

LABORATORIO VIRTUAL MÓVIL DE ROBOTS PARALELOS

Adrián Peidró, Luis Payá, Vicente Román, José María Marín, Arturo Gil, Óscar Reinoso
Departamento de Ingeniería de Sistemas y Automática
Universidad Miguel Hernández, Avda. de la Universidad s/n, 03202 Elche (Alicante), España
{apeidro, lpaya, v.roman, jmarin, arturo.gil, o.reinoso}@umh.es

Resumen

Este artículo presenta m-PaRoLa, un laboratorio virtual educativo formado por simuladores para analizar robots paralelos. Al estar basados en Javascript, dichos simuladores pueden ejecutarse indistintamente tanto en navegadores de ordenadores de sobremesa como en navegadores de dispositivos móviles (como smartphones y tabletas). De esta manera, m-PaRoLa puede ser usado en metodologías de aprendizaje móvil (o m-learning) para reforzar la enseñanza y el aprendizaje de Robótica o Teoría de Máquinas y Mecanismos. Los simuladores presentados son muy visuales e intuitivos, y permiten al usuario analizar diversos problemas cinemáticos básicos de los robots paralelos.

Palabras clave: m-Learning, robots paralelos, laboratorios virtuales, Javascript, Java

1 INTRODUCCIÓN

El aprendizaje móvil (m-learning) puede definirse como una metodología de enseñanza y aprendizaje que se apoya en dispositivos móviles con conexión inalámbrica a Internet, típicamente smartphones y tabletas, para proporcionar o dar soporte a parte de la formación. Algunos autores asocian diversas ventajas al uso de tales dispositivos móviles en la formación [1,2,4]:

- Posibilidad de aprender en cualquier lugar y momento.
- Aumento de las comunicaciones, de la colaboración, y de la interacción entre estudiantes y con los docentes.
- La mayoría de dispositivos móviles suelen ser más económicos que los portátiles u ordenadores de sobremesa.
- La mayoría de estudiantes están muy acostumbrados al uso de tales dispositivos a todas horas, lo cual puede traducirse en una mayor motivación a integrarlos en su formación.

Sin embargo, el uso de móviles en la enseñanza también tiene diversos problemas [4], como la

usabilidad limitada: las pantallas de dichos dispositivos son demasiado pequeñas, y sus métodos de entrada - típicamente, el teclado en pantalla - son demasiado limitados o incómodos para según qué tareas. También puede resultar problemático el uso inapropiado y ofensivo de los móviles en el aula (especialmente, entre los estudiantes más jóvenes).

Por otra parte, un laboratorio virtual educativo es típicamente un entorno web que permite al usuario realizar diversos experimentos sobre un sistema simulado, con el objetivo de facilitar su comprensión sobre determinados aspectos de dicho sistema. Algunas ventajas de los laboratorios virtuales sobre los laboratorios físicos convencionales son: reducción de costes, flexibilidad, y mayor seguridad [16].

Tradicionalmente, los laboratorios virtuales se utilizaban desde ordenadores de sobremesa, ya fuera a través de navegadores web o de aplicaciones de escritorio. Sin embargo, durante los últimos años, diversos investigadores han comenzado a desarrollar laboratorios virtuales móviles (LVM) con el fin de explorar las ventajas del m-learning en la enseñanza-aprendizaje de diversas disciplinas. Por ejemplo, Glavinic et al. [7] presentaron un LVM para la enseñanza de Diseño de Sistemas Digitales, y analizaron diversos problemas de usabilidad de las interfaces gráficas de dichos laboratorios, que han de ejecutarse en pantallas demasiado pequeñas [8]. Bottentuit Jr. y Coutinho [5] presentaron un LVM para la enseñanza de Química Orgánica. Boshale y Livingston [3] desarrollaron otro LVM para el aprendizaje de Seguridad en Redes. Jardim et al. [10] propusieron una arquitectura para el aprendizaje ubicuo, que permite al usuario estudiar en cualquier lugar y momento, y desde cualquier dispositivo, y aplicaron dicha arquitectura para enseñar Redes de Computadores. Oluwole et al. [12] desarrollaron un LVM para el aprendizaje de conceptos de Física, Química, e Informática. Finalmente, Merabet et al. [11] presentaron un LVM para la enseñanza de Biometría, Electrónica, y Lenguajes de Programación.

Debido a su fuerte carácter multidisciplinar, la robótica es una de las áreas en las que más laboratorios educativos virtuales y remotos se han desarrollado [13,16,18]. Sin embargo, los laboratorios virtuales *móviles* sobre robótica son menos habituales.

Considerando las tendencias y ventajas del m-learning, y considerando que la robótica es una de las disciplinas más importantes en las que será necesario formar a los actuales y futuros ingenieros, este artículo presenta los primeros pasos en el desarrollo de m-PaRoLa (mobile Parallel Robotics Laboratory, <http://arvc.umh.es/mparola>), un LVM educativo para el aprendizaje de la cinemática de robots y mecanismos paralelos. Actualmente, m-PaRoLa consta de dos simuladores Javascript que permiten al usuario simular y estudiar diversas nociones cinemáticas de dos robots paralelos: el mecanismo de cinco barras y el robot 3RRR.

