[[1]](#footnote-1)

Simulación de Teleoperación de Robot Móvil en Red con Retardo Variable

Rodrigo Méndez
Universidad Técnica Federico Santa María
rodrigo.mendez.11@alumnos.usm.cl

*Abstract*— El retardo de tiempo variable y la pérdida de paquetes degradan el rendimiento de un sistema de teleoperación basado en protocolos de internet. La mayoría de los trabajos realizados hasta el momento en éste tema proponen soluciones de control en tiempo real u otro sistema para obtener un mejor desempeño para las tareas que se desean realizar, en donde el robot, basado en las acciones que recibe y el retardo medido en la red, intenta estimar la acción deseada, pero en general no se le entrega mayor información al usuario. En este informe se presenta un sistema en donde el usuario dispone de una estimación constante del estado actual del robot, basado en el retardo medido en la red, de modo que las acciones a ejecutar se puedan determinar según la información que se tiene de la última actualización en conjunto con la estimación. Se propone una idea de cómo se organizaría el diagrama de flujo de la interacción entre el usuario y el robot, y se presentan los resultados obtenidos de la simulación realizada con el esquema propuesto.

# INTRODUCCIÓN

En los últimos años la teleoperación ha obtenido mayor atención debido a la investigación que se realiza en lugares en que el ser humano no puede alcanzar o donde no puede permanecer mucho tiempo debido a distintas características del medio. Las mayores dificultades en la teleoperación son el retardo variable de tiempo, la pérdida de paquetes y las desconexiones. La mayoría de los sistemas de control que se proponen intentan compensar los efectos producidos por los fenómenos mencionados anteriormente [1-3], pero no se considera el beneficio que puede generar el operador si es que éste tiene mayor información o si es capaz de estimar lo que realmente está sucediendo en cada instante. Modelar la dinámica del operador y las condiciones de la red en la cual se está trabajando son tareas muy difíciles, ya que estas pueden variar bastante de un momento a otro. Existen algunos métodos que obtienen buenos resultados pero éstos limitan las variaciones del medio y asumen un retardo constante.

 En este informe se presenta un método en el cual es posible entregar mayor información al operador, el cual está encargado de controlar un robot en una red con un retardo variable considerable. Ésta estrategia comienza midiendo el round trip time (RTT) de la red y basado en éste se intenta predecir el próximo RTT para así poder presentar una estimación del estado del robot y de éste modo proporcionar mayor información al usuario de modo que éste sea capaz de prever situaciones que se desean evitar o simplemente tener en cuenta que lo que se está mostrando no es exactamente el estado actual, si no que retrasado.

# Descripción del sistema

Se describe una vez más el sistema o método.

## Diagrama de Flujo

Para el caso del usuario se tiene lo siguiente



Figura 1

Una vez comenzada la interfaz e inicializada las variables a utilizar, se obtienen los comandos desde el mando, se envían en un datagrama los comandos y registra el tiempo en que fueron enviados, luego se revisa si hay algún datagrama entrante, si es que existe, se obtiene la información de estado del robot y se actualiza el estado del robot. Se calcula la diferencia de tiempo entre que se envió el último comando y se recibió el estado desde el robot. Luego se actualiza el estado del robot estimado basado en la diferencia de tiempo calculada.

Para el caso del robot no se presenta el diagrama de flujos ya que en éste caso la única tarea de éste es recibir los comandos y enviar el estado.

## Estimación del Retardo

Para la estimación del retardo se mide el round trip time, es decir, el tiempo que demora en completar un ciclo, desde que el comando es enviado desde el cliente, hasta que éste recibe respuesta del servidor correspondiente al mensaje anteriormente enviado. Al medir el retardo de ésta forma no es necesario que el cliente con el servidor estén sincronizados en tiempo, lo cual no es siempre posible, por lo tanto es mejor medir el tiempo que toma el ciclo completo a que medir los retardos por separado (cliente a servidor y viceversa).

## Modelo de Proceso utilizado

Para el desarrollo de este método se escogió un robot móvil uniciclo, el cual posee dos grados de libertad (v = velocidad tangencial, w = velocidad angular).

Las ecuaciones que representan vector de estado del robot y el modelo de proceso, o el movimiento del robot se presentan a continuación

 (1)

 (2)

 En la figura 2 se puede apreciar la geometría del robot móvil (uniciclo) y su modelo cinemático.



Figura 2

# Simulación

Para implementar la simulación se utilizó MATLAB, en donde se configuró un joystick para maniobrar al robot, una sesión como cliente y otra como servidor, en donde la primera representa al usuario u operador y la segunda al robot, y se creó un retardo variable, y finalmente la visualización del estado del robot recibida y la estimación del estado del robot en el estado actual.

## TCP y Retardo

El cliente, o en este caso el usuario, manda los comandos para realizar las acciones deseadas mediante TCP al servidor o robot, y una vez el robot recibe los comandos éste manda al cliente su estado.

Para generar el retardo se utiliza se realiza lo siguiente.



