

Simulación de Plataformas Robóticas de Movimiento para Aplicaciones de Realidad Virtual Mediante Filtros Digitales

Sergio Casas, Cristina Portalés*, Silvia Rueda, Marcos Fernández

Instituto de Robótica y Tecnologías de la Información y la Comunicación, Universitat de València, C/ Catedrático José Beltrán 2, 46980, Paterna, Valencia, España.

Resumen

El uso de plataformas robóticas de movimiento en simuladores de vehículos y aplicaciones de Realidad Virtual es relativamente habitual. Sin embargo, el ajuste de los algoritmos que controlan su funcionamiento, denominados algoritmos de *washout*, no es sencillo y requiere de numerosas pruebas hasta obtener una apropiada fidelidad de movimiento. Disponer de herramientas que permitan simular plataformas de movimiento puede permitir simplificar esta tarea. Es por ello que este trabajo presenta un método para la caracterización y simulación de manipuladores robóticos mediante filtros digitales de segundo orden, sencillo de implementar y ajustar a partir de una caracterización previa. El simulador se construye con el objetivo de permitir la simulación rápida de manipuladores robóticos y se ejemplifica con una plataforma de dos grados de libertad, aunque el método propuesto podría emplearse en otros dispositivos. En las pruebas realizadas se valida la precisión y velocidad de la simulación, concluyéndose que se obtiene una fidelidad satisfactoria y una velocidad de simulación elevada que permite emplear el simulador como sustituto del *hardware* real con algoritmos de *washout*.

Palabras Clave:

Plataformas de movimiento, simuladores, filtros digitales, realidad virtual, robótica, tiempo real.

1. Introducción

Un manipulador robótico es un sistema diseñado para desplazar elementos o coger piezas, habitualmente accionado de forma electro-mecánica. Estos dispositivos se utilizan en múltiples áreas de conocimiento (aplicaciones industriales (Vogel, Fritzsche, & Elkmann, 2016), cirugía (Cleary, 2016), entrenamiento de equipamiento (Lau, Chan, & Wong, 2007), rehabilitación médica (Fung et al., 2004), etc.), además de en aplicaciones de Realidad Virtual (RV) para la generación de claves gravito-inerciales en simuladores de vehículos (Slob, 2008), que es el área en la que se centra este trabajo. Los manipuladores robóticos que se centran en generar movimiento sobre una base rígida móvil se denominan habitualmente plataformas de movimiento y, aunque son esencialmente el mismo tipo de dispositivo, se diferencian de los manipuladores utilizados en otras áreas como la cirugía o la industria de manufactura en su diseño y finalidad.

Cada uso concreto de estos dispositivos conlleva unas particularidades especiales. Mientras que el uso en aplicaciones quirúrgicas requiere de alta precisión o el uso en la industria de manufactura precisa, además, de un apropiado nivel de repetibilidad, el uso en simulación de vehículos y RV suele requerir

velocidad y aceleración en detrimento de otras características como la precisión, la repetibilidad o la destreza (*dexterity*).

Para el control de alto nivel de los movimientos de una plataforma robótica en aplicaciones de RV se diseñan unos algoritmos específicos conocidos habitualmente como algoritmos de generación de claves gravito-inerciales (*Motion Cueing Algorithms* – MCA, en inglés) o algoritmos de *washout* (Casas, Olanda, & Dey, 2017). Aunque su uso en RV y simulación de vehículos es relativamente habitual, no es fácil diseñar y ajustar estos algoritmos para que generen las señales apropiadas para hacer funcionar correctamente estos dispositivos (Reid & Nahon, 1988). Por ello, es habitual tener que realizar numerosas pruebas sucesivas para lograr ajustar los algoritmos de *washout* y obtener el comportamiento deseado para el manipulador robótico (Grant & Reid, 1997). La tarea se puede automatizar parcialmente pero requiere de un número aún mayor de pruebas para obtener un comportamiento satisfactorio, como se realiza en Casas, Coma, Portalés, and Fernández (2016). Este proceso es costoso, puesto que el manipulador robótico tiene que ser primero construido (si no existe ya) y luego comprobado exhaustivamente para poder asegurar que se comporta adecuadamente para la aplicación escogida. Sin embargo, durante la fase de pruebas, se pueden realizar movimientos severos y potencialmente dañinos que pueden dañar la plataforma de movimiento, y lo que es más importante, provocar daños humanos en caso de fallo del *software* o *hardware* que se está probando. Para resolver estos problemas, es importante disponer de herramientas que permitan acortar las fases

* Autor en correspondencia.

