

## Simulador ROV multiplayer para escenarios petroleros submarinos

Rodrigo Beltran-Labra<sup>1</sup>, Miguel Pérez-Ramírez<sup>1</sup>, Carlos Alberto Ochoa-Zezzatti<sup>2</sup>,  
Norma J. Ontiveros-Hernández<sup>3</sup>

<sup>1</sup> Instituto Nacional de Electricidad y Energías Limpias, Cuernavaca, Morelos, México

<sup>2</sup> Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chihuahua, México

<sup>3</sup> Instituto Tecnológico de Zacatepec, Zacatepec, Morelos, México

blro130524@upemor.edu.mx, mperez@ie.org.mx, alberto.ochoa@uacj.mx,  
njoh\_314@yahoo.com.mx

**Resumen.** Se presenta la versión multijugador de SimROV, un simulador basado en Realidad Virtual e Inteligencia Artificial. Incluye un escenario submarino donde un operador de un ROV puede realizar algunas maniobras a las instalaciones de varios pozos de petróleo. El sistema incluye consolas para un instructor y para uno o más estudiantes, todos comparten un escenario en común, cada participante tiene control sobre su propio ROV aunque pueden observar e interactuar con los otros. El instructor puede introducir corrientes marinas y fugas de combustible, los estudiantes necesitan solucionar el problema, pero sus acciones son detectadas para que puedan ser guiados por el sistema mediante el uso de algoritmos de inteligencia artificial.

**Palabras clave:** realidad virtual, inteligencia artificial, ROV, multiplayer, simulador.

## Simulator ROV Multiplayer for Oil Submarine Scenarios

**Abstract.** A multiplayer version of SimROV is presented based on Virtual Reality and Artificial Intelligence. It includes a submarine scenario where an operator of an ROV can analyze some maneuvers to the installations of several oil wells. The system includes consoles for an instructor and for one or more students, and all of them share a common scenario, each participant has control over its own ROV although can observe and interact with others. The instructor can introduce marine currents and fuel leaks, the students need to solve the problem, but their actions are detected to be guided by the system through the application of artificial intelligence algorithms.

**Keywords:** virtual reality, artificial intelligence, ROV, multiplayer, simulator.

## 1. Introducción

El objetivo de este trabajo es presentar la versión multijugador de SimROV, un simulador de un ROV (Vehículo de Operación Remota por sus siglas en inglés). Además de permitir compartir el mismo escenario entre un instructor y diferentes estudiantes a través de una red, el sistema incluye algoritmos de inteligencia artificial. Por el momento estos algoritmos perciben las acciones de los estudiantes dentro del entorno virtual y proporcionan ayuda para que los estudiantes sean guiados para lograr satisfactoriamente la meta asignada por el instructor. Entre otras cosas se verifica la trayectoria de los movimientos de cada ROV, de esta manera cuando se trabaja en una maniobra específica como el cierre de una válvula para detener una fuga de combustible, se compara la trayectoria de los alumnos con la trayectoria que se espera que siga el estudiante, en caso de que el camino real esté fuera de un rango permitido, los sistemas enviarán al alumno algún mensaje para ayudarlo a volver al camino correcto.

Además de lo anterior, el ROV para trabajos submarinos está equipado con dos brazos y 2 cámaras de video. No está demás agregar que aun sin pensar en la realización de acciones para responder a una maniobra de contingencia, SimROV también puede ser utilizado para que un estudiante aprenda a utilizar y controlar un ROV. Esto incluye no solo el guiar o dirigir el ROV hacia un punto determinado sino saber controlarlo aun en la presencia de corrientes marinas, las cuales dificultan su maniobrabilidad. El control de los brazos también requiere de cierta pericia que puede ser obtenida con la ayuda de SimROV, la dificultad que se presenta aquí, es que hay que aprender a usar los controles para realizar una acción, lo cual implícitamente demanda la realización de actividades en paralelo: mover los diferentes controles de cada brazo, además de guiar al ROV dentro del ambiente virtual.

