



Universidad
Zaragoza

Trabajo Fin de Grado

Entrenador virtual para el juego del billar a tres
bandas

Virtual trainer for three-cushion billiard

Autor/es

Ignacio Díez Ruiz

Director/es

José Ángel Castellanos Gómez

Escuela de Ingeniería y Arquitectura
2018

Entrenador virtual para el juego del billar a tres bandas

RESUMEN

Conocer con precisión cuál va a ser el comportamiento de las bolas de billar antes de ejecutar el tiro es vital para realizar una carambola en el billar a tres bandas. Por ello, a lo largo de este estudio, se dará solución a este problema, haciendo uso de las competencias adquiridas por el autor a lo largo de su formación en el Grado que culmina con este proyecto.

El primer paso a dar es conocer el entorno en el que estará englobado el proyecto, así como valorar estudios similares que se hayan realizado con anterioridad, para poder considerar cómo se han abordado problemáticas similares. Partiendo de este punto, se deduce que la carambola más repetida en el billar a tres bandas es en la que la bola jugadora recorre una trayectoria por la que choca con una banda larga, para continuar chocando con una banda corta y terminar en la banda larga opuesta a la primera, completando así el requisito de las tres bandas.

Con esta premisa se decide que esta secuencia de banda larga/corta/larga deberá de ser modelada por el entrenador virtual. De esta manera, se procederá posteriormente al cálculo de la acción óptima de entrada a dicho modelo, para conseguir la respuesta deseada, que será siempre la ejecución final de la carambola.

El billarista en cada tiro que ejecuta posee seis grados de libertad sobre los que puede actuar, los llamados parámetros del golpeo, y cualquier modificación sobre los mismos se ve reflejado en el comportamiento final de la bola jugadora tras el tiro. Por ello es vital definir y controlar estos parámetros para que el modelo que se realice sea eficaz.

Con todas las variables controladas se procede a elaborar el modelo que secuencia la trayectoria de la bola jugadora a su paso por las bandas y, para ello, se utilizan datos de varios tiros ejecutados por jugadores expertos, de donde se extrae toda la información sobre el comportamiento de la bola jugadora para que el entrenador virtual posea la información suficiente como para elaborar el modelo mencionado.

Una vez que el programa elabora el modelo, se le programa para que calcule la entrada necesaria para conseguir la salida deseada, en función de la posición en la que se encuentren las bolas sobre la mesa de billar en una situación de juego cualquiera.

Siendo ya el programa capaz de calcular la acción óptima para conseguir la carambola, se elabora una interfaz gráfica a través de la cual se permita una comunicación clara y eficaz entre el billarista y el entrenador virtual, dándose de esta manera por concluido el proyecto.



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. IGNACIO DÍEZ RUIZ

con nº de DNI 72896547 W en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
GRADO, (Título del Trabajo)

Entrenador virtual para el juego del billar a tres bandas

Virtual trainer for three-cushion billiard

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 24 de agosto de 2018

Fdo: IGNACIO DÍEZ RUIZ

ÍNDICE DE CONTENIDOS

| | |
|---|----|
| 1. Introducción | 11 |
| 1.1 El billar a tres bandas | 11 |
| 1.2 Objetivo y alcance del proyecto..... | 12 |
| 1.3 Análisis de estudios previos | 12 |
| 1.4 Metodología | 13 |
| 1.5 Club de Billar Círculo Amistad Numancia..... | 15 |
| 2. Elaboración de una base de datos | 17 |
| 3. Elaboración del modelo..... | 21 |
| 4. Cálculo de la acción óptima..... | 23 |
| 4.1 Definición de punto inicial y punto final | 23 |
| 4.2 Sistema de referencia para los puntos inicial y final | 24 |
| 4.3 Estrategia de cálculo de la acción óptima..... | 25 |
| 4.4 Cálculo de la acción óptima para todas las secuencias de banda larga/corta/larga | 25 |
| 5. Desarrollo de una interfaz gráfica | 29 |
| 5.1 Información que tiene que solicitar la interfaz gráfica..... | 29 |
| 5.2 Información que tiene que devolver la interfaz gráfica | 29 |
| 5.3 Desarrollo de la interfaz gráfica final | 30 |
| 6. Ejemplos de aplicación del programa final | 35 |
| 7. Conclusiones..... | 39 |
| 7.1 Validez en la puesta en marcha del programa final | 39 |
| 7.2 Metodología utilizada | 40 |
| 7.3 Conclusión final del autor | 40 |
| 8. Bibliografía..... | 43 |
| ANEXO I Parámetros del golpeo en billar | 45 |
| A1.1 La dirección | 45 |
| A1.2 La velocidad..... | 46 |
| A1.3 La altura..... | 47 |

| | |
|---|----|
| A1.4 El efecto | 49 |
| A1.5 La penetración | 50 |
| A1.6 La inclinación..... | 50 |
| ANEXO II Numeración de los puntos característicos | 51 |
| ANEXO III Base de datos elaborada a partir de tiros de jugadores expertos | 53 |
| ANEXO IV Modelo final de la secuencia de banda larga/corta/larga..... | 55 |
| ANEXO V Código del programa final..... | 59 |
| ANEXO VI Ejemplos del correcto funcionamiento del programa final ... | 71 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| <i>Figura 1.1.</i> Dimensiones de una mesa de billar de carambola..... | 11 |
| <i>Figura 1.2. a</i> Diferentes ejemplos a resolver por el entrenador virtual (1) | 14 |
| <i>Figura 1.2 b</i> Diferentes ejemplos a resolver por el entrenador virtual (2) | 14 |
| <i>Figura 2.1</i> Direcciones que relacionará el entrenador virtual..... | 18 |
| <i>Figura 2.2</i> Puntos que caracterizan las direcciones de entrada y salida | 19 |
| <i>Figura 4.1</i> Ejemplos de situaciones de juego a estudiar | 23 |
| <i>Figura 4.2</i> Ejes para permitir la comunicación billarista-programa | 24 |
| <i>Figura 4.3</i> Secuencia horaria con banda superior a partir de punto inicial y punto final | 28 |
| <i>Figura 4.4</i> Diferentes secuencias a partir de punto inicial y punto final | 28 |
| <i>Figura 5.1</i> Interfaz gráfica final al iniciarse | 31 |
| <i>Figura 5.2</i> Ejemplo de representación de los puntos inicial y final..... | 31 |
| <i>Figura 5.3</i> Resultado del cálculo de la acción óptima..... | 32 |
| <i>Figura 5.3</i> Ejemplo de solución imposible | 33 |
| <i>Figura 5.4</i> Ejemplo de mensaje de error | 33 |

| | |
|--|----|
| <i>Figura 5.5</i> Resultado del cálculo de la acción óptima de los puntos de la Figura 5.3..... | 34 |
| <i>Figura 6.1</i> Ejemplo de puntos inicial y final en situaciones de partido .. | 35 |
| <i>Figura 6.2</i> Resultado del cálculo de la acción óptima de los puntos de la Figura 6.1..... | 36 |
| <i>Figura 6.3</i> Propuesta de soluciones alternativas para la Figura 6.1..... | 36 |
| <i>Figura 6.4</i> Ajuste de la posición de los puntos inicial y final de acuerdo con la Figura 6.3 | 37 |
| <i>Figura 6.5</i> Resultado del cálculo de la acción óptima de los puntos de la Figura 6.4..... | 37 |
| <i>Figura A1.1</i> Dirección inicial de la bola jugadora..... | 46 |
| <i>Figura A1.2</i> Recorrido de la bola con velocidad media. | 47 |
| <i>Figura A1.3</i> Ejes principales de la bola..... | 48 |
| <i>Figura A1.4</i> Altura de ataque para alcanzar directamente el estado estacionario..... | 49 |
| <i>Figura A1.5</i> Puntos de golpeo para efecto favorable. | 50 |
| <i>Figura A2.1</i> Sistema de referencia de los puntos de salida..... | 51 |
| <i>Figura A2.2</i> Sistema de referencia de los puntos de ataque..... | 52 |
| <i>Figura A2.3</i> Sistema de referencia de los puntos de llegada | 52 |
| <i>Figura A6.1</i> Código QR con acceso a vídeos explicativos del programa final..... | 71 |

ÍNDICE DE GRÁFICOS

| | |
|--|----|
| <i>Gráfico 3.1</i> Semi-modelos salida/ataque/llegada..... | 22 |
| <i>Gráfico A4.1</i> Semi-modelo 1 (vista 1) | 55 |
| <i>Gráfico A4.2</i> Semi-modelo 1 (vista 2) | 56 |
| <i>Gráfico A4.3</i> Semi-modelo 2 (vista 1) | 56 |
| <i>Gráfico A4.4</i> Semi-modelo 2 (vista 2) | 57 |

ÍNDICE DE TABLAS

| | |
|---------------------------------------|----|
| <i>Tabla A3.1</i> Base de datos | 53 |
|---------------------------------------|----|

*Todos los *Gráficos, Figuras y Tablas* mostrados a lo largo de este documento que no se especifique su procedencia son de elaboración propia.

1. Introducción

1.1 El billar a tres bandas

El billar a tres bandas es una modalidad deportiva del billar de carambola, reconocida en 1998 por el Comité Olímpico Internacional¹. El juego se desarrolla con tres bolas (una blanca, una amarilla y una roja) de 61,5 mm de diámetro sobre una mesa rectangular cuyas dimensiones son de 1,42 x 2,84 metros.

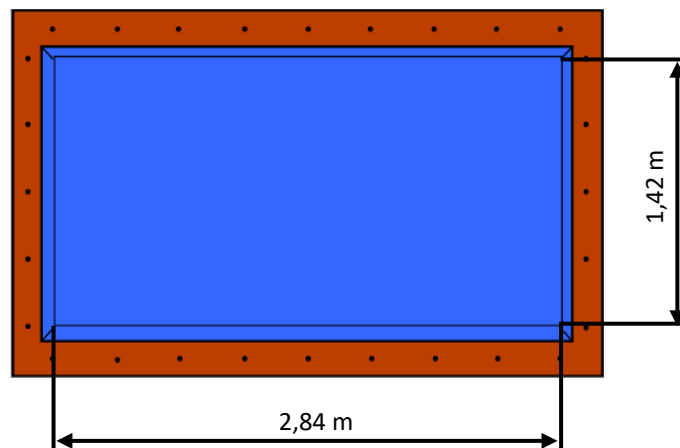


Figura 1.1. Dimensiones de una mesa de billar de carambola

El objetivo del juego consiste en conseguir más carambolas que el rival, en una serie determinada de entradas². Para realizar una carambola en el billar a tres bandas, el jugador debe conseguir que su bola (sobre la que golpea con el taco) llegue a contactar con las otras dos bolas, con la condición de que, antes de contactar con la segunda bola objetivo, la bola jugadora ha de haber chocado previamente con un mínimo de tres bandas a lo largo de su trayectoria. Con estas premisas se pueden dar diversas secuencias de tiro, ya que puede darse que el jugador prefiera golpear primeramente hacia una bola objetivo, con la intención de que posteriormente golpee tres bandas para finalizar chocando con la segunda bola objetivo. También el jugador puede decidir atacar hacia una banda, para que la bola jugadora golpee primero tres bandas y posteriormente choque con las dos bolas objetivo, al igual que cualquier otra secuencia siempre que se cumpla con la condición expuesta anteriormente.

¹ World Confederation of Billiards Sport. WCBS, *History*. Sint-Martens-Latem (Bélgica). [Internet]. 2015 [Consultado 7 Ago 2018]. Disponible en: <https://www.wcbs-billiards.org/history/>

² En billar una entrada significa cada salida a la mesa de los contrincantes.

1.2 Objetivo y alcance del proyecto

El billarista precisa conocer cómo será el comportamiento que tengan las bolas tras los diversos choques entre ellas y con las bandas, para poder conocer con exactitud cuál será el punto al que ha de hacer dirigir su bola, y cómo habrá de hacerlo, para que finalmente consiga realizar la carambola de manera satisfactoria.

Es aquí en donde se encuentra el objetivo de este proyecto, el de diseñar un entrenador virtual que le indique al billarista, mientras realiza sus entrenamientos, cómo y dónde ha de hacer dirigir su bola para que, de esta forma, consiga maximizar la probabilidad de carambola en el billar a tres bandas.

1.3 Análisis de estudios previos

El billar es un deporte que ha sido sometido a muchos estudios con objetivos parecidos al que se plantea en este proyecto. Puede destacarse el trabajo realizado en 2001 por Larsen³, quien desarrolla un programa informático que ayuda al jugador a mejorar su destreza en el billar, pero en la modalidad de pool en este caso, y únicamente ante una serie de posiciones precargadas previamente en el programa. Al igual que el estudio de Smith⁴, elaborado en 2006, en donde se desarrolla un programa que ayuda a los billaristas, o a los robots que sean programados para jugar al billar, a elegir entre la mejor opción de tiro en cada situación de partido, en base a diferentes métodos estadísticos, aunque también está centrado en las modalidades de pool y snooker (ambas modalidades comparten como objetivo de juego el embocar diferentes bolas en seis troneras colocadas alrededor de la superficie de juego, por lo que el resultado no sería extrapolable a una disciplina del billar de carambola, como es la modalidad de tres bandas).

También se puede destacar el proyecto de Bahr⁵, realizado en 2012, quien desarrolla un programa que simula un juego virtual de la modalidad de pool de Bola 8, basado en conceptos mecánicos para lograr caracterizar el choque entre las bolas. O el reciente estudio de Fragkiadaki⁶, del año 2016, por el que a través de las imágenes de diferentes partidos tomadas por una cámara el programa logra aprender el comportamiento de las

³ Lars Bo Larsen, Peter M. Jensen, Kenneth Kammergaard, Lars Kromann, *The Automated Pool Trainer - a Multi Modal System for Learning the Game of Pool*. [Internet]. Aalborg (Dinamarca): Aalborg University, 2001

⁴ Michael Smith, *Running the Table: An AI for Computer Billiards*. [Internet]. Edmonton (Canadá): Department of Computing Science, University of Alberta, 2006

⁵ Jens-Uwe Bahr, *A computer player for billiards based on artificial intelligence techniques*. [Internet]. Kiel (Alemania): Universität Kiel, 2012

⁶ Katerina Fragkiadaki, Pulkit Agarwal, Sergey Levine, Jitendra Malik, *Learning visual predictive models of physics for playing billiards*. [Internet]. Berkeley, California (EE.UU.): University of California, Berkeley, 2016

bolas de billar, para poder deducir, a partir de una situación cualquiera de partido, y un tiro predeterminado, cuál será la situación final de las bolas sobre la mesa de billar, aunque se trabaja en todo momento en una disciplina de pool.

El estudio encontrado con mayor similitud a este proyecto en cuanto a objetivo y problemática a abordar es el estudio elaborado por Joris van Balen⁷ en el año 2009, en donde el problema es resuelto mediante la realización de un modelo mecánico, que es construido a partir de aproximaciones matemáticas de todos los parámetros que afectan al golpeo, que van desde la fuerza que se le imprime a la bola en el momento del golpeo hasta los coeficientes de rozamiento que existen entre la bola, el tapete y las bandas. Finalmente, y como se muestra en “*Appendix F*” de dicho estudio, únicamente uno de 21 ejemplos llevados a la práctica termina con una carambola correctamente realizada. Se puede observar que el fallo más común al aplicar esta estrategia de resolución se basa en la acumulación de errores cometidos en cada una de las aproximaciones matemáticas llevadas a cabo para poder cuantificar todos los parámetros involucrados en un tiro de billar.

Además, también se dispone del trabajo previo publicado por el autor de este proyecto en el libro *Mecánica de una rodada en el billar a tres bandas*⁸.

1.4 Metodología

Con todo ello, y para dar respuesta al objetivo planteado, se partirá de la premisa de que la secuencia de tiro más repetida en esta disciplina deportiva es en la que la bola jugadora realiza una trayectoria que comienza golpeando primeramente con una banda larga, posteriormente chocará con una banda corta para terminar contactando con la banda larga opuesta, y así completar la secuencia de tres bandas necesaria para que la carambola sea correcta. Esta secuencia será válida tanto si el jugador decide cumplir el requisito de que su bola contacte con las tres bandas antes de haber chocado con alguna bola objetivo (como se muestra en la *Figura 1.2. a*), o bien si el jugador decide chocar primero con una de las bolas objetivo y después contactar con la secuencia de bandas antes de terminar chocando con la segunda bola objetivo (*Figura 1.2. b*).

⁷ Joris van Balen, *A Virtual Billiard Assistant*. Twente (Países Bajos): University of Twente (Masters Thesis), 2009

⁸ Ignacio Díez Ruiz, *Mecánica de una rodada en el billar a tres bandas*. Beau Bassin-Rose Hill (República de Mauricio): Editorial Académica Española, 2017

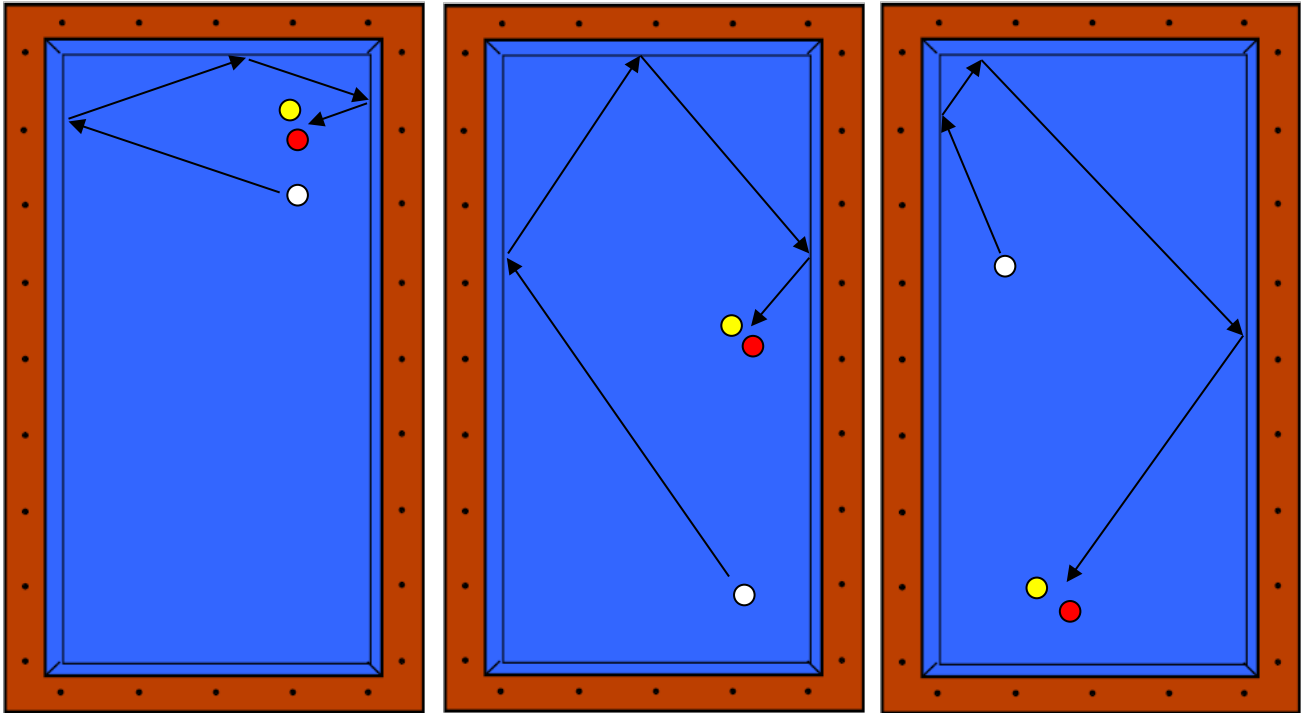


Figura 1.2. a Diferentes ejemplos a resolver por el entrenador virtual (1)

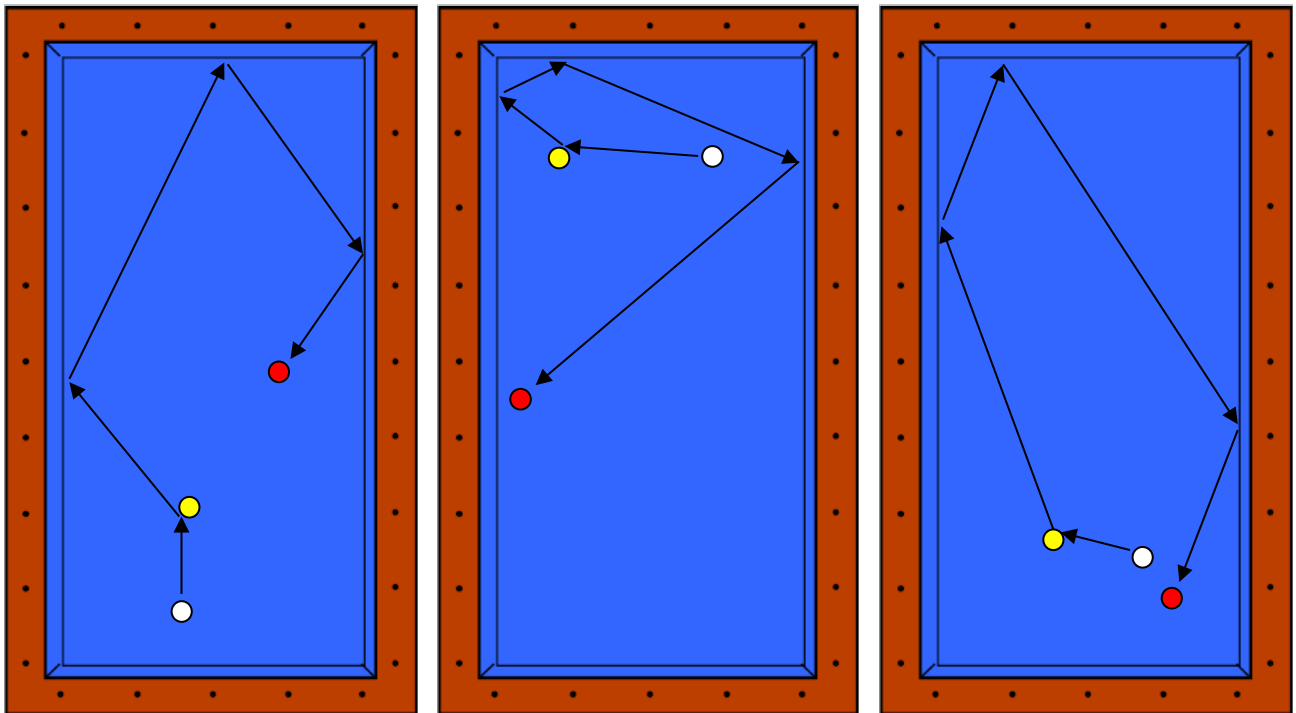


Figura 1.2 b Diferentes ejemplos a resolver por el entrenador virtual (2)

Con los diferentes ejemplos expuestos en la *Figura 1.2* el jugador podrá resolver la mayoría de las posiciones que se pueden dar a lo largo de un partido de billar.

