

Mitigación de Retardo en Teleoperación de Robots Móviles

Sebastián Arriagada Pfeiffer

Resumen—Un aspecto crítico en la teleoperación de robots móviles es el retardo que exista en la red, ya que la retroalimentación que recibimos (imagen, sonido, etc.) podría estar atrasada respecto de la realidad. Esto nos podría inducir a realizar malas maniobras y además, nuestra actuación podría llegar demasiado tarde. En este trabajo se presentará un método para estimar el retardo en la red y mitigar sus efectos.

Index Terms—Robótica móvil, teleoperación de robots, retardo en la red.

I. INTRODUCCIÓN

La teleoperación de robots permite realizar muchas tareas que resultan difíciles y/o peligrosas para las personas. Esto afecta especialmente a nuestro país, el cual es el mayor productor de cobre del mundo [1] y tiene como principal rubro económico la minería. Hoy en día, dado que es necesario ir cada vez más profundo para la recolección de minerales, es que se está cambiando la minería a cielo abierto por minería subterránea. Aquí, la robótica móvil teleoperada resulta de gran utilidad, dado que este tipo de minería presenta muchos peligros para los trabajadores.

En cualquier sistema de teleoperación a través de internet, un factor crítico es el retardo que se produce en el envío de información entre la estación base de teleoperación (EB) y el sistema remoto (SR) que está siendo teleoperado, ya sea éste un robot, una máquina u otro. Ciertamente, si este retardo es muy grande, resultarán muy difíciles las maniobras de teleoperación, ya que la información desde el SR llegará retrasada a la EB y las acciones que el operario decida también llegarán retrasadas al SR.

Respecto a este tema existe una amplia investigación y se han propuesto variadas soluciones. Algunas de éstas proponen utilizar redes neuronales para estimar el retardo de la red como en [2] y [3]. Otros acercamientos toman ideas de la teoría de control automático para actuar ante el retardo, utilizando el *Predictor Smith*, tal como se propone en [3], donde se realiza una combinación de redes neuronales con este predictor y en [4]. Más recientemente, han surgido ideas que proponen caracterizar el retardo en base a la energía del canal y mantener una sincronización constante de la posición del SR y la EB, específicamente en [5] proponen utilizar el *Time Domain Passivity Approach* (TDPA) basado en la potencia, propuesto en [6].

En este trabajo se propone una idea más simple para la estimación del retardo, basada en datos históricos y un método

de predicción de posición tanto en la EB como en el SR para mitigar los efectos del retardo en la red.

II. MARCO DE TRABAJO

En este trabajo consideraremos un robot móvil cilíndrico, al cual podemos asignarle una velocidad lineal, ya sea hacia adelante o hacia atrás, y una orientación. En la figura 1 se muestra una imagen del modelo de robot a utilizar, con su velocidad v y su orientación θ .

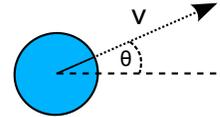


Figura 1: Modelo de robot a utilizar.

Además, consideraremos que el mapa del entorno por el cual se desplazará el robot es conocido. Esto no se aleja de la realidad, ya que, por ejemplo en el contexto de una mina, los túneles de ésta son conocidos. Existen diversas situaciones en las que el mapa del entorno es conocido previamente y por lo tanto, podemos aplicar esta solución.

III. ESTIMACIÓN DE RETARDO

Si bien vimos que se han propuesto algunos métodos bastante sofisticados para realizar la estimación del retardo en la red, en este trabajo utilizaremos un método bastante más simple e intuitivo basado en datos históricos.

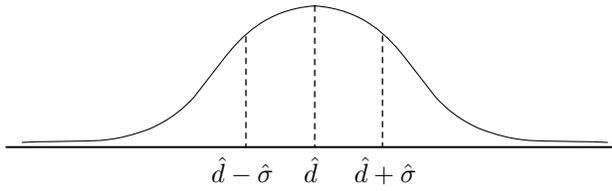
Dada una colección de n mediciones históricas del retardo en la red $\{d_i\}_{i=1}^n$, obtendremos la media utilizando el estimador insesgado de la esperanza

$$\hat{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad (1)$$

Además, utilizaremos un estimador para la varianza lo cual nos dará una idea de qué tan probable es que el retardo esté cerca de la media. El estimador en cuestión será

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (d_i - \hat{d})^2 \quad (2)$$

Con estas estimaciones de media y varianza podemos tener una idea de cómo es la distribución del retardo en la red aproximándola con una función gaussiana tal como se muestra en la figura 2.


