



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
ANTECEDENTES.....	6
JUSTIFICACIÓN.....	10
1. OBJETIVOS.....	12
1.1. Objetivo general.....	12
1.2. Objetivos específicos.....	12
2. MARCO TEÓRICO.....	13
2.1. Tipos de Laboratorios.....	13
2.1.1. Laboratorios Físico.....	13
2.1.2. Laboratorios Virtual.....	14
2.1.3. Laboratorio Remoto online.....	14
2.1.4. Realidad Virtual 3D.....	14
2.2. Modelo de utilización del laboratorio virtual para robótica industrial.....	16
2.3. Sistema operativo para robot (ROS).....	17
2.3.1. Conceptos básicos de ROS.....	18
2.3.2. Lenguajes de programación soportados.....	20
2.3.3. Aplicaciones y librerías.....	21
2.3.4. Herramientas de ROS.....	21
2.4. Interfaz gráfica de usuario (GUI)	21
2.4.1. Interacción dentro la GUI.....	22
2.4.2. Visualización de datos.....	22
2.4.2.1. Visualización 3D de robots.....	22
2.4.2.2. Gráficos de dispersión XY.....	22
2.4.3. Programación de Rutinas de robots.....	23
2.4.4. Información de los fundamentos de Robótica Industrial.....	23
2.5. Modelos de robots en el LVR.....	24
2.5.1. Modelado de brazos robóticos de diferentes DOF.....	24
2.6. Entornos de simulación de robots.....	25
2.6.1. Software de compañías basadas en CAD	26
2.6.2. Software proveído por los fabricantes de robots.....	26
2.6.3. Softwares de programación de varios fabricantes.....	26
2.6.4. Software Simuladores de Robots de propósito general.....	27
2.6.4.1. Simuladores de propósito general comerciales.....	27
2.6.4.2. Simuladores de propósito general de uso libre.....	28

2.7. Modelación Cinemática y Dinámica.....	29
2.7.1. Cinemática Directa e Inversa	29
2.7.2. Espacio de trabajo del robot.....	29
2.8. Control de movimiento en el entorno.....	30
2.8.1. PID loop.....	30
2.9. Hardware externo.....	30
2.9.1. Joystick control.....	31
2.10. Manual de uso del laboratorio virtual.....	31
3. HIPÓTESIS Y VARIABLES.....	32
4. DISEÑO METODOLÓGICO.....	33
5. CRONOGRAMA DE EJECUCIÓN.....	36
6. BIBLIOGRAFÍA.....	37
ANEXOS.....	42

Introducción

La robótica es uno de los campos de la tecnología que (evoluciona) se desarrolla velozmente y encuentra 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 conocer 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 tan 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 desarrollar, 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 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 herramienta 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 universal 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 un manual de usuario el cual contará con la información necesaria para que los docentes pueden elaborar una práctica sobre los fundamentos de la robótica, por ejemplo, una guía para entender los fundamentos de la cinemática inversa en la robótica.

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), dichos lugares no cuentan con información sobre el mismo a pesar su importancia. La primera versión de ROS fue lanzada en el 2007. 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 como 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 Ingeniería, 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 el programa para lograrlo a los estudiantes. 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 SISMLAB 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 en dar 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 saber 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.

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 beneficiarios de los resultados de este proyecto serán los docentes y los estudiantes. Los primeros dispondrán de una herramienta a partir de la cual 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 un beneficiario 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

1.1 Objetivo general

Mostrar la importancia del sistema operativo para robots (ROS) mediante el desarrollo de un laboratorio virtual para el estudio de los fundamentos de la robótica industrial.

1.2 Objetivos específicos

- Identificar las principales características/recursos/estructura de ROS enfatizando aquellas aplicables directamente a la robótica industrial.
- Generar 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.

2. Marco Teórico

Las instituciones académicas, principalmente las de educación superior, 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 instituciones de educación superior, 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. Es una herramienta que puede presentar altos niveles de flexibilidad y la inversión requerida 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.

2.1 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.

2.1.1 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 difícil 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).

2.1.2 Laboratorio Virtual

Es una alternativa a laboratorios presenciales al utilizar recursos computacionales haciendo uso de modelos matemáticos y recursos de visualización como modelos CAD y animaciones gráficas. Con la premisa de acercar a un estudiante a lo más aproximado de un sistema real, pero siendo simulado.

2.1.3 Laboratorio Remoto online

Nacen bajo la necesidad de dar acceso a un conjunto de estudiantes a un espacio de trabajo de forma online que combina recursos de Laboratorios físicos con hardware y recursos de software procediendo al enlace con una página web para su control llevar a cabo el experimento de interés. Su aprovechamiento está basado en la accesibilidad por parte de un usuario en horarios flexibles.

2.1.4 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 Físicos, 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 lo tendrá como elemento integrador. Se debe agregar el hecho relacionado a la inversión baja relativa requerida para tal fin.

2.2 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).

