio simple que permita realizar experimentos relacionados con los aspectos básicos de la cinemática y dinámica de los manipuladores industriales hasta laboratorios virtuales más sofisticados con capacidad de simulación de entornos complejos como sería una estación de trabajo en la cual los robots puedan ser programados para manipular objetos en el entorno del trabajo y en la cual se aplican técnicas complejas de control y posición. Laboratorios virtuales más complejos pueden incluir funciones de comunicación que permitan la interacción con un robot real, lo cual lo convertiría en un laboratorio físico-virtual, que permitiría consolidar aún más la teoría relacionada con la robótica (Hoffmann, 2016).

En la figura 1 se muestra el modelo de utilización del laboratorio virtual propuesto y en la misma se puede apreciar el papel fundamental del profesor quien tendrá la responsabilidad de identificar los diferentes experimentos que pueden ser desarrollados con los recursos disponibles en el mismo. El modelo es una modificación del modelo de laboratorio virtual presentado por Pérez et al (2016).

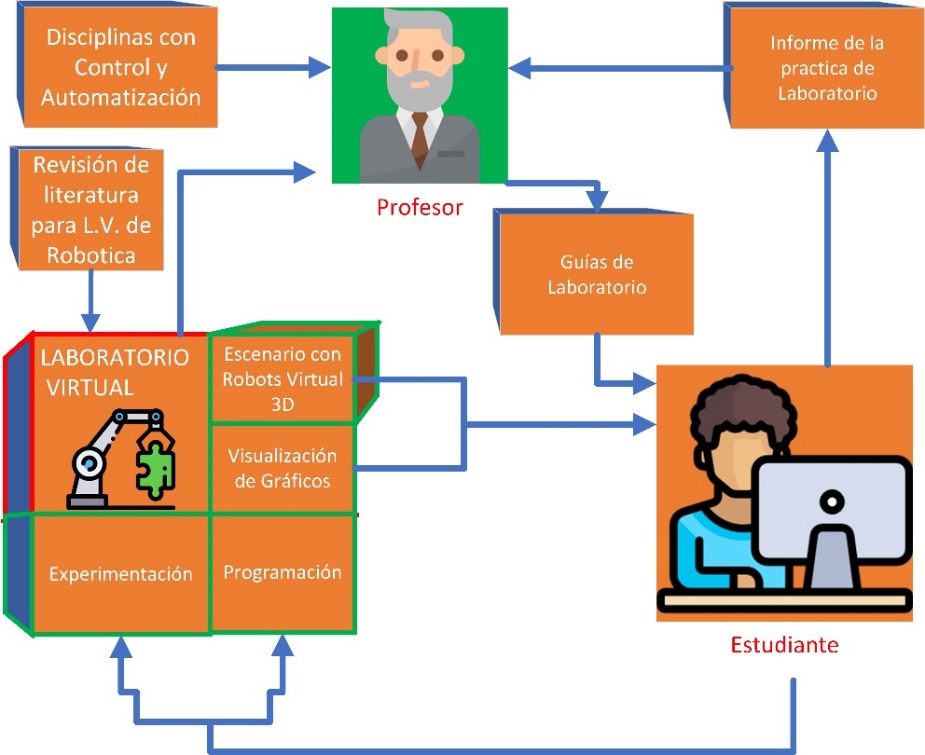


Fig.4.- Modelo de utilización del laboratorio virtual de robótica industrial. (revisarlo)

El laboratorio virtual contará con los recursos básicos que permitirán, al docente en primer lugar, diseñar experimentos que ayuden a los estudiantes a comprender y aplicar los aspectos básicos relacionados con la robótica industrial. También permitirá a los estudiantes realizar las prácticas, indicadas por el profesor, en cualquier lugar, en cualquier momento y las veces que sea necesario.

Los principales elementos del laboratorio virtual son:

* Interface gráfica de usuario (GUI) para (interacción, programación, visualización, etc.)
* Modelos de robots industriales de ABB, KUKA, y DELTA
* Escenarios para la realización de tarea específicas
* Información sobre aspectos fundamentales sobre robótica industrial
* Otros

La interacción entre los diferentes elementos del laboratorio virtual es garantizada mediante el middleware ROS. A continuación, se abordarán los elementos del laboratorio virtual, pero antes de considerará el sistema operativo para robot (ROS) dado que es el elemento de interés y el agente integrador en este proyecto.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* 1. **Interfaz gráfica de usuario (GUI)**

Una interfaz grafica de usuario (GUI, por sus siglas en inglés) facilita la explotación de una aplicación mediante la incorporación de elementos tales como botones, cuadros de lista, controles deslizantes, menús, entre otros. La GUI debe comportarse de una manera comprensible y predecible, de modo que un usuario sepa qué esperar cuando realiza una acción.

El laboratorio virtual de robótica incorporará una interfaz gráfica de usuario que facilitará al estudiante el uso de los recursos de este de una manera transparente sin necesidad de tener que configurar aspectos básicos de ROS tales como su inicialización y el lanzamiento de nodos. La GUI puede ser desarrollada utilizando diferentes lenguajes de programación tales como C++ y Phyton. La interfaz gráfica de usuario del laboratorio virtual contará con los elementos necesarios (botones, sliders, editor de texto, pantalla de visualización, etc.) para realizar las prácticas sobre los fundamentos de la robótica industrial, así como para acceder a documentos de interés tales como libros, guías de laboratorio, entre otros.

Para la creación de una GUI existen diferentes lenguajes de programación tales como Visualbasic, C#, java, Phyton, C++ entre otros, cada uno de estos lenguajes poseen aceptación en creación de GUI, pero existe una preferencia hacia aquellos lenguajes que no son propietarios de una sola plataforma, se prefieren los lenguajes multi-plataformas ya que la GUI puede ser aprovechada en diferentes sistemas operativos donde pueda ser ejecutada la GUI.

Ya que se tiene en cuenta la importancia de los lenguajes multiplataformas es necesario mencionar que para la programación del laboratorio virtual es de importancia elegir un entorno de desarrollo integrado (IDE) de programación que permita aglomerar todas las herramientas necesarias de programación tales como conjuntos de librerias, preprocesador, depurador y compiladores. Se conoce de Visual Studio, Eclipse, NetBeans, CLion, Qt – Creator entre otros como los IDEs mas reconocidos, cada uno con sus fortalezas y bondades aplicadas a ciertos requerimientos de los programadores, siendo mas puntuales en remarcar que también se busca en todo IDE la multiplataforma nosotros para este trabajo nos hemos decidido inclinar hacia el uso de Qt-Creator ya que posee un plugin de Soporte para ROS.

Con Qt Creator se crearan las aplicaciones o nodos de ROS se explorara las altas capacidades que posee QT Creator para la creación de aplicaciones de escritorio cabe destacar que son dos líneas de creación de una aplicación en QT creator una es usando QML el cual es un estándar propio del IDE que garantiza crear aplicaciones dinámicas y más elaboradas capaz de incorporar recursos WEB y entre otras potencialidades de importancia a considerar en este trabajo monográfico, un ejemplo de esto es la capacidad de incorporar un lector de PDF que facilite al estudiante el estudio de los libros seleccionados para el LVR. Otra línea de creación de aplicaciones es usar los Widgets (Slider, botones etc.) de QT escritos en C++ esta línea también beneficia mucho a la visión del LVR debido a que es amigable incorporar las librerías de ROS en C++, a una aplicación utilizando Widgets, un ejemplo de esto es realizar una aplicación de escritorio personalizada que se acople a los requerimientos en el LVR al incorporar las librerías de la herramienta RVIZ y visualizar modelos de Robots.

Debido a que es necesario la transparencia de la GUI para la ejecución de las practicas de los Robots es necesario mencionar que garantizar esto llevara al LVR a tener que ejecutar procesos que den inicio a la ejecución de las interacciones en la GUI, visualización de robots y datos, simulaciones y programación de robots los cuales están basados en nodos que tienen que ser lanzados o eliminados de acuerdo a la lógica implementada en el propio LVR.

~~En la implementación de la interfaz gráfica de usuario destacan dos partes:~~

* ~~Lo relacionado con las prácticas de laboratorio~~
* ~~Lo relacionado con la documentación~~

~~Para el desarrollo del primer aspecto serán utilizados los siguientes recursos de software……~~

~~En los que respecta al repositorio de información, la interfaz contará con un botón que al ser presionado presentará al usuario un navegador mostrando los enlaces a los diferentes documentos disponibles. Para lograr dicho objetivo de utilizará la aplicación xxxx la cual tiene las siguientes características….~~

* + 1. **Interacción dentro la GUI**

La interfaz proporcionará los elementos requeridos para que el usuario pueda utilizar todos los recursos del LVR sin tener que preocuparse en la configuración de ROS y sus respectivos nodos para visualizar datos, visualizar modelos 3D de los robots, para analizar el resultado de la aplicación a su vez el LVR cuenta con una biblioteca digital de libros con fundamentos de robótica, artículos técnicos y tutoriales que facilitaran el entendimiento de los fundamentos de la robótica industrial.

* + 1. **Visualización de datos.**

Dentro de la propia GUI se adoptarán las herramientas o aplicaciones de desarrollo nativas de ROS (Véase apartado 2.1.3), esto es posible debido a la naturaleza de software libre de ROS lo que permite tomar las librerías propias (Código fuente) de estas herramientas que sirven de modo general para depurar, visualizar variables y agilizar el desarrollo de soluciones en una aplicación especifica de robots.