Detectar la posición de cada ROV y los movimientos de los brazos permitirá identificar posibles áreas de mejora en el proceso de aprendizaje del estudiante. Esto también alimenta al modelo de los estudiantes para que SimROV siempre conozca el progreso de los estudiantes y su estado actual.

El resto del artículo está organizado de la siguiente manera. La sección 2 incluye algunos conceptos de simuladores. La sección 3 presenta la arquitectura de SimROV. La sección 4 discute los principales logros del desarrollo de esta versión de SimROV. La sección 5 incluye algunas conclusiones que son seguidas por una lista de referencias.

## 2. Simuladores

Un simulador para entrenamiento es un sistema, que permite adquirir algún tipo específico de habilidad [1]. La literatura sobre el tema de los simuladores distingue dos tipos de simuladores [2]:

- a) Los simuladores orientados al diseño de instalaciones (SODI),
- b) Los simuladores de entrenamiento (SE).

A su vez dentro de los SE se distinguen dos tipos:

1. Los simuladores de alcance total (SAT), que reproducen completamente el entorno de operación. Estos simuladores están asociados a una réplica de la sala de control (SC), tanto en apariencia como en funciones.
2. Los simuladores que reproducen la conducta del proceso (SRCP), pero no reproducen la SC, pero que sí permiten actuar sobre la simulación en tiempo de ejecución.

En lo que respecta a SimROV, este incluye un ambiente virtual y todos los controles para operar un ROV, es decir, incluye una representación del entorno de operación (pozos petroleros en el fondo del mar), y desde su creación fue concebido como un simulador de entrenamiento (SE). Por otra parte, también es un SRCP, lo único que le falta para poder ser considerado un simulador de alcance total (SAT), es la cabina de control. Los controles de un ROV real están basados en joysticks y otros controles, mientras que SimROV por el momento está basado en el uso de teclado y mouse, y un control de videojuegos.

Sin embargo, a falta de la cabina de control, SimROV contiene el modelo del estudiante a fin de personalizar la instrucción, además de estar equipado con algoritmos de inteligencia artificial para monitorear y guiar al estudiante en su actuar durante su respuesta a situaciones de contingencia, que son introducidas por el instructor.

### **3. Arquitectura de SimROV**

La arquitectura del sistema SimROV obedece en gran medida a una serie de requisitos incluidos en la IMCA-International Marine Contractors Association, la cual define un simulador como la creación de ciertas condiciones a través de un modelo, para simular condiciones dentro de una esfera de operación apropiada [3] y clasifica los simuladores en dos clases:

Clase A) Utilizan una máquina de físicos realista y crea una representación visual adecuada para un propósito.

Clase B) Utilizan una máquina de físicos casi realista y crea una representación visual apropiada.

SimROV cumple con la mayoría de los requisitos de un simulador de clase A [4]. La Figura 1 muestra la representación gráfica de la arquitectura de SimROV multiplayer, en dicha figura se puede observar que tanto estudiante como instructor tienen su propia interfaz y son diferentes dado que ambos usuarios tienen funciones diferentes.

A continuación, se muestran los componentes de la arquitectura:

1) *Interfaces*: El sistema permite el acceso a un ambiente virtual en la red. Los usuarios pueden acceder al ambiente virtual de dos formas: como estudiante (operador) y como instructor. La Figura 2 muestra la interfaz principal.

Dentro de la interfaz del instructor se tiene control sobre el ambiente virtual, permite ingresar algunas maniobras y observar las acciones del estudiante a través de sus cámaras en primera persona, además el instructor tendrá acceso a la cámara global del

ambiente para observar las acciones que siguen todos los alumnos en tiempo real. La cámara global sólo está disponible para que el instructor. Posteriormente el avance de las maniobras y de la capacitación en general será guardado dentro de una base de datos, a la cual podrá acceder el instructor.

