

INTEGRACION DE HERRAMIENTAS PARA EL DESARROLLO DE UNA APLICACIÓN CON UN BRAZO ROBOTICO

Mingura Erives, Diego; Márquez Gutiérrez, Pedro Rafael
García Mata, Carmen Leticia; Baray Arana, Rogelio Enrique
Tecnológico Nacional de México Campus Chihuahua
División de estudios de posgrado e investigación
Av. Tecnológico 2909 Col. 10 de Mayo C.P. 31310
Tel. 614 201 20 12 Exts. 2112 y 2114

{m20061538, pedro.mg, carmen.gm, rogelio.ba@chihuahua.tecnm.mx}

RESUMEN.

Se reporta una investigación en el campo de la robótica industrial inteligente, específicamente el robot UR5 de Universal Robots, robot manipulador útil para tareas de “Pick and place” que es parte de la tecnología más reciente en la industria. En ella se utilizan herramientas como ROS, ¡MoveIt!, Gazebo, CoppeliaSim de igual manera con objetivos de investigación en inteligencia artificial se utilizan herramientas como Python, Prolog, ASP, Clingo, OMPL. El objetivo es desarrollar conocimientos y habilidades que permitan trabajar con sistemas mecatrónicos complejos haciendo énfasis en la inteligencia y arquitecturas modulares en un primer alcance conocer las herramientas y ponerlas en funcionamiento.

Palabras Clave: robótica industrial, planeación automática, pick and place, industria 4.0

ABSTRACT.

Research is reported in the field of intelligent industrial robotics, specifically the UR5 robot of Universal Robots, manipulator robot useful for "Pick and place" tasks that is part of the most recent technology in the industry. In it tools such as ROS, MoveIt!, Gazebo, CoppeliaSim are used in the same way with research objectives in artificial intelligence tools such as Python are used, Prolog, ASP, Clingo, OMPL. The goal is to develop knowledge and skills that allow to work with complex mechatronic systems emphasizing intelligence and modular architectures in a first scope to know the tools and put them into operation.

Keywords: industrial robotics, automatic planning, pick and place, industry 4.0

1. INTRODUCCIÓN

En el diseño de sistemas inteligentes robóticos [1,2] se presentan grandes retos a causa de la complejidad de las tareas que estos deben de realizar y como se estructuran los espacios de trabajo [3,4,5,6]. Existen trabajos acerca de cómo mejorar la capacidad de los robots que utilizan agentes que razonan para la planificación de tareas y/o de movimiento en especial el problema de la retroalimentación y monitoreo durante una transición de estados para poder replanificar en caso de error [7]. También como pueden usarse las tecnologías más recientes en almacenes industriales por niveles de automatización siendo más eficiente el uso, selección y adaptación de las herramientas disponibles para entornos industriales [8]. Una herramienta de diseño que resalta entre las demás para robots manipuladores y/o móviles es MoveIt! [9] por tener licencia BSD y estar entre las herramientas dentro del estado del arte en materia tanto para uso comercial como para la investigación, la educación y desarrollos

industriales. Existen herramientas que funcionan en conjunto con MoveIt!, como RViz y Gazebo, dentro de un entorno en ROS. Los robots utilizan el formato URDF (Unified Robotics Description Format), la cual es una especificación XML empleada en la academia e industria para modelar sistemas multi-cuerpo, como manipuladores robóticos. Además, se utilizan algoritmos como RRT (Rapid-exploring Random Tree) y KPIECE (Kinodynamic Motion Planning by Interior-Exterior Cell Exploration) [10].

Se espera que los cobots sean comunes en un futuro con software como MoveIt! el cual evolucionó de un desarrollo para el robot PR2 llamado “arm_navigation” [11]. Existe evidencia del comportamiento del sistema MoveIt! [12] como el uso de estos robots en competencias para los almacenes de la industria 4.0 [13]. Existe un estudio bastante importante que asegura que por lo menos el 44% de los equipos de la competición del primer concurso de Amazon Picking Challenge utilizaron MoveIt! y el 80% algún software de planeación del movimiento, lo cual demuestra la importancia de la presente investigación. Otra alternativa es el empleo de un control híbrido, como el descrito en [14]. La estrategia en el diseño y construcción de estos sistemas es muy similar, y resalta el contraste de planeación contra retroalimentación [15]. En estos sistemas las principales técnicas o algoritmos de planeación del movimiento utilizados son PRM (Probabilistic Road Map) y RRT [16].