Como todas las posiciones de la *Figura 1.2* comparten la misma secuencia de banda larga, banda corta y banda larga contraria, esta secuencia deberá ser modelada por el entrenador virtual para poder cumplir con el objetivo del proyecto. Este modelo, a

diferencia del estudio de van Balen citado anteriormente, será construido a partir de datos recogidos de diferentes tiros de jugadores expertos en billar, que serán procesados de acuerdo a diferentes conceptos de aprendizaje automático, para que el propio programa reconozca y comprenda el comportamiento del modelo.

Posteriormente, a partir del modelo ya generado, se deberá programar el cálculo de la acción óptima (cómo y dónde la bola debe entrar a la secuencia de bandas), para poder conseguir la respuesta deseada (que la bola termine llegando finalmente al punto necesario para completar la carambola).

Por último, se diseñará una interfaz gráfica para que el jugador pueda interactuar con el entrenador virtual de manera clara y eficaz.

Tanto para la elaboración del modelo como para el cálculo de la acción óptima y el desarrollo final de la interfaz gráfica se utilizará la herramienta de software matemático Matlab®, en su versión R2018a (el uso de esta versión permite la utilización de la aplicación App Designer, incorporada de forma nativa en el propio programa).

Para poder asegurar la utilidad del entrenador virtual a diseñar, hay que conocer todos los parámetros que influyen en cualquier golpeo de billar. En un tiro el billarista puede actuar sobre seis parámetros diferentes, que son la dirección de la bola jugadora, la velocidad inicial con la que sale despedida, la altura, el efecto, la penetración y la inclinación con las que el billarista golpea a la bola. Estos seis parámetros del golpeo son definidos y detallados en el **Anexo I** de este documento.

1.5 Club de Billar Círculo Amistad Numancia

El C.A. Numancia es un club de billar de Soria que compite a nivel profesional en disciplinas tanto de pool como de carambola, y que cuenta con un equipo en la Primera División de la Liga Nacional de Billar a Tres Bandas. Además, el club, en colaboración con el Departamento de Deportes del Excmo. Ayuntamiento de Soria, cuenta desde el año 2009 con una escuela de billar que consta de seis clases semanales a diferentes grupos en función del nivel de juego.

Las instalaciones del Club están situadas en la Calle Collado nº 23 de Soria, y cuentan con tres mesas de carambola de alta competición (marca SAM modelo Élite, con bandas Kleber, paños HRD y juego de bolas Super Aramith Por Cup Prestige), al igual que con una mesa de pool Dynamic III.

La directiva del Club ha puesto a disposición de este proyecto tanto el material necesario como personal técnico y jugadores profesionales que realicen las medidas y pruebas oportunas.

El programa final desarrollado será utilizado por los propios jugadores del club para realizar sus entrenamientos, al igual que por el cuerpo técnico para conseguir mejorar y maximizar el aprendizaje de los alumnos de la escuela de billar.

2. Elaboración de una base de datos

Como se ha mencionado anteriormente, para que el programa consiga calcular cómo y dónde la bola jugadora deberá iniciar la secuencia de bandas, éste debe conocer el comportamiento exacto de la bola jugadora a lo largo de su trayectoria, para poder asegurar que el cálculo final que realice corresponda exactamente con el que finalmente termine completando la carambola de manera satisfactoria.

Por todo ello, el sistema deberá conocer un modelo en el que se relacionen todas las variables sometidas a estudio, para así asegurar la eficacia del mismo.

Para conseguir dicho modelo, el sistema contará con una base de datos que le será introducida con el objetivo de que el propio programa, basándose en técnicas de aprendizaje automático, pueda analizarla y deducir las relaciones que existen en ella. Esta base de datos será elaborada a partir de la información recogida observando el comportamiento de la bola jugadora en diferentes tiros, que les serán solicitados a jugadores expertos en esta disciplina.

Los jugadores expertos serán jugadores del C.A. Numancia, pertenecientes a la especialidad de carambola. Los datos serán tomados en la Mesa 1 del propio club. La temperatura del tapete de la mesa estará controlada en todo momento, estando por encima de los 30°C (para así disminuir la humedad y el rozamiento de las bolas con el tapete), y la humedad de la sala se mantendrá en todo momento por debajo del 30% de humedad relativa. El taco que se utilizará para golpear a la bola jugadora será un Adam de la Serie 900, que cuenta con una flecha Super Profesional con un diámetro de suela⁹ de 12 mm.

Para iniciar la toma de datos, se debe conocer cuáles son exactamente los datos que precisa el sistema para poder satisfacer el objetivo del estudio. En el **Anexo I** incluido en este documento se detalla el control que se llevará a cabo de los diferentes parámetros del golpeo, por lo que, y tal y como se explica en el mismo, todos los tiros que se ejecuten para la construcción de la base de datos tendrán el mismo valor de velocidad, penetración e inclinación, que el efecto que se le imprima a la bola jugadora será favorable en todo momento, y que la bola jugadora, antes de chocar con la primera banda ya habrá alcanzado la condición de no deslizamiento. Con todo ello, el único parámetro de golpeo sobre el que se va a actuar será la dirección de la bola jugadora.

⁹ La suela es la parte del taco con la que se golpea a la bola

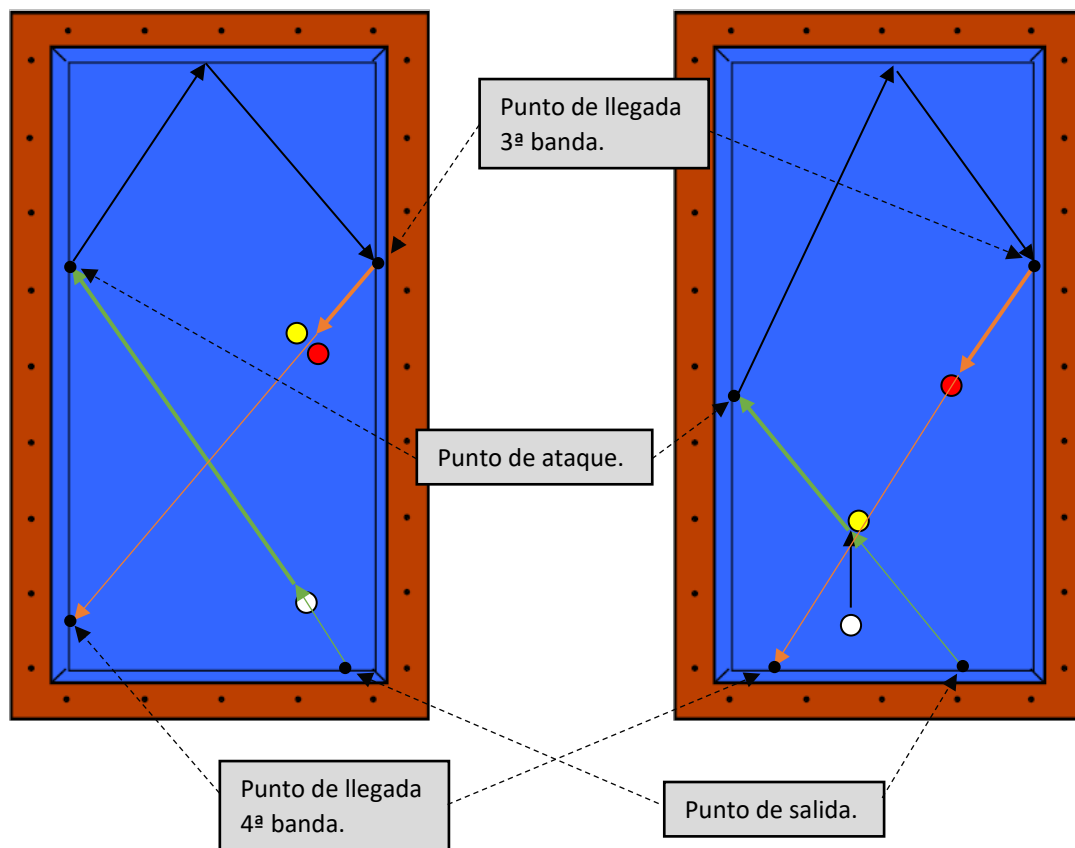


Figura 2.2 a

Figura 2.2 b

Figura 2.2 Puntos que caracterizan las direcciones de entrada y salida

Con todo ello, el modelo únicamente tendrá que relacionar los cuatro puntos mostrados en la *Figura 2.2*, para que el entrenador virtual final pueda deducir la dirección exacta con la que se ha de entrar a la secuencia de bandas, para asegurar que en la dirección de salida se encuentren las dos bolas objetivo (*Figura 2.2 a*) o la segunda bola objetivo (*Figura 2.2 b*).

Las marcas (rombos) que existen a lo largo de las diferentes bandas servirán de ayuda para poder recoger los datos, y posteriormente para que el sistema indique al billarista el punto exacto al que ha de atacar para no errar en la carambola. Por ello, la distancia existente entre dos rombos consecutivos en la misma banda sobre la mesa de billar será tomada como unidad de medida a lo largo de este proyecto (considerándose por lo tanto para el resto del proyecto que $1 \text{ rombo} = 35,5 \text{ cm}$).

Cabe mencionar que hasta ahora, todas las figuras que han sido mostradas en este documento, únicamente contemplan la secuencia de banda larga/corta/larga realizada en un sentido horario, y apoyándose en la banda larga superior. El entrenador virtual final será capaz de resolver la secuencia de banda larga/corta/larga, tanto en un sentido horario como antihorario, al igual que apoyándose tanto en la banda corta superior de la mesa como en la banda corta inferior. Pero, y dadas las simetrías que poseen las

mesas de billar, únicamente será modelada la secuencia realizada en un sentido horario con banda superior, por lo que únicamente será necesario tomar datos de esta secuencia.

El sistema de referencia con el que se va a numerar los diferentes rombos a lo largo de la mesa, para los cuatro puntos característicos detallados anteriormente está detallado en el **Anexo II** de este documento.

Una vez se dispone de un sistema de referencia para poder numerar los diferentes datos obtenidos, se puede proceder a la toma de datos. Para ello, tal y como se ha comentado anteriormente, se solicitará a jugadores expertos que realicen diferentes tiros, en los que la bola jugadora se posicionará en diversos lugares sobre la mesa y será golpeada desde los diversos puntos de salida hacia los diversos puntos de ataque (mostrados en la **Anexo II**). Una vez que la bola jugadora haya completado la trayectoria para cada tiro, se anotará el punto de llegada sobre el que habrá golpeado la bola jugadora, tanto en la tercera como en la cuarta banda a lo largo de su recorrido.

El objetivo final es que en la base de datos que se le acabe introduciendo al programa exista información que abarque todo el rango de puntos de salida y de ataque que las dimensiones de la mesa permite. Por ello, y para realizar una ordenada toma de datos, se partirá del primer punto de salida definido, y se procederá a atacar a cada uno de los puntos de ataque mostrados en el **Anexo II**. Una vez se tenga la información de los puntos de llegada para todos los puntos de ataque, se volverá a repetir la operación para el siguiente punto de salida.

Con todo ello, para cada tiro que se realice, se tendrá la información exacta de la dirección de entrada de la bola jugadora a la secuencia de bandas, y cuál ha sido la dirección de salida de la misma.

Debido a la posibilidad de que exista algún tipo de error accidental en cualquier medida que pueda llegar a distorsionar el resultado final del modelo, cada medida será repetida varias veces y por varios jugadores, anotándose finalmente el resultado más repetido.

Para poder medir con mejor precisión tanto el punto de llegada sobre la tercera banda como el punto de llegada sobre la cuarta banda, se han elaborado dos reglas de papel para posicionarlas sobre las bandas, y que dividen los rombos en diez partes iguales, creando así un sistema de numeración decimal en el que la unidad de medida corresponde a la distancia entre dos rombos consecutivos a lo largo de la misma banda.

Con los resultados finales de las mediciones se ha elaborado la base de datos final, que aparece detallada en el **Anexo III** de éste documento.

3. Elaboración del modelo

Una vez se dispone de la base de datos con toda la información necesaria de los diferentes tiros ejecutados por jugadores expertos, se puede proceder a elaborar el modelo que identifique y comprenda la relación de estos datos, para, de esta forma, poder saber con antelación cuál será la dirección de salida de la bola jugadora de la secuencia de bandas para cualquier dirección de entrada posible.

La base de datos consta de 82 entradas, una por cada tiro diferente ejecutado, y cada entrada cuenta con la información de cuatro puntos diferentes, dos de ellos corresponderán con puntos de entrada al modelo (punto de salida y punto de ataque) y los otros dos con salidas del mismo (punto de llegada sobre la tercera banda y punto de llegada sobre la cuarta banda).

Por este motivo se decide que el modelo final sea el resultado de la conjunción de dos semi-modelos diferentes. Cada uno de estos semi-modelos tendrá como entradas los dos puntos de entrada detallados anteriormente, pero únicamente uno de los puntos de salida (un semi-modelo proporcionará la información del punto de llegada sobre la tercera banda mientras que el otro lo hará del punto de llegada sobre la cuarta banda). Por lo que teniendo la información procedente de los dos semi-modelos será posible alcanzar el objetivo final del proyecto

Al limitar el número de variables de cada semi-modelo a tres, siendo dos de ellas completamente independientes (las entradas al semi-modelo) y la tercera dependiente de estas dos (la salida final del mismo), es posible el uso de la aplicación *Curve Fitting Tool*, incorporada en el software matemático Matlab®, que es capaz de relacionar estas tres variables, estableciendo las relaciones matemáticas oportunas entre ellas, para así poder deducir la salida correspondiente para cualquier combinación de las variables de entrada.

Además, es posible representar de forma gráfica cada uno de los semi-modelos en un gráfico de tres dimensiones, en donde cada dimensión corresponde con cada variable del semi-modelo. Los gráficos correspondientes a cada semi-modelo se representan en el *Gráfico 3.1*. En ellos, los puntos dibujados en color azul corresponden con los puntos de la base de datos, utilizada para generar el modelo, mientras que la superficie representada corresponde con cada uno de los semi-modelos.

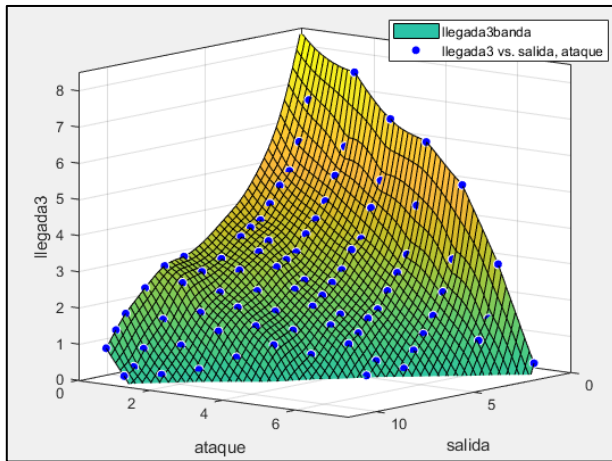


Grafico 3.1 a

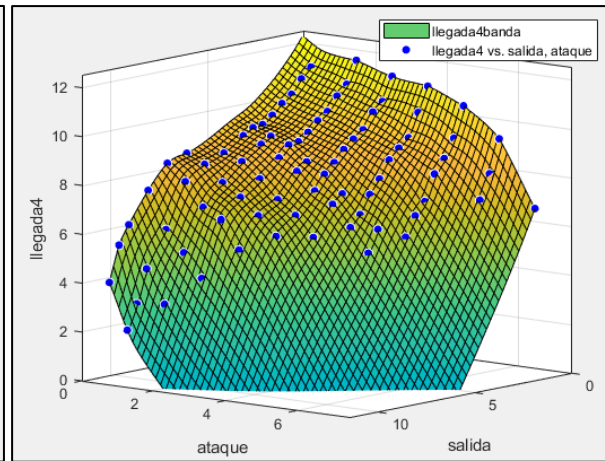


Grafico 3.1 b

Gráfico 3.1 Semi-modelos salida/ataque/llegada

Los ejes de ambos gráficos mostrados son coherentes con la numeración mostrada en el **Anexo II** de dicho documento.

En el **Anexo IV** de este documento se muestran con mayor detalle los gráficos correspondientes a cada semi-modelo.

4. Cálculo de la acción óptima

Una vez se dispone del modelo generado, siendo este capaz de calcular la salida oportuna para cualquier combinación de entradas, es momento de dar respuesta a cuál ha de ser la dirección que deberá llevar la bola jugadora antes de entrar en la secuencia de bandas para conseguir completar la carambola, o lo que es lo mismo, cuál ha de ser la acción de entrada al modelo generado para que devuelva la salida deseada.

4.1 Definición de punto inicial y punto final

Para poder resolver esta pregunta primeramente se debe definir qué información posee el billarista ante una situación de juego cualquiera, y cómo interactuará éste con el modelo para conseguir resolver el problema. Para ello recuperamos las mismas situaciones de partido mostradas tanto en la *Figura 2.1* como en la *Figura 2.2*.

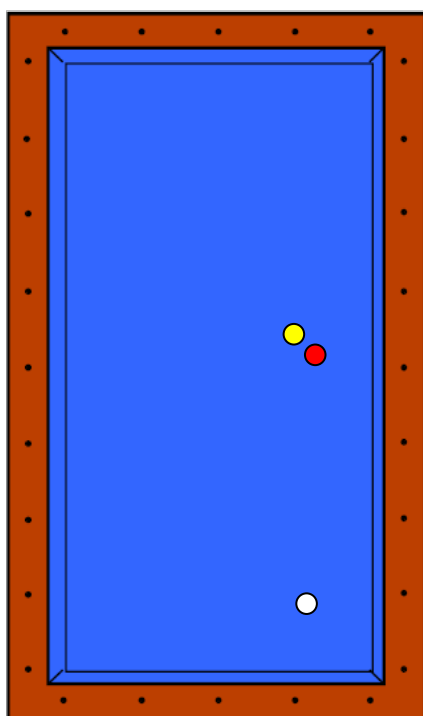


Figura 4.1 a

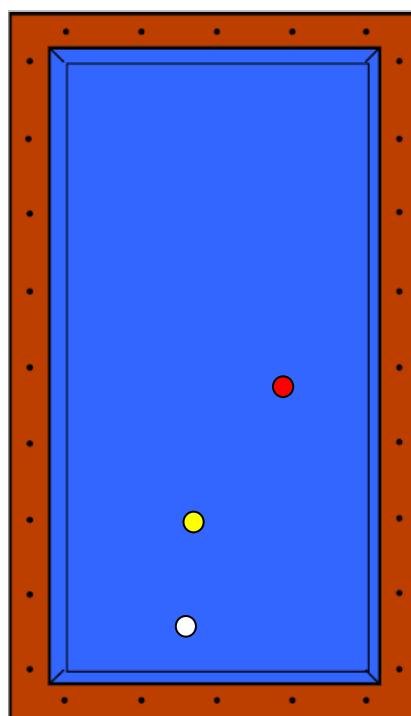


Figura 4.1 b

Figura 4.1 Ejemplos de situaciones de juego a estudiar

El billarista, tal y como se muestra en la *Figura 1.4*, únicamente es conocedor de la posición inicial de las bolas sobre la mesa, por lo que no conoce cuál ha de ser la dirección de salida de la bola jugadora, pero sí sabe que dicha dirección deberá contener un punto intermedio entre las bolas roja y amarilla en la *Figura 1.4 a*, o el punto sobre el que se sitúa la bola roja en la *Figura 1.4 b*, para de esta manera asegurar que se realice la carambola.

Además, el cálculo de la dirección de entrada de la bola jugadora a la secuencia de bandas no puede dar nunca como resultado una dirección que no contenga el punto sobre el que está situada la bola blanca sobre la mesa en la *Figura 1.4 a*, ni que contenga el punto que dista aproximadamente la distancia del diámetro de la bola por debajo de la bola amarilla en el caso de la *Figura 1.4 b*. De no cumplir esta condición en cada caso, la bola jugadora será imposible que recorra la dirección calculada antes de golpear la primera banda, por lo que no será válida.

Por lo tanto, el billarista lo único que conoce a partir de una situación de juego cualquiera, es un punto de la mesa por el que va a pasar la bola jugadora antes de entrar a la secuencia de bandas, y que deberá estar contenido en la dirección de entrada al modelo (a este punto se le denominará punto inicial), y un punto por el que deberá pasar la bola jugadora tras completar la secuencia de bandas, y que deberá estar contenido en la dirección de salida del modelo (a este punto se le denominará punto final).

4.2 Sistema de referencia para los puntos inicial y final

El entrenador virtual deberá conocer estos dos puntos, y para ello, el billarista debe introducirselos al programa. Para realizar esta comunicación entre el billarista y el programa, se definen dos ejes perpendiculares sobre la mesa, uno sobre la banda corta (eje x) y otro sobre la banda larga (eje y), tal y como se muestra en la *Figura 4.2*. La unidad de medida de estos ejes corresponde con la distancia entre dos rombos consecutivos. El billarista deberá por lo tanto introducir las coordenadas tanto del punto inicial como del punto final al programa.