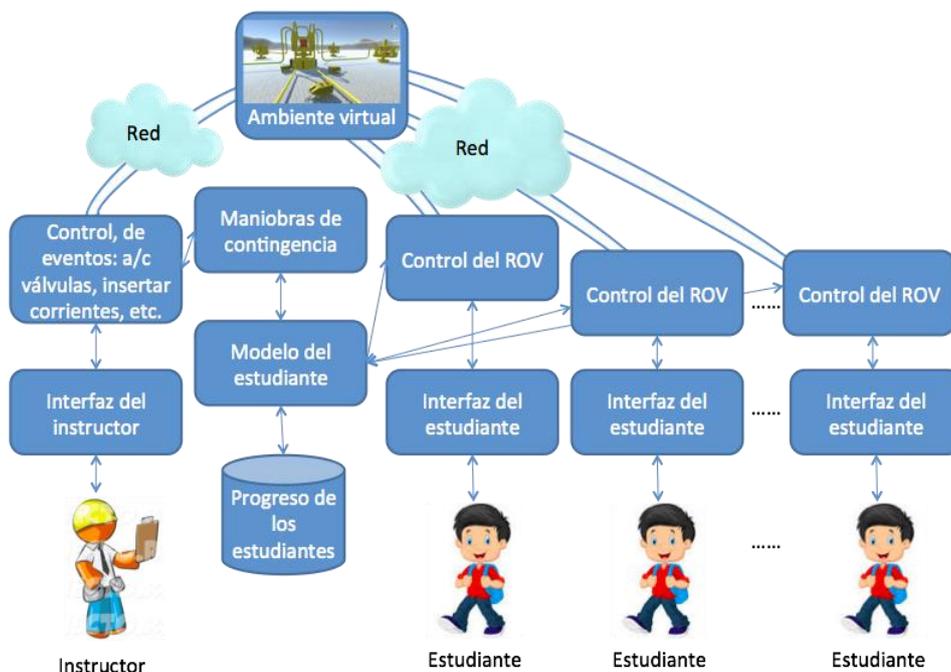


Fig. 1. Arquitectura de SimROV multiplayer.

La interfaz del estudiante le permite controlar el ROV, esto es; controlar su movimiento para trasladarse de un lugar a otro, controlar el brazo izquierdo de apoyo así como su brazo derecho para realizar las maniobras encomendadas por el instructor.

Al implementar el algoritmo de Inteligencia Artificial para calcular la ruta más corta y la base de datos con los recorridos de todas las maniobras, el sistema podrá ir guiando al estudiante durante todo el tiempo de la maniobra para que pueda concluir exitosamente la maniobra asignada.

2) *Ambiente virtual común*: Ambas interfaces pueden ver el mismo ambiente virtual ya que se implementa el método “NetworkManager” de Unity [5] para multijugador y el uso de la red ya sea intranet o internet. Las animaciones de las corrientes marinas y de los movimientos del ROV no están previamente definidas, los brazos y la translación del ROV serán originadas a voluntad por el estudiante a través de los controles definidos para este propósito.

3) *Control del ROV*: El estudiante tiene el control total del ROV, de tal forma que puede moverlo en la dirección que desee a través de las flechas del teclado o un control de videojuegos. Además permite el control sobre los brazos del ROV, de tal manera

que con el brazo derecho pueda sujetarse a algún gancho para evitar el arrastre de las corrientes y de esta manera tomar y manipular objetos con el brazo izquierdo. Con esta funcionalidad el estudiante puede realizar las maniobras asignadas por el instructor.

