

Modelo virtual con comportamiento dinámico del robot humanoide Bioloid PremiumTM

Regular Paper

Dora A. Rodríguez-Vega*, Ulises Zaldívar-Colado *, José V. Nuñez-Nalda[†]

* Corresponding author E-mail: da.rodriguez11@info.uas.edu.mx

Received: 9 november 2014; Available online: 15 september 2015

© 2015 ; licensee AMRob. This is an article distributed under the terms of the Instituto Nacional del Derecho de Autor (www.indautor.gob.mx), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Resumen La simulación de sistemas robóticos que incuya comportamiento dinámico, juega un rol importante en el desarrollo de nuevos sistemas, así como en su control. Existen muchas librerías que funcionan como motores de física: ODE, Bullet, Havok, Newton, PhysX, entre otros, sin embargo en las evaluaciones comparativas entre diversos motores de física se concluye que los mejores para actividades de locomoción en humanoides son PhysX y Bullet. La mayoría de los simuladores de robots comerciales utilizan como motor de física ODE. Este documento presenta el desarrollo de un modelo virtual con comportamiento dinámico para el robot Bioloid PremiumTM haciendo uso de la librería de gráficos OpenGL y del motor de física PhysX. El modelo virtual ayudará a los usuarios a analizar posibles trayectorias del robot sin contar con el robot real y sin el riesgo de daños ocasionados por caídas.

Keywords Humanoides, simulador, Bioloid, motor de física.

1. Introducción

Los simuladores son una herramienta muy utilizada en robótica, ya que ayudan a analizar diferentes alternativas

de controladores sin contar con el robot real y sin los riesgos de daños. Un modelo virtual que solo hace uso de una librería de gráficos, puede mostrar el comportamiento del robot, es posible indicar trayectorias y analizarlas, sin embargo, durante colisiones y autocolisiones los objetos se traslapan, es decir, si se presenta un objeto en el ambiente virtual del robot, el eslabón del robot puede sobreponerse visualmente al objeto sin indicar la colisión, en caso de que la colisión sea entre dos eslabones del robot, éstos se sobreponen visualmente, además el robot se presenta flotando en el ambiente virtual, no tiene masa, por lo que no es posible analizar el comportamiento debido al peso o a la inercia al implementar las trayectorias. En [1] se presenta un modelo virtual para el robot Bioloid PremiumTM con comportamiento cinemático, basado únicamente en la librería de gráficos, dicho modelo se mueve indicando posiciones deseadas sin tomar en cuenta las fuerzas necesarias para lograr dicho movimiento. La Fig. 1 muestra un modelo virtual que no incuye motor de física, se observa que permite autocolisiones.

Un motor de física es una librería que añade al modelo virtual un comportamiento dinámico, es decir, los movimientos se realizan mediante fuerzas y torques, permite detectar colisiones y modelar el comportamiento dinámico del objeto virtual mediante parámetros como forma y material (densidad). Durante el diseño de controladores en bípedos un motor de física nos permite analizar si la trayectoria producirá una caída en el robot.

*D. A. Rodríguez-Vega y U. Zaldívar-Colado, Facultad de Informática, Universidad Autónoma de Sinaloa, Culiacán, Sinaloa, da.rodriguez11@info.uas.edu.mx

[†]Ingeniería Mecatrónica, Universidad Politécnica de Sinaloa, Mazatlán, Sinaloa.

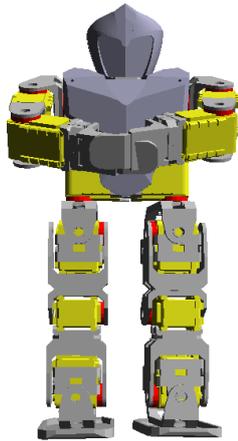


Figura 1. MODELO VIRTUAL QUE NO INCLUYE MOTOR DE FÍSICA

Existen varios motores de física, entre ellos están ODE, Bullet, Vortex, Havok, Newton y PhysX.