2. MARCO TEÓRICO

Para la fundamentación teórica de esta investigación se revisaron los siguientes conceptos y su relación en secuencia de cascada que representa “El gran marco de la célula del desarrollo de las sociedades modernas, la industria y su tendencia”: Industria 4.0 -> Supervisión y diseño industrial->Sistemas CAS->El problema de la integración->sistemas flexibles, políticas, retroalimentación, adaptación e inteligencia->Arquitectura y sistemas de ingeniería->herramientas de operación->manufactura avanzada->metodologías “agile” y manufactura esbelta->ERP [17-26]. Así mismo, se estudió la derivación hacia los sistemas inteligentes: sistemas flexibles, políticas, retroalimentación, adaptación e inteligencia->Sistemas de producción flexible->celda de manufactura flexible->cobots->modelos y arquitecturas de agentes->lógica, computación y algunas limitaciones->cibernética y representaciones de la información->Linux vs. Windows->mecánica, planeación y control [27-41], la derivación de los sistemas mecatrónicos y

sistemas inteligentes: Sistemas mecatrónicos->inteligencia en sistemas mecatrónicos->robótica, máquinas de uso general->robot manipulador y/o móvil, y finalmente, los dispositivos inteligentes: ERP (planeación de recursos empresarial), Planeación y control del movimiento, Robótica móvil y de manipulación.

3. ETAPA I.

En la primera etapa se definió el entorno y se investigaron herramientas de desarrollo [42]. Algunos de los productos que se obtuvieron fueron los siguientes:

- A. Instalación de herramientas de software y selección del robot industrial.
- B. Adopción del lenguaje XML, instalación de la descripción del robot y corrida del modelo en Gazebo.
- C. Descripción PEAS.
- D. Descripción del robot personalizada y RVIZ.
- E. Descripción personalizada del robot y Gazebo.
- F. Configuración personalizada en MoveIt!.
- G. UR5-ROS(MoveIt!)-Gazebo.
- H. Cámara RGB en Gazebo.
- I. Entorno personalizado en Gazebo
- J. Pedestal para el robot UR5 en Gazebo-MoveIt!
- K. Gripper de ventosas en Gazebo.
- L. Uso del "MoveIt! Commander Scripting" con Python
- M. Uso del "Move Group Python"
- N. Librería "MoveIt Grasp"
- O. "El Mundo de los bloques"

Las herramientas que se seleccionaron y utilizaron en la investigación son las siguientes [43-52]:

- a) ROS I Melodic b) MoveIt! 1.0 c) Gazebo 9.0

Herramientas complementarias:

- a) Ubuntu 18.04 LTS b) Windows subsystem for Linux (WSL) c) Xming d) Visual Studio Code

A la hora de seleccionar el robot se deben tener a consideración algunos criterios [53]: a) Alcance y carga útil b) Flexibilidad c) Velocidad d) Espacio y huella e) Ingeniería y desarrollo de proyectos f) Mantenimiento, reparabilidad y disponibilidad

Es importante destacar que los repositorios de ROS-I (ROS Industrial) para el UR5 del Universal Robots se encuentran aún en estatus "Experimental" [54]., por lo que una forma rápida de configurar al UR5 en MoveIt! es a través de MoveIt! Setup Assistant" [55].

El "mundo de los bloques" es un problema en un "mundo de juguete" que nos permite experimentar con el robot para hacer mejoras progresivas para poder pasar al desarrollo a aplicaciones en el mundo real [56].

ETAPA II.

Tanto en la primera como en la segunda etapa, es necesario conocer el entorno y las herramientas de programación orientadas al desarrollo algorítmico y lógico, para lo cual se eligió a Python y Clingo, para personalizar el espacio de trabajo y realizar las interfaces con los controles y comandos, y profundizar en el conocimiento inteligente y de bajo nivel del sistema en desarrollo. Dentro de los principales productos obtenidos tenemos:

- P. El módulo Clingo-Python.
- Q. Revisión de la Industria 4.0
- R. Supervisión y diseño industrial
- S. Sistemas CAS
- T. El problema de la integración
- U. Sistemas flexibles, políticas, retroalimentación, adaptación e inteligencia.
- V. Arquitectura y Sistemas de Ingeniería
- W. Herramientas de Operación
- X. Manufactura Avanzada
- Y. Metodologías "Agile" y Manufactura Esbelta
- Z. ERP
- AA. Sistema de Producción flexible
- BB. Celda de Manufactura Flexible
- CC. Cobots
- DD. Modelos y Arquitectura de Agentes
- EE. Lógica, Computación y algunas Limitaciones
- FF. Cibernética y Representación de la Información
- GG. Linux vs. Windows
- HH. Mecánica, Planeación, y Control