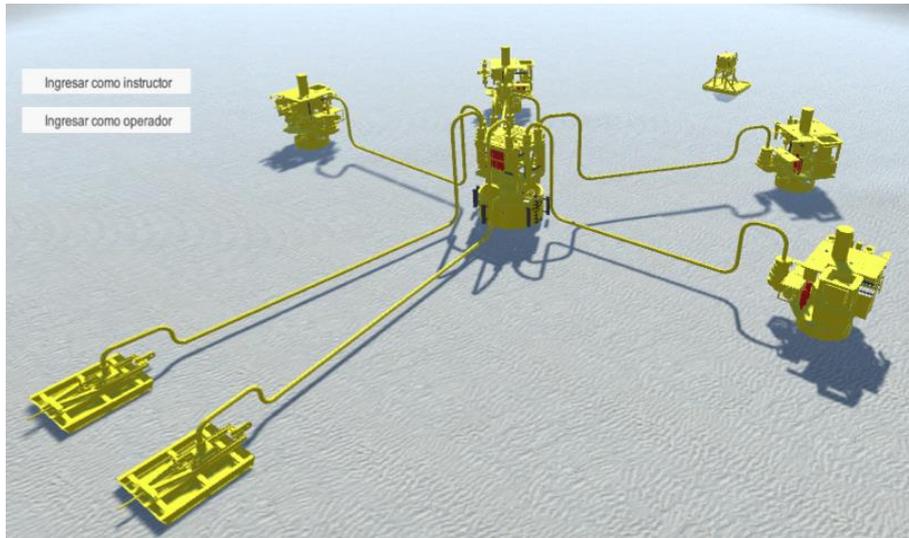


Fig. 2. Interfaz de SimROV multiplayer: Ambiente Virtual.

4) *Control de eventos*: El instructor podrá introducir en el ambiente virtual situaciones de contingencia tales como abrir y cerrar válvulas, inyectar corrientes marinas que puedan afectar al ROV, introducir fugas, etc.

5) *Maniobras de contingencia*: Este módulo contiene información de las maniobras de contingencia que un estudiante puede aprender y a su vez puede registrar su progreso en el aprendizaje en una Base de Datos. Una maniobra de contingencia podría darse por ejemplo por una fuga, que demandaría el cierre de válvulas o incluso el cambio de una sección de tubería, además en este mismo módulo se encuentran alojadas en la base de datos las posiciones de los objetos más importantes del escenario y donde los alumnos pueden llegar a realizar algunas maniobras y es con esto que el sistema puede tomar decisiones para guiar al alumno.

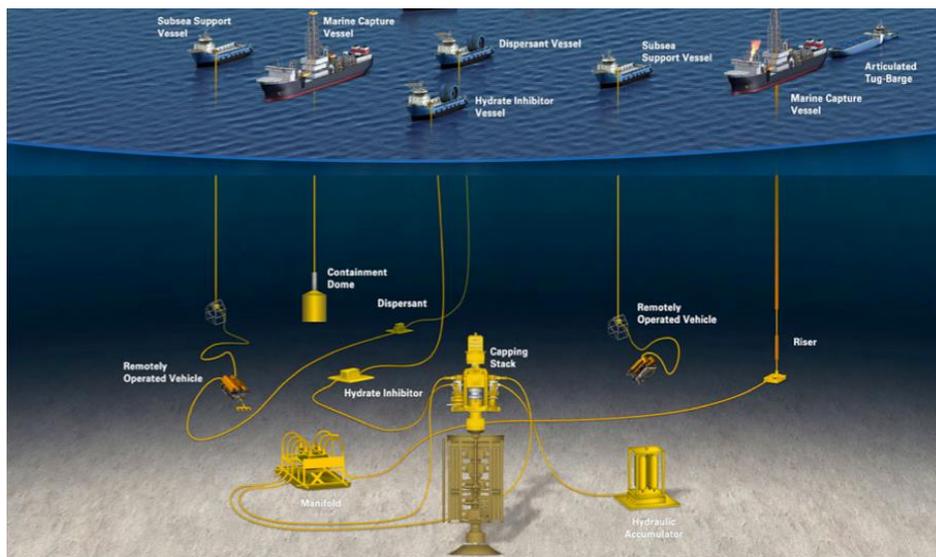
6) *Modelo del estudiante*: A la fecha con la maniobra de contingencia de la introducción de fugas, lo que se puede monitorear son dos cosas principalmente: a) que el usuario sabe controlar al ROV para desplazarse a la válvula que desea cerrar y b) que sabe cómo operar los dos brazos del ROV, de tal modo que puede sostenerse de algún gancho para evitar que la corriente lo arrastre o lo vuelque y de que puede tomar la válvula correspondiente, para cerrarla sin causar daños a otros equipos o al mismo ROV. Así el modelo del estudiante registra en una base de datos la actuación de los estudiantes en estas dos actividades globales. Sin embargo, debido al accionar concurrente de los controles que se requiere para operar los brazos y la navegación del

ROV, también es posible monitorear dichas actividades a fin de obtener una idea precisa de si está aprendiendo a controlar el ROV o no.