* + - 1. **Visualización 3D de robots**

Para garantizar la visualización 3D de los modelos matemáticos de los robots tipo serie considerados en esta propuesta de trabajo monográfico dentro de la propia GUI, es necesario tener una ventana de visualización la cual será logrado utilizando la aplicación RVIZ (Véase apartado 2.1.3) de ROS de la cual se tomarán sus librerías (código fuente) y se adaptarán a los requerimientos del laboratorio virtual. Ver Figura 5. RVIZ es una abreviación para ROS visualización y es una herramienta poderosa para la visualización en 3D. Permite al usuario ver el modelo simulado del robot, obtener información de los sensores del robot, y reproducir la información obtenida de los sensores. Mediante la visualización de lo que ve y hace el robot el usuario puede eliminar los errores de la aplicación robótica desde entradas de sensores hasta acciones planificadas o no planificadas.

* + - 1. **Gráficos de dispersión XY**

De la misma forma en que es posible adoptar las librerías de visualización del modelo de un robot también ROS permite, mediante el uso de la herramienta RQT, la depuración y visualización de los datos provenientes de los nodos de una aplicación dada. Las librerías pueden ser adaptadas a la GUI del laboratorio virtual. RQT\_PLOT permite visualizar datos en gráficos de dispersión XY el cual permitiría hasta mostrar la respuesta del control de un sistema retroalimentado PID de cada unión (Joint) del robot. Ver Figura 5 que muestra la posición con respecto al tiempo de las uniones de un robot industrial.

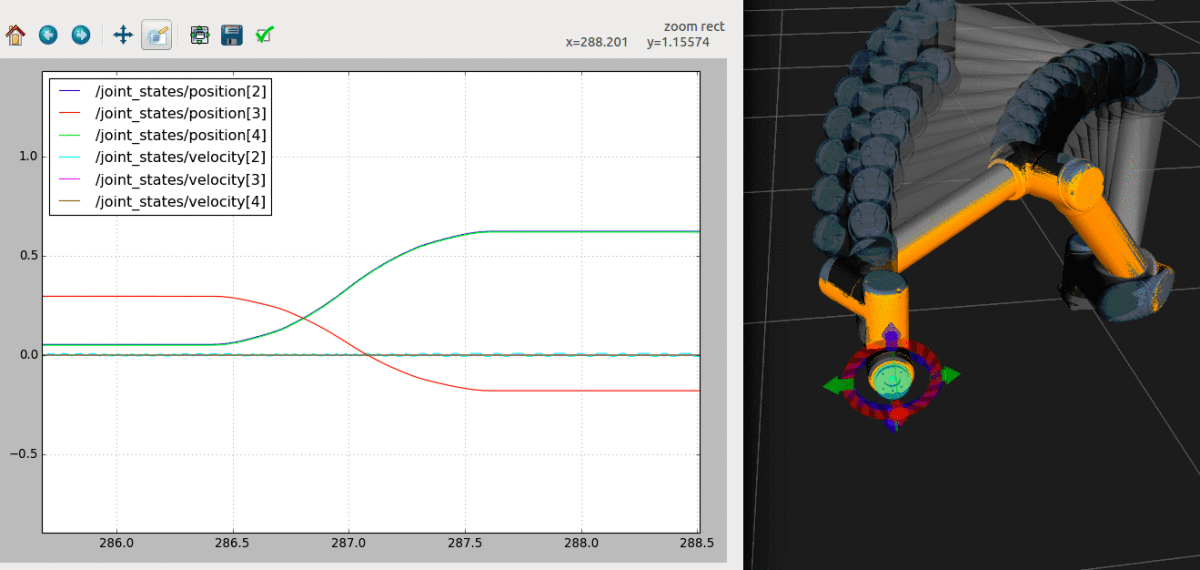


Fig.5. - Muestra de datos en gráficos de dispersión XY de la posición de uniones del modelo de robot virtual mostrado en RVIZ.

* + 1. **Programación de Rutinas de robots**

El software creado por los fabricantes de robot para la programación de estos, tales como Rapid de ABB; KRL de KUKA; Karel de FANUC; o URScript de Universal Robots, adopta ciertas convenciones que son propias para cada tipo de robot. En el laboratorio virtual que se propone se adoptarán ciertas convenciones, utilizadas en el software de programación de robots de ABB, que permitirán describir en un script la programación que será enviada por el nodo de la aplicación GUI al nodo GAZEBO donde se contará con el escenario donde se simulará el comportamiento del robot bajo consideración en una aplicación estándar. Por ejemplo, un robot en una aplicación de paletización.

* + 1. **Información de los fundamentos de Robótica Industrial**

La GUI del laboratorio virtual de robótica permitirá el acceso a información concerniente a la estructura, recursos, y funcionamiento del LVR. Contará además los elementos necesarios para acceder a materiales didácticos tales como libros, videos, enlaces a páginas web que propicien el aprendizaje significativo de los estudiantes.

* 1. **Modelos de robots en el LVR**

En un laboratorio real el estudiante cuenta con un robot físico el cual utiliza para realizar experimentos. En cambio en un laboratorio virtual, generalmente se trabaja con modelos de los robots, siendo la modelación, por lo tanto, un aspecto fundamental en la construcción de este. Existen muchas herramientas para la representación virtual de un robot, una de ellas implica la creación de un modelo utilizando archivos CAD en formatos STL, DAE, que describe la geometría en 3D del mismo. Además del modelo en 3D desarrollado es necesario especificar las propiedades internas, así como las interacciones entre los elementos. Lo anterior se logra utilizando el lenguaje de marcado URDF para modelar el robot y SDF para modelar el robot y el escenario virtual. En la comunidad ROS y GAZEBO a lo anterior se le denomina “modelar el mundo.” El modelo virtual obtenido puede ser controlado en un ambiente de simulación tal como GAZEBO.

* + 1. **Modelado de brazos robóticos de diferentes DOF.**

Las herramientas de modelación de robots, de carácter virtual, permiten gestionar de forma estandarizada las propiedades y representaciones en 3D lo cual permite agilizar el proceso de integración de diferentes diseños de un robot, esto abarca desde realizar un modelo propio o utilizar uno de los existentes, que describen fielmente al robot industrial, proveniente de ABB, FANUC o KUKA. (ROS-Industrial Tutorials, 2017)

1. **CAD:** ElDiseño de robots a base de software CAD consisten en la modelación a detalles del robot bajo ciertos archivos por separados (Union 1 un archivo, Union2 otro archivo) que pueden ser ensamblados y configurados bajo ciertos parámetros de vinculación entre cada sólido.
2. **URDF**: Es un archivo XML que sólo puede especificar las propiedades cinemáticas y dinámicas de un solo robot de forma aislada. El formato URDF define que el robot está compuesto por eslabones rígidos unidos mediante articulaciones, y que se puede definir como una estructura de árbol. Ver figura 4.

URDF soporta la integración de archivos CAD para una mejor visualización del modelo del robot.

* URDF no puede especificar la posición del propio robot dentro de un mundo.
* Carece de fricción y otras propiedades.
* Usado para visualización de un robot virtual en escenarios de visualización tales como RVIZ

1. **SDF**: Es un archivo de lenguaje de marcado XML que contiene la descripción completa de un entorno o escenario de carácter virtual desde el nivel del mundo hasta el nivel del robot, incluyendo (SDF Format Specification, 2014), el usar este lenguaje en el LVR permitirá modelar los propios escenarios de trabajo dentro del ambiente de gazebo valiéndose de los siguientes parámetros:

* **Escena**: Iluminación ambiental, propiedades del cielo, sombras.
* **Física**: Gravedad, paso del tiempo, motor de la física.
* **Modelos**: Colección de enlaces, objetos de colisión, articulaciones y sensores.
* **Luces**: Punto, espacio y fuentes de luz direccionales.
* **Plugins**: plugins soportado por gazebo, plugins del mundo, del modelo, del sensor y del sistema.

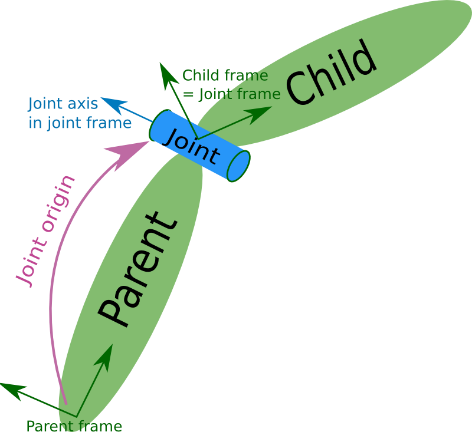


Fig.6. - Representación de la modelación en URDF de dos eslabones y una sola unión.

* 1. **Entornos de simulación de robots**

La simulación de robots es la capacidad de modelación matemática de carácter virtual bajo herramientas de software con el fin de realizar experimentos en pro de apropiarse del conocimiento ligado al comportamiento y estrategias de control y operación de cierto objeto de práctica, sin requerir la construcción o experimentación de un sistema físico, consecuentemente previniendo accidentes y mal operación de un experimento.

El estudio de la robótica basado en la virtualización combina prácticas de varias disciplinas de la ingeniería accediendo a ventajas innegables, las cuales de acuerdo a Mato (2014) son:

* Simulación de nuevos modelos dinámicos permitiendo reconfigurar o actualizar una práctica de laboratorio.
* Introducir al modelo virtual el comportamiento de actuadores bajo un modelo matemático descriptivo.
* Evaluación de técnicas de control de procesos como PID o Fuzzy, etc.
* Simular el comportamiento de trayectorias de un robot antropomórfico.
* Evitar accidentes y daños a infraestructuras físicas.
  + 1. **Software (¿??) de compañías basadas en CAD**

Desarrollados por empresas con gran experiencia en el modelado de sólidos 3D, utilizados para representar sistemas mecánicos y su comportamiento ante análisis de carácter mecánico (análisis de tensiones, deformaciones, resistencia, etc.) análisis cinemático y dinámico. Entre ellos destacan SolidWorks y AutoCAD Estos softwares proporcionan la funcionalidad de realizar simulaciones, pero no poseen las funcionalidades de diseño requeridas para ejecutar el control de robots manipuladores.