Correos electrónicos: scasas@irtic.uv.es (Sergio Casas),
poricris@uv.es (Cristina Portalés), srueda@irtic.uv.es (Silvia
Rueda), marcos.fernandez@uv.es (Marcos Fernández),
URL: irtic.uv.es (Sergio Casas)

de prueba, y al mismo tiempo reducir costes y riesgos. Una de estas herramientas es un simulador de plataformas de movimiento. Este tipo de simuladores permite reducir los riesgos humanos (al reducir la posibilidad de movimientos indebidos en la fase de pruebas), económicos (permitiendo incluso evaluar la idoneidad de una plataforma antes de construirla) y acortar los plazos (permitiendo realizar pruebas de forma simulada mucho más rápido que de forma real). Este último punto es clave, ya que las simulaciones pueden realizarse más rápidamente que el tiempo real. Esto lo diferencia de las maquetas o prototipos (Ortega & Sigut, 2016), que permiten encontrar fallos de diseño, problemas mecánicos y errores en el proceso constructivo, pero no permiten acelerar las pruebas porque ejecutan *hardware* real.

El objetivo de este trabajo será derivar un método de simulación rápido y eficiente para poder emplear plataformas de movimiento simuladas en la verificación y pruebas de algoritmos de *washout* en simuladores de vehículos y aplicaciones de RV.

El resto del trabajo se organiza de la siguiente forma: el capítulo 2 revisa la literatura relacionada, el capítulo 3 está dedicado a describir el manipulador utilizado para ilustrar el procedimiento y el método de simulación, mientras que el capítulo 4 describe las pruebas y resultados obtenidos con el simulador propuesto. Finalmente, en el capítulo 5 se derivan las conclusiones y se señalan posibles líneas futuras de trabajo.

2. Trabajos Relacionados

La simulación de vehículos tiene una ya larga historia que comienza con los simuladores de vuelo y se remonta a las guerras mundiales (Page, 2000). Aunque las plataformas de movimiento tardaron algunos años más en ser utilizadas para incluir movimiento simulado en este tipo de herramientas y la consideración de éstas como verdaderas aplicaciones de RV sólo se materializó cuando los sistemas de generación gráfica mejoraron en los últimos años del siglo pasado, el uso de plataformas robóticas en simulación es generalizado desde hace más de cinco décadas (Royal-Aeronautical-Society, 1979).

Los primeros MCA diseñados para gobernar plataformas de movimiento en simuladores (en este caso aéreos) fueron propuestos por Schmidt and Conrad (1969) en la *National Aeronautics and Space Administration* (NASA). Estos trabajos fueron revisados y mejorados por Reid and Nahon (1985) y Nahon and Reid (1990) en el *University of Toronto Institute for Aerospace Studies* (UTIAS). Este algoritmo se conoce como el *algoritmo clásico*, y se basa en tres principios básicos para generar movimiento y al mismo tiempo evitar alcanzar los topes del dispositivo: (i) reducir la magnitud de los movimientos; (ii) filtrar las aceleraciones de baja frecuencia (que son las que producen movimientos sostenidos de larga duración) mediante filtros pasa-alta; (iii) intentar simular las aceleraciones sostenidas perdidas, mediante ligeras inclinaciones empleando la técnica de *tilt-coordination* (Groen & Bles, 2004; Raymond & Kemeny, 2000). Más adelante, Parrish, Dieudonne, and Martin Jr (1975) propusieron una modificación del algoritmo clásico conocida como *algoritmo adaptativo*, que intenta ajustar dinámicamente el algoritmo para aprovechar mejor el espacio de trabajo del dispositivo. Poco después, la teoría de control óptimo fue aplicada a este campo para proponer el conocido como *algoritmo óptimo* (Kurosaki, 1978; Sivan, Ish-Shalom, & Huang, 1982). Aunque se han propuesto otros algoritmos más recientemente con cierta aprobación, como en Dagdelen, Raymond, Kemeny, Bordier, and Maizi (2009), el algoritmo clásico es todavía el más utilizado por su versatilidad y efectividad,

y porque todavía no se ha llegado a un consenso sobre cuál es la mejor manera de realizar esta tarea. Esto es debido a la falta de una manera estándar de evaluar y ajustar este tipo de algoritmos.