 Figura 2: Estimación de la *pdf* del retardo.

En concreto, el valor que utilizaremos como estimación del retardo será la media estimada \hat{d} .

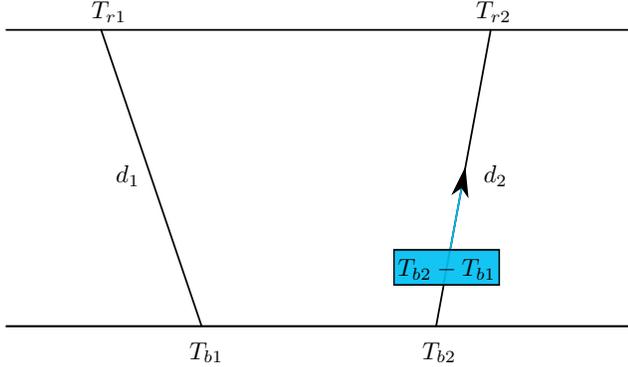


Figura 3: Esquema de medición del retardo en la red.

Para obtener las mediciones históricas del retardo utilizaremos el esquema mostrado en la figura 3. De esta forma, en el SR podemos obtener el retardo completo del enlace como:

$$d_1 + d_2 = (T_{r2} - T_{r1}) - (T_{b2} - T_{b1}). \quad (3)$$

Ciertamente, dado que ambos estimadores son consistentes, es decir,

$$\lim_{n \rightarrow \infty} \text{var}\{\hat{d}\} = 0 \quad \wedge \quad \lim_{n \rightarrow \infty} \text{var}\{\hat{\sigma}\} = 0,$$

mientras mayor sea el número de muestra consideradas en la estimación, mejor será ésta. Sin embargo, si consideramos muchos datos, nuestra estimación será poco sensible a cambios bruscos en los retardos de la red. Por lo tanto, debemos realizar un *trade-off* entre la sensibilidad a cambios y qué tan buena es nuestra estimación.

Dado que, en el sistema implementado el SR envía datos a la EB cada 50[ms], consideraremos 50 datos para las estimaciones. Es decir, el SR debería tener una nueva estimación cada 2.5[s].

IV. PREDICCIÓN DE POSICIÓN EN ESTACIÓN BASE

Dado que el operario en la EB, no conoce la posición real del SR, es necesario hacer algún tipo de predicción de dónde estaría el SR mientras no tenemos datos de su posición real. Esto ayudará considerablemente al operario a maniobrar de mejor manera el SR, ya que al momento de tomar acciones tendrá una idea de la posición del SR.

Para realizar esto en la EB, realizaremos una estimación de acuerdo al modelo de robot presentado en la sección II, el cual sigue el modelo cinemático presentado en la ecuación 4.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_k + \Delta t \begin{bmatrix} v_k \cos \theta_k \\ v_k \sin \theta_k \\ 0 \end{bmatrix} \quad (4)$$

En el sistema implementado para la realización de pruebas, esto se realiza cada 20[ms], y dado que se reciben datos desde el SR cada 50[ms], se realizarán dos de estas predicciones entre una recepción de datos y otra. Esto es, considerando un sistema sin retardo. Ciertamente si agregamos retardo, el número de predicciones entre dos recepciones de datos será variable y dependerá de que tan grande sea el retardo en la red.

V. CORRECCIÓN DE POSICIÓN EN SISTEMA REMOTO

Además de nuestro sistema de predicción en la EB, en el SR se implementa un método de corrección de posición que intentará ajustar la posición del SR a la que el operario observa en la EB. Es decir, si se producen diferencias entre la posición que el operario observa en la EB y la posición real de SR, éste último corregirá su posición para acercarse a la que se observa en la EB incluso cambiando las ordenes recibidas desde la EB de ser necesario.