## 4. Desarrollo

### A. Ambiente Virtual.

Los objetos del escenario fueron creados como modelos 3D y se diseñaron para ser importados a Unity, para darles un aspecto físico se le asignan propiedades tales como RigidBody y Mesh Collider. RigidBody permite a los objetos actuar bajo el control de la física, puede recibir fuerza y torque para que el objeto se mueva de forma realista [6], además es un componente obligatorio para que los objetos sean afectados por la gravedad y de esta manera interactuar con otros objetos a través de Physx que es un motor de física de NVIDIA [7; 8]. Mesh Collider construye una presentación de colisión para el objeto [9], esto es que al no ocupar figuras primitivas se crea una malla con la forma física del objeto para colisionar con los demás objetos en cualquier parte de su estructura sin importar que no se trate de alguna figura primitiva.



**Fig. 3.** Escenario submarino: Marine Well (Tomado de [10]).

El escenario es amplio con el objetivo de que todos los usuarios puedan interactuar entre sí, dicho escenario está basado en un ambiente submarino según la compañía Marine Well mostrado en la Figura 3.

### B. Multiplayer.

Tanto el estudiante como el instructor comparten el mismo ambiente virtual pero con distintos privilegios, por tal motivo se crean dos diferentes consolas; la consola del

estudiante le permite controlar el ROV y observar las maniobras asignadas, la consola del instructor le permite asignar tareas a cada estudiante así como observar qué es lo que está realizando el estudiante en tiempo real a través de su cámara en primera persona.

En el sistema, el instructor está configurado como servidor y los estudiantes como clientes tal y como lo muestra la Figura 4. Dado que el instructor debe también estar presente en la simulación, se configura al instructor como su propio cliente ya que funge como servidor o Host. El anfitrión es un servidor y un cliente en el mismo proceso. El anfitrión utiliza un tipo especial de cliente llamado LocalClient (cliente local), mientras otros clientes son RemoteClientes (clientes remotos) [5].

Todos los estudiantes ejecutan la misma consola, para poder distinguirse de entre todos lo demás se agrega el componente “NetworkIdentity” el cual controla la identidad de un objeto en red, de esta forma se logra que cada estudiante controle su propio ROV.

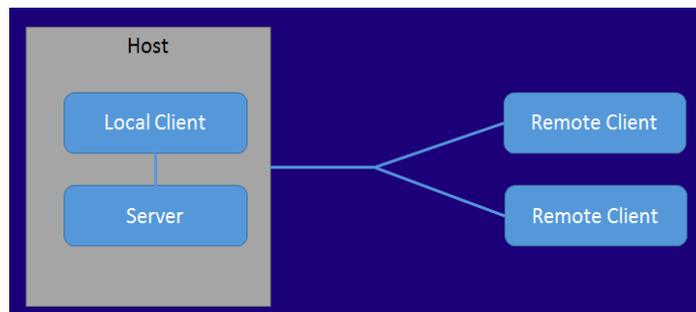


Fig. 4. Configuración de la red (Tomado de [5]).

### C. Algoritmo.

La estructura tiene puntos de referencia que permiten al sistema identificar la posición de los diferentes pozos o lugares donde el estudiante puede aplicar alguna maniobra. Todas las rutas de los diversos puntos del escenario se encuentran guardadas en la base de datos, cuando el instructor asigna una tarea al alumno, el sistema automáticamente construye la ruta más corta desde la posición del ROV hasta el lugar donde deberá hacer la maniobra, posteriormente y a lo largo de todo el recorrido que haga el alumno el sistema le irá guiando, de esta manera evita confusiones y permite el el alumno complete la maniobra satisfactoriamente.

El algoritmo de Dijkstra encuentra la ruta de menor coste dentro de un conjunto de caminos previamente definidos y su funcionamiento se puede observar en la Figura 5, el algoritmo es implementado y codificado como script en lenguaje C#.