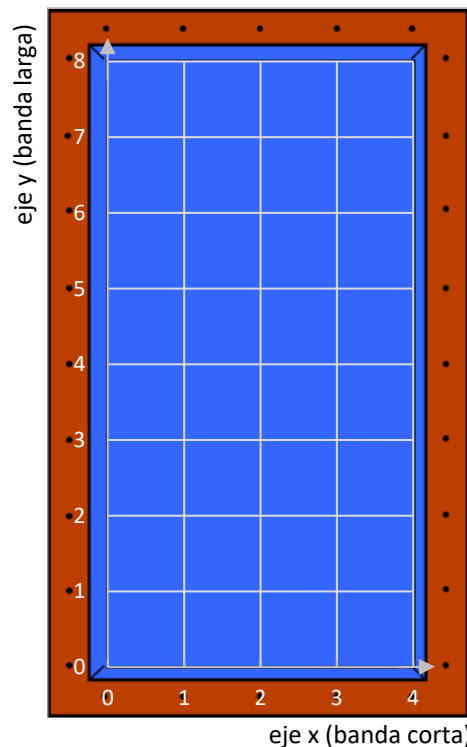


Figura 4.2 Ejes para permitir la comunicación billarista-programa

4.3 Estrategia de cálculo de la acción óptima

Una vez el programa conoce tanto los puntos inicial como final, ya puede realizar el cálculo de la acción óptima a través de la cual se completará la carambola. Para ello, se realizará un barrido entre todos los posibles puntos de ataque que las dimensiones de la mesa permiten; el paso entre un punto de ataque y el siguiente a valorar dependerá de la precisión con la que se quiera obtener el resultado final. Para cada posible punto de ataque, se trazará una línea que lo una con el punto inicial, prolongándose hasta que se corte con la banda en la que se numeran los puntos de salida. Con ello se conseguirá obtener una combinación de puntos de salida y ataque (cuya dirección contiene al punto inicial), o lo que es lo mismo, una acción de entrada al modelo. Esta dirección de entrada será introducida en el modelo para obtener los puntos de llegada sobre la tercera y sobre la cuarta banda, por lo que se construirá la dirección de salida de la bola jugadora de la secuencia de bandas, y se valorará si el punto final introducido inicialmente se encuentra en esa dirección, o en las proximidades de la misma, en un entorno acotado por un error asumible. El error que se permitirá dependerá de la precisión escogida para cada cálculo.

Finalmente, el punto de ataque que cumpla con el requisito mostrado anteriormente será el que consiga satisfacer la carambola, por lo que se habrá conseguido calcular la acción óptima para que el tresbandista logre no errar en el tiro.

4.4 Cálculo de la acción óptima para todas las secuencias de banda larga/corta/larga

Como ya se ha comentado anteriormente, el modelo realizado únicamente resuelve la secuencia de bandas realizada en un sentido horario, y apoyándose sobre la banda corta superior, pero con la secuencia de bandas larga/corta/larga, existen otras tres posibles soluciones al problema. En la *Figura 4.3* se muestra un ejemplo en el que se sitúa sobre la mesa un punto inicial (representado por una bola blanca) y un punto final (representado por una bola roja), y se dibuja sobre la mesa la trayectoria aproximada que realizará la bola jugadora con la secuencia horaria sobre banda superior, tal y como se ha mostrado hasta ahora.

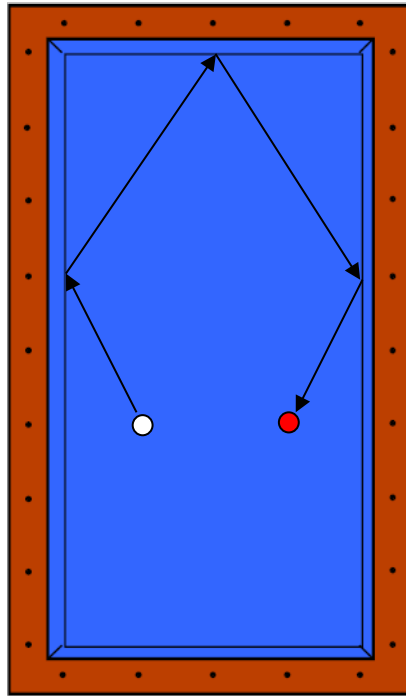


Figura 4.3 Secuencia horaria con banda superior a partir de punto inicial y punto final

En la *Figura 4.4* se muestran las otras tres secuencias posibles para los mismos puntos inicial y final que se han mostrado en la *Figura 4.3*. En la *Figura 4.4 a* se dibuja una trayectoria en la que la bola jugadora seguirá completando la secuencia de bandas apoyándose en la banda corta superior, pero lo realizará en un sentido antihorario, mientras que en la *Figura 4.4 b* y *Figura 4.4 c* la bola jugadora se apoyará en la banda corta inferior, recorriendo la secuencia de bandas en un sentido antihorario y horario respectivamente.

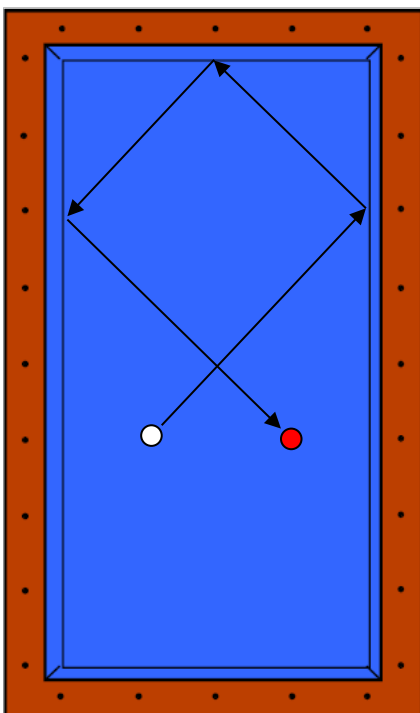


Figura 4.4 a

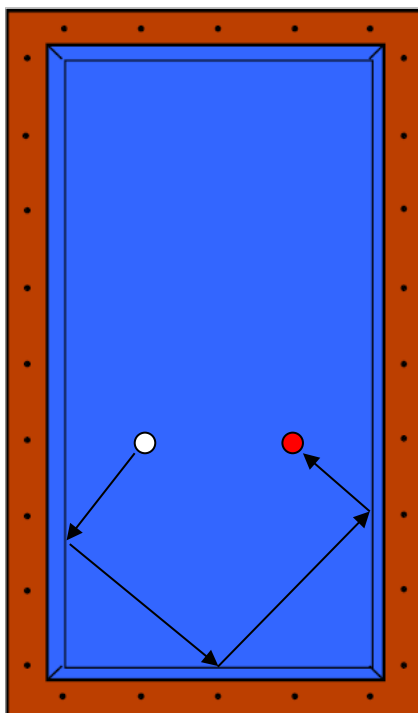


Figura 4.4 b

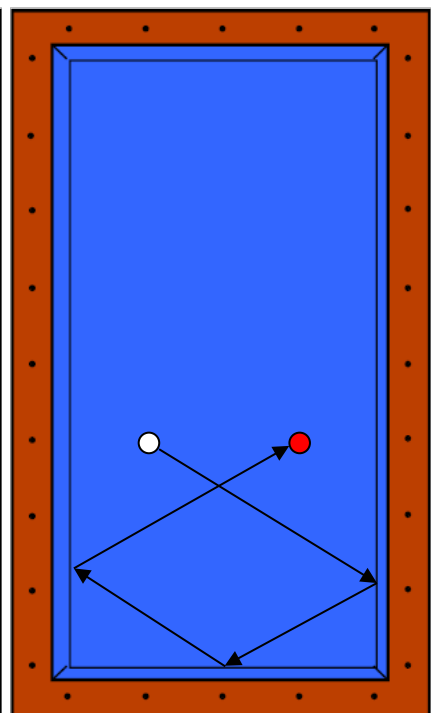


Figura 4.4 c

Figura 4.4 Diferentes secuencias a partir de punto inicial y punto final

Para poder resolver las secuencias mostradas en la *Figura 4.4*, será necesaria primeramente una manipulación de las coordenadas de los puntos inicial y final, aprovechando las simetrías que ofrece la mesa de billar, para poder resolver el problema como una secuencia horaria con banda superior tal y como se ha detallado anteriormente. Una vez el programa muestre el resultado final, éste deberá ser adaptado y manipulado consecuentemente en función de la secuencia que represente. Es decir, sea cual sea la secuencia por la que se desee resolver el problema, se deberán adaptar los puntos inicial y final para que el programa la resuelva en un sentido horario apoyándose en banda superior, y una vez esté resuelto, el resultado final se volverá a adaptar para mostrar el resultado de la secuencia deseada.

Este algoritmo de adaptación estará programado en el propio entrenador virtual, de tal forma que el billarista pueda elegir, primeramente la esquina de la mesa que desee para ser su origen de coordenadas (de acuerdo con la *Figura 4.2*), y, seguidamente, escoger la secuencia de bandas a través de la cual desee resolver el problema. Con todo ello el entrenador virtual llevará a cabo internamente todas las manipulaciones necesarias para resolver el problema y terminar devolviéndole al usuario la solución coherente acorde con la secuencia de bandas que le haya indicado previamente.

5. Desarrollo de una interfaz gráfica

Una vez se ha resuelto el problema del cálculo de la acción óptima, por la que se consigue realizar una carambola en el billar a tres bandas conforme con todas las premisas expuestas, resulta indispensable la realización de una interfaz gráfica, que permita de forma clara y sencilla al billarista interactuar con el entrenador virtual.

Para el diseño de dicha interfaz es necesario primeramente conocer exactamente qué parámetros necesita el entrenador virtual para poder resolver el problema en una situación de juego cualquiera, y qué datos le tiene que devolver el programa al jugador para que sea capaz de ejecutar el tiro conforme a las directrices del entrenador virtual.

5.1 Información que tiene que solicitar la interfaz gráfica

Entre los datos necesarios para que el entrenador virtual sea capaz de resolver el problema se encuentran las coordenadas tanto del punto inicial como final, la secuencia de bandas por la que el billarista desea resolver el problema (tanto si la bola jugadora realizará una trayectoria en un sentido horario o antihorario, como si se apoyará en la banda corta superior o inferior de la mesa), y la precisión que se desea para el resultado final.

Tal y como se ha comentado anteriormente, esta precisión está relacionada con el número de divisiones que se realizan en la banda de ataque, para iterar en busca del resultado correcto, al igual que con el error asumible entre el punto final y la dirección de la bola jugadora tras salir de la secuencia de bandas.

5.2 Información que tiene que devolver la interfaz gráfica

Una vez hayan sido introducidas por parte del billarista las premisas necesarias para el cálculo de la acción óptima, el entrenador virtual puede proceder a realizar el cálculo conforme a lo explicado a lo largo de este documento. Cuando se haya desarrollado el cálculo se deberá devolver al billarista toda la información que éste precisa para poder ejecutar la carambola con éxito. Esta información está compuesta por el punto de ataque sobre la primera banda, definiendo así, junto con el punto inicial anteriormente introducido, la dirección de entrada de la bola jugadora a la secuencia de bandas. Además, el entrenador virtual puede mostrar también el punto de golpeo sobre el que el billarista debe atacar sobre la bola jugadora, ya que, dependiendo de la secuencia elegida, el efecto que se le debe imprimir a la bola jugadora puede ser tanto de derecha como de izquierda, tal y como se detalla en el **Anexo I** de este documento.

Los parámetros del golpeo correspondientes con la velocidad, la penetración del taco en el momento del golpeo y la inclinación del mismo, al haber permanecido constantes a lo largo de la elaboración de la base de datos, tampoco variarán en función de la situación de partido ni del resultado final del entrenador virtual, por ello, no serán detallados en la interfaz gráfica a desarrollar, ya que en ningún caso se contempla ninguna variación en los mismos. La forma con la que el billarista tendrá que atacar a la bola jugadora en relación con estos parámetros del golpeo es la detallada en el **Anexo I**, en donde se indica que se debe atacar en ausencia de penetración y de inclinación del taco respecto del plano horizontal, y con una velocidad media.

Puesto que el entrenador virtual simulará el comportamiento de la trayectoria de una bola de billar entre dos puntos, la interfaz gráfica mostrará también una imagen de una mesa de billar, sobre la que se dibujarán los ejes expuestos en la *Figura 4.2* de esta memoria, dándose la opción, antes de realizar ningún cálculo, de que el billarista compruebe sobre la mesa virtual la situación de los puntos inicial y final que haya introducido. Para la representación de estos puntos se utilizará el mismo criterio utilizado en la *Figura 4.3* y *Figura 4.4*, representándose el punto inicial con un círculo de color blanco y el punto final con uno rojo. Después del cálculo, además de mostrarse estos dos puntos sobre la mesa, se dibujará el punto de ataque calculado y la trayectoria total de la bola jugadora entre los puntos inicial y final. Como se ha comentado a lo largo de esta memoria, la trayectoria exacta de la bola jugadora entre la primera y la tercera banda no ha sido modelada, por lo que será representada con una línea discontinua, indicando de esta forma que la representación llevada a cabo únicamente corresponde con una aproximación de la trayectoria que finalmente seguirá la bola jugadora en este tramo.

5.3 Desarrollo de la interfaz gráfica final

Una vez expuesto todo lo que la interfaz gráfica final debe contener se puede proceder a realizar su diseño. Para ello se utilizará la aplicación App Designer, incluida en la versión R2018a de la herramienta de software matemático Matlab®. En la *Figura 5.1* se puede ver la interfaz gráfica final de la aplicación nada más iniciarse.

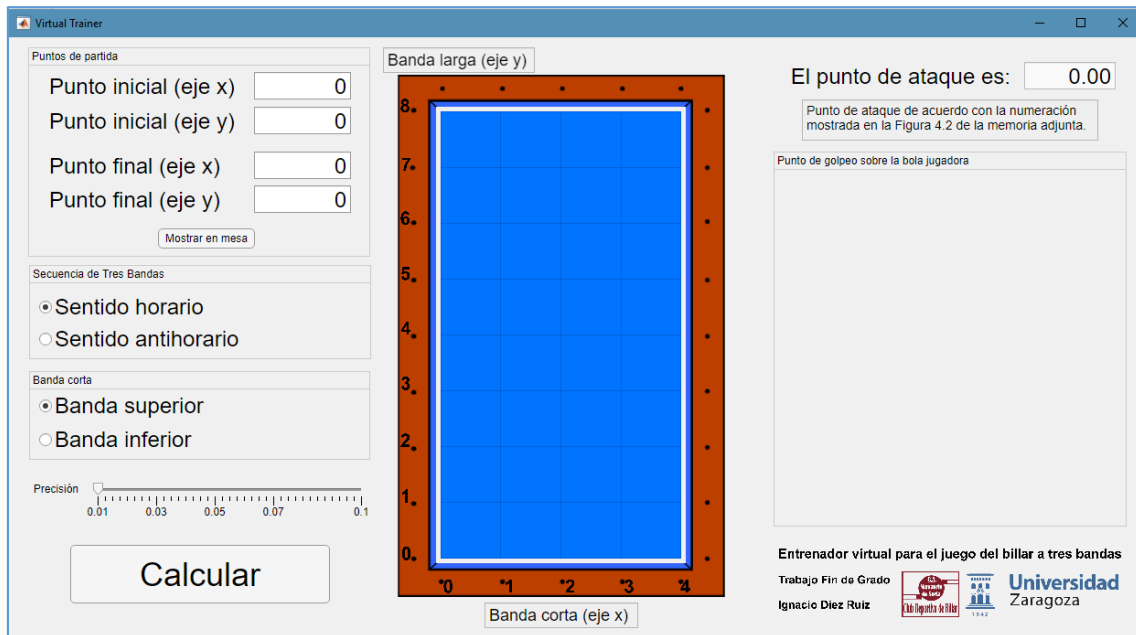


Figura 5.1 Interfaz gráfica final al iniciarse

Como se puede comprobar en la *Figura 5.1*, la interfaz gráfica consta de todos los elementos necesarios para permitir una comunicación eficaz entre el billarista y el programa.

Una vez abierta la aplicación, introducimos las coordenadas de un punto inicial y un punto final cualquiera. Se tomará como ejemplo los puntos inicial y final mostrados en la *Figura 4.3*. Clicando el botón “Mostrar en mesa” se posicionan sobre la mesa virtual dichos puntos, tal y como se muestra en la *Figura 5.2*.

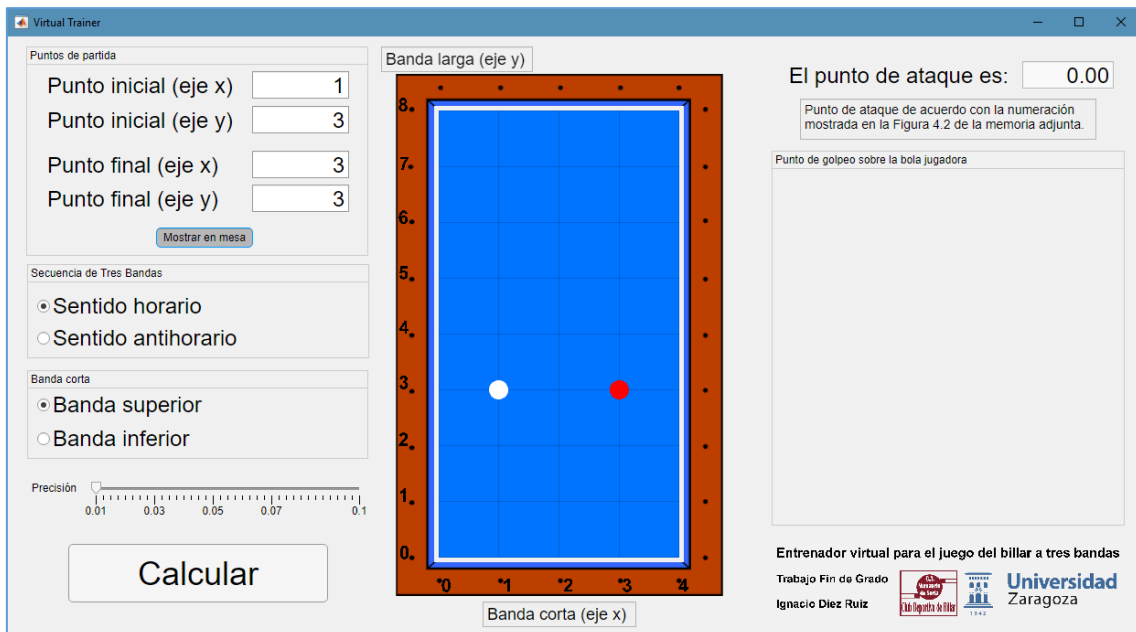


Figura 5.2 Ejemplo de representación de los puntos inicial y final

Una vez el billarista compruebe que ha introducido bien los puntos, deberá elegir la secuencia de bandas que desee que calcule el programa, eligiendo entre sentido horario o antihorario, y escogiendo la banda corta que formará parte de la secuencia de bandas. Además, el billarista puede modificar la precisión con la que desee que el programa calcule el resultado final. Como se ha detallado anteriormente, esta precisión está relacionada con el número de divisiones que el programa realice a lo largo de la banda de ataque para posteriormente valorar cada una de ellas, por lo que si no se precisa una solución fina, se puede cambiar dicho valor (que se encontrará por defecto en 0.01) reduciendo así el coste computacional del cálculo final.

Clicando el botón de “Calcular”, el programa realiza el cálculo de la acción óptima de acuerdo a los parámetros previamente introducidos, tal y como se muestra en la *Figura 5.3*.

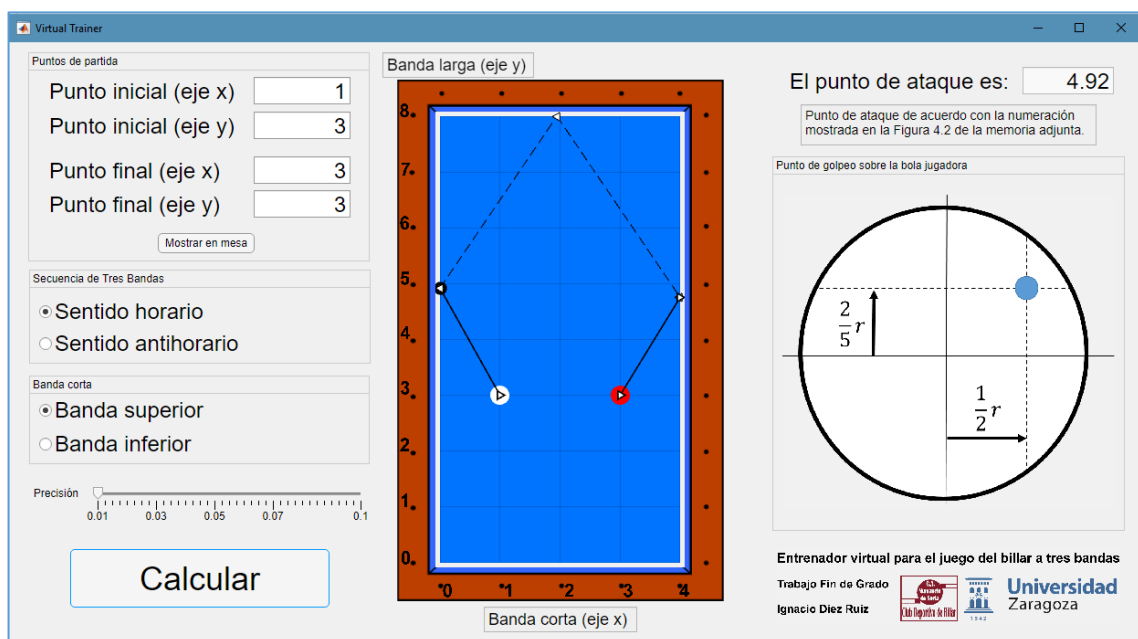


Figura 5.3 Resultado del cálculo de la acción óptima

Tal y como se muestra en la *Figura 5.3* el sistema devuelve el punto de ataque, expresado en el propio sistema de referencia mostrado en la mesa virtual (que a su vez correspondía con el detallado en la *Figura 4.2* de esta memoria). También se representa gráficamente la trayectoria de la bola jugadora entre ambos puntos (las líneas discontinuas muestran que la trayectoria representada en ese tramo es aproximada), al igual que el punto sobre el que el billarista deberá golpear sobre la bola jugadora al ejecutar el tiro, para conseguir el efecto deseado para cada caso.

Existen situaciones de partido en las que resultará imposible realizar la carambola con la secuencia de bandas deseada. Es el caso por ejemplo de la combinación de punto inicial y punto final mostrada en la *Figura 5.4*, en donde se desea resolver el problema con una trayectoria con un sentido horario con apoyo en banda superior.