* + 1. **Software proveído por los fabricantes de robots**

Software propiedad de empresas que permiten acceder al modelado virtual y programación de una planta utilizando los robots comerciales, modelados virtualmente que corresponden a los modelos físicos fabricados por los mismos a base de archivos CAD, esta programación es implementada posteriormente en una planta física, denominando a esta práctica programación desconectada (off-line)

Estos softwares incorporan la funcionalidad más cercana o más fácil para la manipulación de un robot solo de la gama de un fabricante, permitiendo apropiarse de conceptos de una forma superficial (sin conocer la teoría de robots tales como la matemática que describe el movimiento de un robot), en el contexto tradicional es manipular una caja negra al aprender a utilizar un software bajo los estándares establecidos por la empresa propietaria sin poder observar los procesos de cálculos o control que todo robot utiliza sin excepción alguna.

Estos softwares son productos ampliamente probados, son softwares de calidad, ya que con ellos el fabricante provee de una herramienta al usuario para poner en marcha al robot. Los softwares más reconocidos son KukaSim de Kuka, RobotGuide de FANUC y RobotStudio de ABB.

* + 1. **Softwares de programación de varios fabricantes**

La programación de robots industriales se ha visto envuelta en varias evoluciones debido a el problema de que son difíciles de programar. Por lo tanto, los diseñadores de sistemas de fabricación están buscando formas más intuitivas de programar robots, especialmente utilizando los dibujos CAD del sistema de producción que poseen en físico como paletizadoras, banda transportadoras etc. Normalmente estos softwares poseen alianzas con distintas compañías fabricantes de Robots los cuales les permite proveer al usuario final de un software que aloje diferentes robots de diferentes fabricantes facilitando la programación y la centralización de la programación. Octopuz, Delfoi y FastSuite son de las compañías más influyentes en brindar este tipo de soluciones.

* + 1. **Software Simuladores de Robots de propósito general**

Estos simuladores desempeñan un papel importante en la investigación de la robótica como herramientas para probar la eficiencia, la seguridad y la solidez de los nuevos algoritmos. En la realidad, el incursionar en la robótica es caro, pero gracias a los simuladores existentes de robótica es posible crear, programar y testear infinidad robots sin tener que recurrir a una inversión.

Debido a la necesidad de tener una herramienta versátil y de tal forma que en los últimos años hemos sido testigos de los cambios bruscos de la potencia de hardware y de software que benefician al desarrollo de herramientas de software precisas, robustas y fáciles de usar para la simulación de modelos robóticos, sensores y control en entornos virtuales que posean propiedades genéricas de desarrollo

Donde distintos simuladores de robots como V-Rep y Gazebo se posicionan como referentes V-Rep es el simulador comercial por excelencia en cambio Gazebo es conocido como el simulador Software Libre (open-source) más destacable.

* + - 1. **Simuladores de propósito general comerciales**

Existen diferentes softwares de simulación de carácter comercial acá se mencionan los más influyentes.

1. Webots: es un entorno de simulación 3D utilizado para modelar, simular y programar robots móviles. Los modelos simulados tienen muchos atributos diferentes y personalizables, como textura, masa, fricción y forma. Estos atributos son proporcionados por la librería Open Dynamics Engine (ODE) que se utiliza para simular la dinámica rígida del cuerpo. Webots puede simular una gran cantidad de sensores comúnmente utilizados en robótica, como sensores de proximidad, sensores de luz, sensores táctiles, GPS, acelerómetros, láseres y cámaras.
2. MSRS (Microsoft Robotics Studio): Esta basado en Windows es un entorno para control y simulación de robots, MSRS requiere de C# para su programación y simulación de los robots.
3. MATLAB: Matlab posee Simulink donde la comunidad creó cajas de herramientas (Toolbox) en las que se pueden usar bloques definidos para plataformas robóticas y sensores. MATLAB y Simulink se pueden usar conjuntamente, junto con la caja de herramientas gratuita Robotics Toolbox De Peter Corke para diseñar simulaciones con manipuladores de robots. Robotics Toolbox proporciona funciones para generar trayectorias y analizar resultados de simulaciones de modelos virtuales de robots reales. Simulink y MATLAB están disponibles para sistemas Linux, Windows y Mac OSX.
4. V-REP: es un simulador con entorno de desarrollo integrado donde cada objeto/modelo se puede controlar individualmente a través de un script de programación que puede ser escrito usando lenguajes de programación como C/C++, Python, Java, Lua, Matlab or Octave, V-REP se utiliza para el desarrollo rápido de algoritmos, simulaciones de automatización de fábricas, prototipado rápido y verificación y educación relacionada con la robótica.
   * + 1. **Simuladores de propósito general de uso libre (Open source)**

Existen una variedad de simuladores open-source en el campo de la robotica tales como Player/Stage, Gazebo, CARMEN, USARSim, MissionLab, OpenRAVE, OPRoS, Orca, OROCOS, YARP los cuales poseen ciertas características que los hacen destacarse por proveer la reutilización del software pero cada uno tiene sus ventajas con respecto a otro como el tipo de visualización ya sea en 2D o 3D, el lenguaje de programación siendo los más destacados como C++, Phyton, Java, tipo de sistema operativo donde estos pueden ser instalados siendo Linux el que lidera en Simuladores open source. Se mencionarán las características de Gazebo, OROCOS y Player/Stage Por ser los más influyentes de acuerdo a Tsardoulias (2017)

1. Gazebo: Es un simulador 3D Dinamico que provee una simulación realista de la física de los cuerpos rígidos donde los robots pueden interactuar con el mundo, (pueden coger y empujar cosas, rodar y deslizarse por el suelo) se puede crear en gazebo escenarios (mundos) de simulación, variar las características de los contactos con el suelo, los obstáculos e incluso los valores de la gravedad por ejemplo si se describe un mundo con gravedad cero el objeto posicionado en este mundo flotará.

What is Gazebo?

Gazebo is a 3D dynamic simulator with the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. While similar to game engines, Gazebo offers physics simulation at a much higher degree of fidelity, a suite of sensors, and interfaces for both users and programs.

Typical uses of Gazebo include:

* testing robotics algorithms,
* designing robots,
* performing regression testing with realistic scenarios

A few key features of Gazebo include:

* multiple physics engines,
* a rich library of robot models and environments,
* a wide variety of sensors,
* convenient programmatic and graphical interfaces

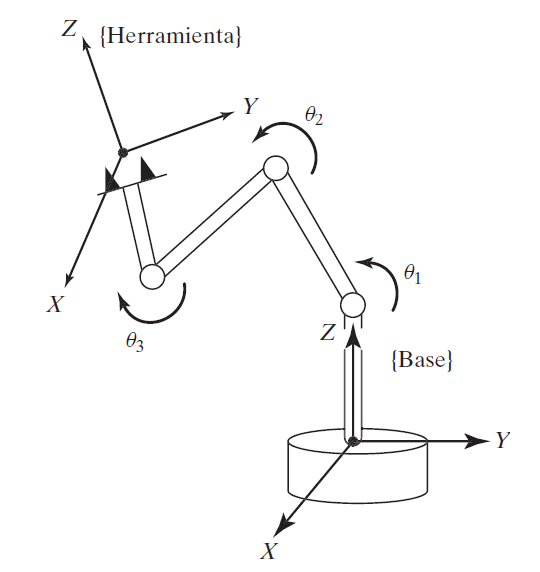
1. OROCOS Proyecto de software o básicamente una colección de bibliotecas portátiles de C ++ orientado al control de robots, su principal fortaleza es que es gratuito y se centra en el control en tiempo real de robots.
2. Player/Stage: Proporciona al usuario un entorno gráfico bidimensional, que tiene las instalaciones para una cámara en perspectiva, lo que lo convierte en un simulador de 2,5 dimensiones. Tiene posee la facilidad para modelar el robot y sus sensores usando scripts simples.

De los anteriores simuladores, la mayoría posee la opción de comunicarse con ROS lo que posibilita la reutilización de software, por ejemplo, software para la planeación de trayectorias, para el desarrollo de escenarios virtuales en los cuales se pueden realizar experimentos con la finalidad de que los estudiantes se apropien de los aspectos básicos de la robótica industrial. Entre los simuladores de código abierto que permiten la comunicación con ROS destacan GAZEBO, OROCOS, USARSim, openRAVE, OPRoS. También hay softwares comerciales que permiten la comunicación y entre ellos tenemos V-REP y MATLAB.

Para poder estudiar el comportamiento de un robot en respuesta a un programa (SCRIPT) que se acople a los lenguajes de programación como RAPID el cual pertenece a la plataforma de programación de robots como ABB se requiere del uso de un simulador donde un robot pueda interactuar con un escenario de acuerdo a las instrucciones del SCRIPT, para lograr esta cercanía a las plataformas de programación comerciales se ha decidido utilizar en este laboratorio virtual de robótica el uso de Gazebo y el uso de los nodos de ROS para el control del robot virtual.

* 1. **Modelación Cinemática y Dinámica**

La cinemática del robot estudia el movimiento de este con respecto a un sistema de referencia sin considerar las fuerzas que intervienen. Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas articulares. Dentro de la cinemática se estudian la posición, velocidad, aceleración y todas las derivadas de mayor orden de las variables de posición (respecto al tiempo o a cualquier otra variable).





**El problema cinemático** directo consiste en determinar cuál es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot.

Dado que un robot se puede considerar como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones, se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia.

**La resolución del problema cinemático directo** consiste en encontrar las relaciones que permiten conocer la localización espacial del extremo del robot a partir de los valores de sus coordenadas articulares.

De esta forma, el problema cinemático directo se reduce a encontrar una matriz de transformación homogénea T que relacione la posición y orientación del extremo del robot respecto del sistema de referencia fijo situado en la base de este. Esta matriz T será función de las coordenadas articulares.