Este artículo está organizado de la siguiente manera. La sección 2 presenta los robots paralelos y algunas herramientas de simulación para estudiarlos. Las secciones 3 y 4 presentan los mecanismos de cinco barras y 3RRR, respectivamente, y describen los simuladores web desarrollados para estudiar la cinemática de dichos robots. Finalmente, la sección 5 presenta las conclusiones y trabajos futuros.

2 ROBOTS PARALELOS

Los robots paralelos (Figura 1b) son manipuladores robóticos en los que el efector final está conectado a la base fija a través de dos o más cadenas cinemáticas dispuestas en paralelo. Esta es una arquitectura alternativa a los manipuladores serie utilizados más habitualmente en la industria, en los que el efector final está controlado por una única cadena cinemática abierta (Figura 1a). Los robots paralelos ofrecen diversas ventajas sobre los de tipo serie, como una mayor rigidez estructural, una mayor capacidad de carga por unidad de masa, y altas prestaciones dinámicas (pueden alcanzar aceleraciones muy elevadas). No obstante, también presentan algunos inconvenientes: los robots paralelos pueden alcanzar configuraciones singulares en las que no es posible controlar completamente el movimiento de su efector final, su espacio de trabajo es más pequeño, y su problema cinemático directo es más complejo.

PaRoLa (Parallel Robotics Laboratory, <http://arvc.umh.es/parola>, [14,15]) es un laboratorio virtual educativo que permite a los estudiantes experimentar con las simulaciones de diversos robots y mecanismos paralelos, con el objetivo de facilitar la comprensión de su cinemática, espacio de trabajo, singularidades, y dinámica. PaRoLa consta de una colección de applets Java creados mediante la herramienta de desarrollo Easy Java Simulations [6]. Como la mayoría de laboratorios virtuales desarrollados hace unos años, el objetivo original de PaRoLa era permitir a los usuarios ejecutar directamente desde el navegador los applets Java

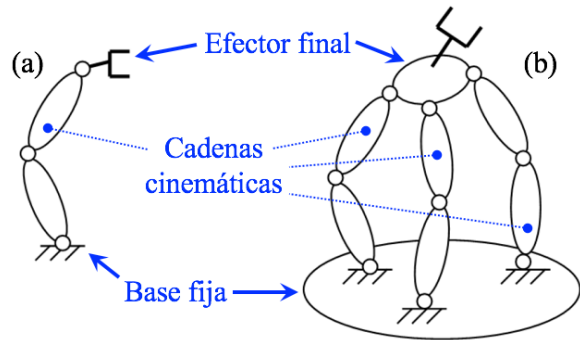


Figura 1: Robots de tipo serie (a) y paralelo (b).

embebidos en una página web, sin tener que descargar y ejecutar los ficheros comprimidos .jar. Sin embargo, la mayoría de navegadores más populares (Chrome y Firefox, entre otros) han ido abandonando su soporte a applets Java durante los últimos años, de forma que, actualmente, es imposible ejecutar directamente en un navegador dichos applets Java embebidos en páginas web. Por otro lado, incluso aunque descargásemos los simuladores en formato .jar para ejecutarlos en un ordenador de escritorio, éstos no funcionarían en dispositivos móviles, lo cual limita su rango de aplicación.

Desde la versión 5.0, Easy Java Simulations (<http://www.um.es/fem/EjsWiki>) permite desarrollar simuladores tanto en lenguaje Java como en Javascript. La principal ventaja de programar simuladores en Javascript es que dichos simuladores pueden estar embebidos en páginas web y ejecutarse directamente desde un navegador, sin tener que descargarlo. Además, los simuladores Javascript pueden ejecutarse indistintamente desde navegadores de ordenadores de sobremesa o de dispositivos móviles, favoreciendo su portabilidad y su uso en estrategias de m-learning.

Considerando todo lo anterior, este artículo presenta m-PaRoLa: un LVM basado en Javascript, desarrollado mediante Easy Java/Javascript Simulations. m-PaRoLa permite al usuario experimentar con robots paralelos en simulación, con el objetivo de estudiar y aprender sobre diversos aspectos cinemáticos de dichos robots, desde cualquier dispositivo (incluyendo dispositivos móviles). En las siguientes secciones presentaremos los simuladores disponibles actualmente en m-PaRoLa, que permiten simular el mecanismo de cinco barras y el robot paralelo 3RRR. Dichos simuladores están disponibles en <http://arvc.umh.es/mparola> y funcionan en diferentes dispositivos, incluyendo smartphones y tabletas con diversos sistemas operativos (iOS, Android) y diversos navegadores (Chrome, Firefox). En las siguientes secciones, también describiremos cómo hemos resuelto diversos problemas que surgieron al intentar implementar

varias funcionalidades en los simuladores, debido a las reducidas dimensiones de las pantallas de los dispositivos móviles y a sus limitados métodos de entrada (factores que dificultan la implementación de demasiadas funcionalidades).

3 SIMULADOR DEL MECANISMO DE CINCO BARRAS

Esta sección describe el mecanismo de cinco barras, los problemas cinemáticos básicos asociados al mismo, y el simulador Javascript desarrollado para estudiar dichos problemas.