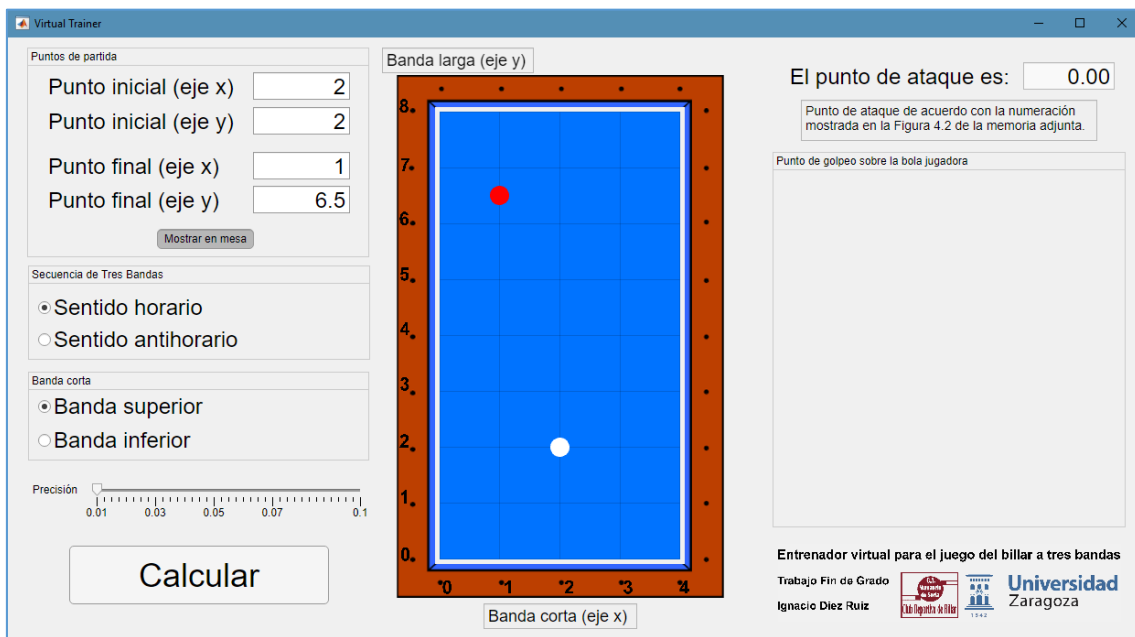


Figura 5.3 Ejemplo de solución imposible

Tal y como se muestra en la *Figura 5.3*, se le solita al programa que calcule la acción óptima para el que la bola jugadora, partiendo del punto inicial mostrado, realice la secuencia de tres bandas en un sentido horario y con contacto en la banda superior, para terminar llegando al punto final detallado. Pero no existe para este caso ningún punto de ataque que pueda llegar a satisfacer dichas condiciones, debido a que el punto final no está contenido en ninguna dirección de salida posible partiendo del punto inicial introducido y con la secuencia de bandas detallada. Para estos casos el programa devuelve un mensaje de error cuando se le solicita el cálculo, tal y como se muestra en la *Figura 5.4*.

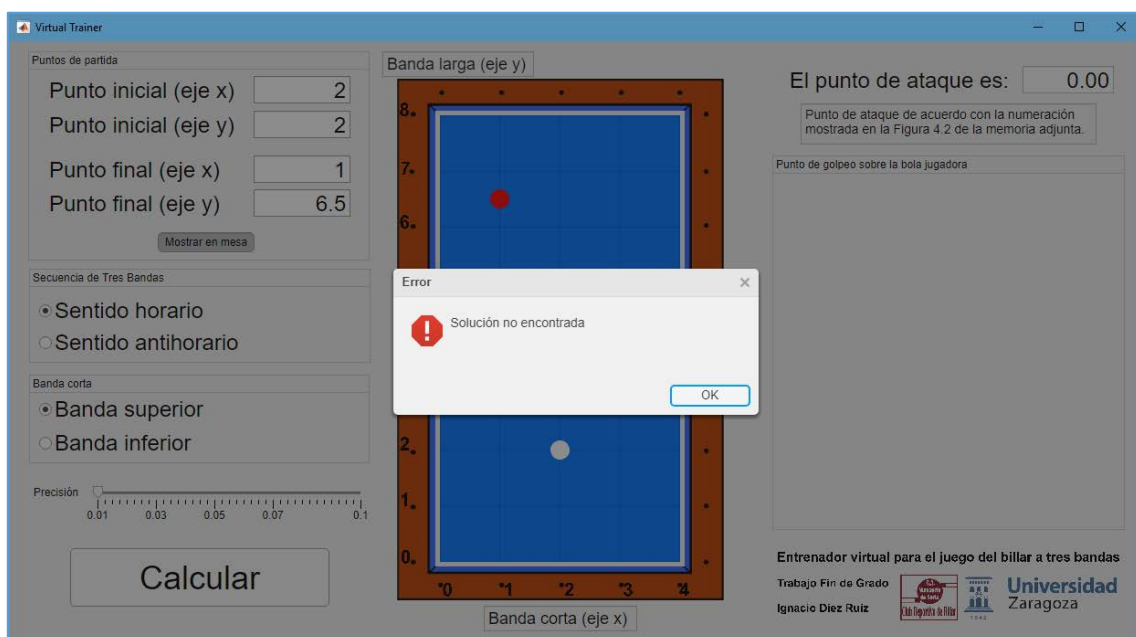


Figura 5.4 Ejemplo de mensaje de error

Sin embargo, aunque sea imposible resolver el problema con los puntos mostrados en la *Figura 5.3* a través de la secuencia deseada, se puede comprobar que con otras secuencias el cálculo de la acción óptima es posible, tal y como se muestra en la *Figura 5.5*.

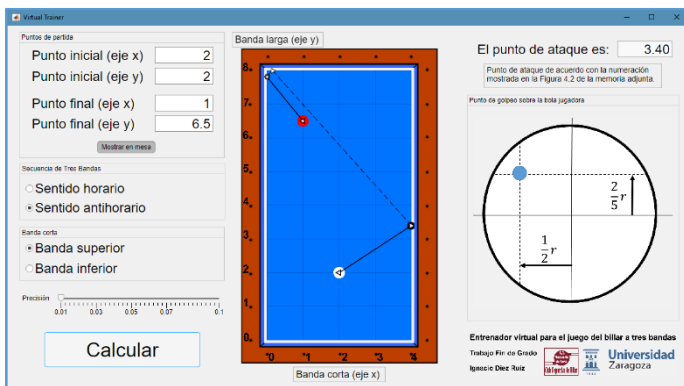


Figura 5.5 a

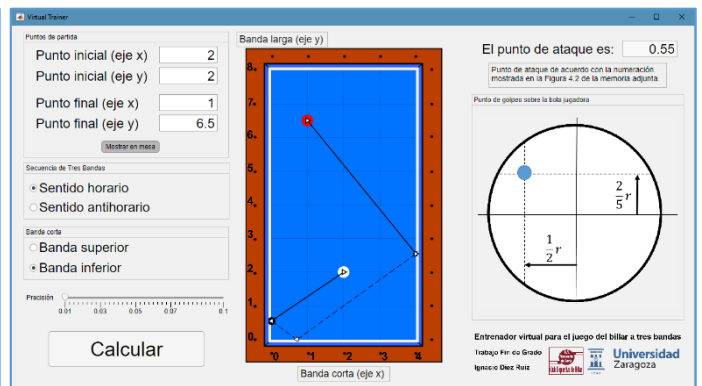


Figura 5.5 b

Figura 5.5 Resultado del cálculo de la acción óptima de los puntos de la *Figura 5.3*

Aun con todo ello, es posible comprobar que existen configuraciones sobre la mesa que resultan imposibles de resolver mediante ninguna de las cuatro posibilidades.

El código del programa final se muestra en el **Anexo V** de este documento.

6. Ejemplos de aplicación del programa final

Una vez se dispone ya del programa final se procederá, a modo de ejemplo, a resolver alguna situación de partido haciendo uso del propio entrenador virtual. Para ello se van a recuperar las posiciones mostradas en la *Figura 4.1*, y se resolverá el problema de acuerdo a la secuencia de bandas mostrada en la *Figura 2.1*. Sobre la mesa, se marcarán con un aspa tanto el punto inicial como final, ya que se precisa conocer sus coordenadas para introducirlas en el programa, esto se muestra en la *Figura 6.1*.

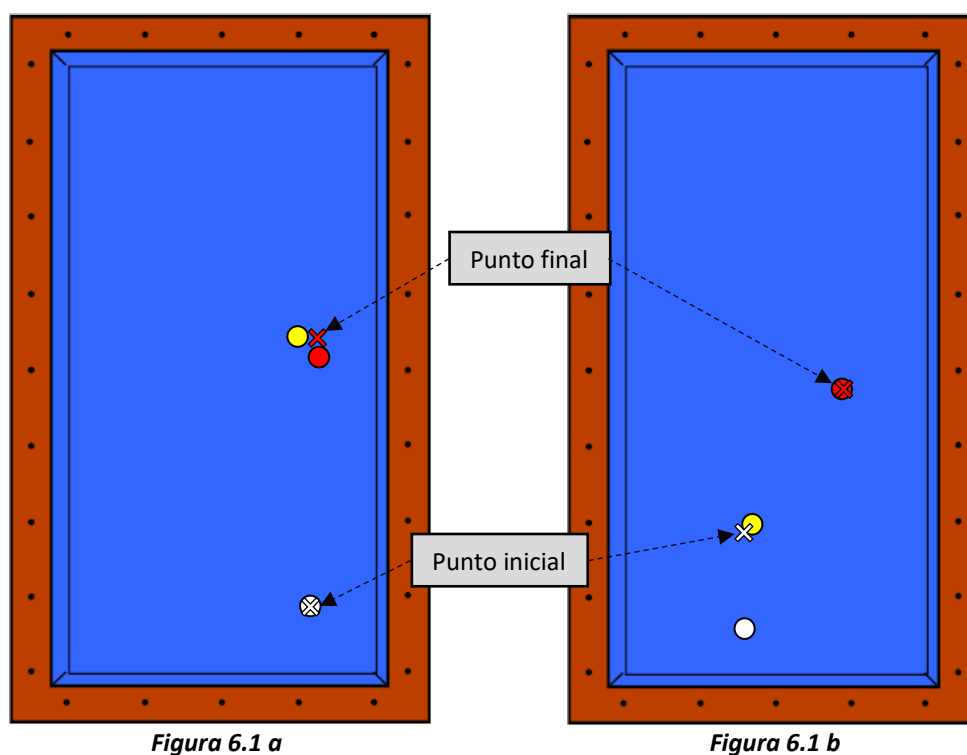


Figura 6.1 a
Figura 6.1 b
Figura 6.1 Ejemplo de puntos inicial y final en situaciones de partido

Como se puede comprobar en la *Figura 6.1 a*, el punto final se sitúa en un punto intermedio entre las dos bolas objetivo, ya que, de llegar a ese punto la bola jugadora, tras completar la secuencia de bandas, se conseguirá completar la carambola. En el caso de la *Figura 6.1 b*, se observa que el punto inicial se sitúa en las proximidades de la primera bola objetivo, esto es debido a que la bola jugadora, tras chocar con esa primera bola objetivo, verá modificada su dirección, pero dicho punto estará contenido en la nueva dirección adquirida por la bola jugadora. La posición exacta del punto inicial es el punto que dista la distancia del diámetro de una bola (61,5 mm), medida desde el centro de la primera bola objetivo y en una dirección perpendicular a la nueva dirección que adquiera la bola jugadora tras el choque con esa primera bola objetivo.

Una vez se conocen tanto los puntos tanto inicial como final, se identifican sus coordenadas sobre la mesa, se introducen en el entrenador virtual y se resuelve el

problema de acuerdo con la secuencia de bandas mostrada en la *Figura 2.1* de sentido horario con banda superior. En la *Figura 6.2* se muestran dichos resultados.

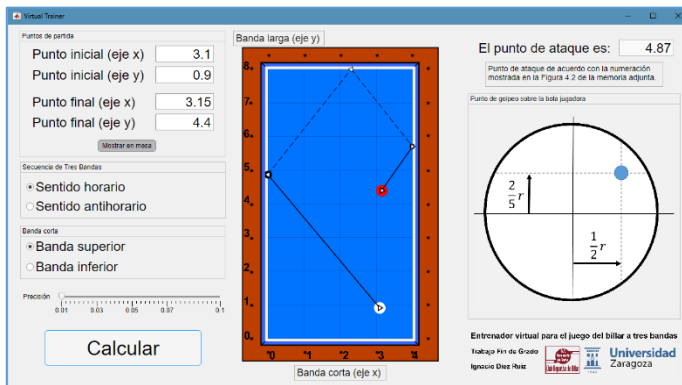


Figura 6.2 a

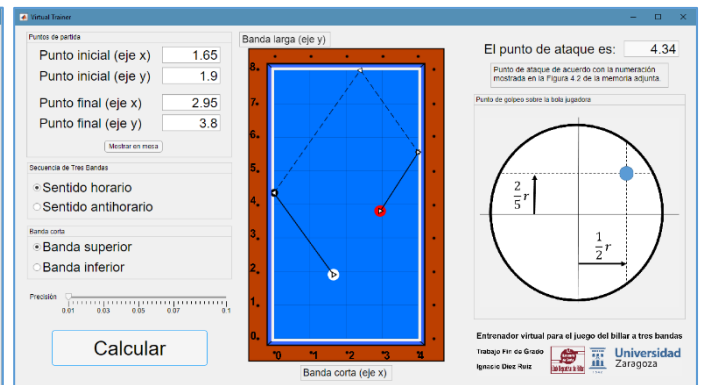


Figura 6.2 b

Figura 6.2 Resultado del cálculo de la acción óptima de los puntos de la *Figura 6.1*

En ambos casos mostrados en la *Figura 6.2* el entrenador virtual devuelve al billarista el punto de ataque, al igual que información sobre la altura y el efecto con la que el jugador deberá golpear a la bola jugadora. Con esta información el tresbandista únicamente tendrá que ejecutar el tiro conforme a estas premisas y a las expuestas en el **Anexo I** de este documento para conseguir la carambola de manera satisfactoria.

Se puede observar que las soluciones propuestas anteriormente para las posiciones de partida mostradas en la *Figura 6.1* no son únicas, sino que se puede llegar a completar la carambola en ambos casos utilizando otras secuencias permitidas por el entrenador virtual, tal y como se detalla en la *Figura 6.3*.

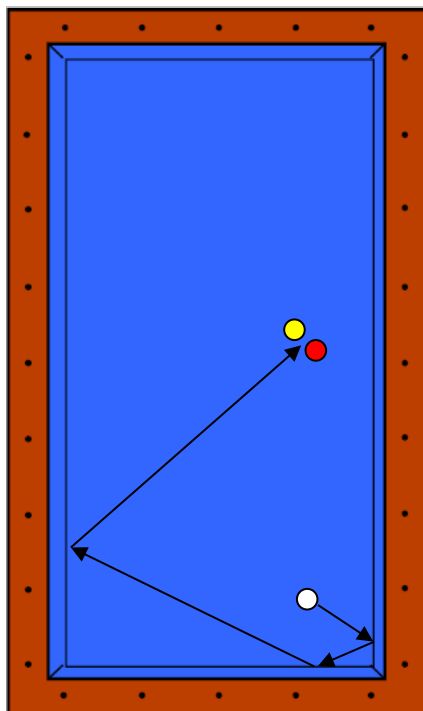


Figura 6.3 a

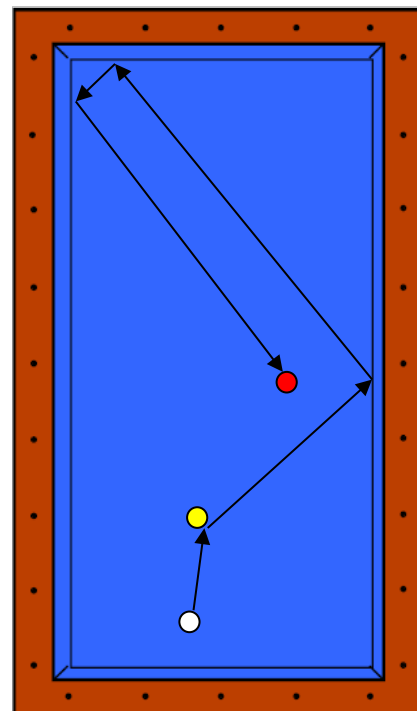


Figura 6.3 b

Figura 6.3 Propuesta de soluciones alternativas para la *Figura 6.1*

Para el cálculo de la acción óptima de acuerdo a la *Figura 6.3*, se deberá modificar ligeramente la posición de los puntos inicial y final detallados en la *Figura 6.1*, tal y como se expone en la *Figura 6.4*.

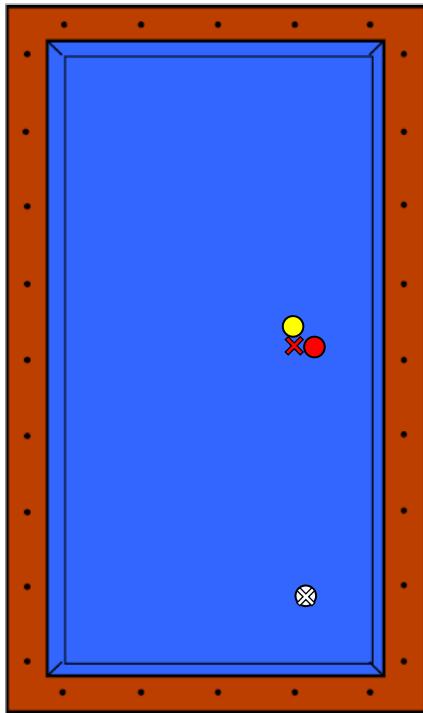


Figura 6.4 a

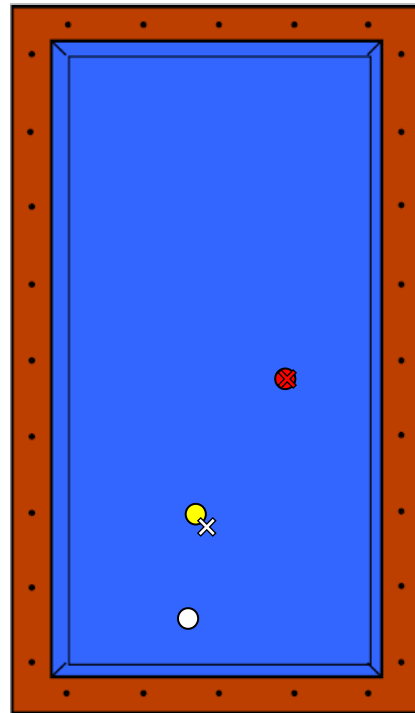


Figura 6.4 b

Figura 6.4 Ajuste de la posición de los puntos inicial y final de acuerdo con la *Figura*

Una vez conocidos los nuevos puntos inicial y final para cada caso se puede resolver el problema de acuerdo con la secuencia seleccionada en cada caso, tal y como se detalla en la *Figura 6.5*.

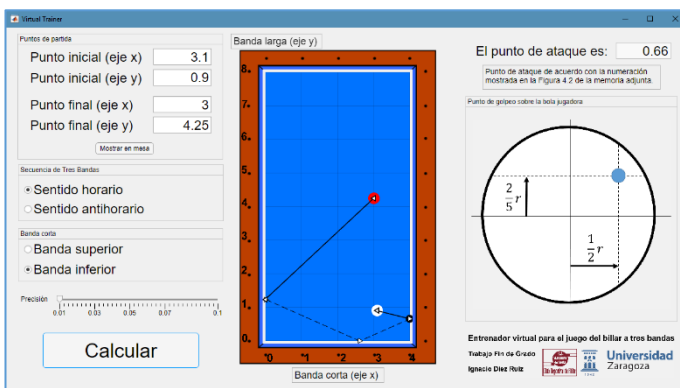


Figura 6.5 a

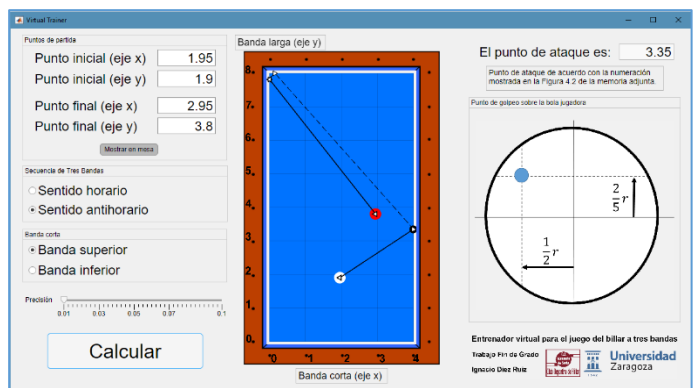


Figura 6.5 b

Figura 6.5 Resultado del cálculo de la acción óptima de los puntos de la *Figura 6.4*

En el **Anexo VI** de este documento se muestra el enlace a diferentes vídeos elaborados demostrando el funcionamiento tanto del modelo generado como de que el programa final cumple con su objetivo.

7. Conclusiones

7.1 Validez en la puesta en marcha del programa final

Una vez diseñado el programa final se puso a disposición de los jugadores del Club de Billar C.A. Numancia en varios entrenamientos de billar a tres bandas.

A lo largo de las pruebas realizadas se comprobó que el modelo generado se aproximaba con gran exactitud a la realidad, debido a que al utilizar las soluciones propuestas por el entrenador virtual, el jugador conseguía realizar la carambola de manera satisfactoria. Para ello resulta imprescindible que el jugador ejecute el golpeo exactamente como el entrenador virtual le comunique, y de acuerdo a las características del golpeo detalladas en el **Anexo I**. Se ha detectado que los jugadores expertos necesitaron varios tiros para asimilar y ejecutar correctamente todos los golpes de acuerdo a estos conceptos, en donde se detectó una evolución en la eficacia de sus tiros conforme avanzaba el entrenamiento. Los primeros tiros ejecutados por parte de la mayoría de los jugadores que utilizaron el entrenador virtual fueron fallados, pero a medida que asimilaban los conceptos de dirección, efecto, altura, velocidad, inclinación y penetración con los que deberían ejecutar los tiros la eficacia de los mismos aumentó, llegando algunos de ellos a conseguir completar prácticamente la totalidad de carambolas conforme a la acción óptima calculada por el entrenador virtual.

