

Teach and Repeat con calibración odométrica para robots omnidireccionales con sensor LiDAR

Sebastián Bedín
Instituto de Ciencias
de la Computación (ICC-CONICET)
Universidad de Buenos Aires,
Buenos Aires, Argentina
Email: sbedin@dc.uba.ar

Javier Civera
IA,
Universidad de Zaragoza,
España
Email: jcivera@unizar.es

Matías Nitsche
Instituto de Ciencias
de la Computación (ICC-CONICET)
Universidad de Buenos Aires,
Buenos Aires, Argentina
Email: mnitsche@icc.fcen.uba.ar

Abstract—En este trabajo, presentamos un método novedoso de *teach-and-repeat* (T&R) para vehículos terrestres autónomos de tracción omnidireccional, basado en la fusión fuertemente acoplada de mediciones de LiDAR y odometría bajo una formulación relativa. Además de la localización del robot, realizamos una estimación en línea de los parámetros intrínsecos de la plataforma, lo que mejora significativamente la robustez y precisión de la localización del sistema. Demostramos la efectividad de nuestro enfoque, incluyendo el resultado de los parámetros de la plataforma y la estimación de la pose, así como el método general de T&R, mediante simulación.

I. INTRODUCCIÓN

Los vehículos terrestres no tripulados se utilizan en diversas formas y en numerosos ámbitos de aplicación, incluyendo el aspirado autónomo o el transporte, entre otros. De particular relevancia es su aplicación en depósitos o fábricas, donde el objetivo es lograr el mayor grado de automatización posible. En este contexto, los robots de tracción omnidireccional poseen una ventaja significativa debido a su movimiento 2D sin restricciones. Dada la naturaleza repetitiva de las tareas de los robots y la conveniencia de definir sus trayectorias a través de la demostración, los métodos de navegación *teach-and-repeat* (T&R) tienen una gran relevancia. Estos aspectos sirven de motivación para nuestro trabajo.

Aunque los sensores LiDAR destacan en robustez en comparación con las cámaras, están limitados en ciertas configuraciones geométricas como pasillos largos o entornos grandes con objetos fuera de su alcance, que son comunes en entornos como los mencionados. Por otro lado la precisión de la odometría está limitada por factores como: el deslizamiento de las ruedas, acentuado en robots de tracción omnidireccional debido a la naturaleza de sus ruedas con rodillos. La deformación de las ruedas por desgaste y cambios a la carga de la plataforma. O la variabilidad entre distintos robots parte de una flota.

En este trabajo, formulamos un método novedoso de T&R para robots omnidireccionales que fusiona, de manera fuertemente acoplada las mediciones de LiDAR y odometría en una representación relativa adecuada para el problema planteado. Al fusionar los dos sensores superamos las limitaciones de ambos. También incluimos parámetros específicos de la plataforma en el estado, para auto-calibrar la odometría. En

nuestros experimentos de simulación, demostramos la efectividad de nuestro enfoque en varios escenarios desafiantes.

II. TRABAJO RELACIONADO

La navegación T&R es una técnica clásica en la robótica autónoma y ha sido abordada a lo largo de los años desde varias perspectivas. La mayoría de las investigaciones se han centrado en robots terrestres equipados ya sea con LiDAR [9], [10], [1] o sensores visuales [14], [15], [20], [17]. Por otro lado, los trabajos que abordan otros tipos de locomoción, como submarinos [8] o vehículos aéreos [16], [12], son menos comunes.

En terminos de sensores a bordo del robot, en la literatura se ha explorado una amplia variedad: cámaras monoculares [20], estéreo [6], LiDAR [9], [19], [10], [1], radar [18] y combinaciones multi-modales como LiDAR-radar [1] o estéreo-inercial [7], [12]. Hasta donde sabemos, el enfoque que formulamos en este trabajo utilizando LiDAR 2D y *encoders* en rueda *mecanum* no ha sido abordado en la literatura. En particular, la auto-calibración simultánea de los parámetros de la plataforma, que proponemos en este trabajo, también es novedosa con respecto a trabajos anteriores.