3.1 EL MECANISMO DE CINCO BARRAS

El mecanismo de cinco barras, mostrado en la Figura 2, es una cadena cinemática cerrada formada por cinco barras (A_1B_1 , B_1P , B_2P , A_2B_2 , y A_1A_2) interconectadas a través de cinco articulaciones de rotación (A_1 , B_1 , P , B_2 , A_2). La barra A_1A_2 es la base fija, mientras que la articulación P es el efector final, el cual está conectado a la base fija a través de las cadenas cinemáticas A_1B_1P y A_2B_2P . Este mecanismo tiene dos grados de libertad: controlando los ángulos θ_1 y θ_2 (por ejemplo, mediante motores eléctricos conectados a las articulaciones A_1 y A_2), es posible controlar la posición (x, y) del punto P en el plano XY .

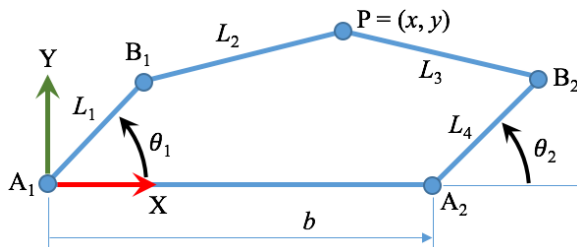


Figura 2: Mecanismo paralelo de cinco barras.

La cinemática directa de este mecanismo consiste en calcular la posición (x, y) del efector final P para valores conocidos de θ_1 y θ_2 . Este problema se reduce a calcular la intersección de dos circunferencias: una centrada en B_1 y con radio L_2 , y otra centrada en B_2 y con radio L_3 . Por tanto, la cinemática directa del mecanismo de cinco barras tiene dos soluciones posibles: para valores dados de (θ_1, θ_2) , el punto P puede encontrarse en la posición mostrada en la Figura 2 o en la posición simétrica de ésta con respecto a la línea B_1B_2 .

La cinemática inversa de este mecanismo consiste en determinar los valores que deben adoptar los ángulos (θ_1, θ_2) para posicionar el efector final P en una posición (x, y) deseada. Para una posición (x, y) dada, existen dos posibles soluciones para el ángulo θ_1 : una de dichas soluciones sería la mostrada en la Figura 2,

mientras que la otra solución sería la que posiciona al punto B_1 en la posición simétrica a la que ocupa en la Figura 2 (simétrica con respecto a la línea A_1P). Esas dos posibles posiciones para B_1 (asociadas a las 2 soluciones de θ_1) son los dos puntos de intersección de dos círculos: uno centrado en A_1 y con radio L_1 , y otro centrado en P y con radio L_2 . Análogamente, habría dos posibles valores del ángulo θ_2 que posicionan el efector final en una posición deseada (x, y) : una solución sería la mostrada en la Figura 2, mientras que la otra solución sería la que posiciona al punto B_2 en la posición simétrica a la que ocupa en la Figura 2 (simétrica con respecto a la línea A_2P). En conclusión: dada una posición deseada (x, y) para el punto P , hay dos posibles soluciones para el ángulo θ_1 , que son independientes de las dos posibles soluciones para el ángulo θ_2 . Por tanto, la cinemática inversa de este mecanismo tiene cuatro soluciones diferentes, que son las cuatro posibles combinaciones de las dos soluciones de θ_1 con las dos soluciones de θ_2 .

El espacio de trabajo de este mecanismo es el conjunto de puntos alcanzables por el punto P . Cuando el movimiento de P se restringe a la cadena cinemática A_1B_1P , P puede alcanzar todos los puntos dentro de una región anular \mathcal{A}_1 encerrada entre dos circunferencias concéntricas centradas en A_1 : una de estas circunferencias tiene radio (L_1+L_2) (correspondiendo a la cadena cinemática A_1B_1P totalmente extendida), mientras que la otra tiene radio $|L_1-L_2|$ (correspondiendo a la cadena cinemática A_1B_1P totalmente plegada en sí misma). De manera análoga, cuando el movimiento de P se restringe a la cadena cinemática A_2B_2P , P puede moverse por la región anular \mathcal{A}_2 encerrada entre dos circunferencias concéntricas centradas en A_2 : una con radio (L_3+L_4) y otra con radio $|L_3-L_4|$. Por tanto, el espacio de trabajo del mecanismo de cinco barras es la intersección de las regiones anulares \mathcal{A}_1 y \mathcal{A}_2 . Los bordes de tal espacio de trabajo se muestran en línea de color azul en el panel p1 de la Figura 3.

Las singularidades de este mecanismo son las configuraciones en las que las barras B_1P y B_2P son paralelas, bien porque se encuentran en prolongación, o bien una plegada sobre la otra. Cuando el mecanismo se encuentra en una configuración singular, no es posible controlar el movimiento del efector final en la dirección perpendicular a dichas barras. Las configuraciones singulares forman curvas que dividen el espacio de trabajo en diferentes regiones libres de singularidades.

