

Simulation of Hexapod Robot Motion in Virtual Reality Environments: A CAD-Based Approach

Sofia Rodríguez-García, Luis Enrique Hernández-Ramírez, Adriana Morales-Martínez, Juan Carlos Gómez-López, Alejandro Jiménez-Villarreal

Departamento de Ingeniería Mecánica, Universidad Nacional Autónoma de México (UNAM), Ciudad Universitaria, Coyoacán, CDMX, México; Laboratorio de Inteligencia Artificial y Robótica, Instituto Politécnico Nacional (IPN), Edificio 9, Unidad Profesional Adolfo López Mateos, Cuernavaca, Morelos, México

Abstract

The aim of this work is the development of an animation of a hexapod robot in virtual reality environments through the use of Unity software. The movement of the robot is 18 degrees of freedom, which are based on servo motors that are controlled by the use of a microcontroller. It also shows the necessary processes to transfer the 3D design of the hexapod robot made in the use of CAD (Computer Aided Design) software to a virtual reality environment without losing the physical characteristics of the original design. Finally, the results obtained from the simulation of movement and responses of the hexapod robot to disturbances such as roughness of the floor, uneven ground and gravity among others are presented, allowing a correct evaluation to possible designs of robots before being elaborated.

Indexed keywords: Computer Engineering, Advanced Computing, Technology, Open Access

Article History: Received: 08 October 2025 | Accepted: 27 December 2025 | Published: 11 January 2026

Animation; hexapod; virtual; Computer aided design.

1. Introducción

En los últimos años, el desarrollo y la investigación en la disciplina de la robótica ha experimentado una evolución exponencial debido a su capacidad de realizar tareas repetitivas precisas y de alta velocidad (Bonilla et al., 2018; Lawrence, 1993). La cinemática, dinámica, sistemas de enlace y de comunicación son áreas de investigación que permiten el desarrollo de robots para entornos industriales, robots utilizados en la cirugía médica hasta el desarrollo de robots móviles autónomos con diferentes aplicaciones en exteriores o interiores que pueden operar tanto en entornos naturales definidos de forma perfecta como en lugares de poco acceso y no estructurados donde el trabajo de un ser humano y de la vida como se la conoce representa un riesgo elevado (Desbats et al., 2006; Mukherjee, 2014; Sánchez et al., 2017).

En la actualidad no se puede verificar el diseño ni el movimiento final de un robot sin el uso de *software* costoso; los algoritmos establecidos mediante el modelamiento matemático no se pueden probar en un diseño previo en este tipo de *software* lo que conlleva a fallas en los movimientos reales una vez implementado el robot (Carlson et al., 2005). Existen varias metodologías utilizadas para el desarrollo de plataformas robóticas móviles, en el caso de la telemedicina se plantea el desarrollo de un diseño conceptual del sistema con sus requerimientos y restricciones y, finalmente, se realiza un modelo sobre una plataforma 3D donde se permita generar una retroalimentación mediante la simulación



de cargas mecánicas y desplazamiento al diseño final del prototipo (Enríquez et al., 2018; Faria et al., 2015).

La ingeniería en conjunto con los sistemas para diseño asistido por computador en conjunto han logrado el desarrollo de ordenadores que soportan simulaciones cada vez más reales y complejas en las diferentes áreas de investigación e industria (Chamorro et al., 2018). En función de este avance, se diseñan herramientas de forma continua para replicar las características físicas que se encuentran en la realidad hacia ambientes virtuales. Una de estas herramientas es el motor gráfico desarrollado por Unity Technologies, Unity3D, el cual permite desarrollar escenarios virtuales para un amplio número de plataformas y sistemas operativos (Al Mashagbeh et al., 2015).

El desarrollo de un robot requiere del adecuado manejo de diversos factores tales como mecánicos, ambientales y económicos que pueden influir en su desempeño. El material del robot, la calidad de los motores y sensores, el nivel de carga de las baterías e incluso la iluminación medioambiental pueden marcar la diferencia entre un robot operativo y uno no funcional. Dependiendo de las habilidades que quieran desarrollarse puede ser deseable el poder manejar estos factores; las habilidades se generan a partir de experimentos de movimiento. Sin embargo, si se desea enfocarse en mejorar habilidades de algoritmos de control, lo ideal sería removerlos por completo (Andaluz, Quevedo, et al., 2016).

La simulación en entornos reales consiste en un simulador que permite a los robots el movimiento en un entorno tridimensional animado. El *software* tradicional remueve una gran cantidad de factores físicos difíciles de controlar y se enfoca en los procesos, mientras que al realizar una animación en Unity se busca brindar un marco de trabajo común para el desarrollo de habilidades como la programación, la creación de algoritmos de movimientos de alta calidad que puedan compararse de manera objetiva y precisa, y que incluso pueda servir como herramienta de apoyo durante la creación de un robot físico (Freund et al., 1999).