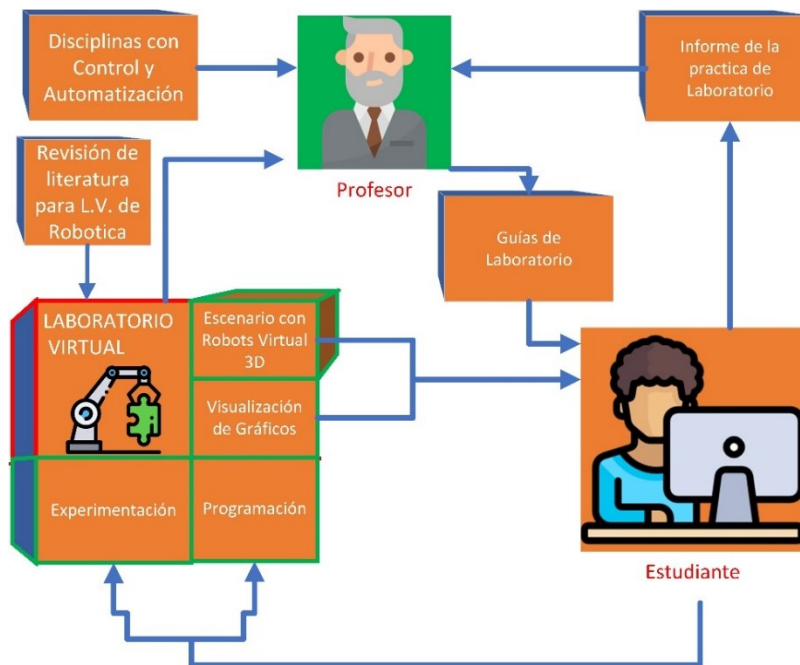


Fig.1.- Modelo de utilización del laboratorio virtual de robótica industrial. (revisarlo)

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.

2.3 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 desarrollado en el 2007 en los laboratorios de Inteligencia Artificial de Stanford. En el 2008 cedieron el derecho al desarrollo y continuación de este al instituto de investigación de robótica Willow Garage, donde la filosofía es proporcionar bibliotecas y herramientas para ayudar a los desarrolladores de software a crear aplicaciones para robots

ROS en esta propuesta de trabajo monográfico, será la herramienta por promover y la cual nos proveerá de los recursos computacionales necesarios para desarrollar un Laboratorio Virtual de Robótica, para el estudio de los fundamentos de la robótica industrial, que permita valorar la potencialidad del middleware ROS.

Una característica importante de ROS es su código abierto (Open Source) lo cual permite el desarrollo a gran escala de complejos sistemas robóticos, ya sea físicos o simulados **esto debido a que ROS cuenta con una amplia colección de herramientas, librerías y convenciones que han sido estructuradas y desarrolladas por muchos colaboradores a nivel mundial.**

Al diseñar un sistema robótico es necesario la búsqueda de trabajos relacionados, como lo define Willow Garage (2010), un roboticista encuentra un trabajo relacionado con el desarrollo de un sistema de robots que le sirva para incursionar en las áreas académicas o de desarrollo de interés del propio roboticista, este sigue las

indicaciones descritas en el documento base para tratar de obtener los mismo resultados, pero este no puede dar con los detalles puntuales para que el software reutilizado que él está utilizando le permita obtener los mismo resultado volviéndose el desarrollo del proyecto tedioso y complicado, lo cual obliga al roboticista escribir o rediseñar la idea inicial del proyecto por la falta de información, este se ve en la tarea de buscar nuevas alternativas para hacerse de la solución que esperaba obtener al usar software reutilizado, debido a los plazos estipulados de entrega de resultados en cada trabajo de desarrollo este se ve obligado a reducir incluso la calidad del trabajo que este esperaba obtener y esto deja como resultado la finalización del proyecto con retrasos, o incluso proyectos fallidos donde el roboticista si publica una guía de su trabajo elaborado acarreará a los siguientes desarrolladores que se avalaran a su trabajo los mismos problemas que él tuvo al desarrollar este proyecto todo esto se puede resumir a la falta de estandarización de los trabajos con robótica que si bien sabemos estos incluyen desarrollo de software y hardware.

Como resultado, ROS ha creado convenciones o buenas prácticas de desarrollo de software para fomentar la reutilización de código para robots. En otras palabras, a como lo define ROS (s.f.) fue diseñado específicamente para que grupos puedan colaborar y construir sobre el trabajo del otro, y definir así una arquitectura robótica totalmente sólida y funcional.

2.3.1 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 lo requerido según sus necesidades. Es decir, el usuario puede seleccionar las partes que necesita para implementar su solución.

En su operación, ROS presenta una red de procesos que se ejecutan simultáneamente en una estructura peer-to-peer y los conceptos básicos relacionados son los siguientes:

1. **Nodos** (nodes): Son los procesos que realizan el cálculo o procesamiento de la información.
2. **Maestro** (master): Proporciona la información necesaria para que todos los nodos sean identificados y se comuniquen.
3. **Mensajes** (messages): forma de comunicación entre nodos.
4. **Tópico** (topic): es un nombre utilizado para identificar el contenido de un mensaje.
5. **Servicios** (services): Es el 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.

