Estrategia de búsqueda y seguimiento de objetos sobre el lecho marino con un vehículo submarino autónomo

Hugo Curti* Gerardo Acosta† Oscar Calvo‡ Maximiliano Suarez

Resumen

En este artículo se presenta el problema de buscar un objeto sumergido utilizando un vehículo submarino autoguiado, y en caso de que el mismo se pierda, utilizar una estrategia de recuperación. La solución propuesta, basada en una máquina de estados implementada en lenguaje C++ y corriendo en un sistema GNU/LINUX, se desempeñó correctamente en simulación. Para realizar estas pruebas debió desarrollarse un ambiente de simulación que también se describe brevemente. Todo el diseño fue hecho teniendo presente la posibilidad de ser trasladado a un submarino real. Los mencionados resultados, como así también unas conclusiones preliminares y líneas de trabajo futuro, completan el trabajo.

1 Objetivos principales

Como definición inicial, se considerará que el seguimiento de oleoductos en general con un vehículo submarino autoguiado (AUV) comprende la detección del oleoducto y la alimentación de esta información directamente al sistema de control del vehículo. Este sistema de control guiará al submarino a lo largo del oleoducto en forma paralela y a una distancia predeterminada, sin la intervención del usuario, como se aprecia en la figura 1. El interés en resolver este problema es adquirir datos acerca del estado de estos oleoductos que están dispuestos en el lecho submarino. Por ello al seguimiento del oleoducto se lo suele denominar también como *adquisición*. Si por alguna causa el AUV deja de adquirir, esto es, sus sensores dejan de percibir al oleoducto, ello quiere decir que se ha apartado de la ubicación real del mismo. En esta situación se hace necesaria una recuperación de la trayectoria, también denominada *readquisición*.

Figura 1: Submarino autónomo siguiendo un oleoducto. (Cortesía de la empresa MARIDAN APS)

En la actualidad la inspección de oleoductos se lleva a cabo (por ejemplo) con vehículos a control remoto (ROV) manejados desde plataformas o bien desde embarcaciones de mayor porte. Las principales desventajas de esta aproximación son el control del vehículo que es fuertemente perturbado por el cable que lo une a la embarcación o plataforma, en una suerte de pesado cordón umbilical, y el costo que trae aparejado. Ambos in-

*INCA - Fac. de Cs. Exactas - Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina; hcurti@exa.unicen.edu.ar

†Investigador CONICET / INTELYMEC - Fac. de Ingeniería - Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina; ggacosta@fio.unicen.edu.ar

‡Departamento de Física - Facultad de Cs. Exactas - Universitat de les Illes Balears, España; oscar@galiota.uib.es

convenientes están relacionados y crecen exponencialmente con la profundidad a la que se encuentra el oleoducto a inspeccionar.

Con un AUV estas limitaciones desaparecen al no tener más un cordón umbilical a la superficie y poder navegar en forma autónoma. A su vez, la navegación puede realizarse en modo más suave, obteniéndose datos de inspección de mayor calidad. La única limitación por la profundidad es la presión que soporte el mismo submarino, pero este problema aparece a profundidades muchísimo mayores.

Para estas pruebas en simulación se han planteado dos formas de adquisición, vinculadas a dos tipos de sensores diferentes:

Seguimiento Acústico Basado en un sonar multi-haz

Seguimiento Magnético Basado en un sistema de sensores magnéticos

Ambos tipos de sensado y un archivo de datos históricos sobre la posición original del oleoducto conforman la entrada de un módulo de fusión de sensores que da idea de la posición relativa del oleoducto con respecto al AUV. La existencia previa de los datos históricos es de gran ayuda, ya que permite establecer una zona en la que se espera encontrar al oleoducto con mayor probabilidad, denominada corredor.

2 Solución propuesta

Para resolver el problema de seguir un oleoducto de cerca capturando datos y de recuperar su seguimiento cuando éste se pierde del alcance de los sensores se optó por una máquina de estados como primera aproximación. En efecto, se ha comprobado que este método provee una manera robusta y simple de implementar el módulo Autotracker (MAT). No sólo brinda las funcionalidades básicas requeridas sino que probó funcionar bien en simulación, como se describe más abajo.

El desarrollo hecho considera las siguientes hipótesis de trabajo:

- El problema de control del AUV está resuelto completamente con un módulo denominado *autopiloto* que recibiendo las coordenadas absolutas en latitud, longitud y profundidad por las que debe pasar el submarino es capaz de conducirlo allí con un mínimo error. Estas coordenadas se denominarán puntos de paso (WP). Este sistema de control será lo suficientemente sofisticado como para evitar obstáculos a partir de sensores frontales en el submarino.