Dado que el SR es un sistema físico, no le es posible cambiar su posición a la posición observada en la EB instantáneamente. Por lo tanto, se realizará una proyección de dónde creará el operario que estará el SR, dada la posición actual y el comando a ejecutar, dentro de un periodo de tiempo T dado por el retardo de la red.

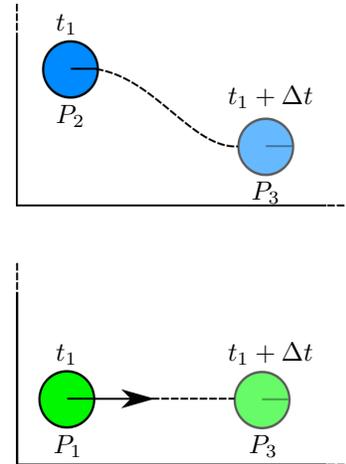


Figura 4: Esquema de corrección de posición en el SR. Arriba se muestran las posiciones del SR (celeste), abajo se muestran las posiciones en la EB (verde).

En la figura 4 se muestra cómo funciona el sistema de corrección en el SR. En el instante t_1 , el operador cree que el robot está en la posición P_1 (en verde) y realiza la acción “seguir derecho”, por lo que el pretende que dentro de un periodo de tiempo Δt el robot esté en la posición P_3 (en verde). Sin embargo, en el instante t_1 el robot real (mostrado en celeste en la figura 4) se encontraba en una posición distinta, P_2 . Por lo tanto, el SR cambia la acción a ejecutar para alcanzar la posición P_2 deseada por el operario.

Para realizar este procedimiento, cada vez que el operario decide realizar una acción, ésta se envía al SR incluyendo la posición, orientación y velocidad que el operario creía que el SR tenía en el momento en que se tomó la acción. Con esta información y con la estimación del retador, el SR predice cual

es la posición P_3 (de la figura 4) a la cual el operario desea ir. Obtenido este punto, se calcula la diferencia de ángulo que existe entre la orientación actual del robot y el punto P_3 , tal como se muestra en la figura 5. La orientación es corregida gradualmente a la velocidad angular ω que permita el sistema físico del robot, y la velocidad lineal se mantiene constante hasta hacer cero la diferencia de ángulo.

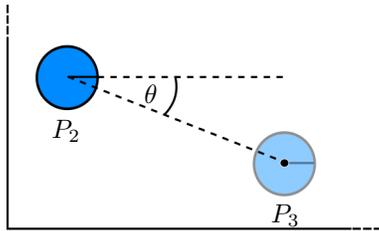
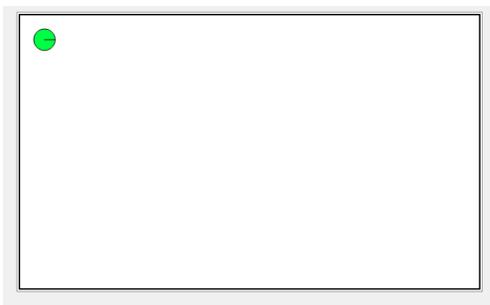


Figura 5: Esquema de corrección de posición en el SR.

VI. RESULTADOS EXPERIMENTALES

Para probar el sistema de mitigación expuesto anteriormente, se crea una interfaz de simulación en C++ utilizando el *framework* Qt [7]. Con ella se realizan dos pruebas a diez voluntarios para probar la efectividad de nuestro sistema de mitigación.

Con este *set-up* se crean dos aplicaciones, una que simula la EB y otra que simula el SR, conectadas a través de un enlace TCP al cual se agrega un retardo aleatorio con distribución gaussiana de media 600[ms] y varianza 100[ms]. Este retardo se agrega tanto a los datos que viajan desde la EB a SR como a los que viajan en la otra dirección (desde SR a EB). Por lo tanto el enlace completo (ida y vuelta) tendrá un retardo medio de 1200[ms].