III. MÉTODO

El método de navegación propuesto se basa en la técnica T&R presentada en [11], [12]. En T&R, se demuestra primero una trayectoria objetivo durante la fase de aprendizaje (*teach*) y luego es seguida en forma autónoma por un vehículo durante la fase de repetición (*repeat*). La localización incremental es utilizada durante ambas etapas para construir un mapa compuesto por *keyframes*. Durante la fase de *teach*, se generan *keyframes* los cuales definen la trayectoria objetivo. Durante la fase de *repeat*, el mapa construido en forma incremental se relaciona continuamente contra el mapa de referencia construido durante la fase previa. Así, el robot puede ser controlado para seguir la trayectoria objetivo.

En este trabajo, extendemos la técnica básica T&R para dirigir robots terrestres que cuentan con *encoders* de ruedas y un sensor LiDAR, mientras seguimos la misma formulación relativa del problema donde el mapa está compuesto por transformaciones relativas encadenadas. En este contexto, la

información odométrica que proviene de los *encoders* de las ruedas se incluye en la estimación siguiendo el conocido enfoque de preintegración [5], [4], que se complementa perfectamente con la formulación relativa propuesta. Además, para mejorar la información odométrica, incluimos la estimación en línea del radio de las ruedas. Para tratar los datos del sensor LiDAR, elegimos un enfoque simple y efectivo basado en el algoritmo de registro de escaneo *Iterative Closest Point* (ICP) [3]. Al calcular la transformación relativa a partir de un par de nubes de puntos LiDAR correspondientes a *keyframes* consecutivos, obtenemos una medición de pose relativa. Asimismo mostramos también cómo se puede obtener la incertidumbre correspondiente del algoritmo ICP.

IV. LOCALIZACIÓN INCREMENTAL

El objetivo de la localización incremental es estimar la pose actual del robot en relación con el *keyframe* más reciente del mapa actual, mientras se registran simultáneamente un conjunto de *keyframes* que describen la trayectoria del robot y se calibran los parámetros intrínsecos de la plataforma. Como se mencionó anteriormente, utilizamos un enfoque relativo, lo que significa que el estado a estimar se define como la pose relativa entre el *keyframe* actual y el anterior en la trayectoria:

$$\mathbf{x} = \left\{ \left\{ \mathbf{T}_i^{i+1} \right\}_{i=1 \dots n-1}, \mathbf{r} \right\}, \quad (1)$$

donde

$$\mathbf{T}_i^{i+1} = \begin{bmatrix} \mathbf{R}_i^{i+1} & \mathbf{t}_i^{i+1} \\ \mathbf{0} & 1 \end{bmatrix} \in \text{SE}(3)$$

representa la transformación relativa entre dos *keyframes* consecutivos denotados como i y $i+1$, y $\mathbf{r} \in \mathbb{R}_{>0}^4$ representa el vector de los radios de las ruedas desconocidos, en nuestro caso de un robot de tracción omnidireccional de cuatro ruedas.

A. Iterative Closest Point

Dado un par de nubes de puntos \mathbf{P} y \mathbf{Q} en dos marcos de referencia locales diferentes, el algoritmo ICP (Iterative Closest Point) nos permite encontrar una transformación relativa $\mathbf{L} \in \text{SE}(3)$ que alinea \mathbf{P} con \mathbf{Q} con un error mínimo. En otras palabras, nuestro objetivo es encontrar la transformación \mathbf{L} que satisface la siguiente ecuación:

$$\mathbf{L} = \underset{\mathbf{L}'}{\operatorname{argmin}} C(\mathbf{Z}, \mathbf{L}') , \quad (2)$$

donde $\mathbf{Z} = [\mathbf{P} \quad \mathbf{Q}]$ y

$$C(\mathbf{Z}, \mathbf{L}) = \sum_{j=1}^n [(\mathbf{R}\mathbf{p}_j + \mathbf{t} - \mathbf{q}_j) \mathbf{n}_j]^2 \quad (3)$$

corresponde a la métrica de error *punto-a-plano* definida en [3]. \mathbf{R} y \mathbf{t} se corresponden, respectivamente, con la parte de rotación y translación de \mathbf{L} . Las nubes de puntos $\mathbf{P} = \{\mathbf{p}_j\}_{j=1 \dots n}$, $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1 \dots n}$ contienen puntos 3D $\mathbf{p}_j, \mathbf{q}_j \in \mathbb{R}^3$, y $\mathbf{n}_j \in \mathbb{P}^3$ representa la normal de un plano construido a partir de los k vecinos más cercanos de \mathbf{q}_j .