Los diseños de prototipos robóticos desarrollados en los *software* de Diseño Asistido por Computador, CAD, como SolidWorks, Autodesk Inventor, entre otros, no brindan todas las condiciones gráficas necesarias para realizar una simulación realista de un modelo 3D, lo que impide realizar pruebas del diseño del prototipo en un entorno casi real permitiendo obtener datos de funcionalidad. La necesidad de simulación enfocada a la interacción entre el operador humano, el robot y un ambiente más real e inmersivo, ha hecho que la función de simulación de los *software* CAD no sean suficientes para lograr dicho objetivo, en virtud que los CAD no tienen una alta calidad de gráficos en comparación con las existentes en animaciones digitales profesionales en otros *software* de simulación (Andaluz, Chicaiza, et al., 2016).

Los principales motores gráficos comerciales como Unity poseen las siguientes características:

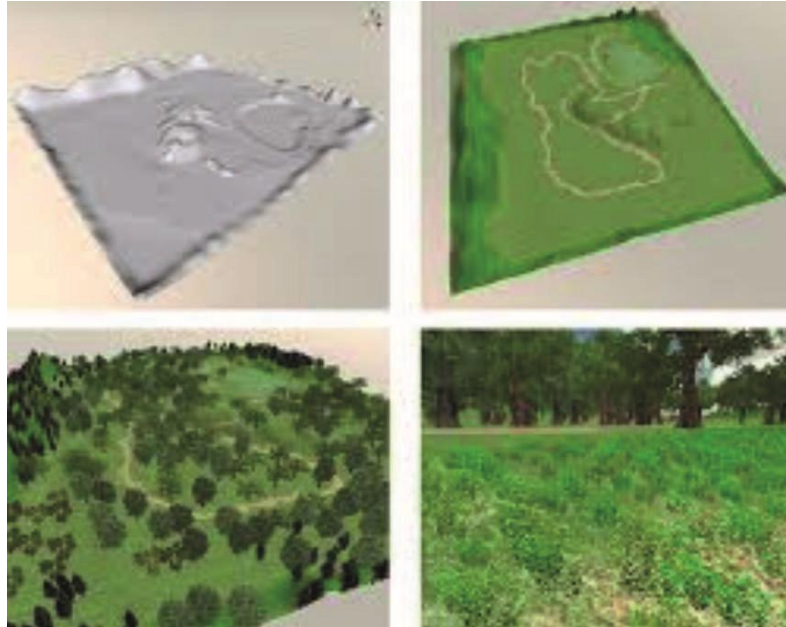
- Calidad gráfica en la simulación.
- Procesamiento en las variables físicas del entorno (gravedad, colisiones, deformaciones entre otras).

Características que un modelo 3D necesita para una interacción más realista de robotmedio; en comparación con los motores gráficos existentes Unity ofrece la mayor integración de *software* y *hardware*, e.g., Leap Motion, Oculus, Novint Falcon, ROS, entre otros, además que el uso de Unity es gratuito (Christ et al., 2013).

Unity 3D posiblemente es la tecnología que tiene mayor crecimiento en la actualidad como un motor de desarrollo gráfico en 3D, sin embargo, en la actualidad se ha incorporado

el desarrollo de juegos en 2D, Unity ofrece la posibilidad de crear escenarios abiertos totalmente editables y manipulables (García et al., 2017), tal como se muestra en la *Figura 1*.

Figura 1. Ambientes con variables físicas desarrolladas en Unity



2. Metodología

La importación de un modelo 3D realizado en cualquier *software* CAD a Unity no es directa, puesto que es necesario realizar modificaciones intermedias para integrar el funcionamiento del modelo diseñado.

A. Diseño mecánico del robot hexápodo

El proceso de simulación inicia con la creación del modelo 3D del robot, considerando los aspectos de modelación y aplicación. En el modelado se diseña la estructura tridimensional de cada una de las partes y elementos que conforman el prototipo robótico, a partir del uso de un *software* de Diseño Asistido por Computador (CAD, por su sigla en inglés), el cual tiene como objetivo desarrollar la fisonomía del robot tomando en cuenta las consideraciones que facilitan su movilidad para posteriores efectos de animación (Pacchierotti et al., 2014).