Para terminar con la descripción de este mecanismo, cabe decir que el diseño del mecanismo de cinco barras está definido por cinco parámetros geométricos: la posición b de A_2 a lo largo del eje X , y las cuatro longitudes $\{L_1, L_2, L_3, L_4\}$ de las barras $\{A_1B_1, B_1P, B_2P, A_2B_2\}$. Todos esos parámetros

geométricos influyen en la forma del espacio de trabajo y de las curvas singulares.

3.2 SIMULADOR JAVASCRIPT DEL MECANISMO DE CINCO BARRAS

El simulador del mecanismo de cinco barras está disponible en la URL <http://arvc.umh.es/mparola/five-bar/five-bar.html>. La Figura 3 muestra una captura de pantalla de este simulador ejecutándose en Chrome, en un smartphone con Android. Como muestra esta figura, el simulador desarrollado consta de cuatro paneles:

- El panel p1 muestra una representación esquemática del mecanismo de cinco barras.
- El panel p2 muestra el plano (θ_1, θ_2) , dentro de los rangos $-\pi \leq \theta_i \leq \pi$.
- El panel p3 es simplemente un selector que permite al usuario elegir entre simular la cinemática inversa o la directa.
- En panel p4 indica en todo momento el valor actual de todos los parámetros cinemáticos relevantes: los cinco parámetros geométricos (b, L_i) , los ángulos (θ_1, θ_2) , y la posición (x, y) .

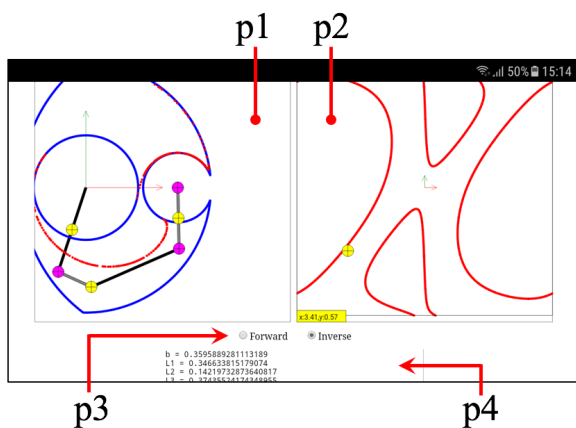


Figura 3: Simulador Javascript del mecanismo de cinco barras.

En laboratorios virtuales no móviles, diseñados para ser utilizados mediante ordenadores de sobremesa (como PaRoLa), el gran tamaño de las pantallas de dichos ordenadores permite que las interfaces gráficas de usuario de tales laboratorios contengan múltiples ventanas y menús que permitan configurar con facilidad las simulaciones y los experimentos. Sin embargo, los LVM están mucho más limitados en este aspecto, ya que deben ejecutarse en las pequeñas pantallas de los dispositivos móviles [8], lo que obliga a optimizar sus interfaces con el fin de proporcionar al usuario todas las funcionalidades requeridas de forma intuitiva y respetando las limitaciones de dichos dispositivos (pantallas pequeñas y métodos de entrada limitados), lo cual puede ser complicado. Esto impide

utilizar demasiados menús y ventanas en las interfaces de los LVM, alentando el uso de diseños minimalistas que favorezcan los gráficos sobre los textos. A continuación, describiremos las capacidades del simulador desarrollado, explicando cómo se ha optimizado su interfaz gráfica de usuario con el fin de implementar múltiples funcionalidades en el poco espacio disponible en la pantalla de un móvil.

3.2.1 Simulación de la cinemática directa

En primer lugar, el usuario ha de elegir simular la cinemática directa en el panel p3. Entonces, debe proporcionar los ángulos (θ_1, θ_2) , para que el simulador resuelva la cinemática directa y calcule la posición (x, y) del punto P, mostrando la solución gráficamente en el panel p1 y textualmente en el panel p4. El usuario puede proporcionar los ángulos (θ_1, θ_2) directamente arrastrando los pequeños círculos amarillos situados en el centro de las barras A_1B_1 y A_2B_2 , en el panel p1. De manera alternativa, el usuario también puede especificar dichos ángulos arrastrando el círculo amarillo en el plano (θ_1, θ_2) en el panel p2, lo cual permite variar ambos ángulos de forma simultánea. Como se ha explicado en la sección 3.1, la cinemática directa del mecanismo de cinco barras tiene dos soluciones. Para cambiar entre estas dos soluciones (y que se muestre una u otra en el panel p1), el usuario debe pulsar sobre el círculo amarillo situado en el efector final (punto P).

3.2.2 Simulación de la cinemática inversa

Asúmase ahora que el usuario elige simular la cinemática inversa en el panel p3. En tal caso, se debe proporcionar la posición deseada (x, y) para el efector final, para que el simulador resuelva los ángulos (θ_1, θ_2) , mostrando la solución gráficamente en los paneles p1 y p2, y textualmente en el panel p4. Para proporcionar la posición deseada (x, y) , el usuario ha de arrastrar el círculo amarillo situado sobre el efector final P (recuérdese que, cuando se simula el problema cinemático directo, la función de dicho círculo es precisamente cambiar entre las dos soluciones posibles de dicho problema).