B. Residuo de LiDAR

En el contexto de la localización de robots basada en LiDAR, el algoritmo ICP puede utilizarse para determinar el movimiento del robot a partir de un par de escaneos láser adquiridos desde dos posiciones distintas. Para establecer la correspondencia entre los puntos en ambas nubes de puntos, se lleva a cabo un proceso de emparejamiento seleccionando el vecino entre los puntos de ambas nubes (donde se ignoran los puntos que no se emparejan).

Modelamos el residuo del sensor LiDAR de la siguiente manera:

$$\mathbf{r}_{\Delta\mathbf{L}} = \log \left((\Delta\mathbf{L}_i^{i+1})^{-1} \mathbf{T}_i^{i+1} \right)^\vee \quad (4)$$

donde $\Delta\mathbf{L}_i^{i+1}$ se obtiene utilizando el algoritmo ICP (como se describe en la sección anterior) a partir de lecturas de nubes de puntos adquiridas en los *keyframes* i e $i+1$. Como semilla inicial para la minimización de ICP, utilizamos la transformación relativa obtenida de las lecturas odométricas integradas entre i e $i+1$ (ver siguiente sección).

Para modelar la incertidumbre de $\Delta\mathbf{L}$ (donde eliminamos los índices por claridad), adoptamos el enfoque descrito en [2]. Específicamente, tenemos en cuenta la incertidumbre inicial en las lecturas de LiDAR propagando su covarianza a través de una aproximación de primer orden:

$$\Sigma_{\Delta\mathbf{L}} = \mathbf{J}_{\mathbf{Z}}^{\Delta\mathbf{L}} \Sigma_{\mathbf{Z}} \left(\mathbf{J}_{\mathbf{Z}}^{\Delta\mathbf{L}} \right)^\top, \quad (5)$$

donde $\Sigma_{\mathbf{Z}}$ es la covarianza de las nubes de puntos, que se encuentra usualmente en la hoja de datos del sensor LiDAR.

Si bien $\Delta\mathbf{L}$ no presenta una forma cerrada, ya que se obtiene como resultado de un algoritmo de minimización (2), $\Delta\mathbf{L}$ y \mathbf{Z} están relacionados por una función implícita. Además, sabemos que el gradiente de la función de costo C es cero en \mathbf{L} , por lo tanto su gradiente es nulo:

$$\frac{\delta C|_{\mathbf{L}}}{\delta \mathbf{L}} = \mathbf{0}^T \quad (6)$$

Como resultado, como se demuestra en [2], podemos aplicar el teorema de la función implícita y finalmente obtener $\mathbf{J}_{\mathbf{Z}}^{\Delta\mathbf{L}}$:

$$\mathbf{J}_{\mathbf{Z}}^{\Delta\mathbf{L}} = - \left(\frac{\delta^2 C(\mathbf{L}, \mathbf{Z})}{\delta \mathbf{L}^2} \right)^{-1} \frac{\delta^2 C(\mathbf{L}, \mathbf{Z})}{\delta \mathbf{Z} \delta \mathbf{L}} \Big|_{\mathbf{L}=\Delta\mathbf{L}} \quad (7)$$

C. Preintegración de Odometría

Para incluir información odométrica en la estimación, seguimos el enfoque de preintegración propuesto originalmente para mediciones inerciales [5]. Primero, definimos una lectura de los *encoders* \mathbf{q} como:

$$\mathbf{q} = [\varphi_{fr} \quad \varphi_{fl} \quad \varphi_{rl} \quad \varphi_{rr}]^T \quad (8)$$

donde $\varphi_{fr}, \varphi_{fl}, \varphi_{rl}$ y φ_{rr} representan los incrementos rotacionales para las ruedas delanteras-derecha, delanteras-izquierda, traseras-izquierda y traseras-derecha, respectivamente, según lo medido por los *encoders*.

Luego, definimos el modelo cinemático de la plataforma de tracción omnidireccional $\mathbf{K}(\mathbf{c})$ como:

$$\mathbf{K}(\mathbf{c}) = \frac{1}{4} \begin{bmatrix} r_{fr} & r_{fl} & r_{rl} & r_{rr} \\ r_{fr} & -r_{fl} & r_{rl} & -r_{rr} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{r_{fr}}{d} & \frac{-r_{fl}}{d} & \frac{-r_{rl}}{d} & \frac{r_{rr}}{d} \end{bmatrix} \quad (9)$$

con

$$\mathbf{c} = \begin{bmatrix} \mathbf{r} & d \end{bmatrix}^\top$$

$$d = \frac{d_1 + d_2}{2}$$

donde d_1 y d_2 representan la separación horizontal y lateral de las ruedas, respectivamente.