Aun así, durante los entrenamientos se dieron algunas configuraciones para las que no se podía utilizar el programa desarrollado, ya que no existía ninguna secuencia de banda larga/corta/larga que posibilitara la realización de la carambola, siendo necesario que el jugador ideara otra solución al margen del programa para conseguir realizar la carambola.

También se detectó que, en las configuraciones en donde se obliga a que la bola jugadora choque con las bandas cerca de los rincones (a una distancia menor a 0,2 rombos aproximadamente), el modelo generado se vuelve más impreciso. Esto puede ser debido a que se estará obligando a que la bola jugadora contacte en breve espacio de tiempo con dos bandas, impidiendo de esta forma que la bola jugadora alcance un estado de rodadura estacionario entre banda y banda, desvirtuando ligeramente las soluciones propuestas del entrenador virtual generadas con el modelo que posee. Si bien, en la mayoría de estas situaciones, la carambola se ha seguido completando de manera satisfactoria, pero en vez de llegar al punto final deseado se observaba una pequeña desviación. Cabe recordar que el diámetro de una bola de billar es de 61,5 mm, por lo que, en la mayoría de los casos, el margen de error para poder completar una carambola con éxito es de unos 12 cm, el equivalente a 0,34 rombos.

Con todo lo anterior, se recomienda que el billarista entrene previamente la morfología de su golpeo, para adaptarla a las exigencias del entrenador virtual, y una vez utilice el

programa final, evite en la medida de lo posible las configuraciones que obliguen a la bola jugadora a golpear cerca de los rincones mientras recorre la secuencia de bandas.

7.2 Metodología utilizada

A lo largo de este proyecto, y para conseguir completar el objetivo final del mismo, se ha utilizado la herramienta de software matemático Matlab® (en su versión R2018a). Gracias a esta herramienta se ha podido resolver tanto la generación del modelo necesario mediante el análisis de datos procedentes de tiros de jugadores expertos, como el cálculo de la acción óptima y el desarrollo de una interfaz final que permitiera la comunicación billarista-programa.

Al principio del proyecto se adelantaba que, para modelar el juego del billar a tres bandas, se utilizarían herramientas de aprendizaje automático. Después de valorar algunas de ellas, tales como árboles de decisión, regresiones logísticas o redes neuronales, y de acuerdo con la naturaleza del problema y de la base de datos que se había elaborado, se decidió utilizar la aplicación *Curve Fitting Tool*, incorporada en Matlab®, con la que se consiguió un modelo que se asemejaba en gran medida a la realidad, por lo que se dio por válido y se continuó con el proyecto.

En relación con las diversas simplificaciones que se han llevado a cabo para terminar abordando el problema, puede llegar a surgir la duda de la posibilidad de llegar a conseguir el modelo a través de conceptos trigonométricos, pero esta solución quedó descartada inicialmente por diversos motivos. El hecho fundamental por el que no sea recomendable tratar el problema de esta forma es debido a la utilización de un efecto diferente al nulo, como se aconseja en el estudio de *Mecánica de una rodada en el billar a tres bandas*. Esto produce que la bola, al chocar con rotación contra una banda, no se comporte de forma trigonométrica. Cuantificar esta rotación en términos trigonométricos (o mecánicos) conllevaría el error de no considerar la pérdida de rotación de la bola a lo largo de su trayectoria, y de considerarse no dejaría de ser una aproximación de la realidad, y podría conllevar diversos errores finales, como ya se observó en el estudio anterior de Joris van Balen.

7.3 Conclusión final del autor

Una vez finalizado el proyecto es posible comprobar que se ha cumplido el objetivo de manera satisfactoria, habiendo sido logrados todos los requisitos del proyecto.

Aun con todo ello, no hay que olvidar las limitaciones que el programa final tiene, siendo la principal de ellas el hecho de que únicamente se pueda utilizar para configuraciones en donde la bola jugadora recorra la secuencia de banda larga, banda corta para terminar en la banda larga opuesta a la primera. Es cierto que esta es la configuración más repetida en el billar a tres bandas, pero existen ocasiones en donde es imposible

resolver la carambola con esta secuencia, por lo que el entrenador virtual diseñado no serviría.

La principal mejora que se le puede aportar, por lo tanto, al programa diseñado es el convertirlo en un programa capaz de resolver cualquier secuencia de tiro, para poder de esta manera conseguir la carambola para cualquier posición de las bolas sobre la mesa.

Como trabajos futuros se podría investigar con la incorporación de una cámara cenital sobre la mesa, que recoja y procese continuamente la posición y el movimiento de las bolas a lo largo de un partido de billar, unido con diversos sensores incluidos en el taco, que recojan información para cada tiro de los seis parámetros del golpeo, y conseguir de esta manera que el propio programa vaya aprendiendo la totalidad del juego del billar conforme recibe toda la información procedente de cada tiro ejecutado, pudiendo de esta manera construir un modelo mucho más completo que el realizado, y por lo tanto permitiendo desarrollar un programa de mayor utilidad y con mayor variedad de posibilidades.

En relación al aprendizaje experimentado a lo largo de la creación y desarrollo de este proyecto, cabe destacar, y dejando a un lado la aplicación final desarrollada, el trayecto seguido a lo largo del mismo, en donde se han tenido que solucionar problemas que han ido surgiendo a lo largo del propio proyecto, del mismo modo que se han tenido que utilizar aplicaciones y programas de los que no se disponía una experiencia especializada, o simplemente se desconocía su existencia (como es el caso de la aplicación App Designer).

Por todo ello, y para conseguir un manejo preciso de los mismos, se requiere un gran periodo de tiempo y esfuerzo, que muchas veces no es tangible en el resultado final, pero sin el que sería imposible llegar al resultado final. El aprendizaje, por lo tanto, adquirido por el autor a lo largo de estos proyectos no es únicamente profesional, sino que sirve para demostrarse a sí mismo la capacidad de resolver los diversos problemas que han ido surgiendo, haciendo uso de todas las competencias que ha adquirido a lo largo de su etapa formativa.

8. Bibliografía

- World Confederation of Billiards Sport. WCBS, *History*. Sint-Martens-Latem (Bélgica). [Internet]. 2015 [Consultado 7 Ago 2018].
Disponible en: <https://www.wcbs-billiards.org/history/>
- Joris van Balen, *A Virtual Billiard Assistant*. [Internet]. Twente (Países Bajos): University of Twente, 2009 [Consultado 7 Ago 2018]
Disponible en: https://essay.utwente.nl/59134/1/scriptie_J_van_Balen.pdf
- Lars Bo Larsen, Peter M. Jensen, Kenneth Kammergaard, Lars Kromann, *The Automated Pool Trainer - a Multi Modal System for Learning the Game of Pool*. [Internet]. Aalborg (Dinamarca): Aalborg University, 2001 [Consultado 7 Ago 2018]
Disponible en:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.137.982>
- Michael Smith, *Running the Table: An AI for Computer Billiards*. [Internet]. Edmonton (Canadá): Department of Computing Science, University of Alberta, 2006 [Consultado 7 Ago 2018]
Disponible en: <https://www.aaai.org/Papers/AAAI/2006/AAAI06-156.pdf>
- Jens-Uwe Bahr, *A computer player for billiards based on artificial intelligence techniques*. [Internet]. Kiel (Alemania): Universität Kiel, 2012 [Consultado 7 Ago 2018]
Disponible en: <https://www.informatik.uni-kiel.de/~mh/lehre/abschlussarbeiten/bsc/Bahr.pdf>
- Katerina Fragkiadaki, Pulkit Agarwal, Sergey Levine, Jitendra Malik, *Learning visual predictive models of physics for playing billiards*. [Internet]. Berkeley, California (EE.UU.): University of California, Berkeley, 2016 [Consultado 7 Ago 2018]
Disponible en:
<https://pdfs.semanticscholar.org/f208/7f10a582c63f075dfb72da8c801ddf94c5f7.pdf>
- Ignacio Díez Ruiz, *Mecánica de una rodada en el billar a tres bandas*. Beau Bassin-Rose Hill (República de Mauricio): Editorial Académica Española, 2017

ANEXO I

Parámetros del golpeo en billar

Como se ha mencionado a lo largo de la memoria, en cada tiro de billar el jugador puede actuar sobre seis parámetros diferentes para modificar las características del golpeo sobre la bola jugadora. Estos seis parámetros tienen que estar bien definidos y caracterizados, porque la modificación de cualquiera de ellos, ya sea a lo largo de la toma de datos o en el uso final del entrenador virtual, conduciría a errores al no corresponderse el modelo desarrollado con la realidad.

Los seis parámetros del golpeo son los siguientes:

A1.1 La dirección

El primer parámetro de golpeo es el de la dirección que lleva el taco en el momento del golpeo (incluyendo en el término dirección tanto la dirección como sentido), que determinará la dirección con la que sale despedida la bola jugadora tras el tiro. Esta dirección se puede observar en la *Figura 1.2* de la memoria, en donde se puede ver representada en la primera flecha que dibuja la trayectoria de la bola jugadora.

Si distinguimos entre las dos secuencias de tiro con las que trabaja el entrenador virtual, encontramos la secuencia en la que la bola jugadora golpea primero las tres bandas antes de chocar con las dos bolas objetivo (que se denominada con el nombre de *bricol* o *lujo*¹⁰), y la secuencia en la que la bola jugadora golpea primero una de las bolas objetivo para chocar después con la secuencia de bandas y terminar en la segunda bola objetivo. En el primer caso (*Figura A1.1 a*) la dirección con la que sale despedida la bola jugadora tras el golpeo ha de contener directamente al punto de la primera banda larga sobre el que se ha de atacar para que se satisfaga la carambola, siendo éste punto deducido por el entrenador virtual. En el segundo caso (*Figura A1.1 b*), la dirección que tome la bola jugadora determinará la toma de bola (punto de contacto entre la bola jugadora y la primera bola objetivo), y ésta toma de bola será la que determine la posterior dirección que llevará la bola jugadora tras el choque con la primera bola objetivo, y que deberá contener al punto de la primera banda larga deducido por el entrenador virtual que complete la carambola.

¹⁰ En billar, un bricol o un lujo hacen referencia a un tiro en donde la bola jugadora primeramente contacta con un número determinado de bandas y posteriormente con las bolas objetivo.

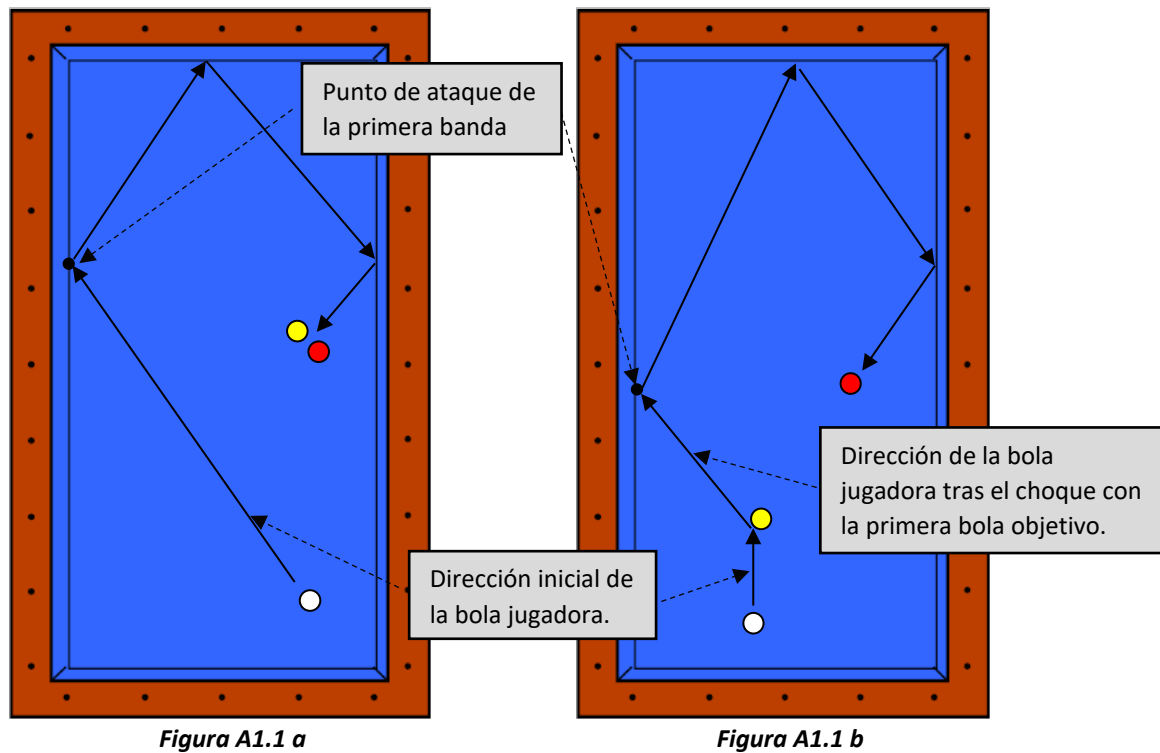


Figura A1.1 Dirección inicial de la bola jugadora

A1.2 La velocidad

En el billar se entiende este concepto como la velocidad de traslación inicial que se imprime a la bola en el momento del golpeo, considerando la bola como una masa puntual concentrada en el centro de la misma. El comportamiento que tiene la bola tras chocar con las bandas es diferente en función de la velocidad que lleve la bola, por lo que es un parámetro que se deberá controlar en todo momento para asegurar el perfecto funcionamiento del entrenador virtual.

El billarista no dispondrá habitualmente de ningún dispositivo que le permita cuantificar numéricamente el valor exacto de la velocidad inicial de la bola jugadora, pero sí que dispone de métodos cualitativos que le permiten conocer y entrenar este parámetro del juego, tal y como se define en *Mecánica de una rodada en el billar a tres bandas*¹¹, en donde se utilizan diferentes velocidades patrón para poder valorar la velocidad inicial con la que sale despedida la bola jugadora. Tal y como se cita en el estudio anteriormente mencionado, la velocidad óptima para realizar una carambola en el billar a tres bandas es la denominada velocidad media, y se caracteriza por ser la velocidad necesaria equivalente para que la bola jugadora recorra la longitud exacta del largo de

¹¹ Ignacio Díez Ruiz, *Mecánica de una rodada en el billar a tres bandas*. Beau Bassin-Rose Hill (República de Mauricio): Editorial Académica Española, 2017

tres mesas de billar antes de detenerse por el efecto del rozamiento entre la superficie o paño y la bola (*Figura A1.2*).

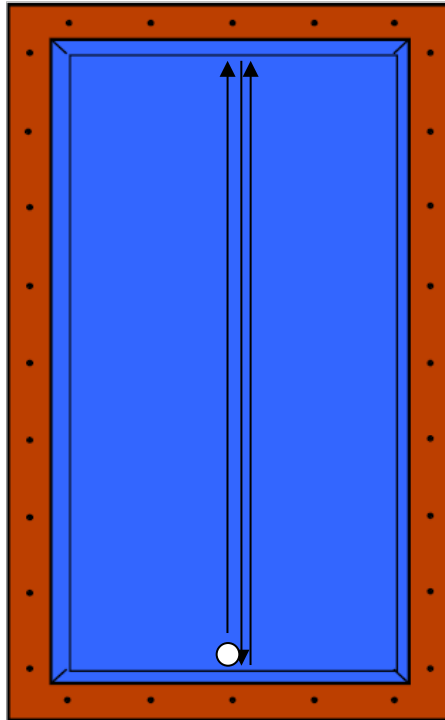


Figura A1.2 Recorrido de la bola con velocidad media.

Por ello, ésta será la velocidad con la que trabaje el entrenador virtual, por lo que el billarista deberá atacar a la bola con esta velocidad para conseguir realizar la carambola.

A1.3 La altura

Una bola de billar es una esfera perfecta de 61,5 mm de diámetro, por ello puede girar en torno a cualquier eje que pase por su centro. Para poder caracterizar dicha rotación, el giro total que experimente la bola se expresará como combinación de tres giros sobre tres ejes principales que pasan por el centro de la bola. Estos ejes principales serán los siguientes.

El primer eje a considerar será el eje paralelo a la dirección que lleve la bola jugadora en cada instante de la trayectoria (*Figura A1.3 a*). Para encontrar los dos ejes principales restantes, trazamos un plano perpendicular al primer eje por el centro de la bola (*Figura A1.3 b*) y distinguimos entre el eje vertical y el eje horizontal, que serán los ejes principales restantes para poder caracterizar cualquier giro que experimente la bola sobre su centro de gravedad.

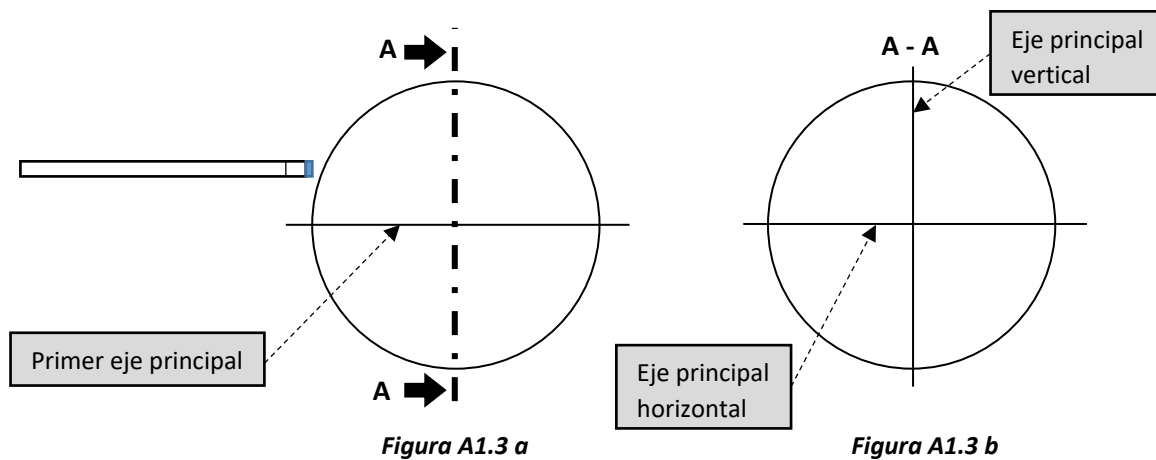


Figura A1.3 Ejes principales de la bola.

Teniendo en cuenta lo mostrado anteriormente, en billar, cuando se habla de altura de una bola, se está cuantificando la rotación que posee la bola respecto al eje horizontal. De tal forma, que si el jugador golpea a la bola con el taco por encima de este eje horizontal, la bola jugadora saldrá despedida con una rotación inicial con un sentido acorde al del movimiento de traslación de la bola. Si por el contrario, el jugador ataca a la bola por debajo de este eje, la bola inicialmente adquirirá una rotación opuesta al movimiento de traslación. Esta rotación es siempre transitoria, ya que la bola, al tener rozamiento con el paño, tenderá a alcanzar el estado de no deslizamiento, en el que el giro respecto a este eje vertical sea coherente con la velocidad de traslación que lleve (cumpliéndose que $\omega = v \cdot r$; siendo ω la velocidad angular de la bola respecto al eje horizontal, v la velocidad lineal de traslación de la bola y r el radio de la misma).

El comportamiento de la bola durante los choques con las bandas se puede ver alterado si la bola jugadora contacta con las bandas antes de haber superado este estado transitorio y alcanzado el estado estacionario, por lo que será importante asegurar que la bola jugadora, al contactar con la primera banda, lo haga habiendo alcanzado ya la condición de no deslizamiento.

El entrenador virtual estará diseñado y parametrizado sin tener en cuenta estos posibles estados transitorios, suponiendo que la bola, en el instante de contacto con la primera banda, ya ha alcanzado el estado estacionario.

En el caso de que el jugador no sea capaz de distinguir y de asegurar dicho requisito, se le aconseja atacar a la bola a una altura de $\frac{2}{5}r$ sobre el eje horizontal, ya que, y tal y como se detalla en el estudio de *Mecánica de una rodada en el billar a tres bandas*, con

ello se consigue que la bola adquiera una rotación inicial sobre el eje horizontal que coincide directamente con la rotación de no deslizamiento (*Figura A1.4*).

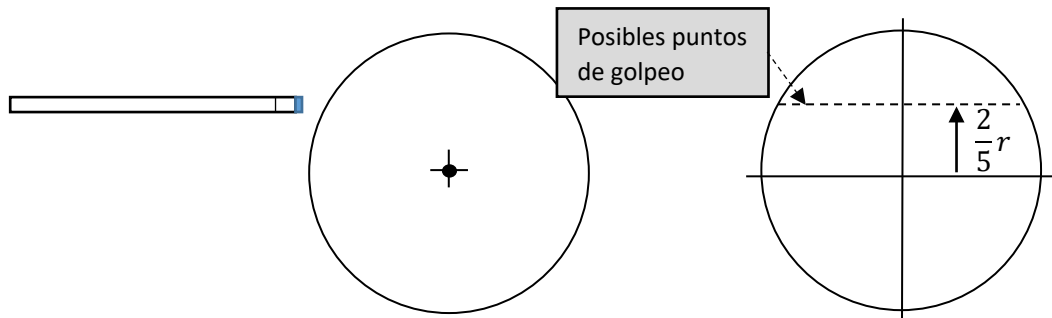


Figura A1.4 Altura de ataque para alcanzar directamente el estado estacionario.

A1.4 El efecto

Si tal y como se ha detallado anteriormente, la altura de una bola cuantifica la rotación de la misma respecto al eje horizontal, el efecto lo hace respecto al eje vertical (*Figura A1.3 b*). De esta forma, dependiendo de si el jugador golpea la bola jugadora por la derecha de este eje o por la izquierda, la bola adquirirá una rotación de derecha o de izquierda respectivamente, siendo mayor esta rotación cuanto más se aleje el punto de ataque a dicho eje.

Tal y como se demuestra en el estudio de *Mecánica de una rodada en el billar a tres bandas*, golpear con efecto favorable homogeniza la respuesta de la banda al contacto con la bola para diferentes ángulos de entrada, al igual que amplía el rango de figuras por las que se puede llegar a conseguir la secuencia de banda larga, banda corta y banda larga opuesta, por lo que el entrenador virtual será parametrizado partiendo de la base de que el billarista golpeará su bola con efecto favorable.