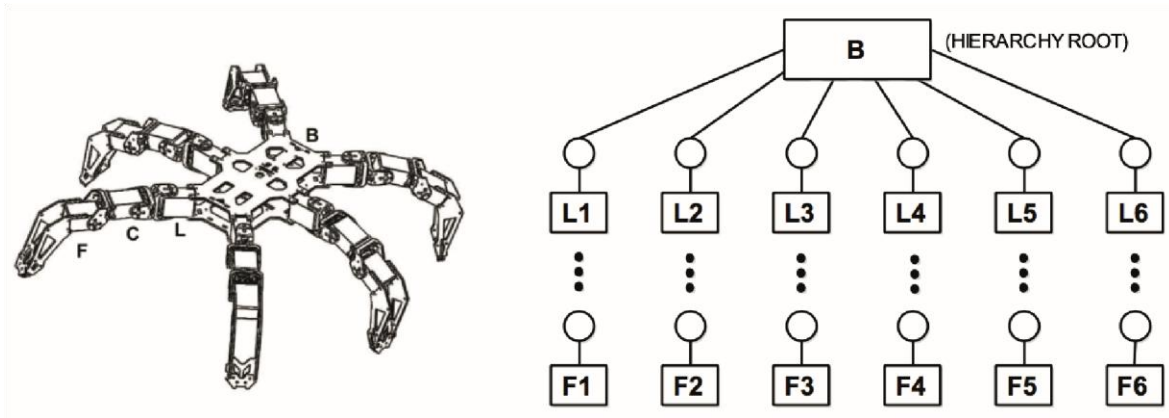
Un *software* de diseño mecánico asistido por computador es una herramienta de modelado sólido paramétrico, basado en operaciones que aprovecha la facilidad de aprendizaje de la interfaz gráfica, en esta herramienta se puede crear modelos sólidos en 3D totalmente asociativos con o sin restricciones, mientras utiliza al mismo tiempo las relaciones automáticas o definidas por el usuario para capturar la intención del diseño. Un modelo sólido geométrico contiene toda la geometría de superficie y alámbrica necesaria para describir detalladamente las aristas y las caras del modelo (Najmaei et al., 2016).

En el presente estudio se desarrolla un robot hexápodo que está conformado por servo motores de la marca Dynamixel serie AX-12A, los cuales tienen la capacidad de realizar un seguimiento de su velocidad, la temperatura, la posición del eje, tensión y carga. El algoritmo de control utilizado para mantener la posición del eje del actuador AX-12A se

puede ajustar de forma individual para cada servo, permitiendo controlar la velocidad y la fuerza de la respuesta del motor. El control de la gestión y de la posición del sensor se maneja por medio de una función de micro controlador del servo (Ponce et al., 2018).

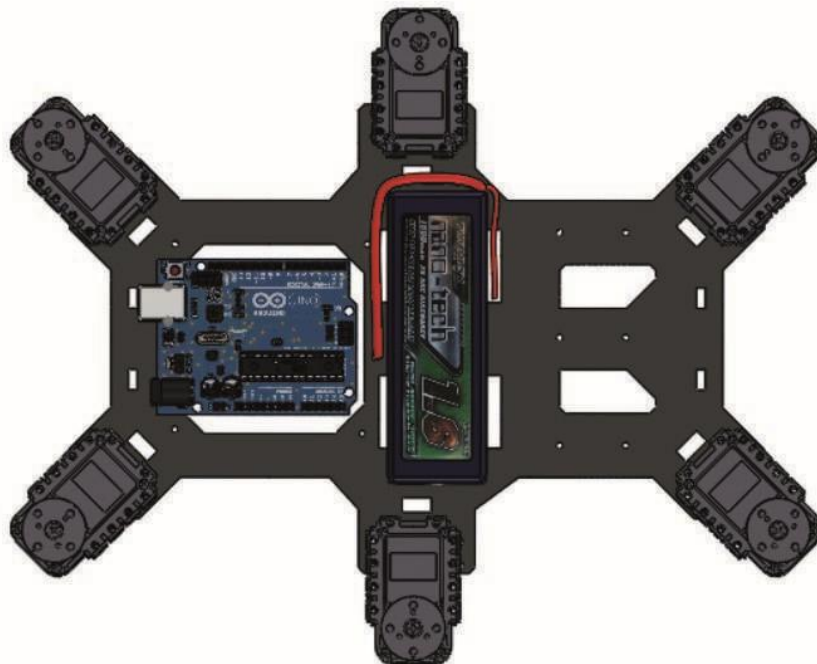
En la simulación del modelo 3D se debe considerar la jerarquía de los elementos que conforman el modelo (Najmaei et al., 2016). Esta consideración representa un conjunto de partes rígidas que están unidas a otras (anteriores y posteriores) por medio de ligaduras llamadas articulaciones. Este tipo de sistemas normalmente no tiene grados de libertad relacionados con las articulaciones, sino solamente con las rotaciones que obedecen a la mecánica del modelo 3D, tal como se muestra en la *Figura. 2*.

Figura 2. Esquema de Jerarquía del robot Hexapodo



El diseño del robot está basado en la locomoción de una araña y trata en su máxima expresión de igualar su movimiento, este diseño consta de una base con una pieza central en la cual se ubica el sistema de control, así como la alimentación –batería li-Po de 11.1v–, se fijan seis servo motores que servirán como locomoción principal para las extremidades del robot según se muestra la *Figura 3* (Yoshida et al., 2016).

Figura 3. Base central del Robot Hexápodo, batería y micro controlador



Cada una de las extremidades del robot posee tres grados de libertad que incluyen el servo motor ubicado en la parte central, tal como se muestra en la *Figura 4*; el diseño tiene el objetivo de permitir al robot desplazarse en superficies lisas o irregulares, superficies boscosas, entornos con rocas o arena.