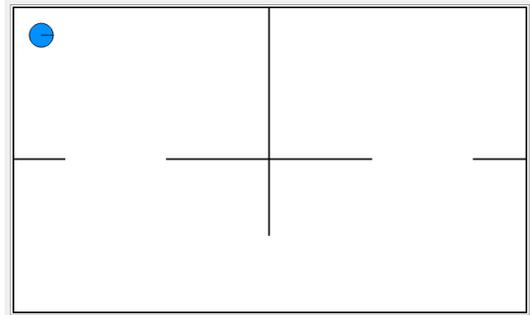
(a) Aplicación cliente (EB).



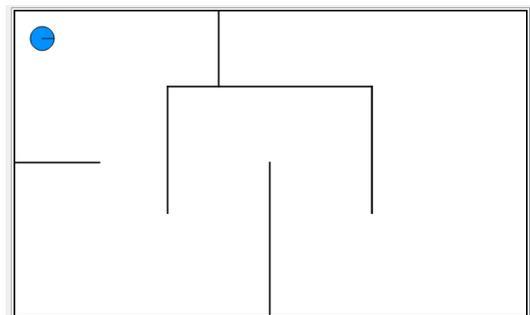
(b) Aplicación servidor (SR).

Figura 6: Aplicaciones de sistema de simulación.

En la figura 6 se muestran una imagen de cada una de las aplicaciones con el modelo del robot expuesto en la sección II. Éstas cuentan con dos mapas distintos, los cuales se muestran en la figura 7.



(a) Mapa 1.



(b) Mapa 2.

Figura 7: Mapas a utilizar.

La prueba para cada uno de los voluntarios consiste en dos minutos de prueba sin retardo en el mapa 1 de la figura 7a. Esto es para que la persona se acostumbre a las maniobras y controles del robot. Transcurridos los dos minutos de práctica, se realizan dos pruebas utilizando el mapa 2 de la figura 7b. La primera consiste en realizar el circuito utilizando un enlace con retardo y sin el sistema de mitigación descrito. En la segunda, se realiza el mismo circuito, pero esta vez utilizando el sistema de mitigación del retardo en la red. En ellas se cronometra el tiempo que el sujeto demora en maniobrar el robot desde el punto inicial (posición en la que se encuentra el robot) hasta el punto final marcado con rojo en las figuras 7, y además, se contabiliza el número de choques que se producen en el trayecto.

Cabe señalar que el circuito para ambas pruebas (con y sin sistema de mitigación) se realiza dos veces. Los resultados expuesto aquí consideran los tiempos y número de choques de la segunda realización. Esto es, para que el sujeto de prueba se acostumbre a maniobrar el robot con las condiciones de retardo que presenta el sistema.

Participante	1	2	3	4	5	6	7	8	9	10
Sin sistema	92	81	101	95	131	88	98	91	110	97
Con sistema	53	55	63	48	59	39	49	45	51	55

Cuadro I: Tiempo en segundos que demora cada uno de los participantes en realizar el circuito sin sistema de mitigación (segunda fila) y con él (tercera fila).

Participante	1	2	3	4	5	6	7	8	9	10
Sin sistema	7	5	10	5	14	6	8	8	10	7
Con sistema	2	2	3	1	2	0	1	0	1	2

Cuadro II: Número de choques de cada uno de los participantes realizando el circuito sin sistema de mitigación (segunda fila) y con él (tercera fila).

En la figura 8 y en el cuadro I se muestran los tiempos que demoraron los diez voluntarios en realizar el circuito, con y sin el sistema de mitigación. En ella se pueda apreciar claramente que utilizando el sistema de mitigación descrito los tiempos del recorrido disminuyen considerablemente.

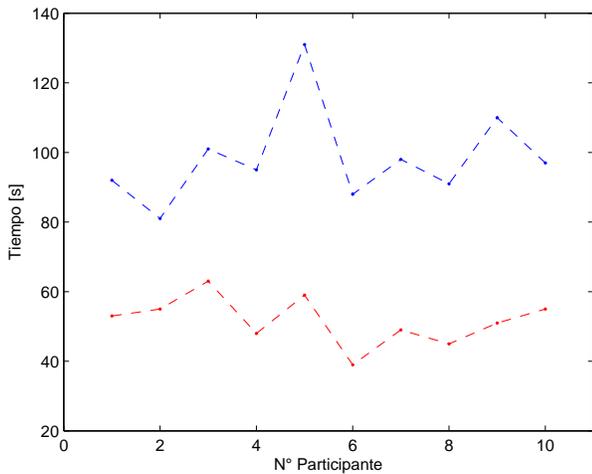


Figura 8: Modelo de robot a utilizar.