La arquitectura básica de comunicación de ROS, usando 3 nodos como ejemplo simplificado y el propio nodo máster, es mostrada en la figura 2. Para establecer comunicación con diferentes nodos es necesario la utilización de registros para indicar al máster que tipos de mensajes se recibe y emite entre los diferentes nodos.

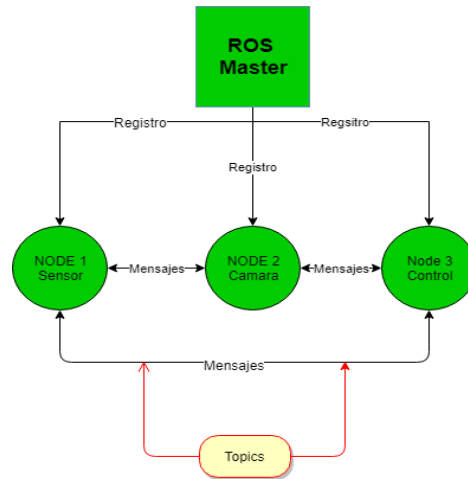


Fig.2.- Esquema de funcionamiento de la Arquitectura de ROS

El middleware ROS funciona bajo comunicación peer-to-peer para los procesos que pueden ser soportados tales como nodos, mensajes, tópicos, servicios y base de datos (bags) propias de ROS. Por ejemplo, en la Figura 2 un nodo puede estar procesando información de un sensor, otro la información de una cámara y el otro ejecuta el procesamiento del control de ciertos actuadores, para ello cada proceso se comunica usando mensajes. 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.

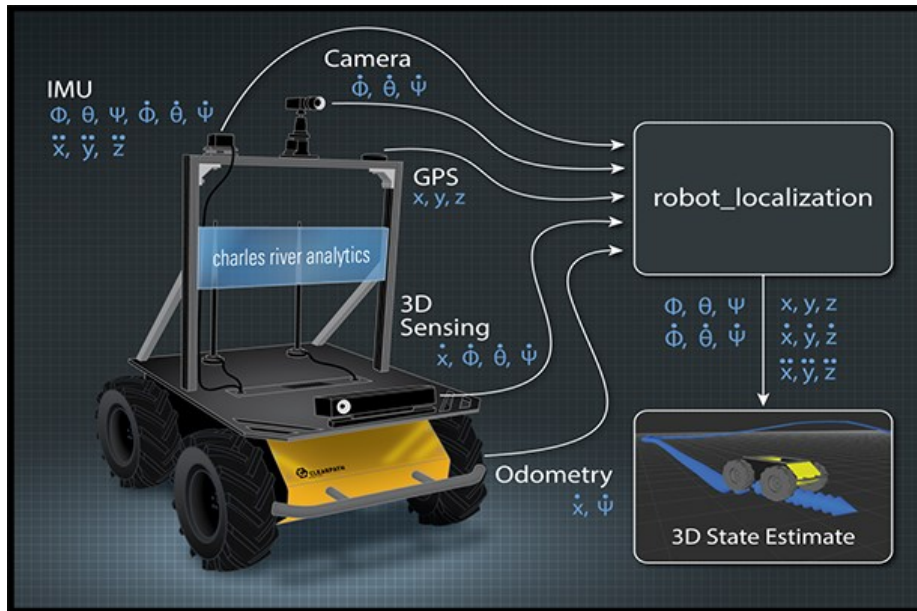


Fig.3. - Nodo robot_localization recibiendo tópicos de un robot físico y mostrando su localización en el ambiente virtual.

En la figura 3 se muestra el nodo robot_localization el cual recibe información vía mensaje de varios sensores bajo los tópicos: IMU, Camera, GPS, Detección 3D (Sensing 3D) y odometria (odometry) donde la mayoría de estos mensajes comparten coordenadas de posicionamiento el cual el nodo procesa 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.

2.3.2 Lenguajes de programación soportados

En el desarrollo del software, ROS permite el uso de distintos lenguajes de programación. De manera oficial soporta Python, C++ y LISP, Java podría ser soportada en el futuro pero en la actualidad se encuentra en una fase experimental, apoyada por Google.

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 suscribirse a un tópico, o entre otro proceso de ROS como Servicios o almacenamiento de datos (Bags). Con ROS es posible utilizar nodos creados con otros lenguajes de programación tales como Phytion y C++, los cuales son utilizados para la comunicación entre distintas aplicaciones, por ejemplo, para enviar la información de la posición del actuador de un robot y que este ejecute el movimiento (mediante simulación).

2.3.3 Aplicaciones y librerías

ROS ofrece herramientas para la visualización, análisis y registro de datos de una aplicación (tanto en un robot físico como simulado). Todas las aplicaciones que se comentan a continuación son nodos en el sistema de suscripción de ROS, con sus topics y servicios para permitir entrada y salida de datos a procesar.