¡Para comprender el funcionamiento de un generador de rutas automático como el de MoveIt! y la implementación de estructuras externas como Clingo y ASP (Answer Set Programming) es imprescindible conocer los conceptos de planificación y diagnóstico [55, 57-60]. Una forma que se está investigando para conectar MoveIt! con Clingo es a través de ROS con un nodo llamado "ROSoClingo" el cual permitiría un uso modular en la robótica. Cabe mencionar que al igual que se conecta el simulador CoppeliaSim con Clingo a través de Python, se anticipa que será posible en las nuevas versiones de MoveIt! que se están actualizando de una manera más sencilla y rápida [61]. A través de la interfaz de Python en MoveIt! es fácil y limpio el código para mandar órdenes para tareas de "pick and place" [62-64]. Al igual que en CoppeliaSim [65], Python nos permite aumentar la capacidad del sistema al utilizarlo para agregar librerías disponibles como Clingo, entre otras de visión artificial y *machine learning*.

4. ETAPA III.

En la tercera etapa es necesario contar con la integración de las herramientas informáticas para poder comenzar a programar tareas de "pick and place", y desarrollar el código necesario para que se comuniquen entre sí. Una vez desarrollada la vinculación

se procede a programar la tarea tipo “pick and place” en su versión derivada del mundo de los bloques, que resulta perfecto para trabajar con este tipo de sistemas, ya que permite trabajar y mejorar mediante experimentación y simulación. Los principales productos obtenidos son:

- II. Experimento Coppeliasim-Python-Clingo.
- JJ. Experimento Coppeliasim-OMPL
- KK. Experimento Coppeliasim-OMPL algoritmos

4.1. Experimento Coppeliasim-Python-Clingo

En este experimento dentro del simulador de Coppeliasim 4.0 se pudo probar exitosamente la integración de las aplicaciones y el robot UR5, lo que permitió llevar de un mundo inicial en el mundo de los bloques a un resultado deseado final. Así, se resolvió exitosamente este tipo de tarea. El razonador de Clingo resuelve los pasos que hay que seguir para llevar el mundo de un estado a otro pasando este plan mediante Python y Lua al robot UR5 para que ejecute el plan.

4.2. Experimento Coppeliasim-OMPL

En este experimento en Coppeliasim 4.0, se probó con el UR5 la librería OMPL (Open Motion Planning Library). Cabe mencionar que su integración a Coppeliasim es reciente, lo cual es un eje para la investigación, ya que el framework MoveIt! se basa principalmente en esta librería y son algoritmos dentro del estado del arte. Se tienen a la disposición 24 algoritmos diferentes (Fig. 1) para experimentar y probar su rendimiento y el robot evite colisionar con los objetos dentro de la escena. Comenzando con su posición inicial de la herramienta, se le da un punto final deseado y el algoritmo busca las soluciones posibles para alcanzar el estado final deseado sin colisionar con los objetos en su entorno, enseguida acorta la lista de opciones para buscar la ruta más corta y ejecutar la solución encontrada.

4.3. Experimento Coppeliasim-OMPL algoritmos

En este experimento con el simulador de Coppeliasim 4.0 se incluye una función de visualización de la trayectoria para apreciar la solución que encuentra el planificador de movimiento.

Se prueban los diferentes algoritmos dando buenos resultados los siguientes, RRT, KPIECE1, EST, LBTRRT, PDST, STRIDE, TRRT, de los cuales destaca LBTRRT en las pruebas.

Al intentar probar los siguientes algoritmos de manera común el simulador se cierra: CForest, FMT

La librería OMPL es un planificador de movimiento en común entre MoveIt! y Coppeliasim [66,67]. Cabe destacar que efectivamente en MoveIt! esta mejor integrada la librería ya que en Coppeliasim hay que escribir unas líneas de código especificando estas acciones. Se exploran algunas propuestas ya que la investigación encontró un área de mejora y en constante expansión del conocimiento [68]. Se utilizaron ejemplos de Potassco [69], Robotica con Python [70] y Coppeliasim y Lua [71] así como cinemática inversa del robot [72] y algoritmos como LBTRRT [73]. Mediante lenguajes de acción es posible modelar un problema de planificación y/o diagnóstico para el robot en Clingo [60], [74], [75] y trabajar en conjunto con el planificador

de movimiento de MoveIt! o Coppeliasim con la librería OMPL para la consecución de tareas ya sea de ensamble o movimiento para “pick and place”.