En [2] se presenta un análisis comparativo entre tres motores de física: Newton, ODE y PhysX; en la mayoría de las pruebas PhysX tuvo el mejor comportamiento. En [3] se muestra una comparación entre PhysX, Bullet, JigLib, Newton, ODE, Tokamak y True Axis, concluyendo que PhysX tiene el mejor método integrador, aunque Bullet presenta el sistema de detección de colisiones más robusto. Por otra parte en [4] el análisis comparativo entre PhysX, Havok, ODE y Bullet, en cuanto a detección de colisiones en cuerpos rígidos muestra que los mejores resultados los proporciona PhysX.

En [5] se muestra una evaluación de motores de física con enfoque a simulación dinámica de robots caminantes, resaltando la importancia de la correcta simulación durante el caminado, así como la detección de colisiones, ya que cada paso del robot tiene una colisión con el terreno donde se mueve, situaciones en las que PhysX resultó bien evaluado, sin embargo, se indica que también es importante el modelo de fricción durante el caminado, ya que previene el deslizamiento no deseado de los pies del robot, prueba en la que Bullet tuvo un mejor desempeño.

En [6] se propone una metodología para evaluar motores de física en tareas de ensamble virtual háptico, mostrando resultados comparativos entre Bullet y PhysX, precisamente los mejores evaluados en [2-5] concluyendo que PhysX se ve afectado fuertemente por la complejidad de las formas, por lo que para ensamblado con piezas de formas no complejas, PhysX proporciona mejores resultados, pero para formas complejas como las cóncavas, Bullet tiene un mejor desempeño.

2. Trabajos relacionados

En esta sección se presentan algunos simuladores con comportamiento dinámico para robots.

WEBOTSTM [7] es un simulador de robots comercial, proporciona algunos modelos de robots y permite modelar otros, los robots son modelados mediante VRML y los controladores se programan en C++ o Java.

SimRobot [8], desarrollado por la Universidad Bremen, soporta diferentes tipos de robots y ambientes modelados en XML, con facilidad para programar controladores. UCHILSIM [9] desarrollado por la Universidad de Chile diseñado para la Liga de cuadrúpedos de la Robocup. Los robots son modelados en VRML. UBERSIM [10] desarrollado por Carnegie Mellon University modela los robots mediante clases.

También hay simuladores para robots específicos como Virtual RE [11] desarrollado por Singapore Polytechnic, basado en UberSim y diseñado para el robot Robo-erectus Junior-BX; OPENHRP [12] para el robot HRP-2 y iCub simulator [13] para el robot iCub.

Gazebo [14] es un simulador que fue diseñado para analizar estrategias de trabajo colaborativo entre robots.

Para todos los simuladores mencionados, su comportamiento dinámico está basado en el motor de física ODE.

Por otra parte Robotics Developer Studio [15] basa su comportamiento dinámico en el motor de física PhysX, sin embargo, sólo para un conjunto definido de robots.

3. Descripción del humanoide

El robot a modelar es el humanoide Bioloid PremiumTM de la compañía RobotisTM, en su configuración A, contiene una unidad de control denominada CM-530 basada en el microcontrolador ARM Cortex, 18 motores dynamixel AX-12, pesa 1.7 Kg. y mide 39.7 cm. La Fig. 2, muestra la imagen del robot, los 18 grados de libertad se distribuyen seis en cada pierna y tres en cada brazo.



Figura 2. Robot humanoide Bioloid PremiumTM configuración A.

4. Arquitectura del sistema

Para el desarrollo del software, se siguió la arquitectura presentada en la Fig. 3.

Se inicia convirtiendo los modelos proporcionados por el fabricante, RobotisTM, en formato IGS (Initial Graphics Exchange Specification), a archivos en formato TRI mediante el software de intercambio de formatos Deep ExplorationTM y la herramienta MakeTRI[16]. La conversión de partes mecánicas 3D a malla poligonal genera un archivo de texto con extensión TRI, este archivo

contiene la descripción de los vértices y caras de la malla poligonal de la parte mecánica. Los archivos en formato TRI, son utilizados por la librería del motor de física PhysX para crear las formas (Shapes); una o varias formas se utilizan para crear un actor. Los actores se conectan mediante uniones rotacionales de un solo grado de libertad. El movimiento se logra aplicando torques a cada actor.