Figura 4. Extremidad del Robot Hexápodo



El resultado final del diseño del robot se muestra en la *Figura 5*, en donde se puede apreciar que cada uno de los servo motores de la base están sujetos a cada una de las extremidades; la ubicación de las mismas permite tener un movimiento omnidireccional o no holonómico; el material de toda la estructura del robot se ha configurado en polimetilmetacrilato debido a características como peso y resistencia a deformaciones (Kenneally et al., 2016).

Figura 5. Extremidad del Robot Hexápodo



B. Simulación en un entorno en realidad virtual

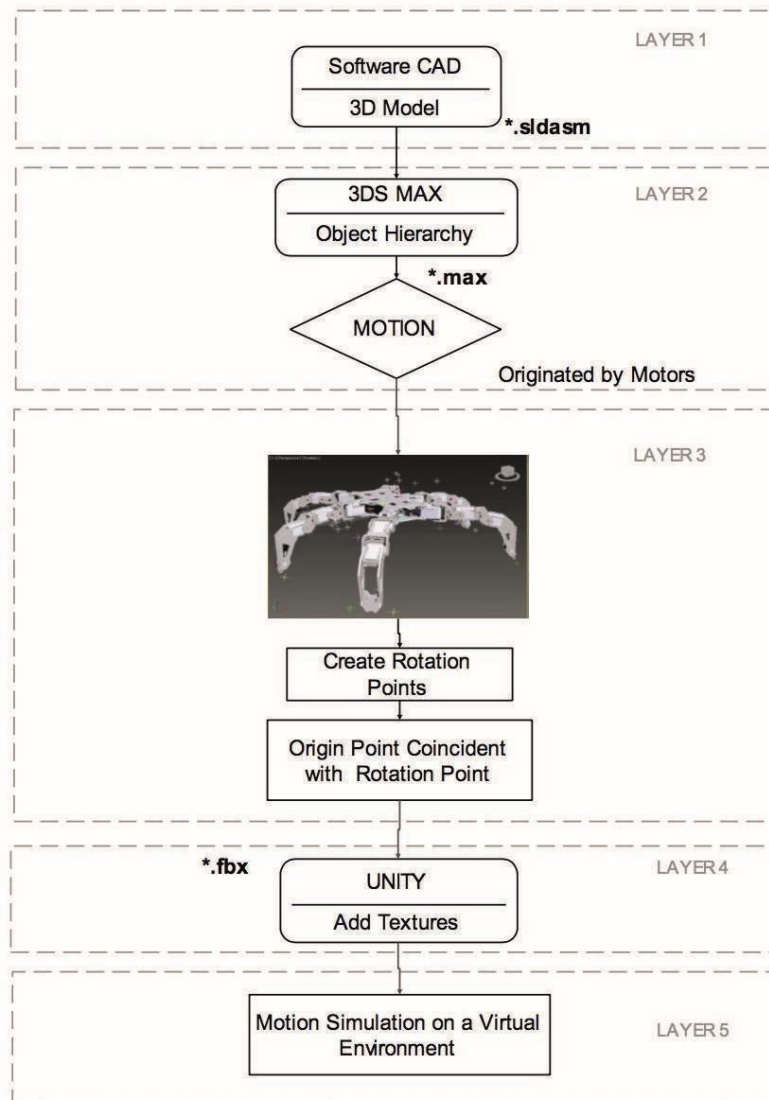
En la elaboración de una simulación del diseño es necesario usar un *software* que permite la conversión de archivos. Mediante la jerarquía se define los requerimientos que deben tener los elementos, esto permite realizar el movimiento controlado del diseño 3D de una manera idéntica a un entorno real, la mayoría de las piezas del modelo 3D se desplazan y rotan con respecto a otras, por lo cual se debe considerar técnicas de animación (Cully et al., 2015).

La simulación en función de jerarquías considera un elemento rígido o estático que formaría un sistema de referencia centrado en el origen de cada articulación. Por consiguiente cada vez que la posición de una articulación perteneciente a la una jerarquía se debe:

- Trasladar el sistema de coordenadas hasta el nuevo punto.
- Girar el sistema de coordenadas según el estado de la articulación.

El proceso de creación de una simulación de movimiento de cualquier modelo 3D, generado en un *software* CAD y relacionarlo con variables físicas, se puede dividir en seis capas: creación del modelo 3D, jerarquías de los componentes, movimiento rotación, texturas y simulación, como se detalla en la *Figura 6*.