- La ubicación del oleoducto vendrá estimada con un porcentaje de error asociado, en las tres dimensiones y relativo a la posición del AUV, caracterizado por $(h,\ v,\ a)$. Esta ubicación estimada es calculada por el módulo de fusión de sensores mencionado en la sección anterior.

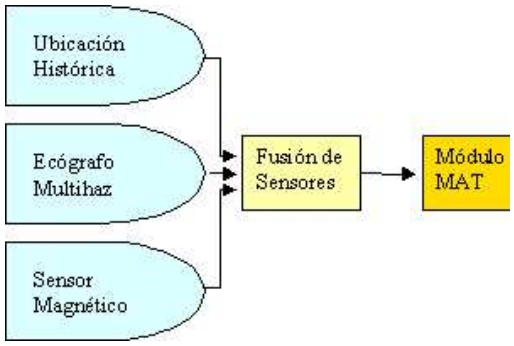

Figura 2: Contexto de funcionamiento del MAT.

Los dos módulos mencionados (el autopiloto y el de fusión de sensores) se comunican entre sí y con el MAT descripto en este artículo mediante un esquema de mensajes que puede apreciarse en la figura 2.

A continuación se describen las diferentes máquinas de estados que implementan el MAT.

2.1 Máquina Principal

Tiene dos estados: *Operate* (operar) e *Idle* (espera). En el estado *Idle* el MAT está corriendo, pero se encuentra en un modo de espera. Cuando se recibe un mensaje de operar, se pasa al estado *Operate*, en el que se inicia la máquina Operar. El MAT permanecerá en este estado hasta que se concluya la misión o se reciba un mensaje de interrupción, en cuyo caso volverá al estado *Idle*.

2.2 Máquina Operar

Esta máquina implementa la mayor parte de la funcionalidad del MAT. Comienza en el estado *FindStart* en el cual el AUV es guiado a las coordenadas en las que se inicia la misión. Una vez allí, el MAT pasa al estado *Search* (búsqueda). Este estado inicia la máquina Buscar, que implementa una revisión completa del lugar cerca de las coordenadas de inicio, se verá en párrafos siguientes, y que concluye con éxito si el oleoducto fue encontrado, o fallo en caso contrario. Si la salida de este estado es exitosa, la máquina pasa al estado *BackToStart*, en el cual el AUV es llevado al punto en el que se inició la búsqueda, y retorna paralelo al oleoducto hasta el punto donde se lo encontró efectivamente. En esta situación, el MAT pasa al estado *Track* (seguimiento). En este estado el AUV es guiado paralelo al oleoducto hasta que se alcance el punto final de la misión, o sus sensores dejen de detectarlo. Si el oleoducto ha sido perdido, una nueva búsqueda debe iniciarse en esas coordenadas. Si esta nueva búsqueda no es exitosa, el MAT pasa al estado *Skip* (salto) en el cual se le provee al AUV una trayectoria, con dirección estimada a partir de los datos históricos y distancia previamente establecida de acuerdo a cada misión. En estas nuevas coordenadas se inicia una nueva búsqueda. Así como la distancia del salto, la cantidad de veces que se puede reiniciar una búsqueda cuando el oleoducto se pierde y la cantidad de veces que se pueden combinar saltos y búsquedas son parámetros que se configuran para cada misión, de acuerdo a las características particulares de la misma. Todos los estados tienen un chequeo de tiempo, por lo que si se excede un tiempo precalculado en alguno de los estados, la máquina pasa al estado *Timeout* que finaliza la misión con error. También hay un estado *Error* al que la máquina pasa cuando se alcanza el número máximo de intentos de búsqueda. En este estado se finaliza también la misión con error.

2.3 Máquina Buscar

Esta máquina inicia en el estado *FirstHalf* (primera mitad) en la que el AUV es guiado desde el punto de inicio de la búsqueda, formando un ángulo predeterminado con la dirección supuesta del oleoducto. Cuando el AUV alcanza el lado izquierdo del corredor de búsqueda (visto desde arriba), dobla una cierta distancia perpendicular en sentido horario y pasa al estado *Line* (línea), donde vuelve a doblar en sentido horario otros noventa grados, quedando paralelo al primer recorrido, pero en sentido contrario. Cuando alcanza el lado derecho del corredor, vuelve a doblar noventa grados ahora en sentido antihorario, y así sucesivamente. Se dibuja un recorrido similar al de una cortadora de césped como se verá en la sección 3, una cantidad de pasadas que también es configurable. Luego de esto la máquina entra en el estado *LastHalf* (última mitad), en el cual el AUV recorre una línea hasta que alcanza el punto de finalización de búsqueda. Si este punto se alcanza sin encontrar el oleoducto, la máquina termina con una búsqueda infructuosa. En cambio si el oleoducto se encuentra en cualquiera de sus estados, la máquina finaliza inmediatamente el con una búsqueda exitosa.