En la actualidad se siguen haciendo avances y desarrollos en software de robótica como es el caso del lanzamiento de MoveIt! 2.0 y ROS II así como grandes aportaciones por parte de la comunidad lo que permitirá en un par de años la fácil implementación en la industria de estas tecnologías. [76-79].

5. CONCLUSIONES Y TRABAJO FUTURO

El desarrollo de aplicaciones prácticas en el ámbito de la robótica exige la integración de diferentes paradigmas y plataformas que permitan una rápida y completa solución a un problema en particular. Existen en el mercado una amplia gama de herramientas para tal efecto y se requiere su estudio comparativo para su adopción. Las alternativas van desde lenguajes de programación imperativos (como C++), basados en lógica (Prolog, CLP, ASP), orientados a objetos (Java, Python) y funcionales (LISP, Scala). Estos tienen la ventaja que al ser de propósito general otorgan la mayor libertad de implementación, pero así mismo, el mayor trabajo. Esto, sin embargo, no debe verse como limitante, ya que proporcionan un mecanismo de solución de problemas puntuales y el puente entre diferentes enfoques. Por otro lado, existen sistemas de simulación que aceleran la prueba de concepto, ya que incluso, poseen bibliotecas de física, que permite observar el desempeño en entornos más apegados a la realidad final. Sin embargo, algunos simuladores se entregan como sistemas monolíticos y no pueden ser conectados a otras plataformas de desarrollo. Existen esfuerzos por desarrollar un sistema estándar que posibilite la creación de cualquier aplicación robótica, como sistemas operativos, pero su curva de aprendizaje e instalación es muy grande en general.

Este artículo describió un estudio realizado con el fin de explorar diferentes plataformas y paradigmas en un problema de robótica complejo, como lo es la manipulación de objetos, en particular, del tipo “pick and place”. El enfoque utilizado fue desde una perspectiva de alto nivel de abstracción (planificación) y no desde el control de bajo nivel. Se estudió y experimentó con algunas plataformas de acceso libre, se describió la forma de utilización y se analizó su facilidad de integración en el contexto del problema base.

El trabajo desarrollado permitió encontrar algunas ideas de investigación futuras, ya que se detectaron algunas áreas de posible mejora que están en constante expansión [68]. Por ejemplo, se debe trabajar con mayor profundidad e integración las herramientas proporcionadas por Potassco como Clingo [69], Python como una interfaz entre herramientas de vanguardia [70], Coppeliasim en entorno Windows [71] apoyándose de la cinemática inversa del robot [72] y algoritmos de planificación del movimiento avanzados como el caso de LBTRRT [73]. Mediante lenguajes de acción es posible modelar un problema de planificación y diagnóstico bajo Clingo [60, 74, 75] y trabajar en conjunto con el planificador de movimiento de MoveIt! o Coppeliasim empleando la biblioteca OMPL para la planificación de tareas, e.g., de ensamble o movimiento

tipo “pick and place”. ¡Continuamente se siguen haciendo avances y desarrollos en la plataforma de desarrollo para robótica como es el caso particular del sistema MoveIt! 2.0 y ROS II así como grandes aportaciones por parte de la comunidad, lo que permitirá en uno cuantos años una mejor adopción en la industria de estas tecnologías [76-79]. El reto está en la aplicación de estas tecnologías integrándolas con software elaborado y estandarizado para un manejo ágil, limpio y esbelto en condiciones cada vez más complejas que contribuyan a un aumento en la flexibilidad de los sistemas inteligentes en la robótica de manipulación, colaboración y/o móvil en diferentes entornos industriales y domésticos.