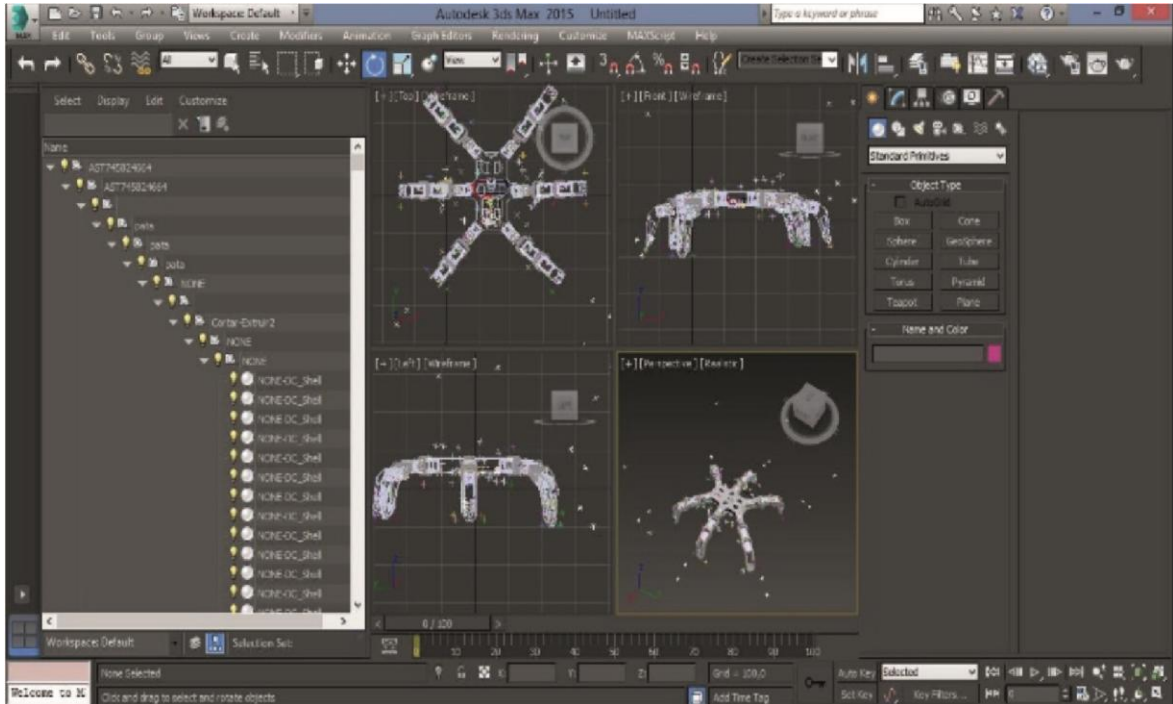
Figura 6. Capas para el proceso de la animación virtual



Una vez finalizado el proceso de diseño del robot en el *software* CAD en este caso específico, SolidWorks genera un archivo con extensión *–sldasm–* el cual será la base para la importación desde el *software* de diseño hasta el de renderizado y modelado, Unity.

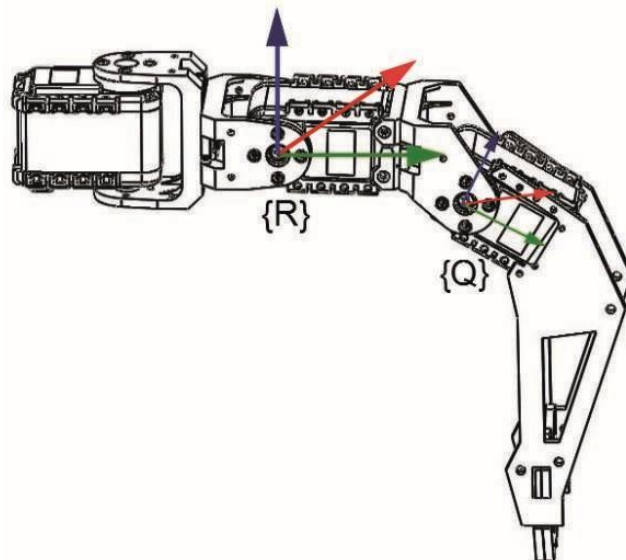
En la etapa de jerarquías, se toma en cuenta que Unity no importa archivos con la extensión obtenidos en la etapa anterior, razón por la cual se usa Auto-desk 3DS MAX en virtud que permite importar el archivo generado por el *software* CAD, y crear jerarquías, al realizar la importación, cada elemento del modelo en 3D mantiene su posición en el espacio de ensamblaje dentro de la estructura del robot, sin embargo, sus relaciones y restricciones de posición con los elementos que le rodean se pierden durante este proceso, para mantener: posición y relación de cada elemento del modelo 3D se utiliza jerarquías, mientras más elementos con detalles de relación y restricción de posición tenga el modelo 3D, existirá un mayor número de jerarquías; establecidas las jerarquías necesarias se obtiene un archivo con una extensión propia de este *software* *–.max–*. tal como se muestra en la *Figura 7* (Ardakani et al., 2018).

Figura 7. Creación de las jerarquías



En la etapa de asignación de movimientos se considera crear un movimiento por medio del uso de los servo motores de cada extremidad del robot, este movimiento genera puntos de rotación que se ubican mediante el uso del *software* 3DS MAX, se establece que el punto de origen de movimiento en cada articulación del modelo 3D debe coincidir con la ubicación de estos puntos, a su vez que deben estar ubicados de preferencia en los ejes de cada motor, tal como se muestra en la *Figura 8*.

Figura 8. Puntos de rotación de una extremidad del robot



El uso de 3DS MAX permite la obtención de un archivo del tipo *-.fbx-* el mismo que puede ser ejecutado en Unity.