Recuérdese de la sección 3.1 que la cinemática inversa de cada cadena cinemática A_iB_iP tiene dos soluciones posibles. Para cambiar entre las dos soluciones de cada cadena, se debe pulsar sobre el círculo amarillo situado en el centro de las barras A_1B_1 o A_2B_2 (mientras se simulaba la cinemática directa, arrastrar dichos círculos permitía variar los ángulos θ_1 y θ_2).

3.2.3 Visualizando el espacio de trabajo y las singularidades

El panel p1 representa en todo momento los bordes del espacio de trabajo, en color azul. Asimismo, también

se muestran las curvas de singularidades en color rojo, en el plano (θ_1, θ_2) (panel p2), así como en el panel p1 mientras se simule la cinemática inversa (las curvas de singularidades mostradas en el panel p1 variarán dependiendo de la solución de la cinemática inversa que se esté visualizando). Mientras se simula la cinemática inversa, el usuario puede comprobar cómo varía la postura del mecanismo a medida que su efector final se arrastra a lo largo del espacio de trabajo, comprobando cómo se extienden o pliegan las cadenas cinemáticas $A_i B_i P$ en las cercanías de los bordes del espacio de trabajo. También es útil visualizar cómo las barras $B_1 P$ y $B_2 P$ permanecen alineadas cuando se posiciona el efector final sobre las curvas rojas de singularidades, en el panel p1. Por otro lado, si el usuario arrastra el efector final fuera del espacio de trabajo, el mecanismo aparenta "romperse" debido a que la cinemática inversa no admite soluciones reales fuera del espacio de trabajo (se trata de posiciones inalcanzables). Algo similar ocurre cuando se simula la cinemática directa: si el usuario cruza las curvas de singularidades en el panel p2, el mecanismo parece "romperse" debido a que no existen soluciones reales a la cinemática directa más allá de dichas curvas (el mecanismo no puede ensamblarse para tales combinaciones de θ_1 y θ_2).

Las formas del espacio de trabajo y de las curvas de singularidades dependen del diseño del mecanismo, de manera que dichas formas cambiarán si el usuario modifica alguno de los cinco parámetros geométricos del mismo: $\{b, L_1, L_2, L_3, L_4\}$. Para modificar cualquiera de estos parámetros, el usuario ha de arrastrar y *soltar* los círculos de color magenta situados sobre las articulaciones A_2, B_1 , y B_2 . Al arrastrar A_2 , varían b y L_4 . Al arrastrar B_1 , varían L_1 y L_2 . Por último, al arrastrar B_2 varían tanto L_3 como L_4 . Por tanto, al arrastrar esas articulaciones, las longitudes mencionadas varían como si el mecanismo estuviera hecho de goma, y el espacio de trabajo y las curvas de singularidades se deforman en consecuencia. Esta funcionalidad es muy útil para estudiar cómo varían el espacio de trabajo y las singularidades al modificar el diseño del mecanismo. Por ejemplo, empleando el simulador puede comprobarse que si b es mucho menor que L_i , y si todos los L_i son muy similares, entonces se obtiene un espacio de trabajo aproximadamente circular bastante amplio, con grandes regiones libres de singularidades para algunas soluciones de la cinemática inversa, lo cual vuelve a tal diseño muy atractivo.

Este método gráfico para variar el diseño del mecanismo (es decir, arrastrando las articulaciones A_2, B_1 , y B_2) tiene un inconveniente, y es que siempre modifica dos parámetros geométricos simultáneamente (las longitudes de las barras unidas por la articulación arrastrada) en lugar de uno, lo cual podría dificultar la selección de un diseño concreto.

Sin embargo, la ventaja del método gráfico es que no requiere menús adicionales que complicarían la interfaz de usuario, que debe ser lo más simple y minimalista posible con el fin de facilitar su uso desde un dispositivo móvil.

4 SIMULADOR DEL ROBOT 3RRR

Esta sección describe el robot paralelo plano 3RRR, así como sus problemas cinemáticos asociados y el simulador Javascript desarrollado para estudiar dichos problemas.

4.1 EL ROBOT PARALELO PLANO 3RRR

El robot paralelo plano 3RRR, mostrado en la Figura 4, es un manipulador formado por una plataforma triangular equilátera ABC (que es el efector final) conectada a una base fija PQR a través de tres cadenas cinemáticas en paralelo: PDA, QEB, RFC. Este robot tiene tres grados de libertad: controlando los ángulos $(\theta_1, \theta_2, \theta_3)$ (por ejemplo, a través de motores eléctricos conectados a las articulaciones P, Q, y R), es posible controlar la orientación ϕ del efector final y la posición (x, y) de su centro G.

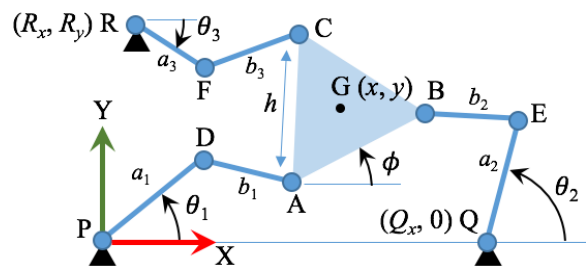


Figura 4: Robot paralelo plano 3RRR.