Conforme se va moviendo el ROV, el sistema detecta si va en la ruta adecuada o se va alejando de su destino, esto es posible gracias al rango de posicionamiento de los puntos de la ruta óptima calculada por Dijkstra. La Figura 6 muestra los puntos de referencia con sus respectivos rangos.

```
 $G = (V, E)$  donde  $V$  es el conjunto de vértices y  
 $E$  el de arcos.  
 $S$  es el conjunto de vértices cuyos caminos  
más cortos al origen han sido ya  
determinados.  
 $V-S$  es el resto de vértices.  
 $d$ : ARRAY de estimaciones de caminos más  
cortos a dichos vértices.  
 $pr$ : ARRAY de predecesores para cada  
vértice.  
<<Inicializar  $d$  y  $pr$ >>  
<<Poner  $S = \emptyset$ >> // aún no hemos estudiado  
ningún vértice  
While  $\{V-S\} \neq \emptyset$  // mientras queden  
nodos sin determinar su camino mínimo al  
origen  
  <<Ordenar los vértices en  $V-S$  y  
  analizar de acuerdo a la menor  
  distancia al origen>>  
  <<Añadir  $u$ , el vértice más cercano en  
   $V-S$ , a  $S$ >> //  $S = S + \{u\}$   
  <<Recalcular la distancia a todos los  
  vértices todavía en  $V-S$  adyacentes a  
   $u$ >>
```

Fig. 5. Pseudocódigo del algoritmo de Dijkstra para la búsqueda de la ruta de menor costo (Tomado de [11]).

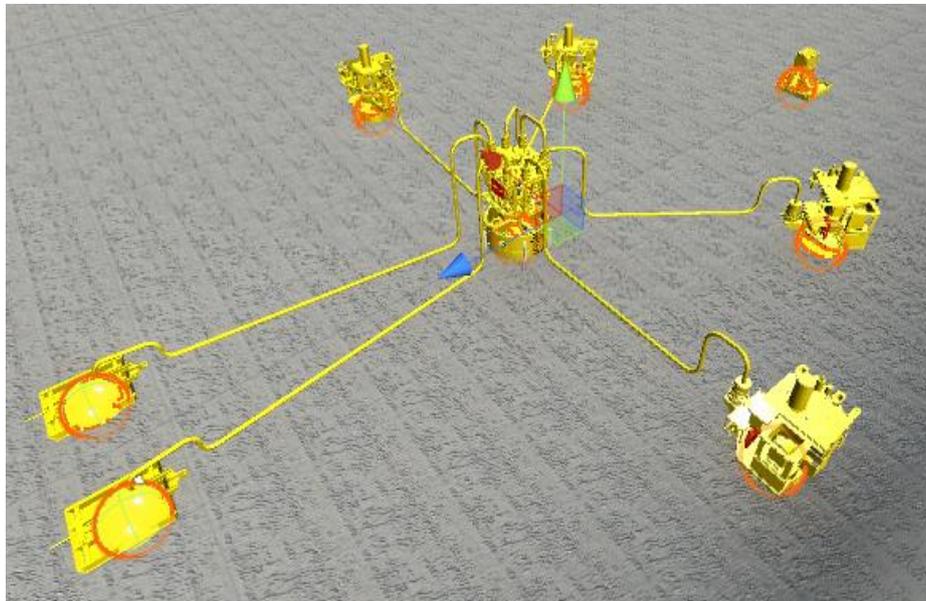


Fig. 6. Puntos de referencia que sigue el algoritmo de Dijkstra.

#### **D. Desarrollo del modelo del estudiante.**

El modelo del estudiante se programó del mismo modo que el instructor sólo restringiendo algunas funciones que tiene éste último como es el acceso a las cámaras de los demás estudiantes, la asignación de maniobras, la creación de corrientes marinas y sobre todo la capacidad de ser el Host de la red. Con lo que respecta a la adaptación de red en Unity se hace uso del componente NetworkManager proporcionado por Unity el cual controla el estado de red del sistema multi-jugador, es usado sin scripting al tener controles que permiten la configuración de sus funciones [5].