- a) **RVIZ**: es un entorno de visualización 3D que permite combinar en una misma pantalla modelos de robots, datos de sensores (cámara, láser, etc.) y otros datos en 3D.
- b) **RQT**: El plugin rqt ofrece la introspección y la visualización de datos provenientes de procesos de ROS, que muestra los nodos y las conexiones entre ellos, lo que le permite fácilmente depurar y entender el sistema, su funcionamiento y como se estructura. Ejemplo Mostrar gráficos de línea XY de datos provenientes de nodos.

2.3.4 Herramientas de ROS

ROS proporciona de una integración con programas y aplicaciones muy populares, como Gazebo, MoveIt, OpenCV y otros proyectos de código abierto. Donde ROS utiliza la información generada por estas aplicaciones externa, desde sensores virtuales, información de análisis de reconocimiento de imagen, etc. Todo esto para proveerle al sistema en desarrollo de mayores funcionalidades.

Un ejemplo es utilizar OpenCV para reconocimiento de imágenes y tomar decisiones de cambio de posición si se encuentra un obstáculo cerca en la simulación o implementación física.

2.4 Interfaz gráfica de usuario (GUI)

Una GUI es una interfaz gráfica de un programa. Una buena GUI puede hacer que los programas sean más fáciles de usar proporcionándole al usuario una apariencia coherente y con controles intuitivos como botones, cuadros de lista, controles deslizantes, menús, etc. La GUI debe comportarse de una manera comprensible y predecible, de modo que un usuario sepa qué esperar cuando realiza una acción.

Para facilitar el uso de la plataforma del laboratorio virtual se incorporará una Interface gráfica de usuario, (GUI por sus siglas en inglés) para garantizar al estudiante de una práctica lo mas transparente posible sin que este tenga que configurar los aspectos básicos de ROS tales como la inicialización de ros, lanzamiento de nodos, etc. Para esto se leería la información de los nodos con la

información de interés del laboratorio virtual, para garantizar la comprensión de robótica por parte del alumno, permitir mostrar gráficas, muestra de datos, etc.

Para garantizar esta GUI podemos utilizar lenguaje orientado a objetos como C++ y Python para crear un aplicación de software. Es necesario dejar claro que en esta propuesta de trabajo se recomienda usar un entorno de desarrollo integrado IDE por sus siglas en inglés, para el desarrollo de Software tanto en la programación y validación de los nodos que garantizaran la funcionalidad y lógica detrás del laboratorio virtual y además de esta misma forma el IDE debe garantizar la creación de la GUI que provean al usuario un entorno amigable donde le permita la interacción, Visualización 3D de modelos matemáticos de robots, programación de rutinas de robots y administración de simuladores externos a la GUI.

2.4.1 Interacción dentro la GUI.

Una interfaz gráfica de usuario proporciona al usuario un entorno familiar en el que trabajar. Este entorno contiene botones, botones, listas, menús, cuadros de texto, etc., que ya son familiares para el usuario, para que pueda concentrarse en usar la aplicación exclusivamente y sin preocuparse en la configuración de ROS y sus respectivos nodos para visualizar datos, visualizar modelos virtuales 3D y Comportamiento de las simulaciones de tal forma que podría considerarse un diseño de software que sea transparente al usuario y permita la rápida familiarización en las funcionalidades de este mismo.

2.4.2 Visualización de datos.

Dentro de la propia GUI de Usuario se adoptarán las herramientas o aplicaciones de desarrollo nativas de ROS (Véase apartado 2.1.3), esta adaptación a la GUI es posible por la naturaleza de software libre que posee ROS el cual permite tomar las librerías propias (Código fuente) de estas herramientas que sirven de modo general para depurar, visualización de variables y agilización del desarrollo de Soluciones en una aplicación específica de robots.

2.4.2.1 Visualización 3D de robots

Para garantizar la visualización de los modelos matemáticos virtuales 3D de los robots tipo serie considerados en esta propuesta de trabajo dentro de la propia GUI es necesario tener una ventana de visualización el cual será garantizado por la herramienta/Aplicación RVIZ (Véase apartado 2.1.3), de ROS del cual se tomarán sus librerías (código fuente) y se adaptara a los requerimientos propuestos dentro del laboratorio virtual. Ver Figura 5.