Únicamente queda entonces definir qué se entiende con efecto favorable. En todas las secuencias mostradas en la *Figura 1.2* de la memoria, la trayectoria de la bola jugadora golpea a las bandas en un sentido horario, y para éste caso, el efecto favorable es el efecto de derecha. En el caso de que se utilizara para completar la carambola una secuencia simétrica (en donde la bola jugadora recorriera la trayectoria en un sentido antihorario), el efecto a utilizar es el efecto de izquierda.

Para asegurar que la bola jugadora sale despedida siempre con la misma cantidad de efecto, se atacará a la bola jugadora a una distancia de $\frac{1}{2}r$ tanto a la derecha del eje vertical como a la izquierda, dependiendo de si se precisa de efecto de derecha o de izquierda.

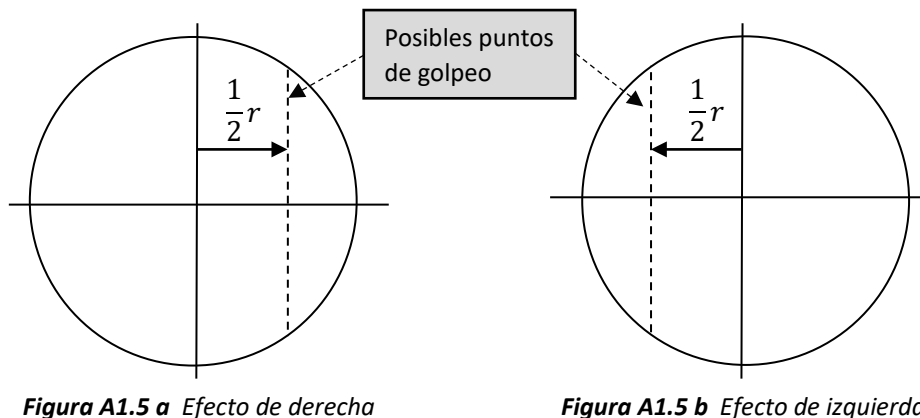


Figura A1.5 a Efecto de derecha

Figura A1.5 b Efecto de izquierda

Figura A1.5 Puntos de golpeo para efecto favorable.

En la *Figura A1.5* se muestran los segmentos que contienen todos los puntos con los que se consigue el efecto favorable, por lo que si estos puntos se combinan con los mostrados en la *Figura A1.4*, con los que se consigue que la bola inicie su recorrido habiendo alcanzado la condición de no deslizamiento, se deducen los dos puntos óptimos de golpeo (uno para conseguir un efecto favorable de derecha y otro para conseguir un efecto favorable de izquierda).

A1.5 La penetración

En billar la penetración determina el tiempo durante el que, en un golpeo, el taco de billar contacta con la bola jugadora. Este es un parámetro importante a tener en cuenta, ya que si el brazo del billarista golpea de forma progresiva a la bola acompañándola en su recorrido inicial, se potenciará la rotación con la que se desee golpear a la bola.

Por ello, y para no incurrir en errores, todos los tiros que se llevarán a cabo se ejecutarán con un golpe seco, es decir, en ausencia de penetración alguna por parte del billarista, quien detendrá el taco una vez que éste haya entrado en contacto con la bola jugadora.

A1.6 La inclinación

Este último parámetro de golpeo determina la inclinación del taco sobre el plano horizontal en el momento del tiro. Está directamente relacionado con el giro que experimenta la bola sobre el primer eje principal detallado en la *Figura A1.3 a*, pero rara vez el billarista precisa golpear con una inclinación no nula del taco, por lo que se impondrá que el taco permanecerá siempre horizontal en el golpeo a la bola jugadora, consiguiendo por lo tanto que la rotación que experimente la bola sobre su primer eje principal será siempre nula.

ANEXO II

Numeración de los puntos característicos

Tal y como se ha mencionado a lo largo de la memoria, el modelo deberá ser capaz de relacionar la dirección de entrada de la bola jugadora a la secuencia de bandas con la dirección de salida de la misma. Ambas direcciones se parametrizan con dos puntos característicos, que coinciden con el corte de cada dirección con las bandas de la mesa. De tal forma, la dirección de entrada está caracterizada por el punto de salida y el punto de ataque, mientras que la dirección de salida de la secuencia de bandas se parametriza por el denominado punto de llegada sobre la tercera banda y por el punto de llegada sobre la cuarta banda.

Pero para poder trabajar con estos valores es necesario definir un sistema de referencia para cada uno de ellos, que permita numerarlos e identificarlos de manera rápida sobre la mesa. Esta numeración se detalla para cada uno de los cuatro puntos característicos en la Figura A2.1, Figura A2.2 y Figura A2.3.

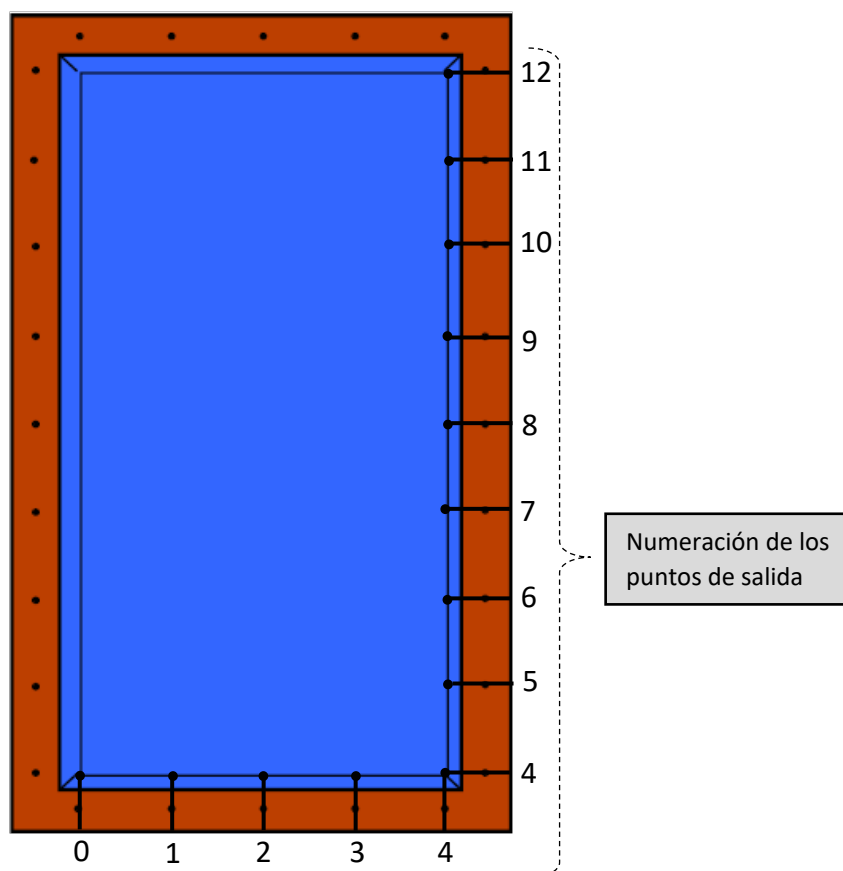


Figura A2.1 Sistema de referencia de los puntos de salida

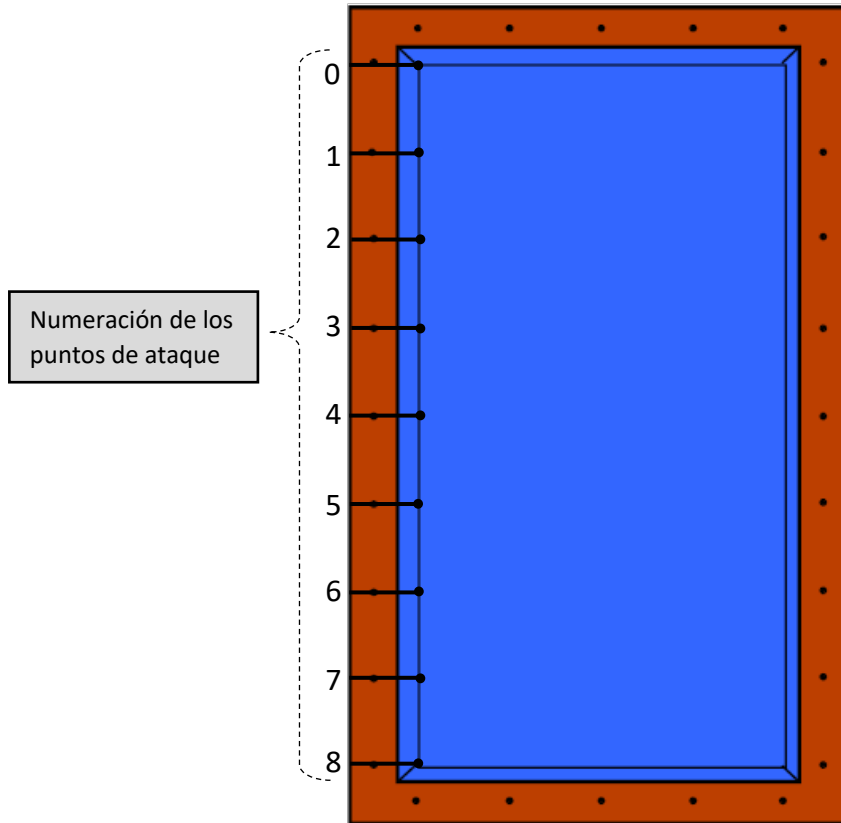


Figura A2.2 Sistema de referencia de los puntos de ataque

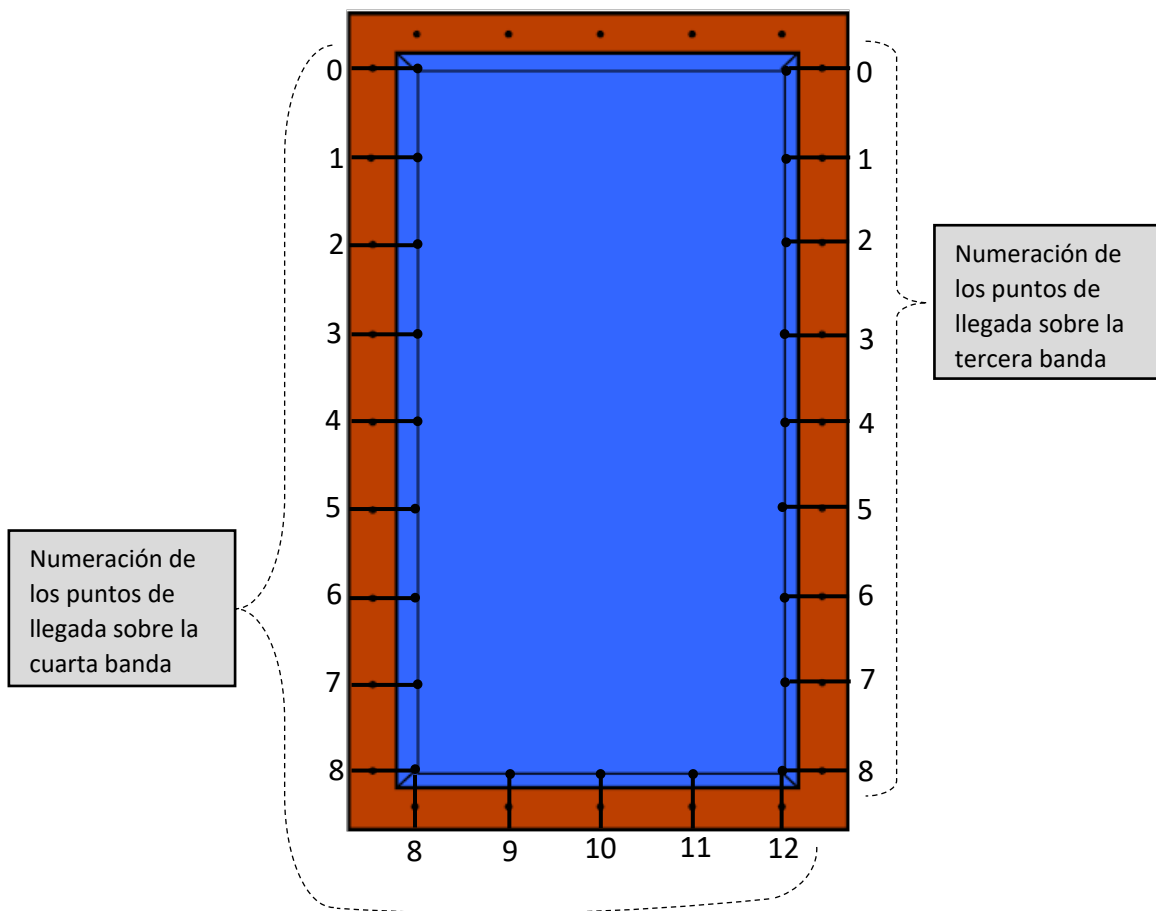


Figura A2.3 Sistema de referencia de los puntos de llegada

ANEXO III

Base de datos elaborada a partir de tiros de jugadores expertos

Para poder elaborar un modelo que relacione la dirección de entrada de la bola jugadora a la secuencia de bandas con la dirección de salida de dicha secuencia, es necesario, tal y como se ha detallado a lo largo de la memoria, conocer una base de datos que arroje información sobre estas direcciones en diversos tiros de billar realizados por jugadores expertos.

A continuación se muestran los resultados finales de las mediciones recogidas en una base de datos conforme a lo estipulado en el apartado 2. *Elaboración de una base de datos* de la memoria adjunta.

| | Punto de salida | Punto de ataque | Punto de llegada 3ª banda | Punto de llegada 4ª banda | | Punto de salida | Punto de ataque | Punto de llegada 3ª banda | Punto de llegada 4ª banda |
|----|-----------------|-----------------|---------------------------|---------------------------|----|-----------------|-----------------|---------------------------|---------------------------|
| 1 | 1,0 | 7,0 | 0,30 | 6,80 | 27 | 3,0 | 2,0 | 3,80 | 9,70 |
| 2 | 1,0 | 6,0 | 2,90 | 9,45 | 28 | 3,0 | 1,0 | 4,60 | 10,20 |
| 3 | 1,0 | 5,0 | 4,95 | 10,60 | 29 | 3,5 | 5,0 | 0,65 | 6,50 |
| 4 | 1,0 | 4,0 | 6,00 | 11,20 | 30 | 3,5 | 4,0 | 1,65 | 7,80 |
| 5 | 1,0 | 3,0 | 6,50 | 11,40 | 31 | 3,5 | 3,0 | 2,60 | 8,80 |
| 6 | 1,0 | 2,0 | 7,65 | 11,85 | 32 | 3,5 | 2,0 | 3,45 | 9,30 |
| 7 | 1,5 | 6,0 | 1,45 | 8,10 | 33 | 3,5 | 1,0 | 4,15 | 9,80 |
| 8 | 1,5 | 5,0 | 2,95 | 9,35 | 34 | 4,0 | 5,0 | 0,20 | 5,70 |
| 9 | 1,5 | 4,0 | 4,30 | 10,20 | 35 | 4,0 | 4,0 | 1,45 | 7,25 |
| 10 | 1,5 | 3,0 | 4,85 | 10,45 | 36 | 4,0 | 3,0 | 2,30 | 8,35 |
| 11 | 1,5 | 2,0 | 5,65 | 10,95 | 37 | 4,0 | 2,0 | 3,00 | 9,00 |
| 12 | 1,5 | 1,0 | 6,80 | 11,45 | 38 | 4,0 | 1,0 | 3,75 | 9,50 |
| 13 | 2,0 | 6,0 | 0,90 | 7,10 | 39 | 4,5 | 4,5 | 0,40 | 6,00 |
| 14 | 2,0 | 5,0 | 2,10 | 8,60 | 40 | 4,5 | 4,0 | 1,10 | 6,50 |
| 15 | 2,0 | 4,0 | 3,00 | 9,25 | 41 | 4,5 | 3,5 | 1,55 | 7,25 |
| 16 | 2,0 | 3,0 | 4,15 | 10,10 | 42 | 4,5 | 3,0 | 2,00 | 7,95 |
| 17 | 2,0 | 2,0 | 4,90 | 10,55 | 43 | 4,5 | 2,5 | 2,35 | 8,35 |
| 18 | 2,0 | 1,0 | 5,70 | 11,05 | 44 | 4,5 | 2,0 | 2,85 | 8,95 |
| 19 | 2,5 | 5,0 | 1,50 | 8,05 | 45 | 4,5 | 1,5 | 3,30 | 9,20 |
| 20 | 2,5 | 4,0 | 2,55 | 8,90 | 46 | 4,5 | 1,0 | 3,60 | 9,45 |
| 21 | 2,5 | 3,0 | 3,35 | 9,45 | 47 | 5,0 | 4,5 | 0,05 | 5,10 |
| 22 | 2,5 | 2,0 | 4,25 | 10,00 | 48 | 5,0 | 4,0 | 0,85 | 6,05 |
| 23 | 2,5 | 1,0 | 4,95 | 10,50 | 49 | 5,0 | 3,5 | 1,30 | 6,90 |
| 24 | 3,0 | 5,0 | 1,05 | 7,00 | 50 | 5,0 | 3,0 | 1,75 | 7,35 |
| 25 | 3,0 | 4,0 | 2,10 | 8,50 | 51 | 5,0 | 2,5 | 2,15 | 8,05 |
| 26 | 3,0 | 3,0 | 3,10 | 9,15 | 52 | 5,0 | 2,0 | 2,70 | 8,50 |

| | Punto de salida | Punto de ataque | Punto de llegada 3ª banda | Punto de llegada 4ª banda | | Punto de salida | Punto de ataque | Punto de llegada 3ª banda | Punto de llegada 4ª banda |
|----|-----------------|-----------------|---------------------------|---------------------------|----|-----------------|-----------------|---------------------------|---------------------------|
| 53 | 5,0 | 1,5 | 3,05 | 8,95 | 68 | 8,0 | 2,0 | 1,25 | 6,45 |
| 54 | 5,0 | 1,0 | 3,40 | 9,20 | 69 | 8,0 | 1,5 | 1,70 | 6,85 |
| 55 | 6,0 | 3,5 | 0,60 | 5,70 | 70 | 8,0 | 1,0 | 2,45 | 7,80 |
| 56 | 6,0 | 3,0 | 1,20 | 6,50 | 71 | 8,0 | 0,5 | 2,85 | 8,45 |
| 57 | 6,0 | 2,5 | 1,65 | 7,05 | 72 | 9,0 | 2,0 | 0,30 | 4,20 |
| 58 | 6,0 | 2,0 | 2,15 | 7,80 | 73 | 9,0 | 1,5 | 0,90 | 5,15 |
| 59 | 6,0 | 1,5 | 2,65 | 8,40 | 74 | 9,0 | 1,0 | 1,55 | 6,00 |
| 60 | 6,0 | 1,0 | 2,90 | 8,65 | 75 | 9,0 | 0,5 | 2,35 | 7,50 |
| 61 | 7,0 | 3,0 | 0,90 | 5,80 | 76 | 10,0 | 1,5 | 0,20 | 3,20 |
| 62 | 7,0 | 2,5 | 1,35 | 6,55 | 77 | 10,0 | 1,0 | 0,85 | 4,55 |
| 63 | 7,0 | 2,0 | 1,80 | 7,20 | 78 | 10,0 | 0,5 | 1,75 | 6,25 |
| 64 | 7,0 | 1,5 | 2,15 | 7,70 | 79 | 10,5 | 1,0 | 0,40 | 3,20 |
| 65 | 7,0 | 1,0 | 2,65 | 8,35 | 80 | 10,5 | 0,5 | 1,35 | 5,50 |
| 66 | 7,0 | 0,5 | 3,00 | 8,70 | 81 | 11,0 | 1,0 | 0,20 | 2,20 |
| 67 | 8,0 | 2,5 | 0,60 | 5,30 | 82 | 11,0 | 0,5 | 0,90 | 4,05 |

Tabla A3.1 Base de datos

Cabe mencionar que cada una de las 82 medidas realizadas ha sido repetida varias veces y por diferentes jugadores antes de ser incluida en la *Tabla A3.1*, anotándose finalmente el resultado más repetido para cada medida. De esta manera, se consigue evitar cualquier posible error accidental que se haya podido cometer a la hora de ejecutar cualquier tiro durante la toma de los datos, y que el modelo que se genere con los mismos pueda verse afectado por ello.

Todas las medidas mostradas en la *Tabla A3.1* están tomadas de manera coherente con la numeración de la mesa expuesta en el **Anexo II** de este documento.

ANEXO IV

Modelo final de la secuencia de banda larga/corta/larga

Como se ha detallado a lo largo de la memoria de este proyecto, el modelo final que utilizará el entrenador virtual constará de la conjunción de dos semi-modelos independientes. Cada uno de ellos tendrá como entradas los puntos de salida y de ataque, pero uno de ellos devolverá la información sobre la llegada de la bola jugadora sobre la tercera banda (semi-modelo 1), mientras que el otro lo hará de la llegada sobre la cuarta banda (semi-modelo 2). Con esta información el programa final únicamente tendrá que construir la dirección que une los puntos devueltos por ambos semi-modelos para conseguir la dirección que adquirirá la bola jugadora tras completar la secuencia de bandas.

Para la elaboración de los semi-modelos mencionados se ha utilizado la aplicación incluida en Matlab® (versión R2018a) *Curve Fitting Tool*. Dichos semi-modelos pueden ser representados de acuerdo con los siguientes gráficos.