Los parámetros geométricos del robot 3RRR son los siguientes (véase la Figura 4): Q_x (posición de la articulación Q a lo largo del eje X), R_x y R_y (posición de la articulación R en el plano XY), h (longitud del lado del efector final equilátero), y $\{a_1, b_1, a_2, b_2, a_3, b_3\}$ (que son las longitudes de las barras $\{PD, DA, QE, EB, RF, FC\}$, respectivamente).

La cinemática directa de este robot consiste en calcular la posición (x, y) y orientación ϕ del efector final, para valores dados de $(\theta_1, \theta_2, \theta_3)$. Este problema tiene seis soluciones en el dominio complejo [9], es decir, cuando x, y , o ϕ son números complejos. Dependiendo de los valores concretos de los parámetros geométricos y de los ángulos $(\theta_1, \theta_2, \theta_3)$, dos, cuatro, o seis de dichas soluciones serán reales (es decir, tendrán parte imaginaria nula).

La cinemática inversa de este robot consiste en calcular los valores de los ángulos $(\theta_1, \theta_2, \theta_3)$ que son necesarios para posicionar al efector final en una

posición (x,y) deseada y con una orientación ϕ deseada. Si (x,y,ϕ) son conocidos, entonces también se conocen las posiciones de las articulaciones $\{A,B,C\}$ y, por tanto, la cinemática inversa de las cadenas $\{PDA, QEB, RFC\}$ puede resolverse de forma idéntica a la cinemática inversa de las cadenas A_1B_1P y A_2B_2P del mecanismo de cinco barras. Por tanto, existen dos soluciones posibles para cada ángulo θ_i ($i = 1, 2, 3$) en el robot 3RRR, y la cinemática inversa del robot completo tiene ocho soluciones diferentes (que se corresponden con las ocho posibles combinaciones de las dos soluciones de cada ángulo θ_i).

El espacio de trabajo del robot 3RRR es el conjunto de posiciones y orientaciones alcanzables por su efector final. Por tanto, es un conjunto tridimensional en el espacio (x,y,ϕ) . Sin embargo, es más fácil visualizar el espacio de trabajo a orientación constante, que es el conjunto de puntos (x,y) alcanzables por el centro G del efector final cuando su orientación ϕ se mantiene fija a un valor dado. Es sencillo ver que el espacio de trabajo a orientación constante del robot 3RRR es la intersección de tres regiones anulares \mathcal{A}_1 , \mathcal{A}_2 , y \mathcal{A}_3 . La región anular \mathcal{A}_1 está encerrada entre dos círculos concéntricos centrados en el punto obtenido al trasladar P mediante el vector \overrightarrow{AG} : uno de dichos círculos tiene radio (a_1+b_1) (correspondiendo a la cadena PDA completamente extendida), mientras que el otro tiene radio $|a_1-b_1|$ (correspondiendo a la cadena PDA completamente plegada). De forma análoga, \mathcal{A}_2 está encerrada entre dos círculos centrados en el punto definido por el vector $\overrightarrow{PQ}+\overrightarrow{BG}$, con radios (a_2+b_2) y $|a_2-b_2|$. Finalmente, \mathcal{A}_3 está encerrado por dos círculos centrados en $\overrightarrow{PR}+\overrightarrow{CG}$, con radios (a_3+b_3) y $|a_3-b_3|$. Al variar cualquier parámetro geométrico del robot, o la orientación ϕ de su efector final, la forma del espacio de trabajo a orientación constante variará. Los bordes de tal espacio de trabajo se representan en color azul en el panel p1 de la Figura 5.

4.2 SIMULADOR JAVASCRIPT DEL ROBOT 3RRR

El simulador del robot 3RRR está disponible en la URL <http://arvc.umh.es/mparola/3rrr/3rrr.html>. La Figura 5 muestra una captura de pantalla de dicho simulador ejecutándose en Chrome en un smartphone con Android. Dicho simulador consta también de cuatro paneles:

- El panel p1 es una representación esquemática del robot 3RRR.
- El panel p2, que sólo se muestra mientras se simula la cinemática directa, muestra el plano complejo del ángulo ϕ , es decir: el eje horizontal representa $\text{Re}(\phi)$, mientras que el eje vertical representa $\text{Im}(\phi)$. Posteriormente

se proporcionarán más detalles sobre la utilidad de este panel.

- El panel p3 permite seleccionar al usuario entre simular la cinemática directa o la inversa.
- El panel p4 muestra los valores actuales de todos los parámetros de relevancia: los parámetros geométricos del robot, los ángulos θ_i , y la pose (x,y,ϕ) del efector final.

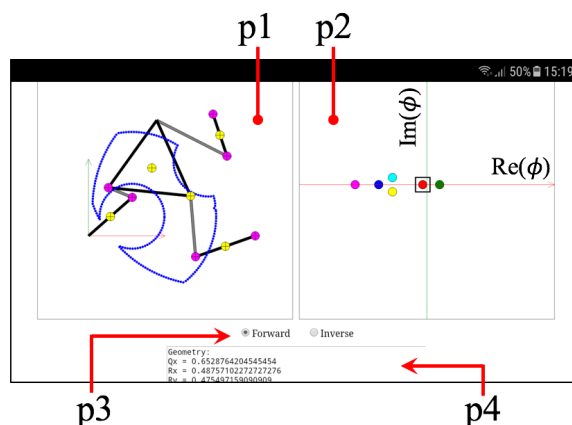


Figura 5: Simulador Javascript del robot 3RRR.