2.4.2.2 Gráficos de dispersión XY

De igual forma que es adoptable las librerías de visualización de un modelo de robot, también ROS posee la herramienta RQT que permite la depuración y visualización de los datos provenientes de nodos de una aplicación específica para robots y

adaptarla a la GUI propia 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

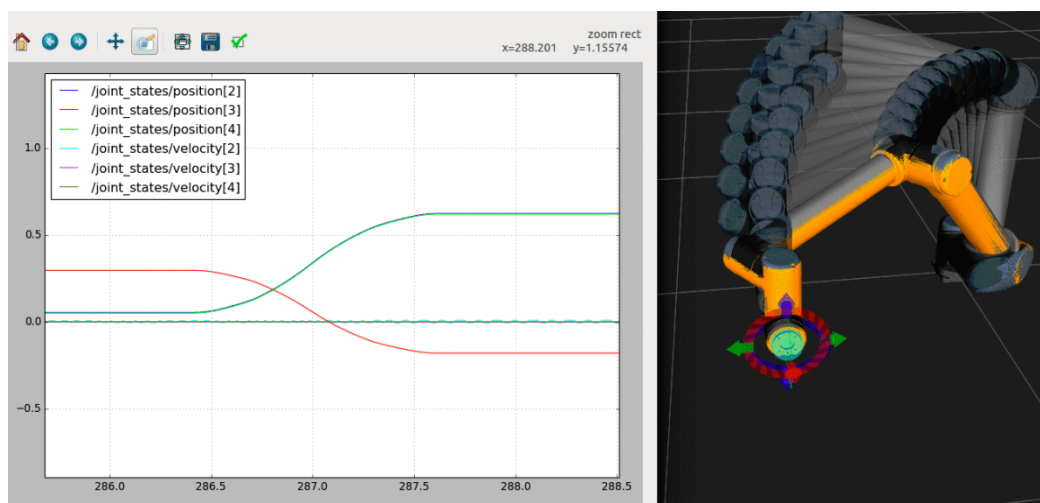


Fig.5. - Muestra de datos en gráficos de dispersión XY de la posición de uniones del modelo de robot virtual mostrado en RVIZ.

2.4.3 Programación de Rutinas de robots

Normalmente softwares de programación de fabricantes de robots adoptan ciertas convenciones de programación de alto nivel propio para cada robot como Rapid de ABB, KRL de KUKA, Karel de FANUC, URScript de Universal Robots, etc. En este laboratorio virtual se propone el adoptar ciertas convenciones usadas dentro de Rapid de ABB para ser capaz de describir en un Script ciertas instrucciones que luego serán enviadas por el nodo de la aplicación GUI y recibidas por el nodo de Gazebo, donde Gazebo poseerá el escenario a simular característico en una aplicación estándar de robótica ejemplo un robot con aplicación de paletizadora, toda esta comunicación de los nodos orquestada por ROS permitirá la simulación de cada una de estas instrucciones que considerara el usuario en su animo de interactuar con el laboratorio virtual de robótica.

2.4.4 Información de los fundamentos de Robótica Industrial

Dentro de la propia GUI será necesario garantizar al usuario el rápido acceso a información puntual relacionada con el funcionamiento del propio Laboratorio Virtual de robótica y de información específica de teoría de los fundamentos de robótica contemplada en el desarrollo de las funcionalidades del propio Laboratorio Virtual de robótica. Además de enlaces de materiales didácticos de ayuda adicionales para comprender mas los fundamentos de robótica tales como libros, videos, enlaces a páginas web, MOCS. Etc.

2.5 Modelos de robots en el LVR

La representación virtual implica la creación de un paquete que describa al robot virtual utilizando archivos CAD en formatos STL, DAE, que describe en un modelo la geometría en 3D de un objeto. Tal creación en 3D modelada tiene el fin de luego ser controlada dentro del entorno de simulación Gazebo. De esta forma no solo basta con el diseño en 3D valiéndonos de un CAD, también es necesario especificar las propiedades internas y grupales relacionadas de un elemento y otro esto se logra utilizando el lenguaje de marcado URDF para modelar el robot y SDF para modelar el robot y modelar el escenario virtual dentro de la comunidad ROS y Gazebo se le denomina el “modelar el mundo”. Por lo cual permitirá el desarrollo del documento guía de uso de ROS y el laboratorio virtual.

2.5.1 Modelado de brazos robóticos de diferentes DOF.

Las herramientas de diseño de modelos de robots de carácter virtual permiten gestionar de forma estandarizada las propiedades y representaciones en 3D agilizando el proceso de integración de diferentes diseños de un robot, esto abarca desde realizar un propio diseño o utilizar los ya existentes que describen fielmente un robot modelado virtualmente que sea de carácter industrial proveniente de ABB, Fanuc o Kuka. (ROS-Industrial Tutorials, 2017)

- a) **CAD:** El Diseño 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 solido.
- b) **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
- c) **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):
 - ✓ 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 del mundo, del modelo, del sensor y del sistema.

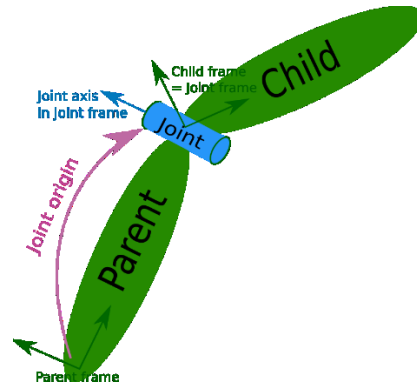


Fig.5. - Representación de la modelación en URDF de dos eslabones y una sola unión.

2.6 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 re configurar 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.

2.6.1 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.