3 Resultados en simulación

El desarrollo completo, tanto de las máquinas de estados como de la simulación fue realizado en ANSI C++, y se implementó y probó en un sistema GNU/LINUX, que también se espera sea el entorno de trabajo en el submarino real.

3.1 Descripción del simulador

El simulador se compone de dos partes principales: un simulador del ambiente propiamente dicho, denominado *ATMEnvs*, y un simulador del bloque de fusión de sensores (magnético, sonar y datos históricos), denominado *SFMSim*. Si bien ambos se inician juntos, se disparan dos procesos separados que solamente se comunican entre sí a través del sistema de mensajes.

El ATMEnvs comienza a funcionar con el AUV en un punto arbitrario (configurable) y atiende los mensajes de trayectoria provenientes del MAT. En función de los puntos por los que debe pasar el submarino (WP) se determina el ángulo de navegación (*heading*) y en función de este ángulo, de la velocidad de navegación deseada y un valor constante de tiempo asociado a cada iteración

del simulador; se calcula en forma paramétrica un avance en el eje X y otro en el eje Y. Luego en cada iteración del simulador se incrementan los valores de posición del AUV en X y en Y de acuerdo al avance calculado. Como resultado se obtiene un desplazamiento en la dirección y la velocidad deseada. En cada iteración se dibuja un nuevo punto y por interpolación de los mismos se traza la línea que marca la trayectoria que sigue el AUV. Al alcanzar un WP se pasa automáticamente a seguir próximo WP. Si no hay más WP válidos el AUV se detiene hasta tanto no reciba un nuevo mensaje de trayectoria del MAT. En cada iteración el simulador emite mensajes de posición de la forma en que lo espera el MAT. Los comandos para dibujar están embebidos en la clase *PipeSim*, y son generados en un lenguaje de scripting que es aceptado por el sistema *grace* vía pipeline de UNIX. El sistema grace es capaz de recibir datos y construir una representación gráfica que se puede imprimir o mostrar por pantalla mediante su cliente de X11 *xmgrace*. Dado que los puntos son incorporados a la gráfica a medida que los comandos son recibidos, se puede conseguir una sensación de tiempo real para la simulación. El simulador no contempla la dinámica propia del AUV, ni el sistema de control de navegación, sino que supone que se trata de un objeto puntual de reacción instantánea, dado que con esta funcionalidad ya se puede probar el MAT a nivel funcional.

El SFMSim también utiliza la clase *PipeSim* para generarse una posición supuesta del oleoducto, el estado del mismo en cada punto y los parámetros (detección, sigmas, etc.). Los estados considerados posibles para el oleoducto en esta simulación son: *enterrado*, sobre el fondo o *normal*, flotante o *libre* e *invisible*. Este simulador también escucha los mensajes de posición provenientes del ATMEnvs y los comandos provenientes del MAT, obedeciendo de la misma forma que el módulo de fusión de sensores real. Luego genera la trayectoria del oleoducto iniciandola a partir del punto del mismo más cercano a la posición actual del AUV. Es de destacar que todo el SFMSim puede anularse en caso de buscar la conectividad con el módulo de fusión de sensores real.

La clase *PipeSim* es la que maneja toda la interconexión con el sistema *grace*, y es la que crea un oleoducto a partir de un archivo gráfico con formato XPM[1]. Es ideal para embeber gráficos en programas. La clase PipeSim también calcula los sigmas, la cobertura de sensores y la detección. Los sigmas (horizontal y vertical) son funciones del estado del oleoducto. Actualmente se definen diferentes constantes para cada sigma en cada estado. La cobertura de sensores es constante y la detección es proporcional (con pendiente negativa) a la distancia entre el AUV y el oleoducto bajo inspección, valiendo 100% cuando el AUV está directamente sobre el oleoducto, con una pendiente más pronunciada cuando el oleoducto está enterrado, con una pendiente menor cuando el oleoducto está libre o normal y siempre vale cero cuando el oleoducto está invisible. La clase PipeSim es utilizada tanto por el ATMEnvs como por el SFMSim en forma independiente uno del otro. La figura 3 muestra un diagrama de bloques del ambiente de simulación.