El movimiento incremental del robot, denotado como $\mathbf{O}(\mathbf{q}, \mathbf{c})$, se calcula entonces de la siguiente manera:

$$\mathbf{O}(\mathbf{q}, \mathbf{c}) = \exp(\mathbf{b}(\mathbf{q}, \mathbf{c})^\wedge) \in \text{SE}(3) \quad (10)$$

$$\mathbf{b}(\mathbf{q}, \mathbf{c}) = \mathbf{K}(\mathbf{c})\mathbf{q} \in \text{se}(3) \quad (11)$$

Para lecturas de *encoder* sucesivas $\mathbf{q}_1, \dots, \mathbf{q}_n$ adquiridas entre los *keyframes* i e $i+1$, podemos entonces construir una medida preintegrada $\Delta \mathbf{T}_i^{i+1}$ como:

$$\Delta \mathbf{T}_i^{i+1} = \mathbf{O}(\mathbf{q}_1, \mathbf{c}) \dots \mathbf{O}(\mathbf{q}_n, \mathbf{c}) \quad (12)$$

$$= \mathbf{O}_1^{n-1} \mathbf{O}(\mathbf{q}_n, \mathbf{c}) \quad (13)$$

Finalmente, el residuo de preintegración de la odometría se puede definir como:

$$\mathbf{r}_{\Delta \mathbf{T}} = \log((\Delta \mathbf{T}_i^{i+1})^{-1} \mathbf{T}_i^{i+1})^\vee \quad (14)$$

D. Estimación del Radio de Rueda

Para realizar la estimación en línea del radios de las ruedas \mathbf{r} , podemos adoptar un enfoque similar al descrito en [5] para estimar los términos de sesgo de la IMU. Específicamente, podemos preintegrar las lecturas de odometría utilizando un valor inicial de $\bar{\mathbf{r}}$, e incorporar un término de corrección para tener en cuenta las actualizaciones a \mathbf{r} durante el proceso de estimación. Así, podemos definir:

$$\bar{\mathbf{c}} = \begin{bmatrix} \bar{\mathbf{r}} & d \end{bmatrix}^\top \quad (15)$$

donde d se obtiene antes de la estimación y se mantiene fijo, ya que no se espera que cambie con el tiempo.

De manera similar a [5], y eliminando los índices en $\Delta \mathbf{T}_i^{i+1}$ por claridad, utilizamos la siguiente aproximación para (13):

$$\Delta \mathbf{T}(\mathbf{q}, \mathbf{c}) \simeq \Delta \mathbf{T}(\mathbf{q}, \bar{\mathbf{c}}) \mathbf{J}_c^{\Delta \mathbf{T}}(\mathbf{q}, \mathbf{c} - \bar{\mathbf{c}}) \quad (16)$$

donde $\bar{\mathbf{c}}$ es el valor de \mathbf{c} en el momento en que se inicializa la medición de preintegración y

$$\mathbf{J}_c^{\Delta \mathbf{T}} = \mathbf{J}_{\mathbf{O}_1^{n-1}}^{\Delta \mathbf{T}} \mathbf{J}_c^{\mathbf{O}_1^{n-1}} + \mathbf{J}_c^{\mathbf{O}} \quad (17)$$

donde

$$\mathbf{J}_{\mathbf{O}_1^{n-1}}^{\Delta \mathbf{T}} = \mathbf{J}_{\mathbf{O}_1^{n-1}}^{\oplus} \quad (18)$$

$$\mathbf{J}_c^{\mathbf{O}} = \mathbf{J}_r(\mathbf{b}) \mathbf{J}_c^{\mathbf{b}} \quad (19)$$

$$\mathbf{J}_c^{\mathbf{b}} = \frac{1}{4} \begin{bmatrix} \varphi_{fr} & \varphi_{fl} & \varphi_{rl} & \varphi_{rr} \\ \varphi_{fr} & -\varphi_{fl} & \varphi_{rl} & -\varphi_{rr} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\varphi_{fr}}{d} & \frac{-\varphi_{fl}}{d} & \frac{-\varphi_{rl}}{d} & \frac{\varphi_{rr}}{d} \end{bmatrix} \quad (20)$$

con $\mathbf{J}_r(\mathbf{b})$ el jacobiano derecho de $\text{SE}(3)$ evaluado en \mathbf{b} .