Denavit y Hartenberg propusieron un método sistemático para describir y representar la geometría espacial de los elementos de una cadena cinemática, y en particular de un robot, con respecto a un sistema de referencia fijo.

Este método utiliza una matriz de transformación homogénea para describir la relación espacial entre dos elementos rígidos adyacentes, reduciéndose el problema cinemático directo a encontrar una matriz de transformación homogénea 4 × 4 que relacione la localización espacial del extremo del robot con respecto al sistema de coordenadas de su base.

¿Qué puede ofrecer el LVR, al docente y estudiantes, para comprender lo antes planteado?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

El segundo, denominado problema cinemático inverso, resuelve la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas. Este problema se plantea de la siguiente manera: dada la posición y orientación del efector final del manipulador, calcule todos los conjuntos posibles de ángulos articulares que podrían utilizarse para obtener esta posición y orientación dadas. (Vea la figura). Éste es un problema fundamental en el uso práctico de los manipuladores.

El problema de la cinemática inversa no es tan simple como el de la cinemática directa. Debido a que las ecuaciones cinemáticas son no lineales, su solución no es siempre sencilla (o incluso posible) en una forma cerrada. Además, surgen preguntas sobre si existe una solución o existen múltiples soluciones.

La existencia o inexistencia de una solución cinemática define el espacio de trabajo de un manipulador dado. La falta de una solución significa que el manipulador no puede obtener la posición y orientación deseadas, ya que se encuentran fuera del espacio de trabajo del manipulador.

¿Qué puede ofrecer el LVR, al docente y estudiantes, para comprender lo antes planteado?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

El modelado cinemático de un robot busca las relaciones entre las variables articulares y la posición (expresada normalmente en forma de coordenadas cartesianas) y orientación del extremo del robot (expresada como matrices de rotación, ángulos de Euler o algún otro de los métodos establecidos en el Capítulo 3). En esta relación no se tienen en cuenta las fuerzas o pares que actúan sobre el robot (actuadores, cargas, fricciones, etc.) y que pueden originar el movimiento del mismo.

Sin embargo, sí incumbe a la cinemática del robot el conocer la relación entre las velocidades de las coordenadas articulares y las de la posición y orientación del extremo, o lo que es equivalente, el efecto que un movimiento diferencial de las variables articulares tiene sobre las variables en el espacio de la tarea. Esta relación queda definida por el modelo diferencial. Mediante él, el sistema de control del robot puede establecer qué velocidades debe imprimir a cada articulación (a través de sus respectivos actuadores) para conseguir que el extremo desarrolle una trayectoria temporal concreta, por ejemplo, una línea recta a velocidad constante.

El modelo diferencial queda concretado en la denominada matriz Jacobiana.

En ambos casos, la **matriz Jacobiana directa** permite conocer una expresión de las velocidades del extremo del robot a partir de los valores de las velocidades de cada articulación. Por su parte, la **matriz Jacobiana inversa** permitirá conocer las velocidades articulares necesarias para obtener un vector concreto de velocidades del extremo.

¿Qué puede ofrecer el LVR, al docente y estudiantes, para comprender lo antes planteado?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

El estudio de la cinemática y dinámica de los robots es fundamental para entender el funcionamiento de los ~~robots~~. El laboratorio virtual que se propone ofrecerá al estudiante realizar experimentos asociados a la cinemática y dinámica de los manipuladores industriales utilizando modelos de robots comerciales.

(Poner una imagen donde se muestre la estructura básica de un manipulador industrial (modelo de eslabones y joints) y destacar los aspectos relacionados con la cinemática y la dinámica.)

En esta propuesta de trabajo monográfico el análisis de la cinemática y algunos aspectos de la dinámica serán de mucha importancia en este trabajo ya que conocer estos conceptos permitirá al lector del documento guía de uso de ROS y al estudiante, profesor, etc que interactúe con el laboratorio virtual propuesto conocer los conceptos básicos y experimentar sobre estos mismos conceptos al utilizar el laboratorio virtual.

En si la modelación de estas cadenas cinemáticas forma parte de los conceptos básicos de robótica industrial la cual consiste en una serie de cuerpos rígidos conectados por juntas, donde un enlace secundario se conecta de vuelta al enlace padre. Normalmente se usan para generar un movimiento de salida deseado o fuerza en un enlace desde la entrada en otro enlace, y son la base de muchos mecanismos.

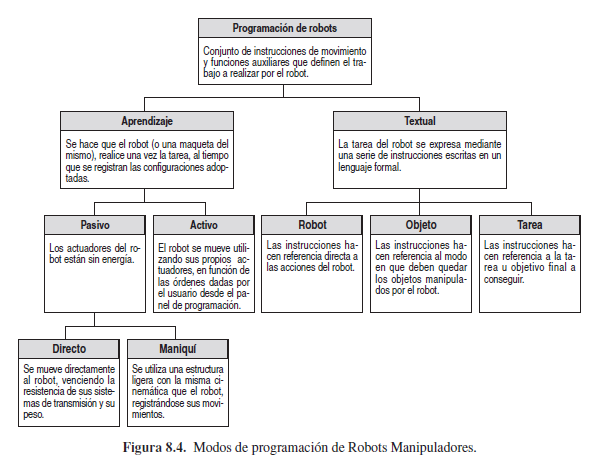
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**PROGRAMACIÓN DE ROBOTS**

Un robot industrial es básicamente un manipulador multifuncional reprogramable, siendo esta capacidad de reprogramación la que permite su adaptación de una manera rápida y económica a diferentes aplicaciones. La programación de un robot se puede definir como el proceso mediante el cual se le indica a éste la secuencia de acciones que deberá llevar a cabo durante la realización de su tarea. Estas acciones consisten en su mayor parte en moverse a puntos predefinidos y manipular objetos del entorno.

Durante la ejecución de un programa se interacciona con la memoria del sistema, leyendo y actualizando el contenido de las variables utilizadas en el programa; con el sistema de control cinemático y dinámico del robot, encargados de dar la señal de mando a los accionamientos del robot a partir de las especificaciones del movimiento que se les proporciona; y con las entradas-salidas del sistema, consiguiéndose así la sincronización del robot con el resto de las máquinas y elementos que componen su entorno.

El sistema de programación es, por tanto, la herramienta con que cuenta el usuario para acceder a las diversas prestaciones del robot, existiendo una relación directa entre las características y posibilidades del sistema de programación y las del robot en sí mismo.



Programar un robot consiste en indicar paso por paso las diferentes acciones (moverse a un punto, abrir o cerrar la pinza, etc.) que éste deberá realizar durante su funcionamiento automático. La flexibilidad en la aplicación del robot y, por tanto, su utilidad dependerá en gran medida de las características de su sistema de programación.

Los escasos intentos de unificar en cierta medida los procedimientos de programación de robots no han tenido hasta la fecha el reconocimiento y la aceptación necesarios, encontrándose que cada fabricante ha desarrollado su método particular, válido únicamente para sus propios robots. Sin embargo, existen algunos sistemas de programación que han servido de modelo para el desarrollo de otros. Tal es el caso del lenguaje AL [FINKEL-74] desarrollado por la Universidad de Stanford en los años setenta y que ha servido de referencia para muchos de los sistemas comerciales existentes. También es necesario indicar que, a pesar de la variedad de sistemas de programación existentes en el mercado, en todos ellos se dan una serie de características comunes y que se explicarán con detalle en este capítulo.

Existen diversos criterios para realizar una clasificación de los métodos de programación de robots. Algunas atienden a la potencia del método, mientras que otras clasificaciones hacen referencia al sistema empleado para indicar la secuencia de acciones a realizar. Este segundo criterio es más ilustrativo a la hora de dar a conocer las alternativas existentes para programar un robot, siendo por esta razón el que se va a seguir.

Según este criterio, un robot puede ser programado mediante lo que se denomina guiado o mediante un procedimiento textual, existiendo robots que conjugan ambos modos. Este primer nivel de división admite un segundo, atendiendo a la potencia y particularidades del método de programación. Las características de cada procedimiento de programación serán más o menos ventajosas según la aplicación a la que se destine el robot.

¿Qué puede ofrecer el LVR, al docente y estudiantes, para comprender lo antes planteado?

El laboratorio debe brindar la posibilidad al estudiante de programar (guiado o textual) al robot y ver la respuesta de su modelo ante el mismo.

¿Programación por aprendizaje o textual?

¿Lenguaje de programación?

Los métodos de programación por guiado son muy útiles e incluso imprescindibles en muchas ocasiones. Así, es difícil programar una tarea de pintura de una carrocería de automóvil mediante un procedimiento diferente a la programación por guiado. Además presentan ventajas, como que no precisan disponer de las coordenadas de los elementos relevantes del entorno de trabajo, no se producen errores de posicionamiento por una incorrecta calibración del robot o su entorno, y pueden resultar fáciles de aprender. Por último, dado que el programador no hace uso directo de las coordenadas de los objetos del entorno, la unidad de control no precisa de las funciones del modelado y control cinemático, desapareciendo, consecuentemente, los posibles problemas asociados a éste como la existencia de soluciones múltiples o los puntos singulares. Sin embargo, presentan una serie de inconvenientes, de los que el más destacable es la necesidad de utilizar al propio robot y su entorno para realizar la programación, obligando a sacar al robot de la línea de producción e interrumpiendo ésta. Otros inconvenientes frecuentes son la inexistencia de una documentación del programa y la dificultad de realizar modificaciones en el mismo, inconvenientes ambos que conducen a una difícil depuración y puesta a punto de las aplicaciones.

Como alternativa a la programación por guiado, el método de programación textual permite indicar la tarea al robot mediante el uso de un lenguaje de programación específico. Un programa se corresponde ahora, como en el caso de un programa general, con una serie de órdenes que son editadas y posteriormente ejecutadas. Existe, por tanto, un texto para el programa.