6. REFERENCIAS

- [1] Pinto-Bustamante, Boris, Bucky Fuller: Entre puentes, domos y palacios de cristal. MasD Revista Digital de Diseño. 08. 8-25. (2011)
- [2] Sosa Compeán, L. B., DISEÑO BASADO EN SISTEMAS COMPLEJOS (Ed. rev.). Monterrey, Nuevo León, México: Editorial Labyrinthos, (2017)
- [3] The Verge., “The 2015 DARPA Robotics Challenge Finals.”, Youtube: <https://www.youtube.com/watch?v=8P9geWwi9e0> sitio visitado enero de 2022
- [4] Boston Dynamics, “Inside the lab: How does Atlas work?” de Boston Dynamics”, Youtube: <https://www.youtube.com/watch?v=EezdinoG4mk&t=1s> sitio visitado el 17 de Agosto del 2021
- [5] Bosch Global, “Meet our Robotics Researchers at Bosch”, Youtube: <https://www.youtube.com/watch?v=7Fm5MUaaxEY&t=6s> sitio visitado 15 de Octubre del 2021
- [6] Aguirre, R. y Moreira, K., Lenguaje y pensamiento. En Vásquez Echeverría, A. (Ed.). Manual de Introducción a la Psicología Cognitiva. (2016).
- [7] A. Kattapur and B. Purushotaman, "RoboPlanner: a pragmatic task planning framework for autonomous robots," in Cognitive Computation and Systems, vol. 2, no. 1, pp. 12-22, 3 2020.
- [8] C. H. Corbato, M. Bharatheesha, J. van Egmond, J. Ju and M. Wisse, "Integrating Different Levels of Automation: lessons From Winning the Amazon Robotics Challenge 2016," in IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4916-4926, Nov. 2018.
- [9] Sharath Jotawar, Manish Soni, and Swagat Kumar. 2017. Motion Planning for an Automated Pick and Place Robot in a Retail Warehouse. In Proceedings of the Advances in Robotics (AIR '17). Association for Computing Machinery, New York, NY, USA, Article 40, 1–6. DOI:<https://doi.org/10.1145/3132446.3134904>
- [10] Hanzhong Tao, Yu Fang, and Zhifeng Zhou. 2019. Research on Trajectory Planning of ROS-based Manipulator. In Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI 2019). Association for Computing Machinery, New York, NY, USA, 374–378. DOI:<https://doi.org/10.1145/3366194.3366260>
- [11] Sachin Chitta, Ioan Sucan, and Steve Cousins. 2012. Ros topics. IEEE robotics and automation magazine 19, 1 (2012), 18–19
- [12] Sharath Jotawar and Nishant Kejriwal. 2016. Experimental video for pick and place using UR5 and Barrett WAM arm. Available: Link 1- <https://youtu.be/paq2pk13CKY> Link 2- <https://youtu.be/CAssP1RjzkI>. (2016)
- [13] Peter R Wurman and Joseph M Romano. 2016. Amazon PickingChallenge 2015. AI Magazine 37, 2 (2016), 97–99.
- [14] N. Correll et al., “Analysis and observations from the first Amazon picking challenge,” IEEE Trans. Autom. Sci. Eng., vol. 15, no. 1, pp. 172–188, Jan. 2018.
- [15] C. Eppner, S. Hofer, R. Jonschkowski, R. Mart “ ‘in-Mart’ in, A. Sieverling, V. Wall, and O. Brock, “Lessons from the Amazon Picking Challenge: Four aspects of building robotic systems,” in Proceedings of Robotics: Science and Systems, 2016.
- [16] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. 2010. Robotics: modelling, planning and control. Springer Science & Business Media.
- [17] Ynzunza Cortés, Carmen Berenice; Izar Landeta, Juan Manuel; Bocarando Chacón, Jacqueline Guadalupe; Aguilar Pereyra, Felipe; Larios Osorio, Martín, “El Entorno de la Industria 4.0: Implicaciones y Perspectivas Futuras”, Conciencia Tecnológica, núm. 54, 2017, Instituto Tecnológico de Aguascalientes, México, Disponible en: <https://www.redalyc.org/articulo.oa?id=94454631006>
- [18] SMC International Training. (2012). Integration and Interaction of Technologies. In SMC International Training (Ed.), "Best practice in training people in field of automation" (Ed. rev., pp. 1–38). Recuperado de http://automotive.bg/wp-content/uploads/2014/03/5_Day-2-SMC.pdf
- [19] Esteban Pérez – López, “Los sistemas SCADA en la automatización industrial”, Tecnología en Marcha. Vol 28, N 4, Octubre-Diciembre 2015. Pag 3-14.
- [20] ARCHITECTURE AND SYSTEMS ENGINEERING: MODELS AND METHODS TO MANAGE COMPLEX SYSTEMS, MIT, 2022
- [21] Bolton, William., Mecatrónica. Sistemas de control electrónico en la ingeniería mecánica y eléctrica. Un enfoque multidisciplinario Quinta Edición. México: Alfaomega Grupo Editor, S.A. de C. V.,. (Mayo 2013).
- [22] Grillo Perelló, Joan - García Jiménez, Raúl “Sistemas de Fabricación Flexible”, Ingeniería Técnica Industrial Electrónica Industrial (ITIEI), Universitat de les Illes Balears – UIB España, Páginas 2 – 5, 2002, [Online]. Disponible en: <https://es.scribd.com/document/347536070/fmsgrillo-garcia-pdf>
- [23] Parrish, David J. “Flexible Manufacturing” Butterworth-Heinemann Ltd. 1993
- [24] Dr. Leoncio Luis Acuña Pinaud, Mag. Cesar Aldo Canelo Sotelo, Ing. Alfredo Rolando Agüero Mauricio, Sr. Gianpierre Cabrejos Meléndez, Sr. Jaime Alejandro Fernández Chávez, Facultad de Ingeniería Industrial y de Sistemas, “Metodología de un sistema de producción flexible en el desarrollo de la pequeña y mediana empresa industrial de manufactura en Lima