El diseño de manipulador robótico más utilizado en simulación de vehículos es el conocido *hexápodo* Stewart-Gough de seis grados de libertad (GdL) (Stewart, 1965), aunque últimamente se han desarrollado algunos estudios destinados a aplicar plataformas de movimiento de menos GdL, por la reducción de costes que ello conlleva (Lozoya-Santos, Tudon-Martinez, & Salinas, 2017; Mauro, Gastaldi, Pastorelli, & Sorli, 2016; Zhang & Zhang, 2013). El aprovechamiento de todos los GdL en un hexápodo es a veces muy escaso por los reducidos desplazamientos máximos simultáneos posibles, especialmente lineales, en este tipo de diseños. Es, por tanto, necesario considerar cuál es el manipulador más apropiado para cada aplicación abriendo la posibilidad a emplear dispositivos de menos GdL pero más potentes (Casas, Coma, Riera, & Fernández, 2015).

Existen diversas aproximaciones en la literatura para simular manipuladores robóticos, aunque muy pocas con el objetivo de emplear estos dispositivos en aplicaciones de RV. Se pueden encontrar algunos trabajos donde se simulan plataformas de movimiento con otros objetivos, como en Selvakumar, Pandian, Sivaramakrishnan, and Kalaichelvan (2010), en el que se simulan tres manipuladores paralelos de 3 GdL mediante el *software* ADAMS (MSC, 2017) con el objetivo de analizar singularidades y comparar diversas opciones de construcción. Otros autores simulan plataformas de más GdL (Hajimirzaalian, Moosavi, & Massah, 2010). En Y.-W. Li, Wang, Wang, and Liu (2003) se realiza una simulación cinemática y dinámica de un manipulador robótico de 3 GdL, empleando ADAMS y una aproximación de tipo Newton-Euler. Estos trabajos tienen como objetivo analizar el comportamiento de las diversas partes del manipulador y comparar los resultados con las soluciones analíticas, por lo que no buscan realizar un simulador que sustituya al *hardware* en pruebas reales.

Existen otras aproximaciones en las que se simula el comportamiento de una plataforma empleando modelos CAD (Hulme & Pancotti, 2004), permitiendo una intervención “*in-the-loop*”, de manera que el simulador no sólo simula el comportamiento del manipulador, sino que puede sustituir al real en cualquier prueba, que es lo que realmente se busca en RV. Este último trabajo emplea una simulación cinemática, por lo que no es suficiente para validar algoritmos de tipo MCA y simula exclusivamente una plataforma *Moog 2000E*.

Una solución que sí permite todo ello, y además permite simular diversos modelos y diseños es la contenida en Casas, Alcaraz, Olanda, Coma, and Fernández (2014), que proporciona un simulador con similar propósito al de este trabajo, cuyo diseño, construcción y validación se describe en detalle. Este simulador emula el comportamiento del manipulador robótico permitiendo una intervención “*in-the-loop*”. En dicho trabajo se emplea la dinámica newtoniana-lagrangiana y modelos CAD. Aunque es un simulador bastante preciso, no es todo lo rápido que sería deseable (especialmente si se quiere emplear para realizar miles de pruebas con algoritmos MCA). Además, este simulador no siempre es sencillo de construir, ya que se ha de partir de un diseño CAD de la estructura del manipulador y aplicar diversas técnicas sobre los objetos del mismo para convertirlo en un modelo físico, comprobando previamente la estabilidad de la simulación.

Una alternativa es tratar la simulación como un problema de identificación de sistemas, de modo que se busque un modelo que sea capaz de responder como lo hace el sistema real dadas las mismas entradas, sin tener en cuenta cómo se producen esas

Download English Version:

<https://daneshyari.com/en/article/8050504>

Download Persian Version:

<https://daneshyari.com/article/8050504>

[Daneshyari.com](https://daneshyari.com)