En la etapa texturas se importa el archivo obtenido en la capa anterior, este archivo no posee texturas *-color-*, por consiguiente mediante el uso de Unity se añaden estas texturas que proporcionan al modelo 3D un acabado casi idéntico al del mundo real.

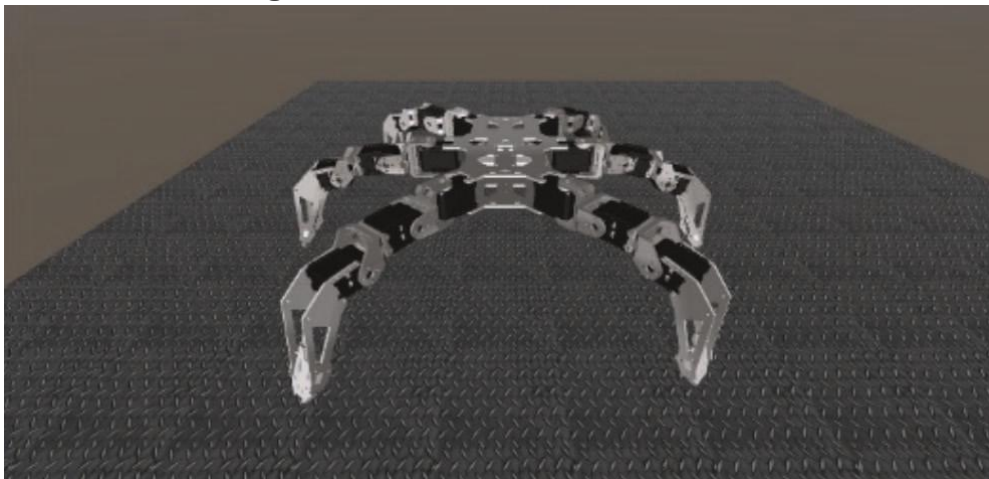
Mediante la capa generación de movimiento, se considera el uso de los puntos de rotación, y se establece el control de movimiento en cada uno de estos puntos en donde la locomoción del mecanismo es generada; establecidos en la capa tipo de movimiento.

En la etapa de simulación, se añaden variables físicas como gravedad rugosidad del suelo, rocas, diferentes tipos de ambientes, al control establecido en la capa generación de movimiento, para obtener una simulación de movimiento del modelo del robot en un ambiente casi real.

3. Resultados y discusión

El control de movimiento se establece en cada uno de los puntos de rotación y translación definidos en los requerimientos del sistema, los mismos generan el movimiento del robot como en el mundo real. Una vez establecido el control de movimiento se da inicio a la etapa de simulación, la misma que permite añadir variables físicas como gravedad, rugosidad del suelo; elementos reales tales como rocas, césped, etc.; tal como se muestra en la *Figura 9*. Estas variables permiten obtener una simulación de movimiento del modelo del robot en un ambiente casi real.

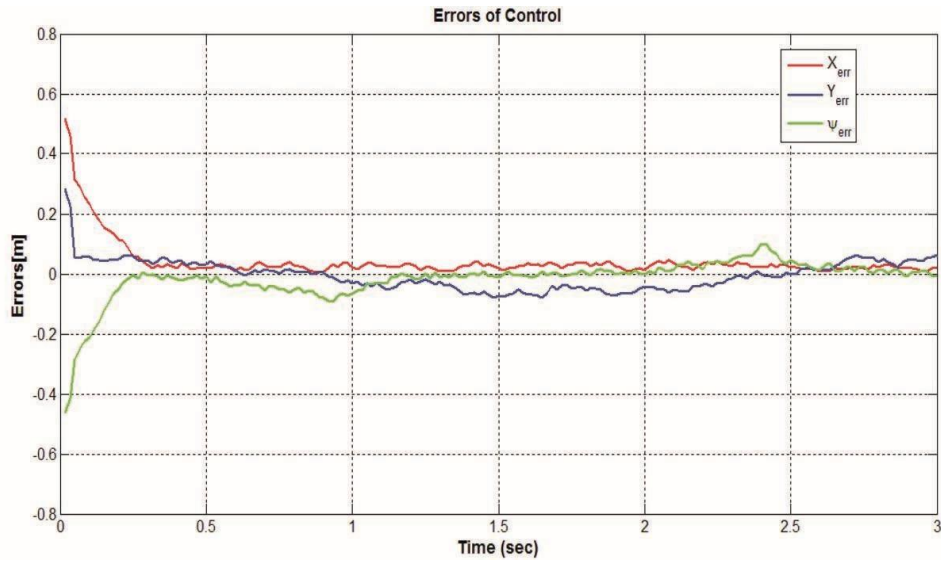
Figura 9. Simulación en un entorno virtual



Las consideraciones tomadas respecto de la simulación en un entorno de realidad virtual junto al robot hexápodo permitieron mediante la retroalimentación del sistema disminuir la complejidad de la construcción de un robot, así como evitar gastos innecesarios sin conocer el desempeño del robot en el mundo real.