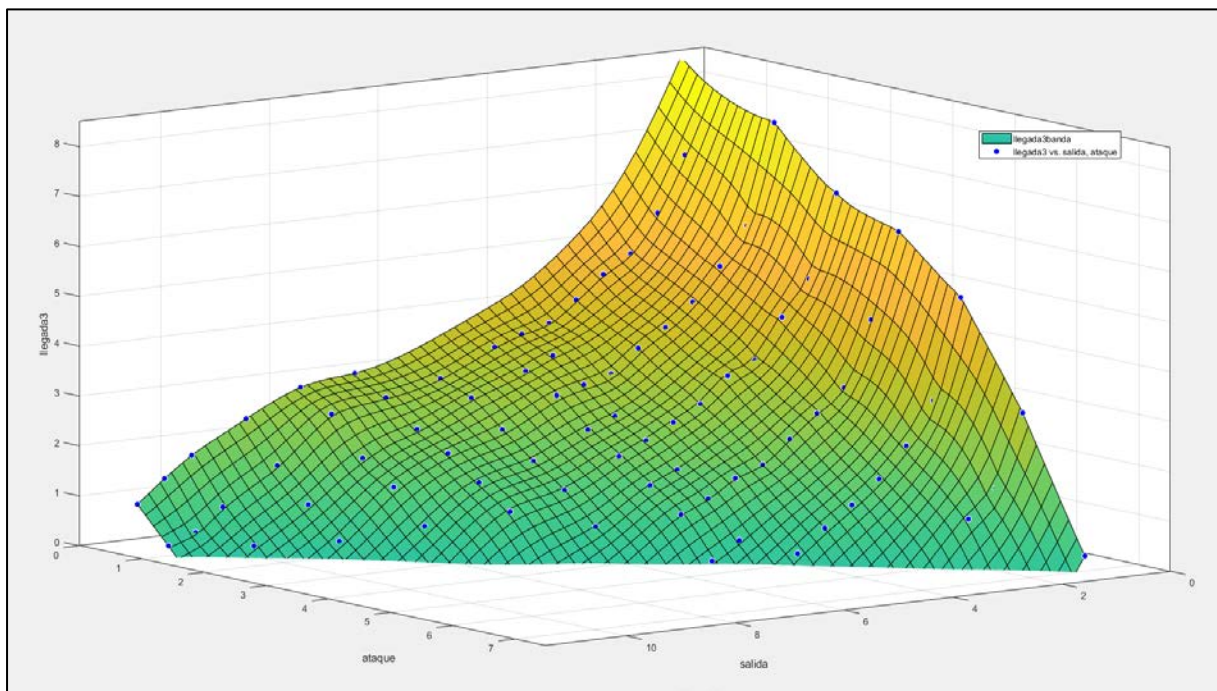


Gráfico A4.1 Semi-modelo 1 (vista 1)

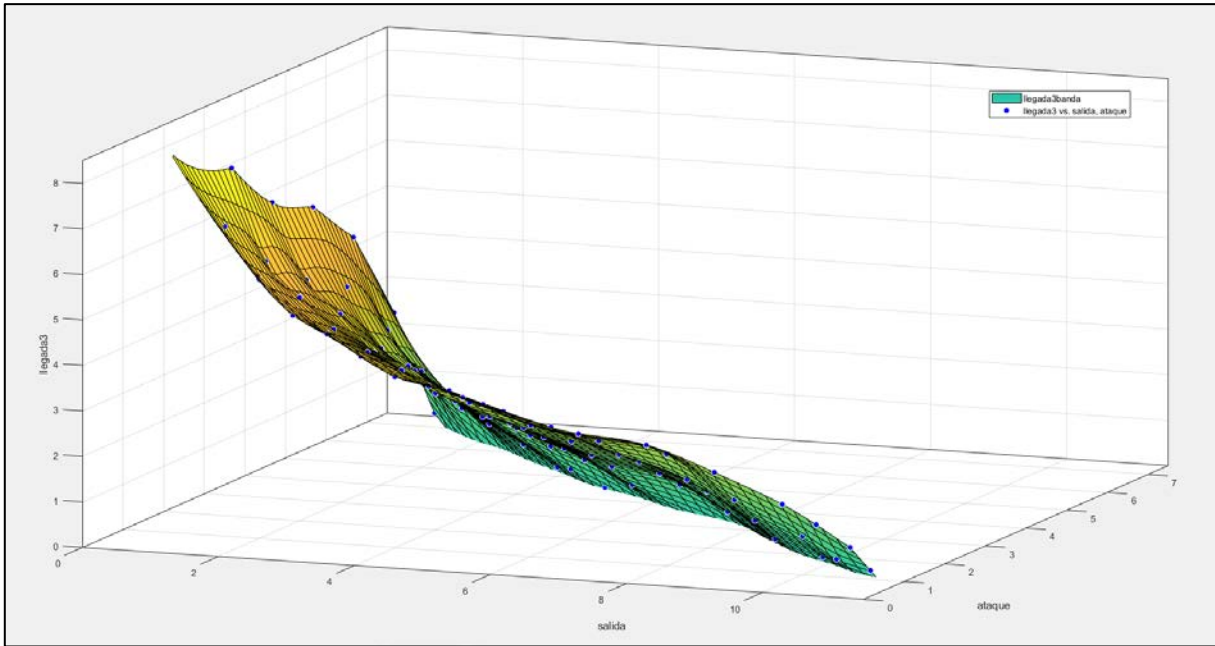


Gráfico A4.2 Semi-modelo 1 (vista 2)

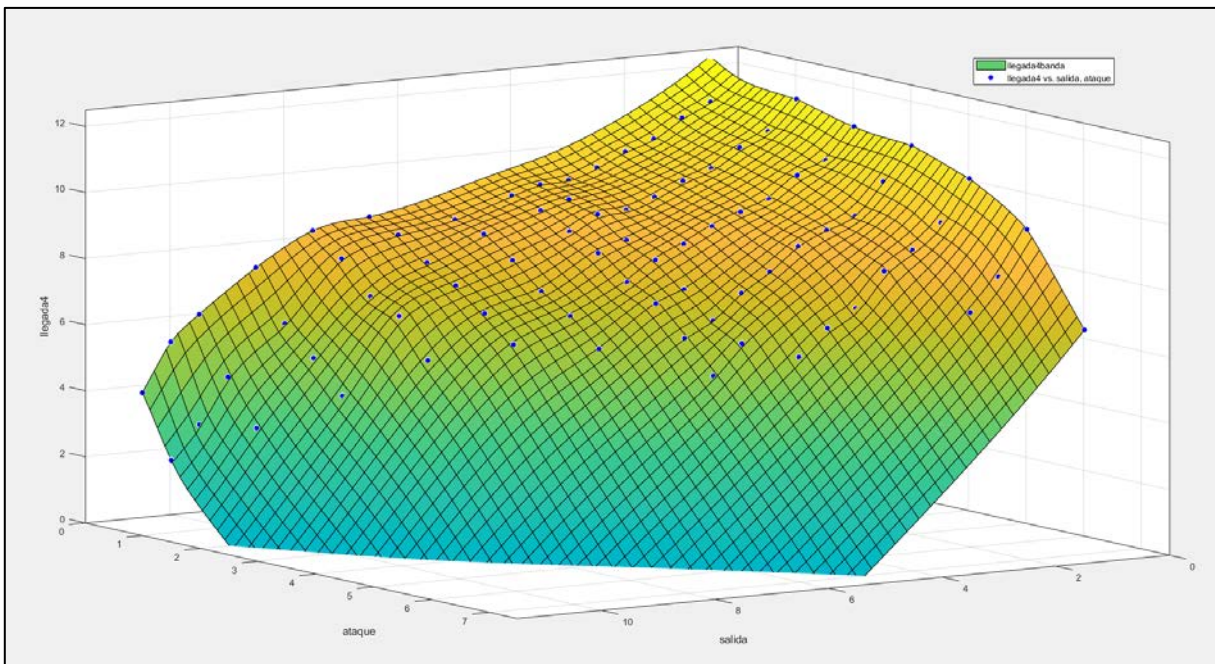


Gráfico A4.3 Semi-modelo 2 (vista 1)

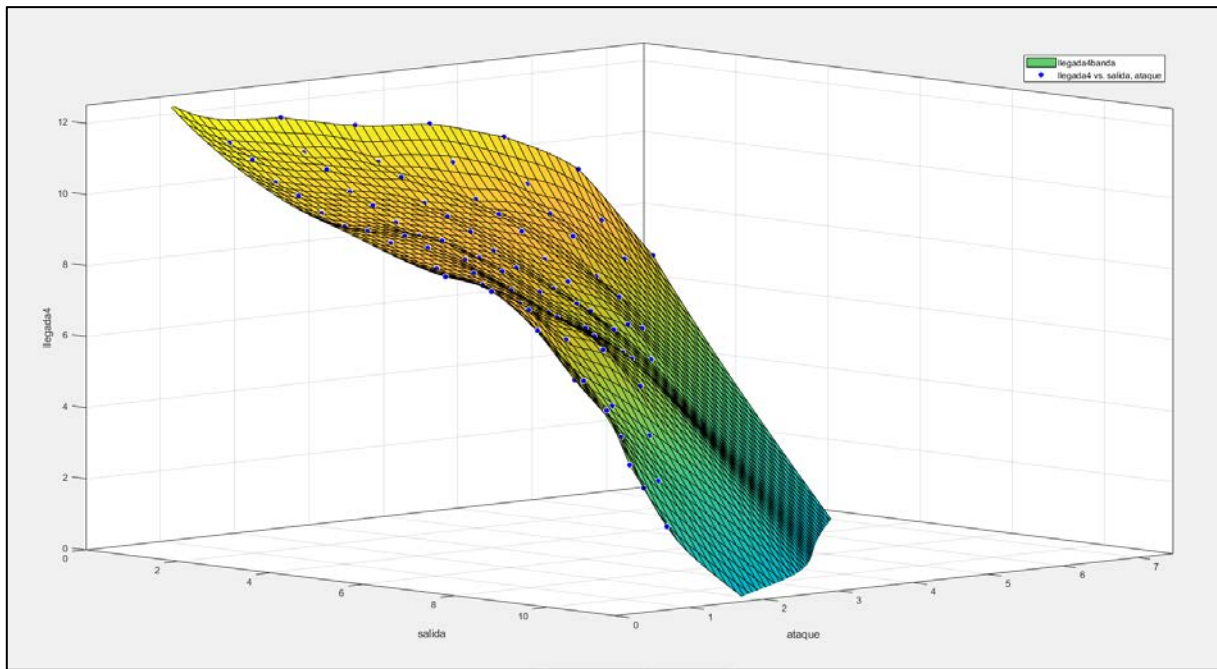


Gráfico A4.4 Semi-modelo 2 (vista 2)

Ambos semi-modelos han sido contruidos forzando a cada uno de ellos a pasar por todos los datos tomados y mostrados en el **Anexo III** de este documento. Con ello, y gracias a la fiabilidad que ofrece el haber repetido varias veces cada medida, se consigue un modelo final de gran precisión, como se demuestra en los últimos capítulos de esta memoria.

ANEXO V

Código del programa final

A continuación se muestra el código del programa final, compatible con la aplicación App Designer, incorporada en Matlab® R2018a.

```
classdef VirtualTrainerApp < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        VirtualTrainerUIFigure          matlab.ui.Figure
        CalcularButton                   matlab.ui.control.Button
        PuntosdepartidaPanel             matlab.ui.container.Panel
        PuntoinicialejexEditFieldLabel  matlab.ui.control.Label
        PuntoinicialejexEditField       matlab.ui.control.NumericEditField
        PuntoinicialejeyEditFieldLabel  matlab.ui.control.Label
        PuntoinicialejeyEditField       matlab.ui.control.NumericEditField
        PuntofinalejexEditFieldLabel    matlab.ui.control.Label
        PuntofinalejexEditField         matlab.ui.control.NumericEditField
        PuntofinalejeyEditFieldLabel    matlab.ui.control.Label
        PuntofinalejeyEditField         matlab.ui.control.NumericEditField
        MostrarenmesaButton              matlab.ui.control.StateButton
        ElpuntodeataqueesEditFieldLabel matlab.ui.control.Label
        ElpuntodeataqueesEditField      matlab.ui.control.NumericEditField
        SecuenciadeTresBandasButtonGroup matlab.ui.container.ButtonGroup
        SentidohorarioButton              matlab.ui.control.RadioButton
        SentidoantihorarioButton         matlab.ui.control.RadioButton
        PrecisinSliderLabel               matlab.ui.control.Label
        PrecisinSlider                   matlab.ui.control.Slider
        TextArea                          matlab.ui.control.TextArea
        PuntodegolpeosobrelabolajugadoraPanel matlab.ui.container.Panel
        UIAxes2                           matlab.ui.control.UIAxes
        UIAxes3                           matlab.ui.control.UIAxes
        UIAxes4                           matlab.ui.control.UIAxes
        TextArea_2                        matlab.ui.control.TextArea
        TextArea_3                        matlab.ui.control.TextArea
        UIAxes5                           matlab.ui.control.UIAxes
        BandacortaButtonGroup             matlab.ui.container.ButtonGroup
        BandasuperiorButton               matlab.ui.control.RadioButton
        BandainferiorButton               matlab.ui.control.RadioButton
    end
    methods (Access = private)

end
methods (Access = private)
    % Code that executes after component creation
    function startupFcn(app)
        % Remove title, axis labels, and tick labels
```

```

%uistack(app.UIAxes,top);
title(app.UIAxes3, []);
xlabel(app.UIAxes3, []);
ylabel(app.UIAxes3, []);
app.UIAxes3.XAxis.TickLabels = {};
app.UIAxes3.YAxis.TickLabels = {};
% Display image and stretch to fill axes
I = imshow('mesa2.png', 'Parent', app.UIAxes3,'XData', [0 5.5],'YData', [0 8.79]);
% Set limits of axes
app.UIAxes3.XLim = [0 5.5];
app.UIAxes3.YLim = [0 8.79];

title(app.UIAxes5, []);
xlabel(app.UIAxes5, []);
ylabel(app.UIAxes5, []);
app.UIAxes5.XAxis.TickLabels = {};
app.UIAxes5.YAxis.TickLabels = {};
% Display image and stretch to fill axes
I = imshow('Logos.png', 'Parent', app.UIAxes5,'XData', [0 6.5],'YData', [0 1.5]);
% Set limits of axes
app.UIAxes5.XLim = [0 6.5];
app.UIAxes5.YLim = [0 1.5];
end
% Callback function: CalcularButton,
% ElpuntodeataqueesEditField, SecuenciadeTresBandasButtonGroup

function CalcularButtonPushed(app, event)
% VARIABLES A UTILIZAR
% Definición y lectura de variables
Xpuntosalida = app.PuntoinicialejexEditField.Value;
Ypuntosalida = app.PuntoinicialejeyEditField.Value;
Xpuntolegada = app.PuntofinalejexEditField.Value;
Ypuntolegada = app.PuntofinalejeyEditField.Value;

precision=app.PrecisinSlider.Value;

% Cambios de sistema de referencia
switch app.BandacortaButtonGroup.SelectedObject
case app.BandasuperiorButton
case app.BandainferiorButton
    Xpuntosalida = 4 - Xpuntosalida;
    Ypuntosalida = 8 - Ypuntosalida;
    Xpuntolegada = 4 - Xpuntolegada;
    Ypuntolegada = 8 - Ypuntolegada;
end

switch app.SecuenciadeTresBandasButtonGroup.SelectedObject
case app.SentidohorarioButton
case app.SentidoantihorarioButton
    Xpuntosalida = 4 - Xpuntosalida;
    Xpuntolegada = 4- Xpuntolegada;
end

```

```

% DEFINICIÓN Y EJECUCIÓN DEL MODELO
%SE INTRODUCE EL MODELO:
% Puntos de salida:
DatosSalida=[1 1 1 1 1 1.5 1.5 1.5 1.5 1.5 1.5 2 2 2 2 2 2.5 2.5 2.5 2.5 2.5 3
3 3 3 3 3.5 3.5 3.5 3.5 3.5 4 4 4 4 4 4.5 4.5 4.5 4.5 4.5 4.5 4.5 5 5 5 5 5 5 5 5 6 6 6 6 6
6 7 7 7 7 7 7 8 8 8 8 8 9 9 9 9 10 10 10 10 10.5 10.5 11 11];
% Puntos de ataque:
DatosAtaque=[7 6 5 4 3 2 6 5 4 3 2 1 6 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4
3 2 1 4.5 4 3.5 3 2.5 2 1.5 1 4.5 4 3.5 3 2.5 2 1.5 1 3.5 3 2.5 2 1.5 1 3 2.5 2 1.5 1 0.5 2.5 2
1.5 1 0.5 2 1.5 1 0.5 1.5 1 0.5 1 0.5 1 0.5];
% Puntos de llegada sobre la tercera banda:
DatosLlegada3=[0.3 2.9 4.95 6 6.5 7.65 1.45 2.95 4.3 4.85 5.65 6.8 0.9 2.1 3 4.15
4.9 5.7 1.5 2.55 3.35 4.25 4.95 1.05 2.1 3.1 3.8 4.6 0.65 1.65 2.6 3.45 4.15 0.2 1.45 2.3 3
3.75 0.4 1.1 1.55 2 2.35 2.85 3.3 3.6 0.05 0.85 1.3 1.75 2.15 2.7 3.05 3.4 0.6 1.2 1.65 2.15
2.65 2.9 0.9 1.35 1.8 2.15 2.65 3 0.6 1.25 1.7 2.45 2.85 0.3 0.9 1.55 2.35 0.2 0.85 1.75 0.4
1.35 0.2 0.9];
% Puntos de llegada sobre la cuarta banda:
DatosLlegada4=[6.8 9.45 10.6 11.2 11.4 11.85 8.1 9.35 10.2 10.45 10.95 11.45 7.1
8.6 9.25 10.1 10.55 11.05 8.05 8.9 9.45 10 10.5 7 8.5 9.15 9.7 10.2 6.5 7.8 8.8 9.3 9.8 5.7
7.25 8.35 9 9.5 6 6.5 7.25 7.95 8.35 8.95 9.2 9.45 5.1 6.05 6.9 7.35 8.05 8.5 8.95 9.2 5.7 6.5
7.05 7.8 8.4 8.65 5.8 6.55 7.2 7.7 8.35 8.7 5.3 6.45 6.85 7.8 8.45 4.2 5.15 6 7.5 3.2 4.55 6.25
3.2 5.5 2.2 4.05];
[xData, yData, zData] = prepareSurfaceData( DatosSalida, DatosAtaque, DatosLlegada3
);
ft = 'thinplateinterp';
[banda3Model, gof3b] = fit( [xData, yData], zData, ft, 'Normalize', 'on' );

[xData, yData, zData] = prepareSurfaceData( DatosSalida, DatosAtaque, DatosLlegada4
);
ft = 'thinplateinterp';
[banda4Model, gof4b] = fit( [xData, yData], zData, ft, 'Normalize', 'on' );

% CÁLCULO DE LA ACCIÓN ÓPTIMA

correcto=1;
AtaqueAUX=8;

while (AtaqueAUX > 0)

    ataque = -1 * AtaqueAUX + 8; %Cambio de referencia para hacer
coincidir con los datos del modelo y Figura A2.2

    PendienteEntrada = (Ypuntosalida - AtaqueAUX ) / Xpuntosalida;
    SalidaAUX = 4 * PendienteEntrada + AtaqueAUX;

    if (SalidaAUX > 8)
        uialert(app.VirtualTrainerUIFigure,'Solución no encontrada','Error');
        correcto=0;
        break
    end

    if (SalidaAUX >= 0)

```

```

        salida = SalidaAUX + 4;           %Cambio de referencia para hacer
coincidir con los datos del modelo y Figura A2.1
    end

    if (SalidaAUX < 0)
        SalidaAUX = -1 * AtaqueAUX / PendienteEntrada;
        salida = SalidaAUX;           %Cambio de referencia para hacer
coincidir con los datos del modelo y Figura A2.1
    end

    %Se introducen las entradas en el modelo:

    llegada3=banda3Model(salida,ataque);

    if (llegada3 < 0)
        uialert(app.VirtualTrainerUIFigure,'Solución no encontrada','Error');
        correcto=0;
        break
    end

    if (llegada3 <= 8)                   % Comprobación de solución dentro de
límites de la mesa
        llegada4=banda4Model(salida,ataque);

        if (llegada4 <= 12)             % Comprobación de solución dentro de
límites de la mesa

            Llegada3AUX = -1 * llegada3 + 8; %Cambio de referencia para hacer
coincidir con los datos del modelo y Figura A2.3

            if (llegada4 <= 8)
                Llegada4AUX = -1 * llegada4 + 8;
                PendienteSalida = (Llegada3AUX - Llegada4AUX) / 4;
            end

            if (llegada4 > 8)
                Llegada4AUX = llegada4 - 8; %Cambio de referencia para hacer
coincidir con los datos del modelo y Figura A2.3
                PendienteSalida = Llegada3AUX / (4 - Llegada4AUX);
            end

            %Se comprueba la solución de la iteración:

            Error = abs(Llegada3AUX - (4 - XpuntoLlegada) * PendienteSalida -
YpuntoLlegada);

            if ( Error < precision*5 )
                break
            else
                AtaqueAUX = AtaqueAUX - precision;
            end
        else
            AtaqueAUX = AtaqueAUX - precision;
        end
    end
end

```

```

        else
            AtaqueAUX = AtaqueAUX - precision;
        end
    end

    if (AtaqueAUX <= 0)
        uialert(app.VirtualTrainerUIFigure,'Solución no encontrada','Error');
        correcto=0;
    end

% REPRESENTACIÓN DE LOS RESULTADOS FINALES

%Mostrar resultado final
if (correcto == 1)
    switch app.BandacortaButtonGroup.SelectedObject
        case app.BandasuperiorButton
        case app.BandainferiorButton
            AtaqueAUX = 8 - AtaqueAUX;
    end

    % Cálculo de los puntos característicos para dibujar en la mesa
    switch app.SecuenciadeTresBandasButtonGroup.SelectedObject
        case app.SentidohorarioButton
            switch app.BandacortaButtonGroup.SelectedObject
                case app.BandasuperiorButton
                    Xataque=-2;
                case app.BandainferiorButton
                    Xataque=2;
            end
        case app.SentidoantihorarioButton
            switch app.BandacortaButtonGroup.SelectedObject
                case app.BandasuperiorButton
                    Xataque=2;
                case app.BandainferiorButton
                    Xataque=-2;
            end
    end

    Xjug=app.PuntoinicialejexEditField.Value-2;
    Yjug=app.PuntoinicialejeyEditField.Value-4;
    Xfin=app.PuntofinalejexEditField.Value-2;
    Yfin=app.PuntofinalejeyEditField.Value-4;
    Yataque=AtaqueAUX-4;

    app.ElpuntodeataqueesEditField.Value = AtaqueAUX;

    % Representación de los puntos característicos para dibujar en la mesa
    p=plot(app.UIAxes4,Xjug,Yjug,Xfin,Yfin,Xataque,Yataque);
    p(1).Marker='o';
    p(1).MarkerFaceColor='w';
    p(1).MarkerEdgeColor='w';
    p(1).MarkerSize=15;
    p(2).Marker='o';