Figura 3

En donde delayPrev es el retardo anterior, delayMean el retardo medio y delayStd es cuanto varia éste retardo medio. Es importante asegurar que los mensajes no se traslapen debido a que es una simulación, y en caso de configurar de forma errónea los retardos es completamente posible que ocurra esto, es por esto que en la generación del retardo se asegura que el siguiente retardo sea a lo menos mayor igual que el anterior.

En la figura 4 se muestra el resultado de la simulación del retardo. Si bien se mide el round trip time en el usuario, las simulaciones se realizaron en dos sesiones de MATLAB, en dos computadores conectados con un cable cruzado, los cuales se comprobó que estuvieran sincronizados para poder realizar la gráfica que se muestra a continuación.



Figura 4

## Control (Joystick)

Se configura un joystick para utilizarlo como mando para así generar una sensación de operación de un robot móvil un poco más cercana a la realidad, donde es posible regular la velocidad (ya sea tangencial o angular) de acuerdo a que tanto se mueve la palanca del joystick.

## Estimación del Retardo

Cómo se mencionó anteriormente el retardo se mide solamente en el cliente, una vez que se recibe la respuesta correspondiente al último comando enviado por el cliente.

Se prueban distintos métodos para estimar el siguiente RTT, se utiliza el promedio, moving average y suavización exponencial.

## Visualización del usuario y entorno

Se crea una pantalla en donde el usuario puede visualizar el último estado actualizado del robot, la estimación de la posición o estado actual del robot, y también es posible visualizar el ambiente en el que el robot se está desplazando (suponiendo que éste ya ha adquirido información del entorno y conoce los obstáculos cercanos).



Figura 5

En la figura 5 se puede ver la interfaz simulada del usuario, en donde la circunferencia roja muestra el movimiento realizado sobre la última actualización del estado recibido desde el robot y la circunferencia azul representa la estimación que se genera a partir de la medición y predicción del siguiente retardo.

# Resultados

Se logra recrear la simulación deseada en el sentido de que posee todas las características básicas que se mencionaron en la sección anterior. Se creó una conexión TCP entre dos computadores conectados por un cable cruzado, cada uno con una sesión de MATLAB, en donde el usuario es capaz de operar al robot mediante el joystick, se calcula el round trip time y se muestra en la interfaz la estimación del estado actual del robot mediante la estimación del retardo.

Debido a ciertas limitaciones del software utilizado para la simulación, si bien se implementa todo lo especificado anteriormente, la simulación no permite estimar de forma concreta si es que se obtiene una ganancia al tener presente la estimación del estado del robot. Esto ocurre principalmente porque la configuración MATLAB para realizar una conexión TCP tiene ciertos parámetros los cuales no son característicos del protocolo, como un Timeout (tiempo máximo que se espera al momento de recibir un mensaje) el cual hay que ajustar, en donde este puede ser mayor a 1 segundo, pero si uno desea un valor menor, el valor más cercano es 0,5 milisegundos. También si bien el protocolo TCP no continúa hasta que todos los paquetes hayan sido entregados, en MATLAB en ciertos casos si se descartan los paquetes (lo cual está directamente relacionado con la configuración de este timeout). Producto de éstas características la estimación y actualización de estados no eran muy fluidas y dependiendo de la velocidad de refrescamiento, el tener una estimación del estado del robot podía llegar a ser molesto.

Sin embargo regulando la velocidad de refrescamiento de la interfaz del usuario acuerdo al retardo medio y variando el valor de Timeout, era posible obtener un mejor desempeño, en donde el tener la estimación beneficia el actuar del operador. No obstante, se deseaba tener una interfaz que ayudara sin importar cual fuera el retardo, por lo que los resultados no son realmente satisfactorios.

En cuanto a la estimación del retardo, no es posible determinar que método es mejor para la predicción del siguiente retardo debido a lo comentado anteriormente.

# Conclusiones

Se logra implementar las características deseadas y en ciertos casos específicos se logra un beneficio con el sistema propuesto. Como se mencionó anteriormente el resultado no es tan satisfactorio como se esperaba. No se presenta ninguna forma de calificar el resultado obtenido ya que es notorio que el funcionamiento no fluido del sistema propuesto es provocado principalmente por las limitaciones que presenta MATLAB al realizar una conexión TCP y tener que ajustar la velocidad de refrescamiento de la interfaz por los factores mencionados anteriormente no es algo que deba ser necesario.

Es esencial realizar futuras pruebas con otros softwares o lenguajes que permitan realizar conexiones entre dos computadores, los cuales sean capaces de simular una mejor representación de los protocolos a utilizar, para así poder determinar cuánto puede llegar a beneficiar realmente éste sistema, ya que si bien no fue del todo satisfactorio, fue posible observar cierta mejora.

References

1. Dan Chen, “Event-based Predictive Control Strategy for Teleoperation via Internet”. International Conference on Advanced Intelligent Mechatronics
2. K.Brady, T.J.Tarn, "Internet-Based Remote Teleoperation", Proceedings of the 1998 IEEE International Conference on Robotics and Automation,

Leuven, Belgium, May 1998

1. H.Kazerooni, " Human-Robot Interaction via the Transfer of Power and Information Signals", IEEE Tran.on Systems, Man and Cybernetics, Vol. 20, No.2. March/April 1990
1. [↑](#footnote-ref-1)