La simulación permite conocer el error en la retroalimentación del sistema de control en el de movimiento de un robot con ciertas consideraciones físicas, por ejemplo, para este caso muy específico se elaboró un control de movimiento con ecuaciones en el método de Euler y un ambiente animado con un suelo metálico provisto de rugosidades, obteniendo el resultado descrito en la *Figura 10* (Pourebrahim et al., 2016).

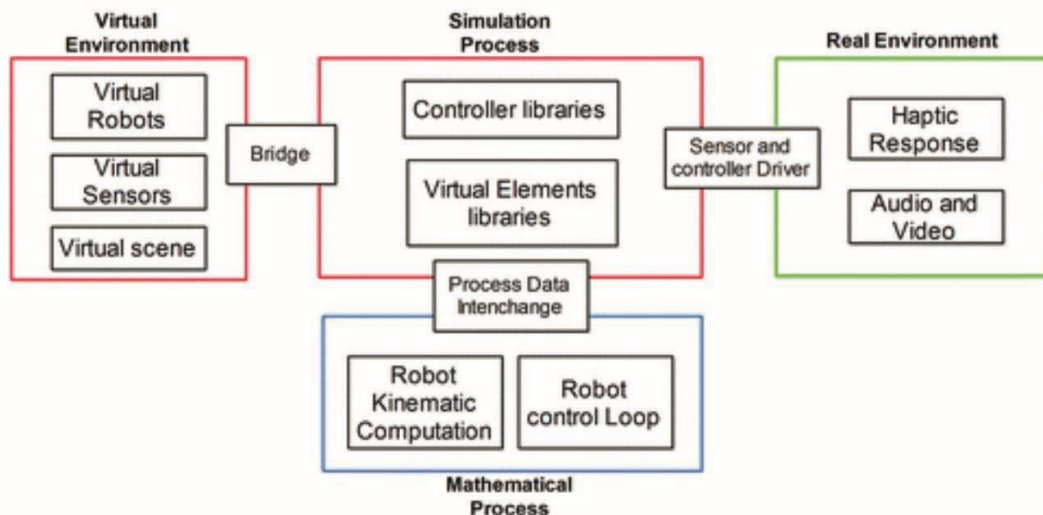
Figura 10. Simulación en un entorno virtual



En la *Figura 10* se presenta el resultado de la simulación del desplazamiento en los ejes X, Y del robot, respecto del tiempo en que el controlador diseñado por *software* busca realizar la corrección de la posición deseada, en la que el robot debe posicionarse, asociando dicho error a la existencia de motores y elementos mecánicos que en el arranque del proceso generan picos de velocidad y, por ende, de desplazamiento que con el transcurrir del tiempo buscan estabilizarse en el *set point* deseado. El sistema ha sido diseñado para ser un lazo retroalimentado cerrado donde el error tiende a ser 0, al igual que en una situación real un controlado de tipo PID en lazo cerrado buscaría corregir la diferencia del error entre el *set point* y la señal adquirida de la posición hasta que tienda a ser 0.

La animación puede ser ejecutada para cualquier tipo de diseño de robot diseñado mediante *software* CAD, un esquema general se presenta en la *Figura 11*.

Figura 11. Proceso general de animación virtual (Andaluz, Chicaiza, *et al.*, 2016)



5. Conclusiones y recomendaciones

La metodología descrita en este artículo permitió la elaboración de un proceso de simulación donde se analiza el comportamiento físico bajo un modelo de control desarrollado por *software* de un prototipo robótico diseñado en 3D mediante el uso de animaciones de entornos reales virtuales.

La utilización de una simulación donde los ambientes de desempeño del robot influyan de forma física en su comportamiento abre varias posibilidades de investigación para la creación de escenarios con elementos o condiciones que en la realidad serían difíciles o imposibles de crear, como puertas que se abren a distancia, paredes que pueden romperse de manera instantánea, ambientes peligrosos o tóxicos, terrenos escabrosos, etc; el uso de esta animación virtual con entornos físicos reales permite generar una retroalimentación continua y mejorar los diseños mecánicos 3D, cálculos de dinámica y cinemática, estrategias de control, elementos mecánicos para sujeción de robots que hayan sido realizado bajo un *software* CAD.

El proceso desarrollado está compuesto por dos capas de software: la simulación y la visualización. Ambas se desarrollaron en Unity de forma que estuviesen desacopladas, de tal modo que es posible, por ejemplo, desactivar la visualización y ejecutar simulaciones enteras de manera inmediata, lo cual es útil para estudiar los resultados como, por ejemplo, el de un algoritmo de movimiento, piezas diseñadas de forma errónea, estructuras sin soporte, mientras que la conjunción de ambas capas brinda la experiencia de usuario completa y brinda la motivación para el desarrollo de habilidades de programación, diseño y elaboración de robots.