2.6.2 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 mas reconocidos son KukaSim de Kuka, RobotGuide de FANUC y RobotStudio de ABB.

2.6.3 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 mas influyentes en brindar este tipo de soluciones.

2.6.4 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.

2.6.4.1 Simuladores de propósito general comerciales

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

- a) 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.
- b) 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.
- c) 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 y robots reales. Simulink y MATLAB están disponibles para sistemas Linux, Windows y Mac OSX.

- d) 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.

2.6.4.2 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 mas destacados como C++, Phyton, Java, tipo de sistema operativo donde estos pueden ser instalados siendo Linux el que lidera en Simuladores open source. Se mencionaran las características de Gazebo, OROCOS y Player/Stage Por ser los mas influyentes de acuerdo a Tsardoulis (2017)

- a) Gazebo: 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 flotara.
- b) 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.
- c) 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 y permitir el uso de reutilización de software tanto en algoritmos de planeación (movimiento) de robots dentro de los escenarios virtuales donde se experimentará la teoría de robots. Estos simuladores que permiten la comunicación son: Gazebo, Orocos, USARSim, openRAVE, OPRoS de lado de los simuladores open-Source y V-REP y Matlab de lado de los simuladores comerciales.

2.7 Modelación Cinemática y 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.

2.7.1 Cinemática Directa e Inversa

El problema de la cinemática directa es calcular la posición y orientación de la trama de la herramienta relativa a la trama base.

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 Efecto final (TCP por sus siglas en ingles)

El problema de Cinematica 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.

- a) **Movelit para análisis de cinemática:** Utiliza una arquitectura basada en plugins para resolver cinemáticas inversas mientras que proporciona una implementación nativa de cinemática directa.
- b) **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.

2.7.2 Espacio de trabajo del robot.

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

2.8 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

2.8.1 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 K_p , K_i y K_d 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

2.9 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 la 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.

2.9.1 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.

2.10 Manual de uso del laboratorio virtual.

Es necesario para el adecuado uso de este laboratorio y trabajo en general dejar un manual de usuario que permita al profesor o estudiante formular experimentos con la interacción de esta herramienta virtual, describiendo de forma específica cada una de sus propiedades que permita 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 respuesta de movimiento de un eslabón, Velocidad, etc

Hipótesis y Variables

Este proyecto está clasificado como de desarrollo ya que contempla el diseño e implementación de un **laboratorio virtual de Robótica permitirá desarrollar competencias y habilidades** necesarias a los estudiantes de ingeniería electrónica en el aprendizaje de la robótica bajo el concepto de una rama multidisciplinaria.

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

Diseño Metodológico

El desarrollo del proyecto se basa en el modelo de la Figura 2 donde se toman en cuenta 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.

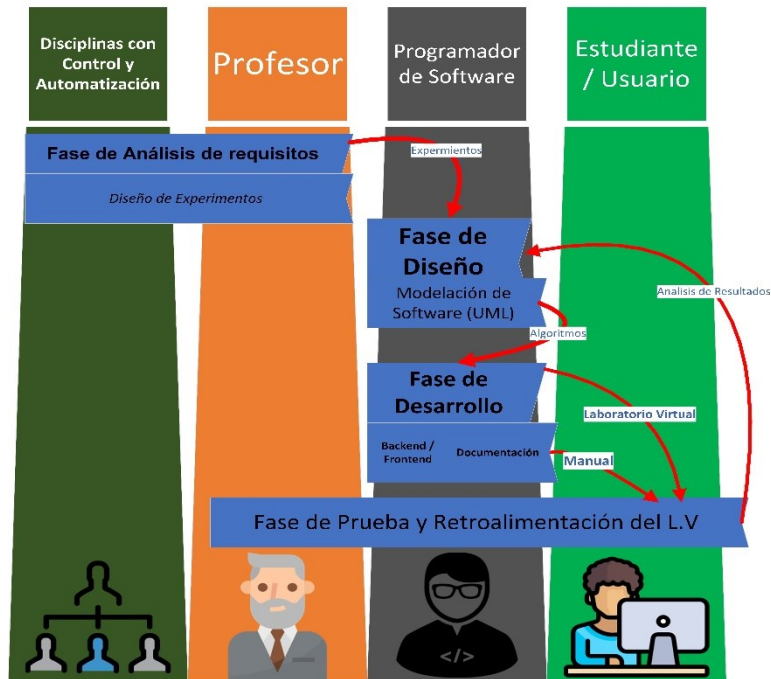


Fig.2.- Diseño metodológico del laboratorio Virtual.