E. Covarianza de Odometría

Modelamos el ruido en los *encoders* incrementales como:

$$\mathbf{q} = \tilde{\mathbf{q}} + \epsilon_{\mathbf{q}} \quad (21)$$

donde $\tilde{\mathbf{q}}$ es la medida real del *encoder* y $\epsilon_{\mathbf{q}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{q}})$. Además, dado que asumimos un valor previamente estimado para d , también consideramos su contribución a la covarianza propagada como $d = \tilde{d} + \epsilon_d$, $\epsilon_d \sim \mathcal{N}(0, \sigma_d)$.

Luego, podemos propagar la covarianza de un solo paso de integración (16) como:

$$\Sigma_{\mathbf{O}} = \mathbf{J}_{\mathbf{O}}^{\mathbf{O}} \Sigma_{\mathbf{q}} (\mathbf{J}_{\mathbf{O}}^{\mathbf{O}})^\top + \mathbf{J}_c^{\mathbf{O}} \Sigma_c (\mathbf{J}_c^{\mathbf{O}})^\top \quad (22)$$

donde $\Sigma_c = \mathbf{I}_5 [0 \ \sigma_d]^\top$ y $\mathbf{J}_{\mathbf{O}}^{\mathbf{O}} = \mathbf{J}_{\mathbf{b}}^{\mathbf{O}} \mathbf{J}_c^{\mathbf{b}} = \mathbf{J}_r(\mathbf{b}) \mathbf{K}(\mathbf{c})$. Nótese que las covarianzas del radio de las ruedas no se consideran en este paso ya que estas variables forman parte de la estimación, a diferencia de las separaciones entre las ruedas, que son información previa. Las covarianzas del radio de las ruedas se obtienen como resultado del proceso de estimación.

La covarianza para un solo incremento puede acumularse a la medición completa preintegrada (13) de manera similar a como se hace en [5].

F. Repetición

Para la fase de repetición, adoptamos una metodología similar a la descrita en [12], aunque con adaptaciones para los diferentes sensores y el tipo de plataforma robótica involucrada. Proporcionamos primero un breve resumen del enfoque estándar de localización en el mapa.

La fase de repetición consta de dos procedimientos de localización distintos: 1) el método de localización incremental, que ya ha sido descrito en la sección anterior, y 2) el proceso de localización global sobre el mapa estimado en la fase de aprendizaje. Para realizar la localización global, seleccionamos el *keyframe* K_m del mapa aprendido que está más cercano al *keyframe* más reciente K_l en la fase de repetición, que luego se designa como el *keyframe* de referencia del mapa. Este *keyframe* está sujeto a un cambio continuo a medida que el robot se mueve y se generan nuevos *keyframes* durante el proceso de localización incremental en la fase de repetición.

Ejecutando el algoritmo de ICP entre los escaneos de ambos *keyframe* K_m y K_l , podemos determinar la transformación relativa \mathbf{T}_m^l entre ellos. Posteriormente, para establecer la pose

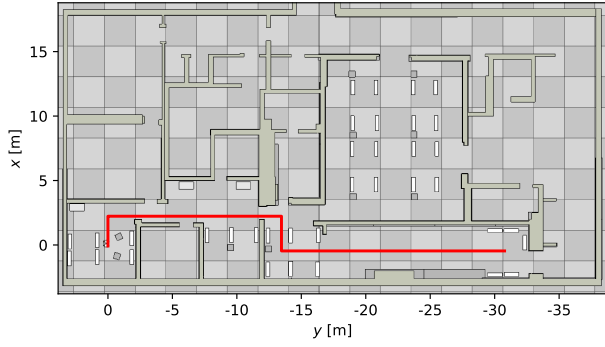


Fig. 1. Vista superior del entorno de simulación y trayectoria de enseñanza.

actual del robot con respecto al mapa anterior, combinamos \mathbf{T}_m^l con la transformación más reciente \mathbf{T}_l^n en la localización incremental en la fase de repetición (última pose en (1)) como:

$$\mathbf{T}_m^n = \mathbf{T}_k^l \mathbf{T}_l^n \quad (23)$$

Esta separación en dos hilos de localización distintos nos permite mantener una pose actualizada \mathbf{T}_k^n en todo momento, incluso durante fallos temporales de la localización global. En tal caso, la pose encontrada previamente \mathbf{T}_k^* se extiende (predice) con transformaciones relativas más nuevas \mathbf{T}_{n+1}^* , \mathbf{T}_{n+2}^* , etc. Esto añade robustez a todo el sistema T&R e incluso permite desviaciones del mapa aprendido si fueran necesarias (por ejemplo, para evitar un obstáculo no presente previamente).