Referencias

- Al Mashagbeh, M., y Khamesee, M. B. (2015). Unilateral teleoperated master-slave system for medical applications. *IFAC-PapersOnLine*, 48 (3), 784-787.
- Andaluz, V. H., Chicaiza, F. A., Gallardo, C., Quevedo, W. X., Varela, J., Sánchez, J. S., y Arteaga, O. (2016). *Unity3D-MatLab simulator in real time for robotics applications*. Paper presented at the International Conference on Augmented Reality, Virtual Reality and Computer Graphics, pp. 246-263.
- Andaluz, V. H., Quevedo, W. X., Chicaiza, F. A., Varela, J., Gallardo, C., Sánchez, J. S., y Arteaga, O. (2016). *Transparency of a bilateral tele-operation scheme of a mobile manipulator robot*. Paper presented at the International Conference on Augmented Reality, Virtual Reality and Computer Graphics.
- Ardakani, M. M. G., Karlsson, M., Nilsson, K., Robertsson, A., y Johansson, R. (2018). *Master-Slave Coordination Using Virtual Constraints for a Redundant Dual-Arm Haptic Interface*. Paper presented at the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- Bonilla, V., Moya, M., Evgeny, A. V., Lukyanov, A., y Leonardo, M. (2018). Modeling and simulation of the Mitsubishi RV-2JA Robot controlled by electromyographic signals. *Enfoque UTE*, 9(2), 208-222.
- Carlson, J., y Murphy, R. R. (2005). How UGVs physically fail in the field. *IEEE Transactions on robotics*, 21(3), 423-437.
- Chamorro, W., y Rueda, P. V. (2018). Linear Quadratic Regulator and Model Predictive Control Applied to a Four-Tank System: A Performance Comparison. *Enfoque UTE*, 9(1), 204-216.
- Christ, R. D., y Wernli Sr, R. L. (2013). *The ROV manual: a user guide for remotely operated vehicles*: Butterworth-Heinemann.
- Cully, A., Clune, J., Tarapore, D., y Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553), 503.
- Desbats, P., Geffard, F., Piolain, G., y Coudray, A. (2006). Force-feedback teleoperation of an industrial robot in a nuclear spent fuel reprocessing plant. *Industrial Robot: An International Journal*, 33(3), 178-186.
- Enríquez, D., Guerrón, G., y Moya, M. (2018). *Mechatronic Mobil System for Telemedicine*. Paper presented at the 2018 International Conference on Information Systems and Computer Science (INCISCOS).

- Faria, B. M., Reis, L. P., Lau, N., Moreira, A. P., Petry, M., y Ferreira, L. M. (2015). *Intelligent wheelchair driving: bridging the gap between virtual and real intelligent wheelchairs*. Paper presented at the Portuguese Conference on Artificial Intelligence.
- Freund, E., y Rossmann, J. (1999). Projective virtual reality: Bridging the gap between virtual reality and robotics. *IEEE transactions on robotics and automation*, 15 (3), 411-422.
- García, J. C., Patrão, B., Almeida, L., Pérez, J., Menezes, P., Dias, J., y Sanz, P. J. (2017). A natural interface for remote operation of underwater robots. *IEEE computer graphics and applications* (1), 34-43.
- Kenneally, G. D., De, A., y Koditschek, D. E. (2016). Design Principles for a Family of Direct-Drive Legged Robots. *IEEE Robotics and Automation Letters*, 1 (2), 900-907.
- Lawrence, D. A. (1993). Stability and transparency in bilateral teleoperation. *IEEE transactions on robotics and automation*, 9 (5), 624-637.
- Mukherjee, J. (2014). *Fast visualisation technique for view constrained tele-operation in nuclear industry*. Paper presented at the Information Science and Applications (ICISA), 2014 International Conference on.
- Najmaei, N., Asadian, A., Kermani, M. R., y Patel, R. V. (2016). Design and performance evaluation of a prototype MRF-based haptic interface for medical applications. *IEEE/ASME Trans. Mechatronics*, 21 (1), 110-121.
- Pacchierotti, C., Tirmizi, A., y Prattichizzo, D. (2014). Improving transparency in teleoperation by means of cutaneous tactile force feedback. *ACM Transactions on Applied Perception (TAP)*, 11 (1), 4.
- Ponce, R., Canchingre, G. M., Velarde, P., y Moya, M. (2018). *Design and Construction of an Automatic Transport System Inside the Home for People with Reduced Mobility*. Paper presented at the 2018 International Conference on Information Systems and Computer Science (INCISCOS).
- Pourebahim, M., Ayati, M., y Mahjoob, M. (2016, 27-29 de julio de 2016). *Design and implementation of PI and fuzzy PID supervisory controllers for a flexible link robot*. Paper presented at the 2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE).
- Sánchez, M., Bonilla, V., Moya, M., y Mosquera, G. (2017). *Analysis and Construction of a Mobile Robot for Search and Mitigation of Low-Scale Fire*. Paper presented at the Information Systems and Computer Science (INCISCOS), 2017 International Conference on.
- Yoshida, K., Wilcox, B., Hirzinger, G., y Lampariello, R. (2016). Space robotics. En *Springer Handbook of Robotics* (pp. 1423-1462): Springer.