La programación textual puede ser clasificada en tres niveles: **robot**, **objeto** y **tarea**, dependiendo de que las órdenes se refieran a los movimientos a realizar por el robot, al estado en que deben ir quedando los objetos manipulados o al objetivo (o subobjetivo parcial) a conseguir.

En el estado actual, la programación de robots se queda materialmente en el primero de ellos (nivel robot), existiendo una gran cantidad de lenguajes de programación textual a este nivel.

De los tres, si es que se considera la programación, ¿cuál será considerado en el LVR y cuáles serán los recursos ofrecidos por el mismo?

Es importante destacar que en la actualidad es muy frecuente que los sistemas de programación de robots tiendan a combinar los dos modos básicos (guiado y textual), permitiéndose desarrollar el programa mediante la escritura de las instrucciones, y utilizando la posibilidad de guiado en línea en aquellos momentos en que sea necesario. Sistemas como RAPID de ABB, VAL II de Staübli y V+ de Adept Technology son ejemplos de esta ambivalencia.

Otras prestaciones que algunos robots incluyen, bien de serie o de manera opcional, en sus entornos de desarrollo son:

* la disponibilidad de herramientas para la programación fuera de línea (sin necesidad de disponer del robot),
* el uso de simuladores de la célula robotizada con posibilidad de generar el programa directamente para el robot real, o
* la existencia de herramientas para el desarrollo de la interfase humano-máquina.

Indudablemente, un método de programación de robots debe incluir la posibilidad de especificar el movimiento del robot. Además del punto de destino, es necesario especificar el tipo de trayectoria espacial que debe ser realizada, la velocidad media del recorrido y la precisión con que se debe alcanzar el punto destino. Incluso en ocasiones puede ser necesario indicar si el movimiento debe realizarse en cualquier caso o debe estar condicionado por algún tipo de circunstancia como, por ejemplo, la medida proporcionada por un sensor.

¿Cómo implementaremos lo anterior?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* + 1. **Cinemática Directa e Inversa**

¿Qué es la cinemática y por qué es importante en el estudio de la robótica industrial?

El problema de la cinemática directa es calcular la posición y orientación de la trama de la herramienta relativa a la trama base (¿sistema de referencia?). ¿A partir de?

El algoritmo de Denavit-Hartenberg (D-H), es un recurso para obtener de manera sistemática la matriz de transformación homogénea de un robot tipo serie, que define la geometría cinemática del robot. Son 4 valores a tomar en consideración en el algoritmo la finalidad de este análisis es para encontrar el valor en coordenadas cartesianas del efector final (TCP por sus siglas en inglés) (¿Y qué tiene que ver con el LVR?)

El problema de Cinemática Inversa se plantea de la siguiente manera: 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. Este es un problema fundamental en el uso práctico de los manipuladores.

1. **MoveIt para análisis de cinemática**: (¿Qué es MoveIt?) Utiliza una arquitectura basada en plugins para resolver la cinemática inversa mientras que proporciona una implementación nativa de cinemática directa. Normalmente estas tareas de resolver la cinemática se lo denominan en el blog de Moveit, Planeacion de trayectorias de un robot. Moveit! (s.f.)
2. **Librerías: Orocos\_KDL** Este paquete soportado por ROS merece una atención especial. Incluye la biblioteca KDL que tiene clases y funciones para los modelos cinemáticos y dinámicos de robots.

The Toolchain allows setup, distribution and the building of real-time software components. It is sometimes refered to as 'middleware' because it sits between the application and the Operating System. It takes care of the real-time communication and execution of software components.

The [Toolchain](http://www.orocos.org/toolchain)provides a limited set of components for application development. The Orocos Component Library (OCL) is a collection of infrastructure components for building applications.

¿Qué tienen que ver MoveIt y Orocos con el laboratorio virtual?

* + 1. **Espacio de trabajo del robot.**

1. Definición
2. Importancia
3. ¿Qué ofrece el LVR?
4. ¿Cómo de logrará 3?

El espacio de trabajo de un robot está definido como el grupo de puntos que pueden ser alcanzados por su efector-final, es de importancia el estudio de este concepto dentro de esta propuesta de trabajo para permitir al estudiante una cercanía a escenarios de estudios en las industrias donde se considera de importancia conocer el espacio de trabajo de un robot para evitar accidentes.

**reuleaux**: es un paquete básico que proporciona herramientas para crear varios tipos de mapas del alcance del robot y encontrar ubicaciones de base óptimas para una tarea determinada especificada mediante poses a través de esos mapas de accesibilidad

* 1. **Control de Movimiento en el entorno**

Para agregar plusvalía a esta propuesta de trabajo monográfico se pretende agregar el control de cada articulación del robot en el laboratorio virtual de forma independiente o “desacoplada” del resto de las articulaciones, usando habitualmente un controlador PID digital en cada articulación.

Sin implementar este control no se obtendrían los resultados apropiados después de ejecutar el experimento en el laboratorio virtual

* + 1. **PID loop**

El paquete del controlador PID es una implementación de un controlador Proporcional-Integral-Derivativo – La librería PID de ROS está diseñado para su uso en un problema de control directo que sólo necesita lanzar un bucle PID.

Las características de esta librería son:

* La reconfiguración dinámica de Kp, Ki y Kd facilita la sintonización directa.
* El filtro de paso bajo en la derivada de error con una frecuencia de corte parametrizada proporciona un término de derivado más suave.
* Soporte para múltiples controladores
* Modos Auto / Manual
* Simulaciones de plantas de primer y segundo orden permiten evaluar las características del controlador.
* Un auto-sintonizador Ziegler-Nichols

Podría considerarse en una ampliación del LVR

* 1. **Hardware externo**

Esta propuesta de trabajo monográfico tiene las intenciones de dejar claro la importancia de aprender a utilizar ROS debido a su capacidad tanto en el área de educación con aspectos teóricos en la robótica y en el área aplicativa de estos mismos conceptos usando un modelo de robot físico.

De tal forma objetivo del ROS es proveer librerías estandarizadas y confiables no solo para modelación de un proceso a base software, sino que la explotación de recursos de hardware que es utilizado en la industria tradicional al crear ROS INDUSTRIAL se proporcionó al diseño de sistemas robóticos interfaces ROS a muchos tipos de equipos industriales, incluyendo PLCs, Controladores de Robots, Servos, Interfaces hombre maquinas por sus siglas en ingles HMI, etc.

* + 1. **Joystick control**

Consiste en un conjunto de potenciómetros y botones con código de identificación que permitirá crear un nodo de comunicación de hardware hacia la aplicación de la interfaz del laboratorio y utilizarse para acercar a un usuario en el movimiento de un eslabón virtual del robot.

* 1. **Manual de uso del laboratorio virtual.**

El laboratorio virtual de robótica, que se propone desarrollar en el trabajo de monografía, contará con un manual de usuario con información suficiente para entender la estructura del LVR y de los recursos que ofrece. El manual es importante para el docente ya que le permitirá utilizar los recursos disponibles para diseñar experimentos relacionados con los manipuladores industriales. De igual informa, el manual contará con la información necesaria para que el estudiante conozca y utilice efectivamente los recursos del LVR en el desarrollo de sus experimentos.

¿Guías de laboratorio?

Consolidar aspectos teóricos (Cinemática del Robot), ingresando programas para movimiento de robots (Rutinas) Obteniendo la representación de datos y movimiento del robot en 3D, Salida de datos en gráficos XY (Gráficos en repuesta de movimiento de un eslabón, Velocidad, etc

**Hipótesis y Variables**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

El laboratorio virtual de robótica será el resultado de un proyecto de desarrollo ya que el producto será una aplicación de software que los estudiantes podrán utilizar para comprender y/o afianzar conocimientos básicos de la robótica industrial tales como estructura de un robot manipulador, cinemática directa e inversa, espacio de trabajo, entre otros.

Variables: Simulaciones, cálculos y modelos matemáticos, Curva de aprendizaje en robótica, multidisciplinar, plataforma virtual.

**Diseño Metodológico**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

El laboratorio virtual de robótica será desarrollado siguiendo las fases del método científico, análisis, hipótesis, síntesis, y validación, según lo establece Peter Bock en el libro Getting it right: R&D Methods in Science and Engineering. A continuación, una descripción breve de cada una de las fases.

El desarrollo del proyecto se basa en el modelo de la Figura 2 donde se toman en cuentan los cuatro protagonistas que intervienen en el desarrollo del laboratorio virtual siguiendo la fase de análisis de requisitos, fase de diseño, fase de desarrollo y fase de prueba y retroalimentación.