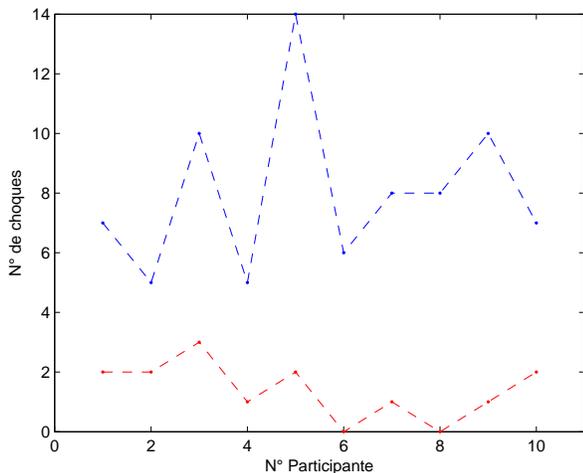


Figura 9: Modelo de robot a utilizar.

Además, en la figura 9 se muestran los choques que obtuvieron los voluntarios al realizar el circuito sin (en azul) y con el sistema de mitigación (en rojo). En este caso también podemos apreciar que los choques disminuyen considerablemente en las pruebas con el sistema de mitigación. De hecho uno de los participantes obtuvo 14 choques durante el recorrido sin el

sistema de mitigación, lo que en un caso real de teleoperación resultaría nefasto.

Finalmente, se les consulto a los voluntarios si se les había hecho más fácil maniobrar el robot utilizando el sistema de mitigación y el 100 % de ellos respondió que si.

VII. CONCLUSIÓN

En este trabajo se presentó un método sencillo pero eficaz para poder mitigar los efectos del retardo en la red al momento de teleoperar sistemas remotos a través de ella. Esto podría resultar extremadamente útil en sistemas que no tengan mucha capacidad de procesamiento para implementar algoritmos más sofisticados. Con el sistema aquí presentado pueden lograr un buen resultado.

Aún existen mejoras que podemos realizar a este sistema, como por ejemplo, mostrar al operario una predicción de donde estaría el robot en un futuro al momento de que él tome una acción de control. Si bien esta propuesta fue revisada y no se obtuvieron los resultados esperados, se pretende visitar como trabajo futuro para poder descartar errores de implementación y/o buscar implementaciones alternativas que permitan obtener un buen desempeño del sistema.

REFERENCIAS

- [1] Edelstein, Daniel L., *Mineral Commodity Summaries - Copper*, U.S. Geological Survey, 2012.
- [2] S.R. Seyed Tabib, Ali A. Jalali, *Modelling and Prediction of Internet Time-delay by Feed-forward Multilayer Perceptron Neural Network*, UKSIM Tenth International Conference on Computer Modelling and Simulation Cambridge UK, 2008.
- [3] H. Kopka and P. W. Daly, *Neural Network-Based Teleoperation using Smith Predictors*, Proceedings of the IEEE International Conference on Mechatronics & Automation, Niagara Falls, Canada, 2005.
- [4] H. Kopka and P. W. Daly, *Real-Time Mobile Robot Teleoperation over IP Networks Based on Predictive Control*, IEEE International Conference on Control & Automation, Guangzhou, China, 2007.
- [5] H. Kopka and P. W. Daly, *Position Synchronization in Bilateral Teleoperation Under Time-Varying Communication Delays*, IEEE/ASME Trans. on Mechatronics, vol. 20, no. 1, Feb. 2015.
- [6] Y. Ye, Y.-J. Pan, and T. Hilliard, *Bilateral teleoperation with time-varying delay: A communication channel passification approach*, IEEE/ASME Trans. on Mechatronics, vol. 18, no. 4, Aug. 2013.
- [7] Cross-platform application and UI development framework for leading desktop, embedded and mobile operating systems, <http://www.qt.io/>.