Es posible tener acceso a datos de cada actor: masa, posición global, centro de masa y tensor de inercia, que pueden ser utilizados para la programación del controlador.

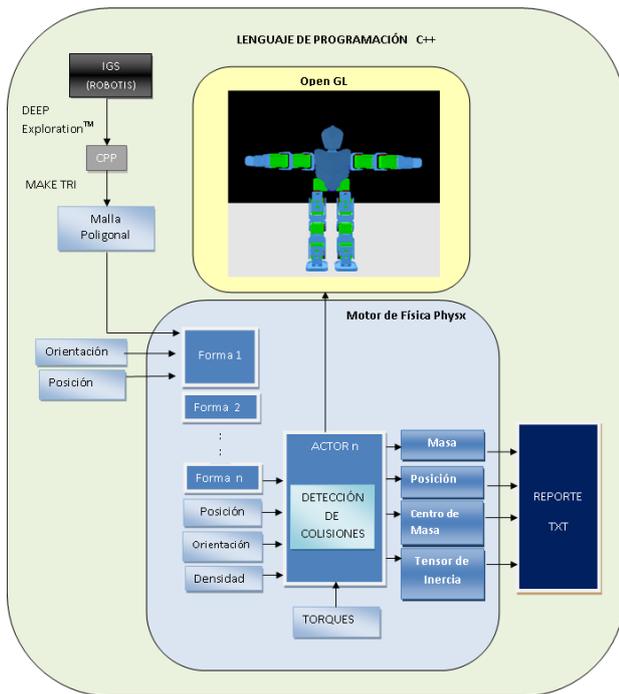


Figura 3. Arquitectura del sistema.

4.1. Formas

Toman la información de la malla triangular y forman una estructura de datos que permite la detección de colisiones.

Para cada parte requerida para ensamblar el robot, el fabricante proporciona en modelo geométrico en el formato IGS (Initial Graphics Exchange Specification) el cual mediante DEEP Exploration es convertido a formato CPP, posteriormente mediante la herramienta MAKETRI, la información de la malla triangular en el archivo CPP es reorganizada y colocada en un archivo TRI. De esta manera, se crearon las formas de cada parte requerida para realizar el ensamble del robot excepto los tornillos. La Fig. 4 muestra ejemplos de formas utilizadas.



Figura 4. Ejemplos de formas.

4.2. Actor

Un actor es un ente individual en el ambiente virtual con marco de referencia propio y que responde a detección de colisiones con otros actores.

Para PhysX un actor es un objeto de simulación, al cual podemos indicarle su posición, orientación y densidad; puede encapsular una o más formas y para moverlo es necesario aplicarle una fuerza o torque. La detección de colisiones asegura que las formas de un actor no se traslapen con las formas de otro actor.

Se definió un actor para cada eslabón del robot Bioloid, es decir, para cada conjunto de elementos actuados por un motor. Los actores son:

Torso, cada brazo está formado por tres eslabones: Hombro 1, Hombro 2 y Codo; cada pierna está formada por seis segmentos: ingle, cadera, 1 cadera 2, rodilla, tobillo 1 y tobillo 2. La Fig. 5 muestra ejemplos de actores.

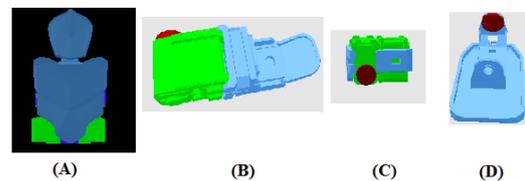


Figura 5. EJEMPLOS DE ACTORES, (A) TORSO, (B) CODO, (C) TOBILLO 1, (D) TOBILLO 2

El valor de densidad seleccionado es de 1350 Kg/m^3 que corresponde a los plásticos de ingeniería indicados por el fabricante.