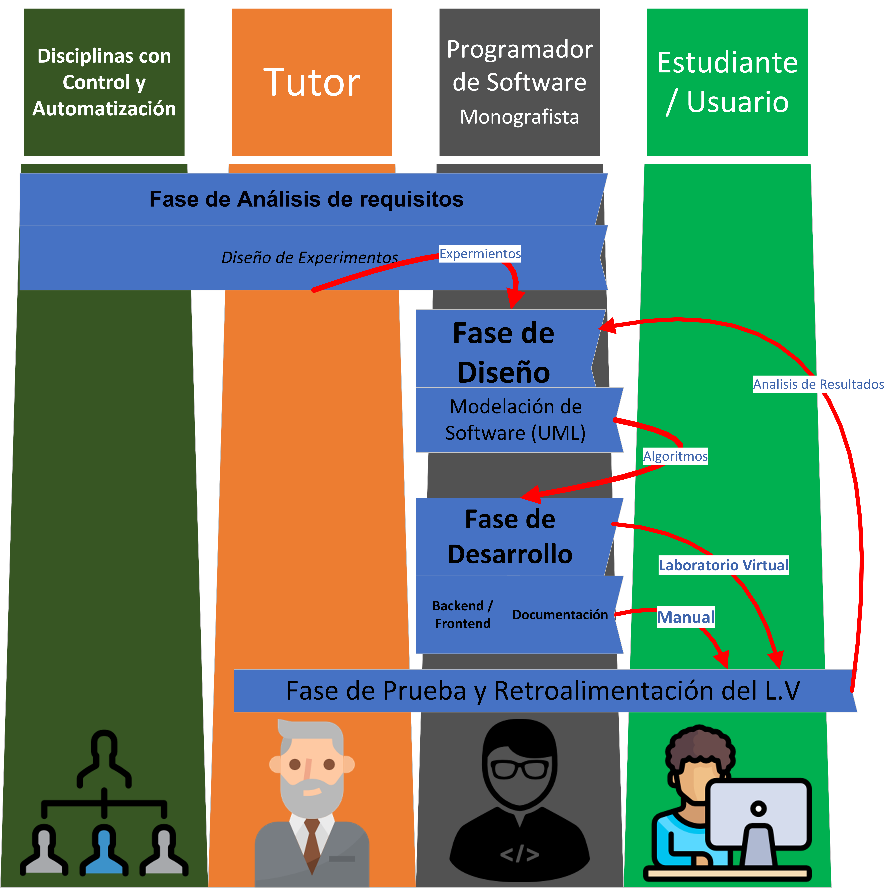


Fig.7.- Diseño metodológico del laboratorio Virtual.

* **Fase de Análisis:**

El laboratorio virtual de robótica estaría diseñado para ofrecer al estudiante un entorno que le permita comprender y verificar aspectos relacionados con la robótica industrial, por lo tanto, en la fase de análisis, con la ayuda del tutor, se determinarán los requerimientos que deberá satisfacer el LVR. Los requerimientos están asociados con los aspectos de la robótica industrial considerados importantes en la formación del estudiante en dicho campo, así como con los aspectos de la interfaz de usuario del LVR.

La fase de análisis contempla la revisión de literatura relacionada con los aspectos fundamentales de la robótica y el desarrollo de laboratorios virtuales incluido el diseño y construcción de las interfaces gráficas de usuario.

Entre otros resultados, de la revisión de literatura, se obtendrá información relacionada a las aplicaciones de software más utilizadas en el desarrollo de laboratorios virtuales en el campo de la robótica industrial. Dichos resultados serán evaluados para determinar los lenguajes de programación, simuladores, etc., apropiadas para el desarrollo efectivo y eficiente del LVR.

* **Fase de diseño**:

El resultado de la fase de diseño será una propuesta de la estructura del laboratorio virtual y la descripción de los recursos a utilizar en su implementación, ROS, lenguajes de programación, simuladores, etc.

* **Fase de desarrollo:**

En esta fase se implementa el laboratorio virtual de robótica y se diseñan y ejecutan los experimentos necesarios para verificar el funcionamiento del sistema. Los resultados de dichos experimentos serán utilizados para evaluar la respuesta del laboratorio virtual. Se implementarán los diferentes escenarios donde se realizarán los experimentos, la interfaz gráfica de usuario, el manual de usuario, y algunos ejemplos de guías de laboratorio.

* **Fase de validación**

Se determinará el desempeño del laboratorio virtual a partir de los resultados obtenidos en los experimentos y se formularán las conclusiones correspondientes. En esta etapa, se preparará el informe de resultados del proyecto.

* + **Construcción del contenido Virtual:** En desarrollo de software es común el uso de los términos Backend y Frontend los cuales aportan una perspectiva de la clasificación de tipo de Software a utilizar en la construcción del propio laboratorio virtual, se contempla el uso de ROS para manejo de datos en Backend y el uso de librerias de software libre para desarrollar GUI del lado de Frontend.
    - **Laboratorio virtual:** Esta aplicación de software permitirá centralizar los procesos vinculados en garantizar la compresión teórica y práctica del estudiante al ejecutar las diferentes pruebas propias de la robótica esto se logrará con:
      * Desplegarse diferentes menús o GUI para introducir datos o configuraciones.
      * Simuladores de terceros que gestionara la aplicación de software permitiendo al estudiante verificar visualmente las practicas teóricas.
      * Menús de interfaz de programación de scripts que permitirán hacer cambio en el comportamiento de las simulaciones.
      * Configuración de herramientas de hardware externas para el movimiento orientado por el usuario de un objeto simulado.
    - **Entorno de Visualización en Simuladores:** Se procederá a evaluar el simulador parametrizando los motores gráficos y motores físicos virtuales (Simular la gravedad) Todo esto con el fin de simular un entorno lo más realista posible (diseño del mundo virtual).
* **Gestión de infraestructura de hardware-software:** Es necesario adquirir el equipo adecuado para ejecutar programas, simulaciones y conexión del hardware externo compatible (Uso de Gamepad USB para mover el robot virtual). (¿Qué significa?)
* **Diseño de las guías de aprendizaje:** Se agregará un documento guía basado en la teoría que se desea manejar en el laboratorio para desarrollar las competencias.
* **Fase de pruebas:** En esta fase ~~es necesario someter~~ el diseño de software a una evaluación dirigida por personal afín a la materia el cual pueda dar retroalimentación como sugerencias o inconformidad del diseño. La fase de diseño es concebida como un ciclo que se acaba cuando se logra alcanzar los objetivos, si es necesario re-diseñar y volver a la fase de desarrollo, esto se decidirá en esta fase de pruebas. (chequear los modelos del ciclo de vida)

~~Si se procede a dar el visto bueno~~ es necesario documentar las experiencias, aspectos técnicos y recomendaciones en el documento de guía de uso del software para el usuario.

Además del lanzamiento todo software necesita de evaluaciones periódicas de su desempeño funcional y operativo estas pruebas de seguimiento permitirán a los desarrolladores futuros del proyecto en pensar en nuevas estrategias de actualizaciones del propio software.

**Cronograma de ejecución**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Marzo** | | | | **Abril** | | | | **Mayo** | | | | **Junio** | | | | **Julio** | | | |  | |
| **Semana** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |  |
| Fase Análisis de Requisitos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fase de Diseños |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fase de Desarrollo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Fase de pruebas |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Lanzamiento de la plataforma |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Elaboración del informe |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Defensa del Trabajo Monográfico |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Bibliografía**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[1] Perez, V., & Mayta, R. (2001). ACTUALIDAD Y PERSPECTIVAS DE LA ROBÓTICA Obtenido de Universidad Nacional Mayor de San Marcos, Lima, Peru Facultad de Ingeniería Industrial: http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/indata/v04\_n1/act

[2] Jara, C., et al. (2011). Hands-on experiences of undergraduate students in Automatics and Robotics using a virtual and remote laboratory. *Computers & Education*, 57(4), 2451–2461. Obtenido de https://doi.org/10.1016/j.compedu.2011.07.003

[3] Candelas, F., Torres, F., Gil, P., Ortiz, F., Puente, S., & Pomares, J. (2004). Laboratorio virtual remoto para robótica y evaluación de su impacto en la docencia. *RIAI: Revista Iberoamericana de Automática e Informática Industrial*, 1(2), 49–57. Obtenido de http://rua.ua.es/dspace/handle/10045/4609

[4] Torres, F., et al. (2006). Experiences with virtual environment and remote laboratory for teaching and learning robotics at the university of alicante. *International Journal of Engineering Education*, 22(4), 766–776. Obtenido de http://rua.ua.es/dspace/handle/10045/6231

[5] Castellanos, F., & Martínez, O. (2010). Laboratorios virtuales (LV) como apoyo a las practicas a distancia y presenciales en Ingenieria. *INGE CUC, 6(1), 267-280.* Obtenido de http://revistascientificas.cuc.edu.co/index.php/ingecuc/article/view/311

[6] Álvarez, V., Paule, R., & Gutiérrez, M. (2008). Presente y futuro del desarrollo de plataformas Web de elearning en educación superior. En V Simposio Pluridisciplinar sobre Diseño y Evaluación de Contenidos Educativos Reutilizables. Universidad Pontificia de Salamanca. Disponible en http://www.web.upsa.es/spdece08/contribuciones/118\_SPEDECE08Revisado.pdf

[7] Bodekaer, M. (2015, October). This virtual lab will revolutionize science class. [Video]. Obtenido de https://www.ted.com/talks/michael\_bodekaer\_this\_virtual\_lab\_will\_revolutionize\_science\_class

[8] Schuster, K., Groß, K., Vossen, R., Richert, A., & Jeschke, S. (2016). Preparing for industry 4.0–collaborative virtual learning environments in engineering education. In *Automation, Communication and Cybernetics in Science and Engineering 2015/2016,* 417 - 427. doi 10.1007/978-3-319-42620-4\_33

[9] Ortega, J., Sánchez, R., González, J., & Reyes, G. (2016). Virtual laboratories for training in industrial robotics. IEEE Latin America Transactions, 14(2), 665-672. doi 10.1109/TLA.2016.7437208

[10] Pinto, M., Barrera, N., & Pérez, W. (2010). USO DE LA ROBÓTICA EDUCATIVA COMO HERRAMIENTA EN LOS PROCESOS DE ENSEÑANZA. Ingeniería Investigación y Desarrollo, 10(1) 15 – 23 Obtenido de http://revistas.uptc.edu.co/index.php/ingenieria\_sogamoso/article/view/912

[11] Schmidgen, H., Dierig, S., & Kantel, J. (2000). The Virtual Laboratory for Physiology. Max Planck Institute for the History of Science, Obtenido de.