```

```

p(2).MarkerFaceColor='r';
p(2).MarkerEdgeColor='r';
p(2).MarkerSize=15;
p(3).Marker='o';
p(3).MarkerFaceColor='k';
p(3).MarkerEdgeColor='k';
p(3).MarkerSize=10;

% Cálculo y representación de las líneas que seguirá la bola jugadora
switch app.BandasCortaButtonGroup.SelectedObject
    case app.BandasuperiorButton
        switch app.SecuenciadeTresBandasButtonGroup.SelectedObject
            case app.SentidohorarioButton
                Xcorta = 4*(4-(Yataque))/(4-(Yataque)+(4-(Llegada3AUX-4)))-2; %
Cálculo aproximado debido a que éste punto no ha sido modelado

                line(app.UIAxes4,
[Xjug,Xataque],[Yjug,Yataque], 'color','k','LineWidth',1.2,'Marker','>');
                line(app.UIAxes4,
[Xataque,Xcorta],[Yataque,4], 'color','k','LineWidth',0.5,'Marker','<','LineStyle','--');
                line(app.UIAxes4, [Xcorta,2],[4,Llegada3AUX-
4], 'color','k','LineWidth',0.5,'Marker','<','LineStyle','--');
                line(app.UIAxes4, [2,Xfin],[Llegada3AUX-
4,Yfin], 'color','k','LineWidth',1.2,'Marker','>');

            case app.SentidoantihorarioButton
                Xcorta = 4*(4-(Llegada3AUX-4))/(4-(Yataque)+(4-(Llegada3AUX-
4)))-2; % Cálculo aproximado debido a que éste punto no ha sido modelado

                line(app.UIAxes4,
[Xjug,Xataque],[Yjug,Yataque], 'color','k','LineWidth',1.2,'Marker','<');
                line(app.UIAxes4,
[Xataque,Xcorta],[Yataque,4], 'color','k','LineWidth',0.5,'Marker','>','LineStyle','--');
                line(app.UIAxes4, [Xcorta,-2],[4,Llegada3AUX-
4], 'color','k','LineWidth',0.5,'Marker','>','LineStyle','--');
                line(app.UIAxes4, [-2,Xfin],[Llegada3AUX-
4,Yfin], 'color','k','LineWidth',1.2,'Marker','<');
            end
        case app.BandainferiorButton
            switch app.SecuenciadeTresBandasButtonGroup.SelectedObject
                case app.SentidohorarioButton
                    Xcorta = 4*((4+8-Llegada3AUX-4)/(4+8-Llegada3AUX-4+4+Yataque))-
2; % Cálculo aproximado debido a que éste punto no ha sido modelado

                    line(app.UIAxes4,
[Xjug,Xataque],[Yjug,Yataque], 'color','k','LineWidth',1.2,'Marker','<');
                    line(app.UIAxes4, [Xataque,Xcorta],[Yataque,-
4], 'color','k','LineWidth',0.5,'Marker','>','LineStyle','--');
                    line(app.UIAxes4, [Xcorta,-2],[-4,8-Llegada3AUX-
4], 'color','k','LineWidth',0.5,'Marker','>','LineStyle','--');
                    line(app.UIAxes4, [-2,Xfin],[8-Llegada3AUX-
4,Yfin], 'color','k','LineWidth',1.2,'Marker','<');

                case app.SentidoantihorarioButton

```



```

        Xcorta = 4*((4+Yataque)/(4+8-Llegada3AUX-4+4+Yataque))-2; %
Cálculo aproximado debido a que éste punto no ha sido modelado

        line(app.UIAxes4,
[Xjug,Xataque],[Yjug,Yataque], 'color','k','LineWidth',1.2,'Marker','>');
        line(app.UIAxes4, [Xataque,Xcorta],[Yataque,-
4], 'color','k','LineWidth',0.5,'Marker','<','LineStyle','--');
        line(app.UIAxes4, [Xcorta,2],[-4,8-Llegada3AUX-
4], 'color','k','LineWidth',0.5,'Marker','<','LineStyle','--');
        line(app.UIAxes4, [2,Xfin],[8-Llegada3AUX-
4,Yfin], 'color','k','LineWidth',1.2,'Marker','>');
        end

    end

% Representación gráfica del punto de golpeo
switch app.SecuenciadeTresBandasButtonGroup.SelectedObject
case app.SentidohorarioButton
    % Remove title, axis labels, and tick labels
    %uistack(app.UIAxes,top);
    title(app.UIAxes2, []);
    xlabel(app.UIAxes2, []);
    ylabel(app.UIAxes2, []);
    app.UIAxes2.XAxis.TickLabels = {};
    app.UIAxes2.YAxis.TickLabels = {};
    % Display image and stretch to fill axes
    I = imshow('EfectoDcha.png', 'Parent', app.UIAxes2,'XData', [0
4], 'YData', [0 3.8]);
    % Set limits of axes
    app.UIAxes2.XLim = [0 I.XData(2)];
    app.UIAxes2.YLim = [0 I.YData(2)];

case app.SentidoantihorarioButton
    % Remove title, axis labels, and tick labels
    %uistack(app.UIAxes,top);
    title(app.UIAxes2, []);
    xlabel(app.UIAxes2, []);
    ylabel(app.UIAxes2, []);
    app.UIAxes2.XAxis.TickLabels = {};
    app.UIAxes2.YAxis.TickLabels = {};
    % Display image and stretch to fill axes
    I = imshow('EfectoIzda.png', 'Parent', app.UIAxes2,'XData', [0
4], 'YData', [0 3.8]);
    % Set limits of axes
    app.UIAxes2.XLim = [0 I.XData(2)];
    app.UIAxes2.YLim = [0 I.YData(2)];

    end
end
end
% Value changed function: MostrarenmesaButton
function MostrarenmesaButtonPushed(app, event)
    %Representación de los puntos tras 'Mostrar en mesa'

    Xjug=app.PuntoinicialejexEditField.Value-2;

```

```

Yjug=app.PuntoinicialejeyEditField.Value-4;
Xfin=app.PuntofinalejexEditField.Value-2;
Yfin=app.PuntofinalejeyEditField.Value-4;
p=plot(app.UIAxes4,Xjug,Yjug,Xfin,Yfin);
p(1).Marker='o';
p(1).MarkerFaceColor='w';
p(1).MarkerEdgeColor='w';
p(1).MarkerSize=15;
p(2).Marker='o';
p(2).MarkerFaceColor='r';
p(2).MarkerEdgeColor='r';
p(2).MarkerSize=15;

end
end
% App initialization and construction
methods (Access = private)
    % Create UIFigure and components
    function createComponents(app)
        % Create VirtualTrainerUIFigure
        app.VirtualTrainerUIFigure = uifigure;
        app.VirtualTrainerUIFigure.Colormap = [0.2431 0.149 0.6588;0.251 0.1647
0.7059;0.2588 0.1804 0.7529;0.2627 0.1961 0.7961;0.2706 0.2157 0.8353;0.2745 0.2353
0.8706;0.2784 0.2549 0.898;0.2784 0.2784 0.9216;0.2824 0.302 0.9412;0.2784 0.3216 0.9608;0.2784
0.3451 0.9725;0.2745 0.3686 0.9843;0.2706 0.3882 0.9922;0.2588 0.4118 0.9961;0.2431 0.4353
1;0.2196 0.4588 0.9961;0.1961 0.4863 0.9882;0.1843 0.5059 0.9804;0.1804 0.5294 0.9686;0.1765
0.549 0.9529;0.1686 0.5686 0.9373;0.1529 0.5922 0.9216;0.1451 0.6078 0.9098;0.1373 0.6275
0.898;0.1255 0.6471 0.8902;0.1098 0.6627 0.8745;0.0941 0.6784 0.8588;0.0706 0.6941
0.8392;0.0314 0.7098 0.8157;0.0039 0.7216 0.7922;0.0078 0.7294 0.7647;0.0431 0.7412
0.7412;0.098 0.749 0.7137;0.1412 0.7569 0.6824;0.1725 0.7686 0.6549;0.1922 0.7765 0.6235;0.2157
0.7843 0.5922;0.2471 0.7922 0.5569;0.2902 0.7961 0.5176;0.3412 0.8 0.4784;0.3922 0.8039
0.4353;0.4471 0.8039 0.3922;0.5059 0.8 0.349;0.5608 0.7961 0.3059;0.6157 0.7882 0.2627;0.6706
0.7804 0.2235;0.7255 0.7686 0.1922;0.7725 0.7608 0.1647;0.8196 0.749 0.1529;0.8627 0.7412
0.1608;0.902 0.7333 0.1765;0.9412 0.7294 0.2118;0.9725 0.7294 0.2392;0.9961 0.7451
0.2353;0.9961 0.7647 0.2196;0.9961 0.7882 0.2039;0.9882 0.8118 0.1882;0.9804 0.8392
0.1765;0.9686 0.8627 0.1647;0.9608 0.8902 0.1529;0.9608 0.9137 0.1412;0.9647 0.9373
0.1255;0.9686 0.9608 0.1059;0.9765 0.9843 0.0824];
        app.VirtualTrainerUIFigure.Position = [50 50 1241 658];
        app.VirtualTrainerUIFigure.Name = 'Virtual Trainer';
        % Create CalcularButton
        app.CalcularButton = uibutton(app.VirtualTrainerUIFigure, 'push');
        app.CalcularButton.ButtonPushedFcn = createCallbackFcn(app, @CalcularButtonPushed,
true);

        app.CalcularButton.FontSize = 36;
        app.CalcularButton.Position = [68 40 284 64];
        app.CalcularButton.Text = 'Calcular';
        % Create PuntosdepartidaPanel
        app.PuntosdepartidaPanel = uipanel(app.VirtualTrainerUIFigure);
        app.PuntosdepartidaPanel.Title = 'Puntos de partida';
        app.PuntosdepartidaPanel.Position = [22 419 374 229];
        % Create PuntoinicialejexEditFieldLabel
        app.PuntoinicialejexEditFieldLabel = uilabel(app.PuntosdepartidaPanel);
        app.PuntoinicialejexEditFieldLabel.FontSize = 24;
        app.PuntoinicialejexEditFieldLabel.Position = [23 172 209 30];

```

```

app.PuntoinicialejexEditFieldLabel.Text = 'Punto inicial (eje x)';
% Create PuntoinicialejexEditField
app.PuntoinicialejexEditField = uieditfield(app.PuntosdepartidaPanel, 'numeric');
app.PuntoinicialejexEditField.Limits = [0 4];
app.PuntoinicialejexEditField.FontSize = 24;
app.PuntoinicialejexEditField.Position = [247 172 106 30];
% Create PuntoinicialejeyEditFieldLabel
app.PuntoinicialejeyEditFieldLabel = uilabel(app.PuntosdepartidaPanel);
app.PuntoinicialejeyEditFieldLabel.FontSize = 24;
app.PuntoinicialejeyEditFieldLabel.Position = [23 134 209 30];
app.PuntoinicialejeyEditFieldLabel.Text = 'Punto inicial (eje y)';
% Create PuntoinicialejeyEditField
app.PuntoinicialejeyEditField = uieditfield(app.PuntosdepartidaPanel, 'numeric');
app.PuntoinicialejeyEditField.Limits = [0 8];
app.PuntoinicialejeyEditField.FontSize = 24;
app.PuntoinicialejeyEditField.Position = [247 134 106 30];
% Create PuntofinalejexEditFieldLabel
app.PuntofinalejexEditFieldLabel = uilabel(app.PuntosdepartidaPanel);
app.PuntofinalejexEditFieldLabel.FontSize = 24;
app.PuntofinalejexEditFieldLabel.Position = [23 85 209 30];
app.PuntofinalejexEditFieldLabel.Text = 'Punto final (eje x)';
% Create PuntofinalejexEditField
app.PuntofinalejexEditField = uieditfield(app.PuntosdepartidaPanel, 'numeric');
app.PuntofinalejexEditField.Limits = [0 4];
app.PuntofinalejexEditField.FontSize = 24;
app.PuntofinalejexEditField.Position = [247 85 106 30];
% Create PuntofinalejeyEditFieldLabel
app.PuntofinalejeyEditFieldLabel = uilabel(app.PuntosdepartidaPanel);
app.PuntofinalejeyEditFieldLabel.FontSize = 24;
app.PuntofinalejeyEditFieldLabel.Position = [23 48 196 30];
app.PuntofinalejeyEditFieldLabel.Text = 'Punto final (eje y)';
% Create PuntofinalejeyEditField
app.PuntofinalejeyEditField = uieditfield(app.PuntosdepartidaPanel, 'numeric');
app.PuntofinalejeyEditField.Limits = [0 8];
app.PuntofinalejeyEditField.FontSize = 24;
app.PuntofinalejeyEditField.Position = [247 48 106 30];
% Create MostrarenmesaButton
app.MostrarenmesaButton = uibutton(app.PuntosdepartidaPanel, 'state');
app.MostrarenmesaButton.ValueChangedFcn = createCallbackFcn(app,
@MostrarenmesaButtonPushed, true);
app.MostrarenmesaButton.Text = 'Mostrar en mesa';
app.MostrarenmesaButton.Position = [142 9 106 22];
% Create ElpuntodeataqueesEditFieldLabel
app.ElpuntodeataqueesEditFieldLabel = uilabel(app.VirtualTrainerUIFigure);
app.ElpuntodeataqueesEditFieldLabel.HorizontalAlignment = 'right';
app.ElpuntodeataqueesEditFieldLabel.FontSize = 24;
app.ElpuntodeataqueesEditFieldLabel.Position = [848 602 248 30];
app.ElpuntodeataqueesEditFieldLabel.Text = 'El punto de ataque es: ';
% Create ElpuntodeataqueesEditField
app.ElpuntodeataqueesEditField = uieditfield(app.VirtualTrainerUIFigure,
'numeric');
app.ElpuntodeataqueesEditField.ValueDisplayFormat = '%.2f';
app.ElpuntodeataqueesEditField.ValueChangedFcn = createCallbackFcn(app,
@CalcularButtonPushed, true);

```

```

app.ElpuntodeataqueesEditField.Editable = 'off';
app.ElpuntodeataqueesEditField.FontSize = 24;
app.ElpuntodeataqueesEditField.Position = [1111 602 100 30];
% Create SecuenciadeTresBandasButtonGroup
app.SecuenciadeTresBandasButtonGroup = uibuttongroup(app.VirtualTrainerUIFigure);
app.SecuenciadeTresBandasButtonGroup.Title = 'Secuencia de Tres Bandas';
app.SecuenciadeTresBandasButtonGroup.SizeChangedFcn = createCallbackFcn(app,
@CalcularButtonPushed, true);
app.SecuenciadeTresBandasButtonGroup.Position = [23 307 374 103];
% Create SentidohorarioButton
app.SentidohorarioButton = uiradiobutton(app.SecuenciadeTresBandasButtonGroup);
app.SentidohorarioButton.Text = 'Sentido horario';
app.SentidohorarioButton.FontSize = 24;
app.SentidohorarioButton.Position = [11 43 185 28];
app.SentidohorarioButton.Value = true;
% Create SentidoantihorarioButton
app.SentidoantihorarioButton = uiradiobutton(app.SecuenciadeTresBandasButtonGroup);
app.SentidoantihorarioButton.Text = 'Sentido antihorario';
app.SentidoantihorarioButton.FontSize = 24;
app.SentidoantihorarioButton.Position = [11 8 225 28];
% Create PrecisinSliderLabel
app.PrecisinSliderLabel = uilabel(app.VirtualTrainerUIFigure);
app.PrecisinSliderLabel.HorizontalAlignment = 'right';
app.PrecisinSliderLabel.Position = [22 154 55 22];
app.PrecisinSliderLabel.Text = 'Precisión';
% Create PrecisinSlider
app.PrecisinSlider = uislider(app.VirtualTrainerUIFigure);
app.PrecisinSlider.Limits = [0.01 0.1];
app.PrecisinSlider.Position = [98 163 288 3];
app.PrecisinSlider.Value = 0.01;
% Create TextArea
app.TextArea = uitextarea(app.VirtualTrainerUIFigure);
app.TextArea.Editable = 'off';
app.TextArea.FontSize = 14;
app.TextArea.BackgroundColor = [0.9412 0.9412 0.9412];
app.TextArea.Position = [868 546 324 45];
app.TextArea.Value = {'Punto de ataque de acuerdo con la numeración mostrada en la
Figura 4.2 de la memoria adjunta.'};
% Create PuntodegolpeosobrelabolajugadoraPanel
app.PuntodegolpeosobrelabolajugadoraPanel = uipanel(app.VirtualTrainerUIFigure);
app.PuntodegolpeosobrelabolajugadoraPanel.Title = 'Punto de golpeo sobre la bola
jugadora';
app.PuntodegolpeosobrelabolajugadoraPanel.Position = [837 124 386 410];
% Create UIAxes2
app.UIAxes2 = uiaxes(app.PuntodegolpeosobrelabolajugadoraPanel);
title(app.UIAxes2, 'Title')
xlabel(app.UIAxes2, 'X')
ylabel(app.UIAxes2, 'Y')
app.UIAxes2.PlotBoxAspectRatio = [0.997101449275362 1 0.997101449275362];
app.UIAxes2.Visible = 'off';
app.UIAxes2.Position = [6 9 374 372];
% Create UIAxes3
app.UIAxes3 = uiaxes(app.VirtualTrainerUIFigure);
xlabel(app.UIAxes3, 'X (Eje banda corta)')

```

```

ylabel(app.UIAxes3, 'Y (Eje banda larga)')
app.UIAxes3.XColor = 'none';
app.UIAxes3.YColor = 'none';
app.UIAxes3.Position = [420 42 370 581];
% Create UIAxes4
app.UIAxes4 = uiaxes(app.VirtualTrainerUIFigure);
app.UIAxes4.FontSize = 16;
app.UIAxes4.XLim = [-2 2];
app.UIAxes4.YLim = [-4 4];
app.UIAxes4.TickDir = 'out';
app.UIAxes4.TickLength = [1e-06 2.5e-07];
app.UIAxes4.GridAlpha = 0.15;
app.UIAxes4.MinorGridColor = [0.1 0.1 0.1];
app.UIAxes4.MinorGridAlpha = 0.25;
app.UIAxes4.XAxisLocation = 'origin';
app.UIAxes4.XColor = 'none';
app.UIAxes4.XTick = [-2 -1 0 1 2];
app.UIAxes4.XTickLabel = '';
app.UIAxes4.XMinorTick = 'on';
app.UIAxes4.YAxisLocation = 'origin';
app.UIAxes4.YColor = 'none';
app.UIAxes4.YTick = [-4 -3 -2 -1 0 1 2 3 4];
app.UIAxes4.YTickLabel = {''; '-3'; '-2'; '-1'; ''; '1'; '2'; '3'; ''};
app.UIAxes4.YMinorTick = 'on';
app.UIAxes4.Color = [0 0.451 1];
app.UIAxes4.XGrid = 'on';
app.UIAxes4.YGrid = 'on';
app.UIAxes4.LabelFontSizeMultiplier = 0.1;
app.UIAxes4.TitleFontSizeMultiplier = 0.1;
app.UIAxes4.Position = [468 84 273 499];
% Create TextArea_2
app.TextArea_2 = uitextarea(app.VirtualTrainerUIFigure);
app.TextArea_2.Editable = 'off';
app.TextArea_2.FontSize = 18;
app.TextArea_2.BackgroundColor = [0.9412 0.9412 0.9412];
app.TextArea_2.Position = [521 13 168 28];
app.TextArea_2.Value = {'Banda corta (eje x)'};
% Create TextArea_3
app.TextArea_3 = uitextarea(app.VirtualTrainerUIFigure);
app.TextArea_3.Editable = 'off';
app.TextArea_3.FontSize = 18;
app.TextArea_3.BackgroundColor = [0.9412 0.9412 0.9412];
app.TextArea_3.Position = [410 620 166 28];
app.TextArea_3.Value = {'Banda larga (eje y)'};
% Create UIAxes5
app.UIAxes5 = uiaxes(app.VirtualTrainerUIFigure);
title(app.UIAxes5, 'Title')
xlabel(app.UIAxes5, 'X')
ylabel(app.UIAxes5, 'Y')
app.UIAxes5.Visible = 'off';
app.UIAxes5.Position = [837 13 386 100];
% Create BandacortaButtonGroup
app.BandacortaButtonGroup = uibuttongroup(app.VirtualTrainerUIFigure);
app.BandacortaButtonGroup.Title = 'Banda corta';

```

```

app.BandacortaButtonGroup.Position = [23 197 373 97];
% Create BandasuperiorButton
app.BandasuperiorButton = uiradiobutton(app.BandacortaButtonGroup);
app.BandasuperiorButton.Text = 'Banda superior';
app.BandasuperiorButton.FontSize = 24;
app.BandasuperiorButton.Position = [11 45 185 28];
app.BandasuperiorButton.Value = true;
% Create BandainferiorButton
app.BandainferiorButton = uiradiobutton(app.BandacortaButtonGroup);
app.BandainferiorButton.Text = 'Banda inferior';
app.BandainferiorButton.FontSize = 24;
app.BandainferiorButton.Position = [11 8 171 28];
end
end
methods (Access = public)
% Construct app
function app = VirtualTrainerApp
% Create and configure components
createComponents(app)
% Register the app with App Designer
registerApp(app, app.VirtualTrainerUIFigure)
% Execute the startup function
runStartupFcn(app, @startupFcn)
if nargin == 0
clear app
end
end
% Code that executes before app deletion
function delete(app)
% Delete UIFigure when app is deleted
delete(app.VirtualTrainerUIFigure)
end
end
end
end
end

```

ANEXO VI

Ejemplos del correcto funcionamiento del programa final

A continuación se muestra un Código QR que enlaza con una carpeta compartida de acceso libre en Google® Drive, en donde se encuentran tres vídeos mediante los que se demuestra la correcta utilización y validez del programa final.



Figura A6.1 Código QR con acceso a vídeos explicativos del programa final

*La Figura A6.1 ha sido generada a través de:

<http://www.codigos-qr.com/generador-de-codigos-qr/>

[Fecha de creación: 23/07/2018]