Para acomodar cada una de las formas en cada actor se indica posición y orientación mediante la matriz homogénea. Para acomodar cada actor se indica posición y orientación mediante otra matriz homogénea.

4.3. Estructura Padres e Hijos

Las matrices homogéneas indican posición y orientación en la forma para constituir un actor y en el actor para formar el robot Humanoide; ésta información es con respecto a los marcos locales, por lo que es necesario identificar la secuencia tanto en las formas como en los actores, para ello se define una estructura de padres e hijos, a cada forma y actor se le indica el padre y con ello el marco referencial para la posición y orientación indicados. La Fig. 6 muestra la estructura padres e hijos para los actores.

4.4. Uniones

Las uniones son una forma de conectar dos actores. Sin ellas, cualquier fuerza (incluida la gravedad) puede separar los actores.

El motor de física puede representar diferentes tipos de uniones, entre ellas están las prismáticas, esféricas y rotacionales. La primera permite movimiento de traslación entre dos actores a lo largo de un eje definido. Las uniones esféricas permiten movimiento en los tres ejes. Las uniones rotacionales sólo permiten movimiento de rotación en un eje definido. La Fig. 7 muestra ejemplos de cada una de ellas.

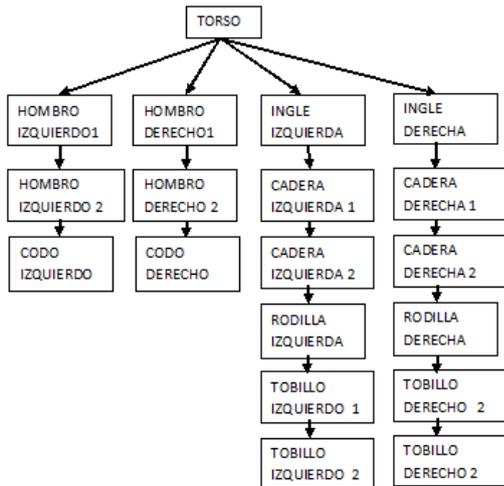


Figura 6. Estructura padres e hijos.

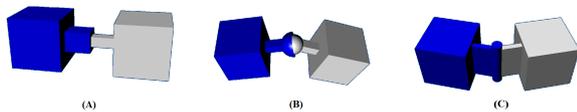


Figura 7. Ejemplos de uniones en PHYSX: (A) prismáticas, (B) esféricas Y (C) rotacionales.

Para el simulador propuesto se seleccionaron uniones rotacionales que solo permiten movimiento en un eje, indicando los dos actores a unir, el eje de rotación, y el punto de unión. Se colocaron 18 uniones entre cada motor y rotor, el eje fue seleccionado de acuerdo a la orientación del motor.

5. Propiedades dinámicas

Al crear los actores (después de indicar formas y densidad) PhysX calcula los parámetros dinámicos masa, posición global, centro de masa y tensor de inercia.

Estos parámetros serán utilizados para la evaluación numérica del modelo dinámico con los datos deseados de posición, velocidad y aceleración y realizar las correcciones requeridas para lograr movimientos estables (control).

6. Resultados

Se realizaron pruebas para verificar que no se traslapan los actores del robot con actores externos, el torso permanece fijo, por lo que el robot da la apariencia de estar 'colgado'.

Las Fig. 8, 9 y 10 muestran el comportamiento del robot al colocar una esfera cerca del brazo izquierdo, derecho y entre las piernas del robot.

La Fig. 11 muestra como cae el robot al soltar el torso que permaneció fijo en los ejercicios anteriores.

Se colocaron dos esferas frente al robot para observar su comportamiento al caer cerca de ellas. Las Figuras 12 y 6 muestran los resultados.

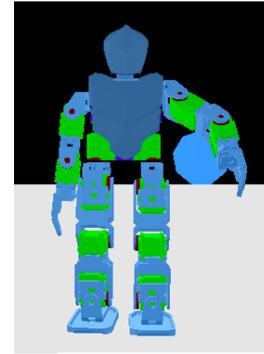


Figura 8. Brazo izquierdo del robot interactuando con esfera.