[12] CNU (2018, abril) Repositorios Nicaragua. En Wellcome to the university repository of the CNU Obtenido de http://repositorio.cnu.edu.ni/

[13] Ramírez, K. (2018, abril) Material del Curso CI-2657. En Lista de Presentaciones Obtenido de http://www.kramirez.net/ci-2657/materialci2657/

[14] COSTARICAMAKERS. (2018, abril) Tag Archives: ROS. YA CONTAMOS PERSONAS… AHORA HAGAMOS ALGO CON PYTHON Obtenido de http://costaricamakers.com/?tag=ros

[15] Valverde, S. (2015). “Robótica inteligente: Implementación de sensores 3D para

desenvolvimiento de robots móviles y vehículos autónomos” Obtenido de https://repositoriotec.tec.ac.cr/bitstream/handle/2238/6932/robotica\_inteligente\_implementaci%C3%B3n\_sensores\_desenvolvimiento.pdf?sequence=1&isAllowed=y.

[16] Rosas, L., Fuentes, M., Samaniego, C., Alvarado, R., & Valencia, J. (2013). MODELADO Y CONTROL DEL ROBOT MÓVIL ROBOTNIK SUMMIT XL. Obtenido de http://cerescontrols.com/wp-content/uploads/2013/10/SUMMIT-XL-Informe-Final.pdf

[17] Justina (2017, mayo) Justina. En user\_manual Obtenido de

https://github.com/RobotJustina/JUSTINA/blob/master/user\_manual/user\_manual.pdf

[18] Eagle X (2018, marzo) Un robot construido por alumnos del TEC. En Vinculación y prestigio Obtenido de https://tec.mx/es/noticias/queretaro/vinculacion-y-prestigio/un-robot-construido-por-alumnos-del-tec

[19] Juárez, G. (2008) IMPLEMENTACIÓN DE UN LABORATORIO VIRTUAL CON LA AYUDA DE LABVIEW, AL CURSO DE CIRCUITOS ELÉCTRICOS 1. Obtenido de http://biblioteca.usac.edu.gt/tesis/08/08\_0148\_ME.pdf

[20] Prieto, R., Zaldivar, U., & Bernal, R. (2010) Creación de un Laboratorio Virtual para Optimizar el uso de un Laboratorio de Robótica Real. Obtenido de https://www.academia.edu/374898/Creaci%C3%B3n\_de\_un\_Laboratorio\_Virtual\_para\_Optimizar\_el\_uso\_de\_un\_Laboratorio\_de\_Rob%C3%B3tica\_Real

[21] Ayala, J., Pupo, L., & Salazar, L. (2016) LABORATORIO VIRTUAL DE INSTRUMENTACIÓN Y CONTROL Obtenido de http://www.informaticahabana.cu/sites/default/files/ponencias/EDU095.pdf

[21] Owen, A. (14 de marzo de 2016). What is the Best Programming Language for Robotics? [Entrada en un blog] Robotiq. Recuperado de https://blog.robotiq.com/what-is-the-best-programming-language-for-robotics

[22] Tellez, R. (2017). A thousand robots for each student: Using cloud robot simulations to teach robotics. En Robotics in Education, 143-155. Springer, Cham.

[12] Pisani, U., Cambiotti, F., Corinto, F., & Romano, G. (2007). SWILAB: A virtual laboratory for electronics. *Proceedings Education and Information Systems: Technologies and Applications (Eista,’03)*, 1-2.

[13 ] Tsardoulias, E., & Mitkas, P. (2017). Robotic frameworks, architectures and middleware comparison. Obtenido de https://arxiv.org/pdf/1711.06842.pdf

~~[13] Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V., & Hart, J. C. (1992). The CAVE: audio visual experience automatic virtual environment. Communications of the ACM, 35(6), 64-73.~~

~~[14] Hoffmann, M., Meisen, T., & Jeschke, S. (2016). Shifting Virtual Reality Education to the Next Level–Experiencing Remote Laboratories Through Mixed Reality. In Automation, Communication and Cybernetics in Science and Engineering 2015/2016 (pp. 293-307). Springer International Publishing.~~

[14] MoveIt! (s. f.) Concepts. En Documentacion Obtenido de http://moveit.ros.org/documentation/concepts/

~~[15] Bourke, C. (2017, Febrero). Computer science, arts students combine for VR course. [Video]. Obtenido de http://news.unl.edu/newsrooms/today/article/computer-science-arts-students-combine-for-vr-course/~~

[16] Candelas, F., Puente, T., Torres, F., Segarra, V., & Navarrete, J. (2005). Flexible system for simulating and tele‐operating robots through the internet. Journal of Field Robotics, 22(3), 157-166. doi 10.1002/rob.20056

[17] Corke, P. I. (1996). A robotics toolbox for MATLAB. IEEE Robotics & Automation Magazine, 3(1), 24-32. doi 10.1109/100.486658

[18] Staranowicz, A., & Mariottini, G. (2011). A survey and comparison of commercial and open-source robotic simulator software. In Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments (p. 56). ACM.

[19] ROS (s. f.) Core components. En Communications Infrastructure Obtenido de http://www.ros.org/core-components/

[20] robot\_localization (2018). [fotografía]. Recuperado de https://www.cra.com/work/case-studies/robotlocalization

[20] Pitzer, B., Osentoski, S., Jay, G., Crick, C., & Jenkins, O. C. (2012, May). Pr2 remote lab: An environment for remote development and experimentation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* 3200-3205. IEEE.

[21] Casañ, G. A., Cervera, E., Moughlbay, A. A., Alemany, J., & Martinet, P.(2015). *ROS-based online robot programming for remote education and training. In Robotics and Automation (ICRA), 2015 IEEE International Conference on 6101-6106.*

[22] Universidad Tecnológica La Salle (2017) Ingeniería en Mecatrónica y Sistemas de Control. En Plan de estudio Obtenido de http://www.ulsa.edu.ni/index.php/ingenieria-en-mecatronica-y-sistemas-de-control

[23] Calvo, I., Zulueta, E., Gangoiti, U., López, J. M., Cartwright, H., & Valentine, K. (2009). *Laboratorios remotos y virtuales en enseñanzas técnicas y científicas*(Vol. 3, No. 3, pp. 1-21). Ikastorratza.

[24] Perez, X., Cerda, A., & Incer, W. (2016) Desarrollar un laboratorio virtual para la realización de prácticas de laboratorio en la disciplina de control automático y automatización industrial

[25] DDE (2015, mayo) Mision. En Todo sobre DDE Obtenido de http://www.dde.uni.edu.ni/UNI\_Direccion\_de\_Desarrollo\_Educativo/index.php/definicion/78-todo-sobre-dde

[26] ROS Industrial (2016, noviembre) ROS QTC Plugin. En Repositorio Obtenido de https://github.com/ros-industrial/ros\_qtc\_plugin

[14] ROS-Industrial Tutorials. (2017, Marzo) En ROS-Industrial Obtenido de http://wiki.ros.org/Industrial/Tutorials

[14] Willow Garage. (2010, Abril) En Comic: Reinventing the Wheel [Entrada en un blog] Obtenido de http://www.willowgarage.com/blog/2010/04/27/reinventing-wheel

[27] Repositorio Universitario de Nicaragua. En Buscador CNU Obtenido de http://repositorio.cnu.edu.ni/

[28] Juárez, G. (2008) IMPLEMENTACIÓN DE UN LABORATORIO VIRTUAL CON LA AYUDA DE LABVIEW, AL CURSO DE CIRCUITOS ELÉCTRICOS 1.

[29] Merino, E. (2015) DISEÑO DE UN SIMULADOR DE COMPILADOR PARA PLATAFORMA MOODLE E IMPLEMENTACIÓN DE UN LABORATORIO VIRTUAL PARA LA ENSEÑANZA DE PROGRAMACIÓN.

[30] Guerrero, L., Gómez, D., Sandoval, E., Thomson, P., Marulanda, J. (2014). SISMILAB, UN LABORATORIO VIRTUAL DE INGENIERÍA SÍSMICA, Y SU IMPACTO EN LA EDUCACIÓN.

[31] Tellez, R. (2016) A thousand robots for each student: using cloud robot simulations to teach robotics.

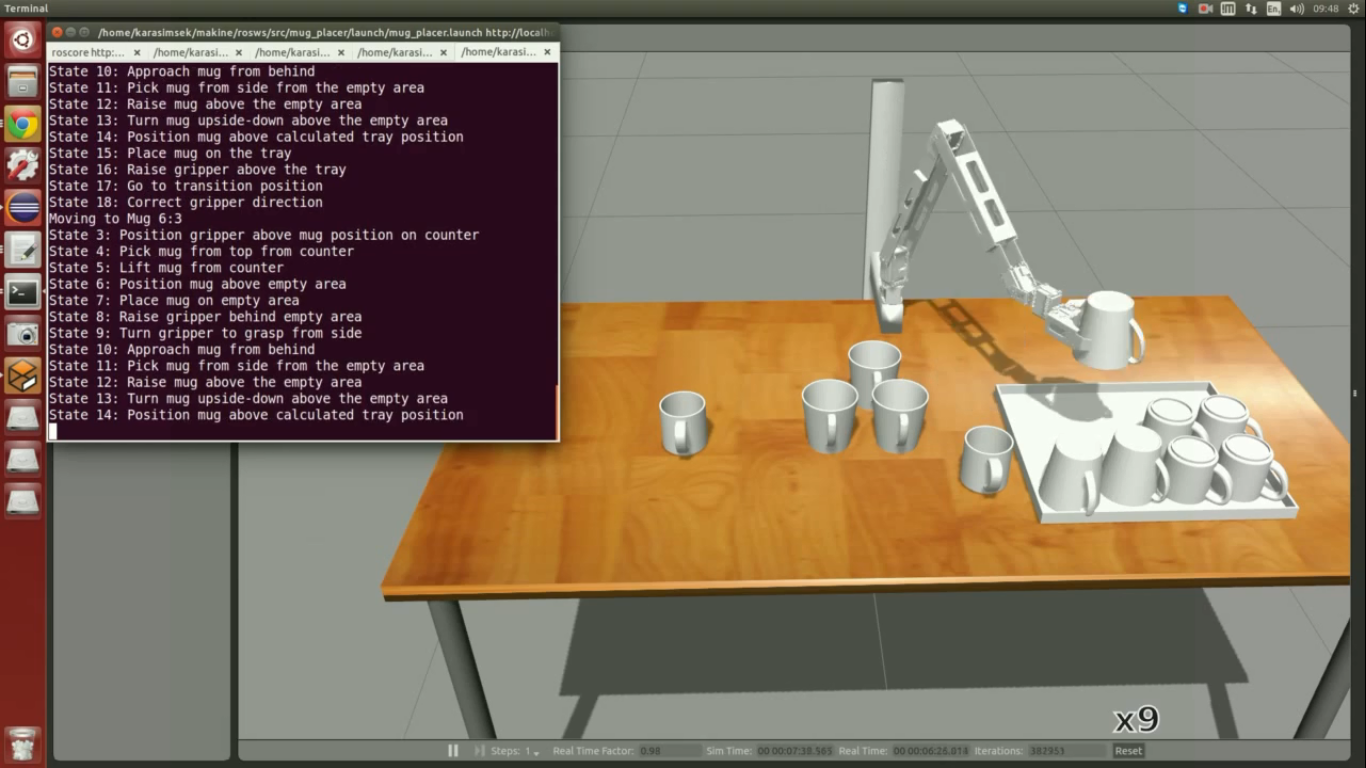
[32] [ARTE: A ROBOTICS TOOLBOX FOR EDUCATION](http://arvc.umh.es/arte/index.html) Obtenido de http://arvc.umh.es/arte/index\_en.html