- **Fase de Análisis de requisitos:**

La creación de una herramienta para mejorar el aprendizaje va ligada **con conocer las demandas del usuario (cliente es diferente de usuario) final**, realizando un estudio de los alcances necesarios para lograr una solución viable que se ajuste de acuerdo a la teoría de robots industriales y a los recursos de software tales como Simuladores, lenguajes de programación, que se desean integrar a la solución. **(La fase de análisis está asociada con el problema a resolver, el objetivo que se persigue, que se ha hecho relacionado al problema bajo consideración, los requerimientos)**

- **Fase de diseño:**

Las fases de diseño del laboratorio tendrán que vincularse con teoría relacionada a la robótica industrial tomando los conceptos básicos sin inferir en la importancia que tienen para la formación del estudiante. La revisión de literatura **jugará** un papel clave en la toma de ideas para adoptar una solución que se **ajustará** a la medida de lo posible de lo estipulado en las disciplinas

~~con automatización y control~~, cabe destacar que en esta parte es indispensable utilizar algoritmos básicos de descripción del desarrollo del modelo del laboratorio virtual, **Ejemplo uso de UML el lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.**

- **Fase de desarrollo:**

- **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 librerías de software libre para desarrollar GUI del lado de Frontend.

- **Laboratorio virtual:** **Esta aplicación de software** permitirá centralizar los procesos vinculados en garantizar la comprensión 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 prácticas 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 Ingeniería. *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_implimentaci%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] Tsardoulis, E., & Mitkas, P. (2017). Robotic frameworks, architectures and middleware comparison. Obtenido de <https://arxiv.org/pdf/1711.06842.pdf>

~~[13] Cruz-Neira, G., 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.~~

~~[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.

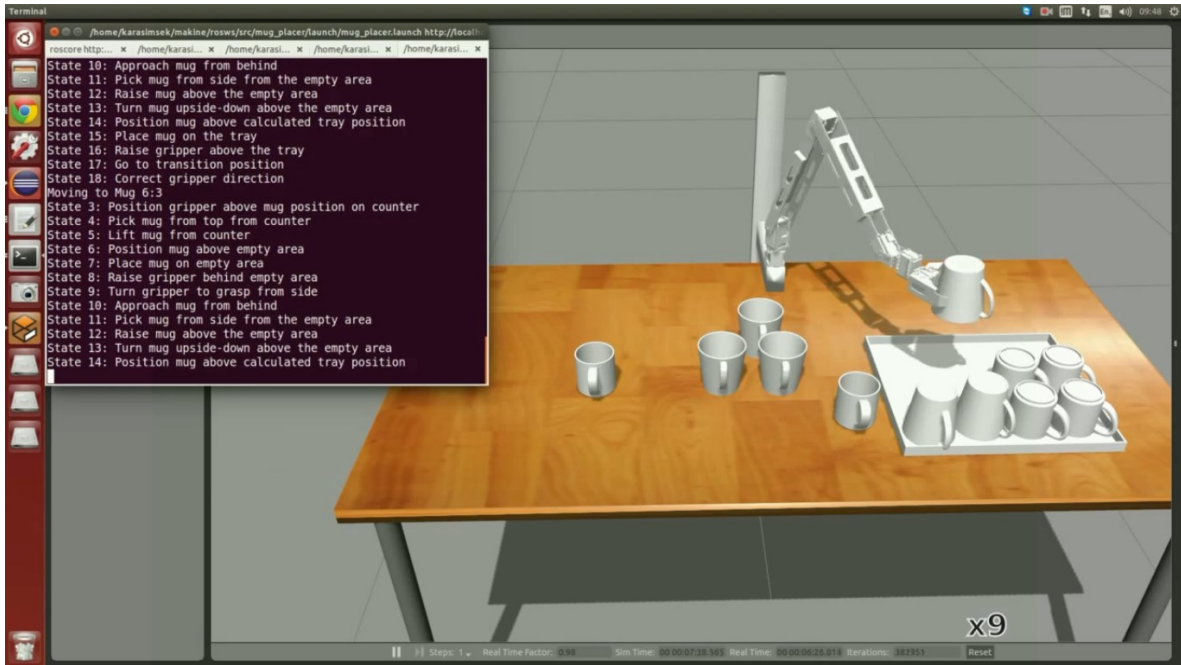
[31] Tellez, R. (2016) A thousand robots for each student: using cloud robot simulations to teach robotics.

[32] ARTE: A ROBOTICS TOOLBOX FOR EDUCATION Obtenido de http://arvc.umh.es/arte/index_en.html