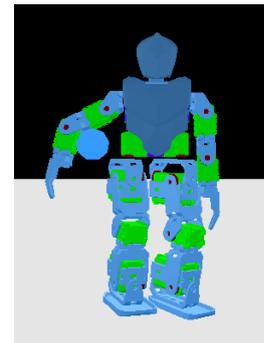


Figura 9. Brazo derecho del robot interactuando con esfera.

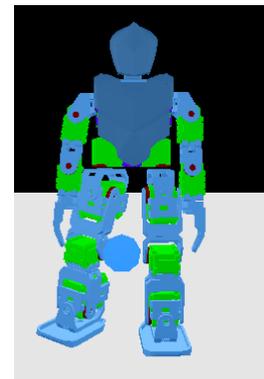


Figura 10. Piernas del robot interactuando con esfera.

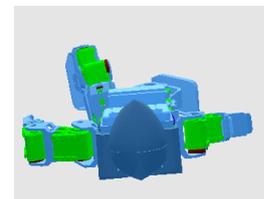


Figura 11. Robot bajo efectos de la fuerza de gravedad.

Se realizó una comparación entre el peso de los actores calculado por PhysX y el real, dando como resultado diferencias entre 0.8 y 18 %.

La metodología utilizada para el desarrollo del modelo propuesto puede ser empleada para la creación de nuevos modelos para otros robots.

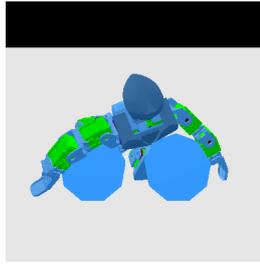


Figura 12. Comportamiento del robot al caer sobre dos esferas.

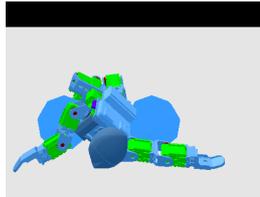


Figura 13. Comportamiento del robot al caer sobre dos esferas.

Para el diseño de un simulador con comportamiento dinámico se requiere:

- 1 El modelo geométrico de cada una de las piezas.
- 2 Datos de posición y orientación de cada pieza para formar un segmento del robot
- 3 Identificar las piezas (formas) que se requieren para cada segmento del robot (actor), ya que el colocar cada pieza (forma) como un actor, implica tener uniones fijas, manejo de colisiones entre ellas; resultando en una simulación inestable y muy lenta.
- 4 Posición y orientación de cada actor.
- 5 Identificar material a utilizar y su densidad para el cálculo de masa.
- 6 Identificar los tipos, ejes y puntos de unión.

7. Conclusiones

Se diseñó un modelo virtual con comportamiento dinámico para el robot Bioloid PREMIUMTM, el cual permite observar el comportamiento real del robot ante un conjunto de trayectorias para sus 18 grados de libertad. Se genera además un reporte con los parámetros dinámicos de los eslabones del robot que serán utilizados en el diseño del controlador.

Como trabajo futuro se indica verificar los datos de centro de masa y tensor de inercia para cada actor mediante el software SolidWorksTM. Para ello, es necesario definir los marcos locales y verificar el centro de masa de los motores, ya que no es una forma isotrópica, es decir, la densidad del motor varía debido a que los elementos internos no están distribuidos de manera uniforme. De la misma manera se pretende implementar el cálculo del punto de momento cero (Zero Moment Point, ZMP) que será de utilidad en el diseño de controladores.

El modelo dinámico de un robot permite conocer las fuerzas y torques necesarios para lograr cierto movimiento con velocidad y aceleración definidas tomando como parámetros masa y tensor de inercia. El modelo virtual con

comportamiento dinámico propuesto permite tener acceso a estos datos, así mismo, para verificar el modelo dinámico calculado matemáticamente es necesario definir una trayectoria para posición, velocidad y aceleración para cada grado de libertad (q_i , \dot{q}_i y \ddot{q}_i) y verificar que el cálculo de fuerzas y torques resulte en el movimiento solicitado. El simulador propuesto permite verificar esto, proporcionando información visual y numérica, misma que será de utilidad en el diseño de controladores.