Metropolitana” 16th LACCEI International Multi-Conference for Engineering, Education, and Technology: “Innovation in Education and Inclusion”, 19-21 July 2018, Lima, Peru.

[25] Administración y dirección de operaciones, Chase Aquilano. Mc Graw Hill. [Online] Recuperado de: <https://www.usmp.edu.pe/publicaciones/boletin/fia/info46/index2.html>

[26] Universidad del País Vasco, DPTO. DE INGENIERIA MECANICA, TECNOLOGIA MECANICA, Modulo V: Otras Tecnologías, “TEMA 21: Sistemas de Fabricación Flexible” Gabriela Torres, Robótica, “Cobots-humanos: innovación codo a codo”, 10 de Diciembre del 2021, URANY,[Online]. Disponible en: <https://urany.net/blog/cobots-humanos-innovacion-codo-a-codo#:~:text=Los%20cobots%20fueron%20creados%20en,general%20controlado%20por%20una%20computadora.>

[27] James E. Colgate Michael A. Peshkin, Northwestern University, U.S. Patente US5952796A, 23 de Febrero de 1996 [Online]. Disponible en: <https://patents.google.com/patent/US5952796>

[28] Jon Van and Tribune Staff Writer, "Mechanical Advantage" Chicago Tribune, 11 de Diciembre de 1996.

[29] Universal Robots, “COBOTS Y ROBÓTICA COLABORATIVA - QUE SON, QUE FUNCIONES PUEDEN REALIZAR Y SUS VENTAJAS”, [Online] Disponible en: <https://www.universal-robots.com/es/cobots-robots-colaborativos/> visitado en marzo de (2022),

[30] International Federation of Robotics, “Japan is World’s number one Robot Maker”, Frankfurt, Tokyo, [Online] Disponible en: <https://ifr.org/ifr-press-releases/news/japan-is-worlds-number-one-robot-maker> visitado en Mar 10, 2022

[31] Association for Advancing Automation, “Collaborative Robots Market Update 2018”, Robotics Online Marketing Team. [Online] Disponible en: <https://www.automate.org/blogs/collaborative-robots-market-update-2018> visitado el 20 de marzo, 2022

[32] Keith Darlington, “La era de los robots colaborativos: los cobots”, [Online] Disponible en: <https://www.bbvaopenmind.com/tecnologia/robotica/la-los-robots-colaborativos-los-cobots/#:~:text=Seg%C3%BAn%20un%20nuevo%20informe%20de,175%25%20de%20su%20nivel%20actual1.> Visitado el 04 de noviembre 2019

[33] James Albus, en respuesta a Henry Hexmoor, [Online] Disponible en: <http://tommy.jsc.nasa.gov/er/er6/mrl/papers/symposium/albuS.txt> visitado el 24 de Abril del 2022

[34] NILS J. NILSSON, Stanford University, “INTELIGENCIA ARTIFICIAL Una nueva síntesis”, 2001

[35] Russell, S. J., Russell, S. J., Norvig, P., & Davis, E. (2010). Artificial Intelligence: A Modern Approach (3ª ed.). New

Jersey, USA: Prentice Hall.

[36] “MODERN ROBOTICS MECHANICS, PLANNING, AND CONTROL” Kevin M. Lynch and Frank C. Park, May 3, 2017.

[37] “Knowledge Representation, Reasoning, and the Design of Intelligent Agents” de Michael Gelfond Yulia Kahl, January 28, 2013

[38] “Integrating ASP into ROS for Reasoning in Robots” por Benjamin Andres, David Rajaratnam, Orkunt Sabuncu y Torsten Schaub.