Al igual que en el simulador del mecanismo de cinco barras, estos paneles contienen diversos elementos de control e interacción, optimizados para ofrecer al usuario múltiples funcionalidades en el poco espacio disponible en las pequeñas pantallas de los dispositivos móviles. Dichas funcionalidades se explican a continuación.

4.2.1 Simulación de la cinemática directa

Si se elige simular la cinemática directa, es preciso proporcionar los valores de los ángulos $(\theta_1, \theta_2, \theta_3)$. Dichos ángulos se han de proporcionar arrastrando los círculos amarillos situados en el centro de las barras PD, QE, o RF, como en el mecanismo de cinco barras. Al variar dichos ángulos, el simulador resuelve la cinemática directa del robot 3RRR, obteniendo la pose (x,y,ϕ) del efector final y representando la postura correspondiente en el panel p1.

Según la sección 4.1, la cinemática directa del robot 3RRR admite seis soluciones en el dominio complejo. Estas seis soluciones se representan en el panel p2, que representa el plano complejo $(\text{Re}(\phi), \text{Im}(\phi))$. Cada solución se representa en este plano mediante un pequeño círculo en un color distinto. A medida que varían los ángulos θ_i , las soluciones describen trayectorias en el plano complejo, y es posible que alguna solución real pase a ser compleja o viceversa. Para visualizar la postura del robot en cada una de las seis soluciones, se debe pulsar sobre la solución deseada en el panel p2 (la solución seleccionada en

todo momento aparece encerrada por un pequeño cuadrado negro en dicho panel). En caso de seleccionar una solución con parte imaginaria no nula, la postura del robot asociada mostrada en el panel p1 será incorrecta, y el robot se mostrará como si estuviera "roto" (lo cual significa que las soluciones complejas no son físicamente posibles).

4.2.2 Simulación de la cinemática inversa

Para simular la cinemática inversa, es preciso proporcionar la posición (x,y) y la orientación ϕ del efector final. Para proporcionar la posición, es necesario arrastrar el círculo amarillo central del efector final, lo cual produce la traslación del mismo sin alterar su orientación. Para variar la orientación ϕ , es necesario arrastrar el círculo amarillo situado sobre la articulación B del efector final: esto provoca la rotación del efector final alrededor de su centro G. Para cambiar la solución de la cinemática inversa de cada cadena cinemática mostrada al usuario, es preciso pulsar los círculos amarillos situados en los centros de las barras {PD, QE, RF} (tales círculos se utilizaban para variar los ángulos θ_i al simular la cinemática directa).

4.2.3 Visualizando el espacio de trabajo a orientación constante

En todo momento, el panel p1 representa en color azul los bordes del espacio de trabajo a orientación constante, para el valor actual de la orientación ϕ . Si ϕ varía por cualquier motivo (al simular la cinemática) o si alguno de los parámetros geométricos del robot varía, la forma del espacio de trabajo a orientación constante varía. Mientras el usuario simula la cinemática inversa, es posible mover el efector final hacia los bordes de dicho espacio de trabajo para visualizar cómo las diferentes cadenas cinemáticas se estiran o pliegan. Como en el mecanismo de cinco barras, al arrastrar el efector final fuera del espacio de trabajo, el robot parece "romperse" porque la posición seleccionada es inalcanzable.

Como se ha indicado más arriba, la forma del espacio de trabajo a orientación constante variará si cualquiera de los parámetros geométricos del robot cambia. Los parámetros geométricos del robot pueden modificarse de forma similar a como se hace en el mecanismo de cinco barras: arrastrando los círculos de color magenta situados sobre diversas articulaciones del robot. Esto provoca que las barras del robot conectadas a la articulación arrastrada se estiren o compriman como si estuvieran hechas de goma. Por ejemplo, al arrastrar Q, los parámetros $\{Q_x, a_2\}$ varían simultáneamente. Al arrastrar R, varían $\{R_x, R_y, a_3\}$. Al arrastrar D, E, o F varían $\{a_1, b_1\}$, $\{a_2, b_2\}$, o $\{a_3, b_3\}$, respectivamente. Por último, al arrastrar A varía tanto el tamaño h del efector final como las longitudes $\{b_1, b_2, b_3\}$.

5 CONCLUSIONES

En este artículo se ha presentado m-PaRoLa, un LVM para estudiar la cinemática de robots paralelos. Al contrario que un laboratorio virtual de sobremesa, uno móvil debe basarse en una interfaz intuitiva y visual que ofrezca al usuario todas las funcionalidades de forma eficaz, teniendo en cuenta las restricciones típicas de los dispositivos móviles en cuanto a usabilidad, como pantallas demasiado pequeñas o métodos de entrada limitados. Esto implica que las interfaces gráficas de usuario de los LVM han de optimizarse en gran medida, utilizando elementos de interacción que ofrezcan unas funcionalidades u otras dependiendo del tipo de análisis o simulación que el usuario desee realizar.