#### **E. Software utilizado.**

Para el diseño de los modelos 3D se utilizó el software de modelado, animación y renderización “3DS Max”. Para el desarrollo de la aplicación y codificación de los scripts se utilizó el motor de videojuegos “Unity3D”.

Finalmente, para poder generar las animaciones dinámicas de acuerdo a la interacción de usuario, se hizo uso de Ageia PhysX [12; 13], es el motor de física de UNITY.

### **5. Discusión de resultados**

En SimROV convergen diferentes tecnologías en una sola herramienta de apoyo al aprendizaje. Este sistema es un simulador casi de alcance total, por otra parte, de acuerdo a la clasificación de IMCA, el sistema es de tipo A. No conforme con ser un simulador, se le trató de integrar la tecnología de Tutores Inteligentes, ya que es una herramienta de aprendizaje, además todo esto, haciendo uso de la tecnología de Realidad Virtual. Esto último evitará la restricción de contar con un ROV real para poder entrenar al personal que va a operar un ROV. En la tecnología de RV se hacen uso de la animación dinámica, es decir, los movimientos no están predefinidos, más bien serán producidos en tiempo real de acuerdo a la interacción de los usuarios y de la operación inherente a los equipos involucrados en el ambiente virtual.

El modelo del estudiante almacena datos sobre si el estudiante logró la meta de la tarea asignada o no. Por ejemplo si logró cerrar la válvula o no, a fin de evitar que continúe algún tipo de derrame, así como el tiempo que le llevó lograr la meta. También se almacena información sobre la presencia e intensidad de corrientes marinas y la diferencia promedio de la ruta de navegación seguida por el estudiante, por ejemplo para alcanzar una válvula que debe cerrar, contra la ruta óptima. Esto será un indicador de si el estudiante es capaz de controlar el ROV en presencia de corrientes marinas o simplemente si es capaz de operarlo en caso de no haber corrientes.

En el primer caso se introducen corrientes de menor intensidad a fin de que el usuario aprenda a operar el ROV dentro de ese contexto, por ejemplo recurriendo a uno de los brazos del ROV para sujetarse, mientras opera con el otro brazo, y en el segundo se despliegan sugerencias de las combinaciones de comandos, que le ayuden a aprender a operar el ROV.

En la arquitectura de la Fig. 1 puede observarse cómo el modelo del estudiante cuenta con un repositorio donde almacena la información descrita arriba, la cual es generada por el módulo del Control del ROV que monitorea las trayectorias seguidas por el estudiante al tratar de lograr una meta dictada por el instructor a través del módulo de Maniobras de Contingencia. El registro de información se realiza al momento de alcanzar la meta establecida o cuando el instructor cancela la misión, quizá debido a que ha transcurrido mucho tiempo sin haberse logrado la meta o para cancelar corrientes e introducir otras de menor o mayor intensidad.

La capacidad multiplayer de SimROV, por el momento quedará reservada para la integración de maniobras colaborativas, donde dos o más operadores puedan unir esfuerzos para realizar determinadas maniobras.

La intención es contar con una herramienta sólida que en un futuro cercano sea de utilidad en el sector petrolero de aguas profundas del país, es decir, en el ámbito real de aplicación. Entre otras posibles aplicaciones para un futuro cercano de SimROV se pueden mencionar las siguientes: a) maniobras de instalación de equipos relacionados con los pozos petroleros; b) maniobras de contingencia tales como la reparación de fugas en tuberías; c) limpieza de petróleo derramado en el mar, a fin de evitar daños a los ecosistemas marinos; d) inspección; e) mantenimiento; y f) reparación.

Usualmente esta tecnología de ROVs viene del exterior, así, este trabajo constituye un primer paso hacia el desarrollo propio en el INEEL dentro de este ámbito.