[39] IONOS Inc., “Linux vs. Windows: cuadro comparativo”, (14 de Enero del 2021), [Online] Disponible en:

<https://www.ionos.mx/digitalguide/servidores/know-how/linux-vs-windows-el-gran-cuadro-comparativo/>

[40] Ioan A. Șucan, Mark Moll, Lydia E. Kavraki, The Open Motion Planning Library, IEEE Robotics & Automation Magazine, 19(4):72–82, December 2012.

<https://ompl.kavrakilab.org> Swagat Kumar, (21 de Septiembre del 2020), “UR5 Robot Simulation with Gazebo and

Moveit”, [Online] Youtube: <https://www.youtube.com/c/SwagatKumarIndia/featured>

[41] Santi Montiel, (29 de Julio del 2020), “Cómo configurar ROS y Gazebo en Windows 10 utilizando WSL”, [Online] Disponible en:

<https://www.santimontiel.eu/index.php/2020/07/29/como-configurar-ros-y-gazebo-en-windows-10-utilizando-wsl/>

[42] David Coleman, Ioan A. Șucan, Sachin Chitta, Nikolaus Correll, (2019), Getting Started, [Online] Disponible en:

http://docs.ros.org/en/melodic/api/moveit_tutorials/html/doc/getting_started/getting_started.html#

[43] Microsoft, (8 de Abril del 2022), “Instalación de WSL.”, [Online] Disponible en: <https://docs.microsoft.com/es-es/windows/wsl/install-win10>

[44] Colin Harrison, (2016), “Xming X Server for Windows”, [Online] Disponible en: <https://sourceforge.net/projects/xming/>

[45] Open Robotics, (2019), “Simulate before you build”, [Online] Disponible en: <http://gazebo-sim.org/>

[46] Open Robotics, (2020), “Documentation”, [Online] Disponible en: ROS: <http://wiki.ros.org/es>

[47] Programación Extrema, (28 de Mayo del 2018), “Instalar Gazebo + ROS en un WSL”, Youtube: <https://www.youtube.com/watch?v=4toU7sS0eRA>

[48] Robotics Fundation, (2014), “Plugins available in gazebo_plugins”, [Online] Disponible en:

http://gazebo-sim.org/tutorials?tut=ros_gzplugins#Pluginsavailableingazebo_plugins

[49] Universal Robots, FZI, (3 de Septiembre 2021) “Universal_Robots_ROS_Driver”, https://github.com/UniversalRobots/Universal_Robots_ROS_Driver, <https://www.youtube.com/watch?v=ayp87SjrwPc&list=WL&index=1&t=127s>

[50] The Construct, (14 de Mayo del 2019), “ROS Developers LIVE-Class #57: Configuring Moveit! for your Arm Robot”, [Online] Youtube:

<https://www.youtube.com/watch?v=f7MRESmhewg>

[51] Kevin Tardif, documentación técnica: “Robots. "Hay uno para cada trabajo:

Cómo elegir el correcto” de Festo Inc. 2020

[52] ROS-INDUSTRIAL, (2019), “ROS-Industrial Conference 2019 - Event Video”, [Online] Disponible en: <https://rosindustrial.org/>

[53] The Construct, (3 de Julio del 2017), “[ROS Q&A] 003 - How to control a Gazebo simulated robot with MoveIt! (UR5) – Tutorial”, [Online]

Youtube:

https://www.youtube.com/watch?v=j6bBxuD_bYs&t=1158s

[54] Abner Guzmán, Student Member, IEEE, José Ilitzky, Cesar Montero, Guillermo Hernández y Erik Yñigo, “Planificación de Orden Parcial: El Mundo de los Bloques”, 2004

[55] Roland Kaminski, Torsten Schaub y Philipp Wanko, University of Potsdam, Alemania Inria, Bretagne Atlantique, Rennes “A Tutorial on Hybrid Answer Set Solving with clingo”.

[56] Potassco, the Potsdam Answer Set Solving Collection, (2022), “Running clingo”, Tools for Answer Set Programming developed at the University of Potsdam, [Online] Disponible en: <https://potassco.org/clingo/run/>

[57] Michael Gelfond y Yulia Kahl, “Knowledge Representation, Reasoning, and the Design of Intelligent Agents The Answer-Set Programming Approach”, Cambridge University Press, Marzo de 2014, Disponible en: (<http://pages.suddenlink.net/ykahl/>)

[58] Dra. Carmen Leticia García Mata, “Prolog y ASP” de la clase: “Control Inteligente Para Sistemas Mecatrónicos”, ITCH, Chihuahua, México, Mayo del 2021