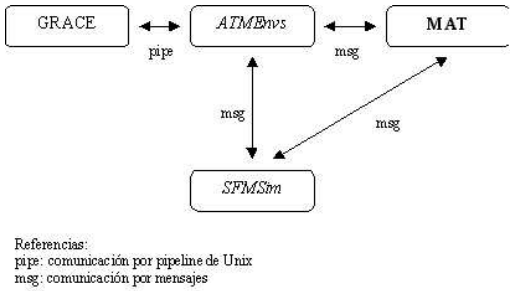

Figura 3: Diagrama de bloques del ambiente de simulación.

3.2 Casos de prueba

El MAT fue probado en simulación realizando la búsqueda inicial cuando comienza la misión. El resultado tal como aparece en la pantalla de la computadora se muestra en la figura 4 , donde se aprecia que el MAT efectúa una búsqueda exitosa del oleoducto, concluyendo en posición paralela a la ubicación simulada del mismo. Los estados por los que pasó fueron *Search* y *Track* de la máquina Operar.

[1]El formato XPM describe un gráfico como una estructura en lenguaje C.

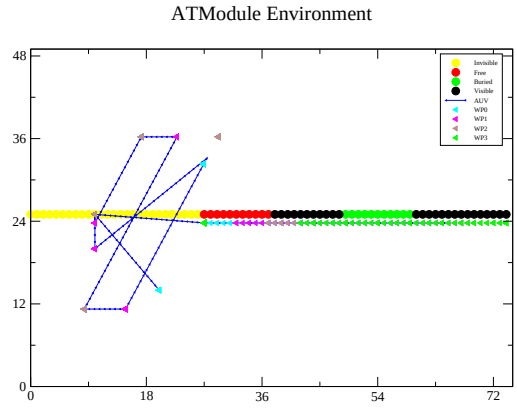

Figura 4: Búsqueda exitosa y seguimiento.

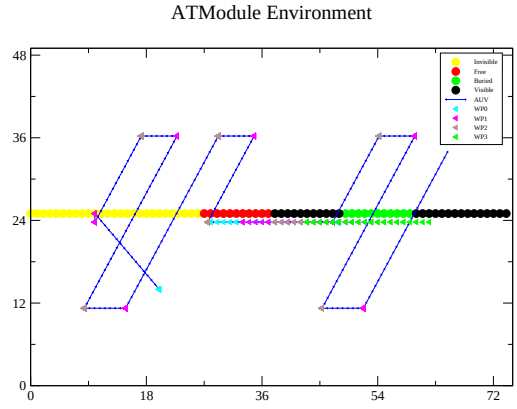

Figura 5: Búsqueda infructuosa, salto y nueva búsqueda.

En la figura 5 se ve una situación similar a la anterior, pero con un resultado de búsqueda infructuosa, por lo que el AUV debe saltar a una nueva posición de búsqueda, efectuando una secuencia entre los estados *Search*, *Skip* y *Search* de la misma máquina de estados.

4 Conclusiones preliminares y trabajo futuro

La aproximación en máquina de estados probó ser eficiente y robusta para abordar el problema planteado. La programación en C++, combinada con un adecuado sistema de mensajes entre módulos, en un sistema operativo como Linux también demostró ser una buena elección.

Como trabajo futuro se propone probar otros métodos para resolver la aplicación, y comparar distintas propuestas. En este sentido se está trabajando sobre un sistema basado en conocimiento que a partir de la lectura de sensores o salida del módulo de fusión de sensores, determine una situación o escenario. En este escenario, el sistema tomará la decisión de cómo generar las trayectorias de búsqueda en caso de ser necesario, y otras decisiones basadas en la experiencia humana en resolver contingencias que puedan presentarse durante una misión. Además de contemplar una mayor cantidad de situaciones, este sistema experto permitirá un crecimiento incremental, dotando de un mejor comportamiento al AUV sin necesidad de reprogramar todo el código, ya que simplemente se irán agregando piezas de conocimiento en forma de reglas. Finalmente, sería deseable probar estos algoritmos sobre un prototipo de submarino o vehículo autoguiado real.

Referencias

[1] BJERRUM A, SLATER T (2001): *Autonomous tracking of submarine pipelines and cables*, Hydrographic Society HYDRO 2001.

[2] ALLEN S (2000): *Pipeline Inspection*, Hydrographic Society HYDROFEST 2000.