## 6. Conclusiones

Se ha presentado un simulador multiplayer basado en Realidad Virtual e inteligencia artificial aplicado a la industria petrolera para escenarios submarinos donde se sitúan los pozos petroleros. El escenario incluye las estructuras metálicas representativas de los pozos petroleros y uno o más ROVs (Remotely Operated Vehicle), para realizar actividades submarinas, cada ROV representan a cada uno de los estudiantes conectados en una red LAN (Local Area Network), cada estudiante tiene control solamente sobre su propio ROV, para poder realizar tareas que le encomiende el instructor.

El simulador ayuda al usuario a realizar con éxito las tareas asignadas ya que lo va guiando a lo largo de todo el proceso y le da recomendaciones en tiempo real para cada una de las maniobras, de esta manera se optimiza el tiempo y permite a los operadores aprender a controlar un ROV real ya que el comportamiento de SimROV es casi de alcance total como se mencionó anteriormente.

Una vez que ya se cuenta con la infraestructura básica sobre la operación del ROV en ambiente de simulación multiusuarios (multiplayer), en un futuro cercano se desea revisar e incluir un conjunto de situaciones de contingencia y otras actividades propias de la operación en un ambiente petrolero en el fondo del mar. Por otra parte, el modelo del estudiante es solo parte del inicio de la integración de algoritmos de inteligencia artificial dentro de SimROV, sin embargo, se desea integrar toda la tecnología de los Sistemas Tutores Inteligentes (STI), a fin de que el sistema pueda ofrecer mayores facilidades como herramienta de apoyo al entrenamiento.

## Referencias

1. Gopinath, C., Sawyer, J.: Exploring the Learning from an Enterprise Simulation. En: Journal of Management Development (1999)
2. Merino, A., Pelayo, S., Rueda, A., Alves, R., García, A., Acebes, F., de Prada, C., Gutiérrez, G., García, M.: Un simulador de alcance total para la formación de los operarios de sala de control de factorías azucareras. En: 2ª Reunión de Usuarios de EcosimPro, UNED, Madrid 24–25 Febrero (2003)
3. IMCA: Guidance on the Use of Simulators. International Marine Contractors Association, Sitio web: <https://www.imca-int.com/?s=simulator> (2016)
4. Olgúin-Tolentino, N. D., Ingram-Ramírez, J. W., Pérez-Ramírez, M., Nava-Ayala, E. R., Hernández-Pérez, M. Y.: Prototype of a ROV simulator using dynamic animation and Virtual Reality. En: CIIDET-Congreso Internacional sobre Innovación y Desarrollo Tecnológico (2016)
5. Unity: Documentation Network Manager. Unity Technologies. Sitio web: <https://docs.unity3d.com/es/current/Manual/UNetManager.html> (2016)
6. Unity: 3D physics reference. Unity Technologies. Sitio web: <https://docs.unity3d.com/es/current/Manual/class-Rigidbody.html> (2016)
7. NVIDIA: Tecnología PHYSX. NVIDIA Corporation. Sitio web: <http://www.nvidia.es/object/nvidia-physx-es.html> PhysX (2016)
8. Craighead, J., Murphy, R., Burke, J., Goldiez, B.: A survey of commercial & open source unmanned vehicle simulators. In: Robotics and Automation, 2007 IEEE International Conference on, IEEE, pp. 852–857 (2007)
9. Unity: Mesh Collider. Unity Technologies. Sitio web: <https://docs.unity3d.com/es/current/Manual/class-MeshCollider.html> (2016)
10. Marine Well Containment Company: Containment System Overview. MWCC's Marine Well Containment System, 1, pp. 1 (2007)
11. Torrubia, G. S., Terrazas, V. L.: Algoritmo de Dijkstra. Un tutorial interactivo. En: VII Jornadas de Enseñanza Universitaria de la Informática (2012)
12. Tumult: Scene Physics Gravity. Tumult Inc. Sitio web: <http://tumult.com/hype/documentation/3.0/#physics> (2016)
13. Craighead, J., Murphy, R., Burke, J., Goldiez, B.: A survey of commercial & open source unmanned vehicle simulators. In: Robotics and Automation, 2007 IEEE International Conference on IEEE, pp. 852–857 (2007)