[59] Benjamin Andres, David Rajaratnam, Orkunt Sabuncu, y Torsten Schaub, University of Potsdam y University of New South Wales, INRIA Rennes, “Integrating ASP into ROS for Reasoning in Robots”

[60] The Construct, 14 de Mayo del 2019, “ROS Developers LIVE-Class #58: Use MoveIt! to Pick an Object.”, [Online] Youtube:

<https://www.youtube.com/watch?v=FeIC8taF9ag&t=3272s>

[61] David Coleman, Ioan A. Șucan, Sachin Chitta, Nikolaus Correll, (2019), “Move Group Python Interface”, MoveIt 1 – Melodic, [Online] Youtube:

http://docs.ros.org/en/melodic/api/moveit_tutorials/html/doc/move_group_pyth

on_interface/move_group_python_interface_tutorial.html#getting-started

[62] Ioan A. Sucan and Sachin Chitta, "MoveIt", [Online] Disponible en: moveit.ros.org.

[63] Coppelias Robotics, Ltd., “CoppeliaSim”, [Online] Disponible en: <https://www.coppeliarobotics.com/>

[64] Coppelias Robotics, Ltd., “OMPL Plugin API reference”, [Online] Disponible en:

<https://www.coppeliarobotics.com/helpFiles/en/simOMPL.ht>

m#enum:Algorith m

[65] Coppelias Robotics, Ltd., “Path/motion planning”, [Online] Disponible en:

<https://www.coppeliarobotics.com/helpFiles/en/pathAndMotionPlanningModules.htm>

[66] David Coleman, Ioan A. Șucan, Sachin Chitta, Nikolaus Correll, Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study, Journal of Software Engineering for Robotics, 5(1):3–16, May 2014. doi: 10.6092/JOSER_2014_05_01_p3.

[67] Potassco, (Diciembre 11 del 2020), “Easy answer set programming, introduction, running first example”, [Online] Youtube: https://www.youtube.com/watch?v=PXk3xYbmy_s

[68] Robótica con Python, (15 de Abril del 2020), “Robótica con Python - Primeros pasos en CoppeliaSim - Uso de RemoteAPI”, [Online] Youtube:

https://www.youtube.com/watch?v=bzyyHDmhq84&list=PLOxPNrO-_c2i7W0iq4S0q4dRZsB_vuZkb&index=7

[69] Leopoldo Armesto, (4 de Febrero del 2020), “Programming with CoppeliaSim API using Lua | CoppeliaSim (V-REP)”, [Online] Youtube:

<https://www.youtube.com/watch?v=q0l0hVGcarU>

[70] Juan Axl Rose King, (04 de Junio del 2021), “IK en CoppeliaSim (Tutorial)”, [Online] Youtube: <https://www.youtube.com/watch?v=dB8ebrjUPdQ>

[71] Oren Salzman y Dan Halperin Blavatnik School of Computer Science, Tel- Aviv University, Israel, “Asymptotically near-optimal RRT for fast, high- quality, motion planning”, 04 de Marzo del 2015.

[72] Dr. Pedro Rafael Márquez Gutiérrez, “Modelos descriptivos y Prolog” de la clase: “Temas Selectos Para ingeniería Mecatrónica”, ITCH, Chihuahua, México, Mayo del 2021.

[73] Julian J. Portillo, Carmen L. Garcia-Mata, Pedro R. Márquez-Gutiérrez, and

Rogelio Baray-Arana A., “Robot Platform Motion Planning using Answer Set Programming”, Instituto Tecnológico de Chihuahua, Chihuahua, Chih., Mex.

[74] Rangel Alvarado, (6 de Enero del 2021), “Part 1/2 - UR5 RobotIQ Gripper Moveit 2.0 - Unable to identify any set of controllers that can...”, [Online]

Youtube:

<https://www.youtube.com/watch?v=lgJM86NhGGQ&t=65s>

[75] Rangel Alvarado, (25 de Enero del 2021), “Pick n Place application files for ROS Noetic and Moveit with an UR5 and RobotIQ 2DoF Hand”, [Online]

Youtube: https://github.com/issaiass/ur5_robotiq_85_picknplace

[76] Angélica García Vega, (2013), “INTELIGENCIA ARTIFICIAL EN LAS EMPRESAS”, [Online]

Recuperado de: <https://www.uv.mx/iiesca/files/2013/01/artificial2002-1.pdf> visitado en Mayo de 2022

[77] PickNik Inc, “Accelerate your robotics development”, [Online] Disponible en: <https://picknik.ai/> visitado en Mayo de 2022