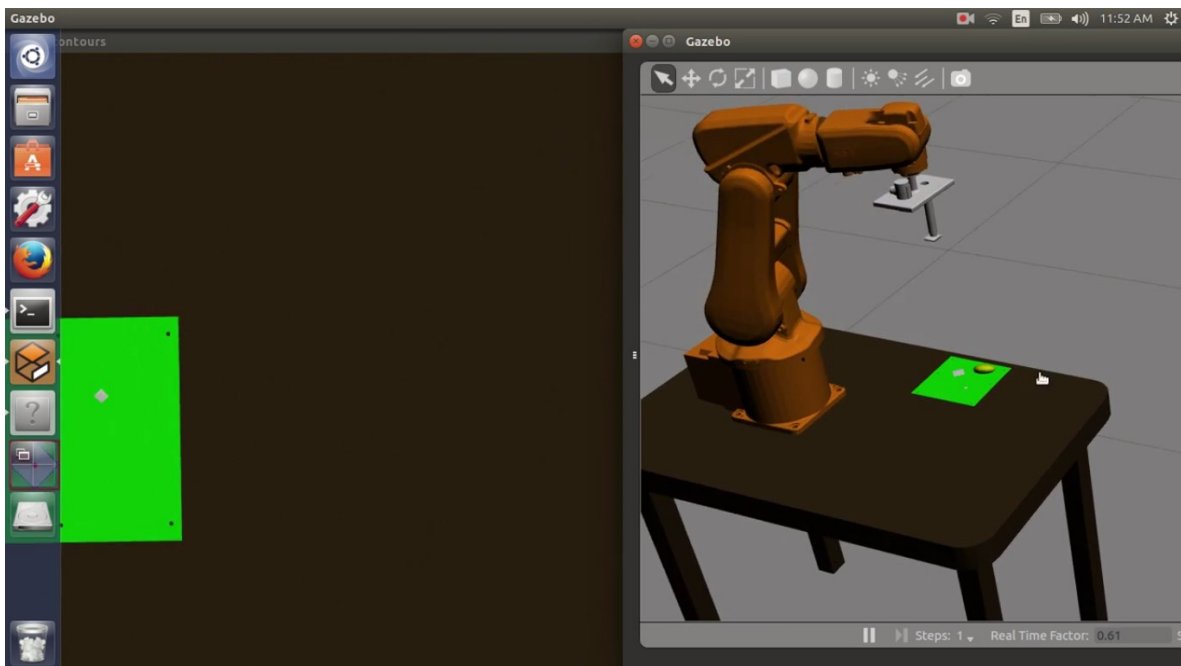
[33] Mato, M. (2014). Simulación, control cinemático y dinámico de robots comerciales usando la herramienta de Matlab, Robotic Toolbox. Obtenido de <http://uvadoc.uva.es/handle/10324/13187>

Anexos

➤ Trabajos relacionados con ROS y Gazebo.

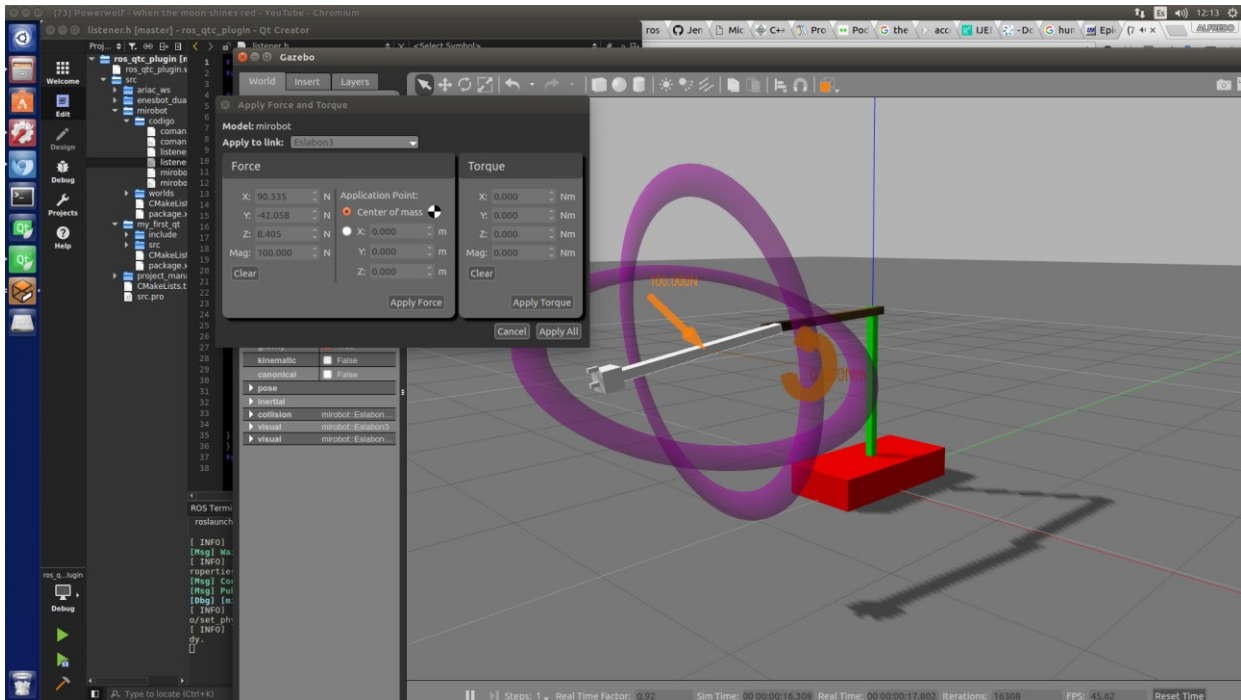


A.3: Captura de pantalla trabajos relacionados en simulación usando ROS y Gazebo para practica de Pick and Place.



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.