[33] Mato, M. (2014). Simulación, control cinemático y dinámico de robots comerciales usando la herramienta de Matlab, Robotic Toolbox. Obtenido de <http://uvadoc.uva.es/handle/10324/13187>

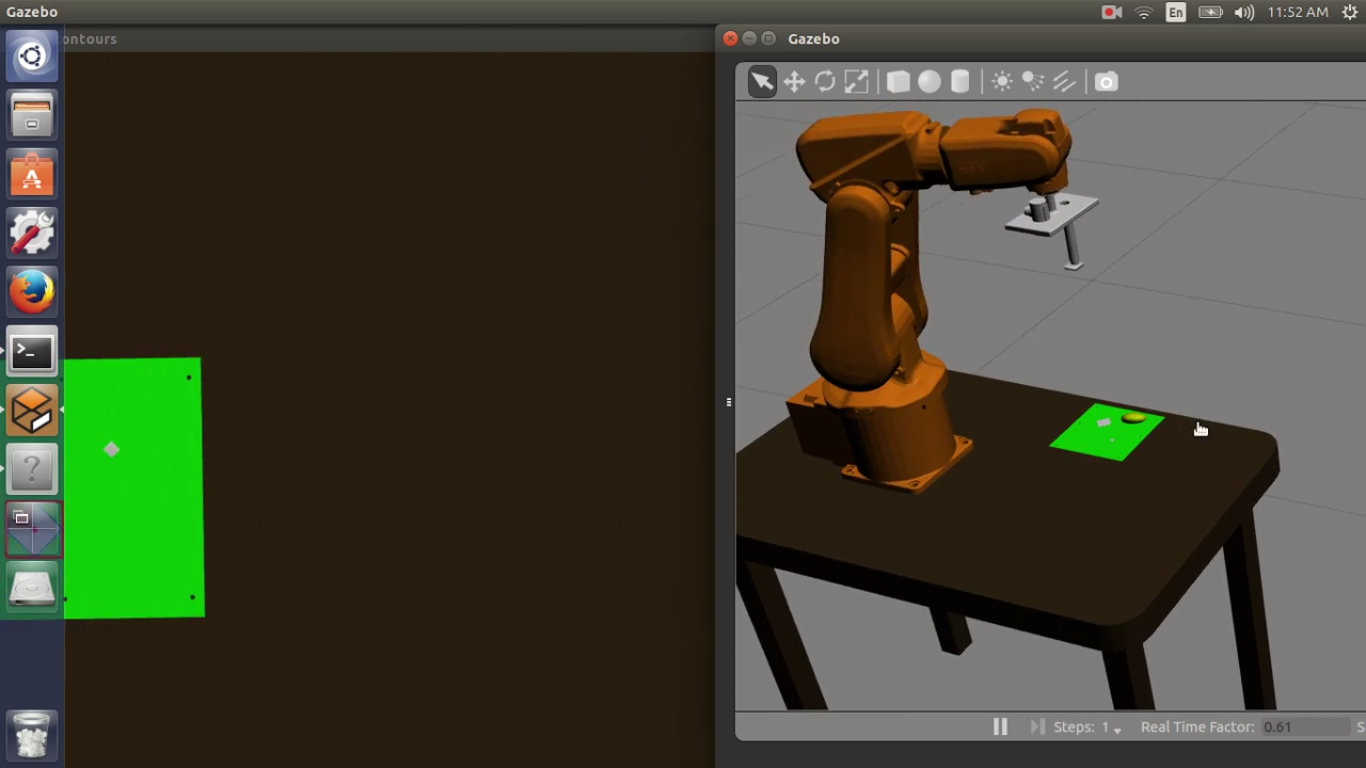
[34] ROBOTIS Co., Ltd. (2017).*ROS Robot Programming*. Obtenido de https://community.robotsource.org/t/download-the-ros-robot-programming-book-for-free/51

**Anexos**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **Trabajos relacionados con ROS y Gazebo.**

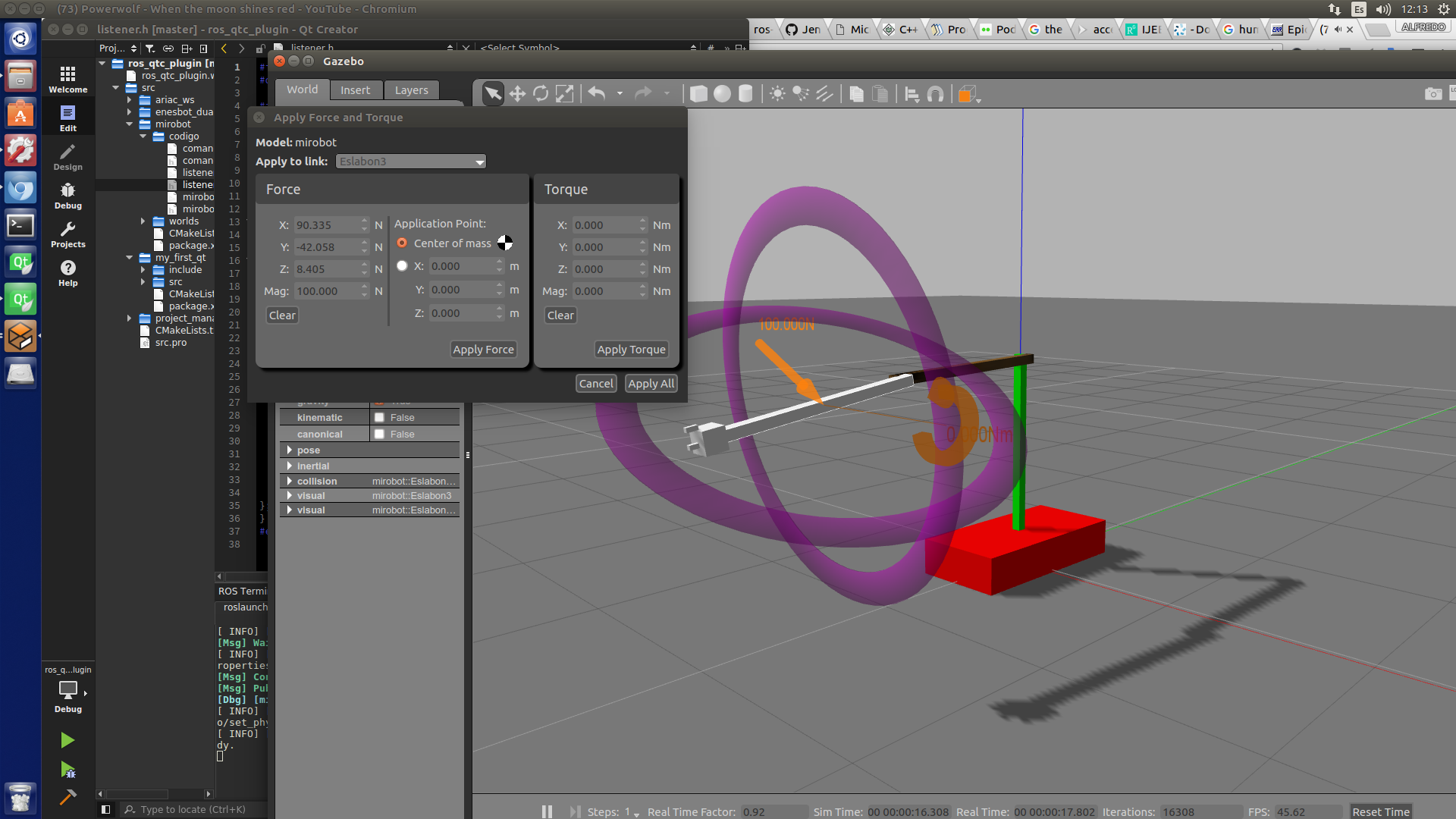


A.3: Captura de pantalla trabajos relacionados en simulación usando ROS y Gazebo para practica de Pick and Place.



A.4: Captura de pantalla trabajos relacionados en simulación usando ROS y Gazebo para practica de Visión simulada.

* **Pruebas realizadas con ROS y Gazebo.**



A.5: Captura de pantalla Pruebas realizadas en simulación usando ROS y Gazebo para practica de aplicar velocidad a un eslabón.