Agradecimientos

Los estudios de doctorado del primer autor son financiados por CONACYT.

Este proyecto se realiza con financiamiento del Programa de Fomento y Apoyo a Proyectos de Investigación (PROFAPI) 2013.

Referencias

- [1] Dora A. Rodríguez-Vega, Ulises Zaldivar-Colado, José V. Nuñez-Nalda, Abraham Briseño-Cerón . *Teleoperación en Línea de un Humanoide Virtual Mediante la Manipulación de un Humanoide Real*. Congreso Mexicano de Robótica 2012, Puebla, México.
- [2] Axel Seugling and Martin Rölin *Evaluation of Physics Engines and Implementation of a Physics Module in a 3d-Authoring Tool* Umeå University 2006
- [3] Boeing, Adrian and Bräunl, Thomas, *Evaluation of Real-time Physics Simulation Systems*, Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia GRAPHITE '07
- [4] Erwin Coumans and Keith Victor *COLLADA physics*. In Proceedings of the twelfth international conference on 3D web technology (Web3D '07). ACM, New York, NY, USA, 101-104.
- [5] Roennau, A; Sutter, F.; Heppner, G.; Oberlaender, J.; Dillmann, R., *Evaluation of physics engines for robotic simulations with a special focus on the dynamics of walking robots* Advanced Robotics (ICAR), 2013 16th International Conference on , vol., no., pp.1,7, 25-29 Nov. 2013
- [6] Germanico Gonzalez-Badillo, Hugo I. Medellin-Castillo, Theodore Lim, James M. Ritchie, Raymond C.W. Sung, and Samir Garbaya *A new methodology to evaluate the performance of physics simulation engines in haptic virtual assembly* Assembly Automation 2014 34:2, 128-140
- [7] Página web WEBOTS <http://www.cyberbotics.com/webots/>
- [8] Laue, Tim and Spiess, Kai and RÁúfer, Thomas *SimRobot A General Physical Robot Simulator and Its Application in RoboCup* Lecture Notes in Computer Science RoboCup 2005: Robot Soccer World Cup IX
- [9] Zagal, Juan Cristóbal and Ruiz-del-Solar, Javier *UCHILSIM: A Dynamically and Visually Realistic Simulator for the RoboCup Four Legged League* Lecture Notes in Computer Science RoboCup 2004: Robot Soccer World Cup VIII2005
- [10] Go, Jared and Browning, Brett and Veloso, Manuela, *Accurate and Flexible Simulation for Dynamic,*

Vision-Centric Robots Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3, AAMAS '04

- [11] Calderon, C.A.A and Mohan, R.E. and Changjiu Zhou, *Virtual-RE: A Humanoid Robotic Soccer Simulator* Cyberworlds, 2008 International Conference on, vol., no., pp.561,566, 22-24 Sept. 2008
- [12] Página WEB OPENHRP
<http://www.openrtp.jp/openhrp3/en/about.html>
- [13] Tikhanoff, V. and Cangelosi, A. and Fitzpatrick, P. and Metta, G. and Natale, L. and Nori, F., *An Open-source Simulator for Cognitive Robotics Research: The Prototype of the iCub Humanoid Robot Simulator*, Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems PerMIS '08,
- [14] Nathan Koenig and Andrew Howard, *Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator* In IEEE/RSJ International Conference on Intelligent Robots and Systems 2004
- [15] Página WEB Robotics Developer Studio
<http://msdn.microsoft.com/en-us/library/bb648760.aspx>
- [16] U. Zaldívar Colado *Planification d'Assemblage en Environnement Virtuel* Ph.D. thesis, Laboratoire d'Ingénierie des Systèmes de Versailles, Université de Versailles Saint Quentin-en-Yvelines, France, 2009.