En el futuro, continuaremos añadiendo más funcionalidades y robots a m-PaRoLa, incluyendo robots remotos, para que los estudiantes puedan experimentar con robots paralelos reales a través de Internet.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Educación (ayuda FPU13/00413) y el Ministerio de Economía (proyecto DPI 2016-78361-R).

English summary

VIRTUAL MOBILE LABORATORY OF PARALLEL ROBOTS

Abstract

This paper presents m-PaRoLa, an educational virtual laboratory consisting of simulators for analyzing parallel robots. Since these simulators are based on Javascript, they can be run indistinctly on the web browsers of both desktop computers and mobile devices (such as smartphones and tablets). In this way, m-PaRoLa can be used in mobile learning (m-learning) methodologies, in order to support the teaching and learning of Robotics or Mechanism and Machine Theory. The presented simulators are very visual and intuitive, and allow the user to analyze some basic kinematic problems of parallel robots.

Keywords: m-Learning, parallel robots, virtual laboratories, Javascript, Java

Referencias

- [1] Alioon, Y. and Delialioglu, O. (2017). The effect of authentic m-learning activities on student engagement and motivation. *British Journal of Educational Technology*.
- [2] Asabere, N.Y. (2013). Benefits and challenges of mobile learning implementation: Story of developing nations. *Intl Journal of Computer Applications*, 73(1), 23-27.
- [3] Bhosale, Y.S. and Livingston, J. (2014). V-lab: a mobile virtual lab for network security studies. *Intl Journal of Computer Applications*, 93, 35-38.
- [4] Bhullar, M.S. (2014). A new method of learning: M-learning (mobile learning). In 2014 9th International Conference on Computer Science & Education, 322-325.
- [5] Bottentuit Jr., J.B. and Coutinho, C. (2007). Virtual laboratories and m-learning: learning with mobile devices. In Proceedings of International Multi-Conference on Society, Cybernetics and Informatics, 275-278.
- [6] Esquembre, F. (2004). Easy java simulations: a software tool to create scientific simulations in java. *Computer Physics Communications*, 156(2), 199-204.
- [7] Glavinic, V., Kukec, M., and Ljubic, S. (2007). Mobile virtual laboratory: Learning digital design. In 2007 29th International Conference on Information Technology Interfaces, 325-332.
- [8] Glavinic, V., Kukec, M., and Ljubic, S. (2009). Digital design mobile virtual laboratory implementation: A pragmatic approach. In C. Stephanidis (ed.), *Universal Access in Human-Computer Interaction. Addressing Diversity*, 489-498. Springer, Berlin, Heidelberg.
- [9] Gosselin, C.M. and Sefrioui, J. (1991). Polynomial solutions for the direct kinematic problem of planar three-degree-of-freedom parallel manipulators. In *Fifth International Conference on Advanced Robotics (ICAR'91)*, volume 2, 1124-1129.
- [10] Jardim, R.R., Lemos, E., Herpich, F., Bianchim, R., Medina, R., and Nunes, F.B. (2014). U-lab cloud: A ubiquitous virtual laboratory based on cloud computing. In *The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, 259-262.
- [11] Merabet, A., Akhrouf, S., Boubetra, D., Belhadj, F., Selmani, L., and Boubetra, A. (2015). Bba virtual laboratory through m-learning. In 2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), 181-184.
- [12] Oluwole, O., Nicolae, G., Olawale, O., and Oludele, A. (2015). Mobile virtual laboratory in Nigeria. *Intl J. of Engineering and Computer Science*, 4, 11417-11421.
- [13] Othayoth, R.S., Chittawadigi, R.G., Joshi, R.P., and Saha, S.K. (2017). Robot kinematics made easy using roboanalyzer software. *Comput Appl Eng Educ*, 25(5), 669-680.
- [14] Peidro, A., Gil, A., Marin, J.M., and Reinoso, O. (2016). A web-based tool to analyze the kinematics and singularities of parallel robots. *J. Intell. Robotic Syst.*, 81(1), 145-163.
- [15] Peidro, A., Reinoso, O., Gil, A., Marin, J.M., and Paya, L. (2015). A virtual laboratory to simulate the control of parallel robots. *IFAC-PapersOnLine*, 48(29), 19-24.
- [16] Potkonjak, V., Gardner, M., Callaghan, V., Mattila, P., Guetl, C., Petrovi, V.M., and Jovanovi, K. (2016). Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education*, 95, 309-327.
- [17] Saenz, J., Esquembre, F., Garcia, F., de la Torre, L., and Dormido, S. (2016). A new model for a remote connection with hardware devices using javascript. *IFAC-PapersOnLine*, 49, 133-137.
- [18] Torres, F., Candelas, F., Puente, S., Pomares, J., Gil, P., and Ortiz, F. (2006). Experiences with virtual environment and remote laboratory for teaching and learning robotics at the University of Alicante. *Int. J. Engng Ed.*, 22, 766-776.



© 2018 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC-BY-NC 3.0 license (<https://creativecommons.org/licenses/by-nc/3.0>).