A partir del conjunto de *keyframes* cercanos al *keyframe* de referencia del mapa, se construye una trayectoria suave que tiene en cuenta las velocidades lineales y rotacionales registradas durante la fase de aprendizaje. Esto permite evaluar la trayectoria en cualquier momento t y obtener un punto de control con el fin de seguir suavemente el camino aprendido (para más detalles ver [12]).

Dado que las contribuciones de este trabajo se centran principalmente en el algoritmo de localización incremental, optamos por una estrategia de control directa basada en un controlador PI de posición/orientación para nuestros experimentos. Dicho controlador opera de manera independiente en cada uno de los tres ejes controlables del robot.

V. RESULTADOS

A. Configuración Experimental

Para evaluar el rendimiento del método T&R propuesto, realizamos una serie de experimentos en el simulador *Coppelia V-Rep*. La plataforma robótica utilizada fue el *KUKA youBot*, que cuenta con un sistema de locomoción omnidireccional compuesto por cuatro ruedas de tipo *mechanum*. Las lecturas de los *encoders* de las ruedas simuladas se obtuvieron convirtiendo la rotación de las ruedas en 2000 *ticks* discretos. Para el LiDAR elegimos el sensor *Hokuyo URG-04LX*, el cual proporciona lecturas a 50Hz hasta una distancia de 4 m sobre un ángulo de 240°. Introducimos ruido en las lecturas del LiDAR perturbando la distancia de cada lectura con ruido

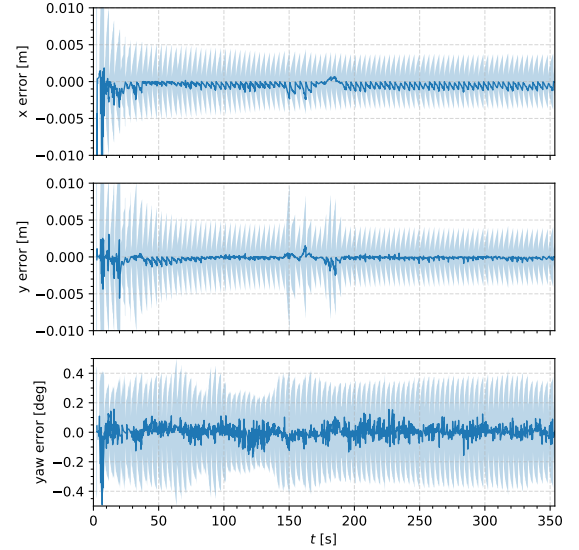


Fig. 2. Errores de pose relativa (en azul oscuro) e incertidumbres estimadas (en azul claro).

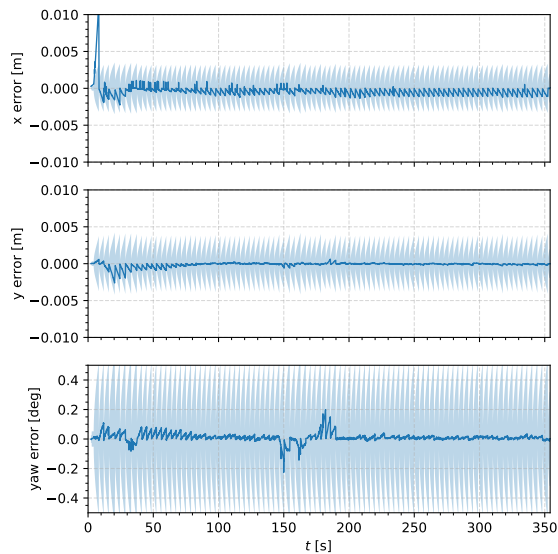
gaussiano, con una incertidumbre del 2% sobre la distancia, como se propuso en [13].

El entorno simulado corresponde a un modelo de una región del *Centro de Conferencias Marcum* de la Universidad de Miami (ver Figura 1). Los experimentos consisten en una fase de demostración inicial sobre el entorno y posteriores ejecuciones con control a lazo cerrado de la fase de repetición.

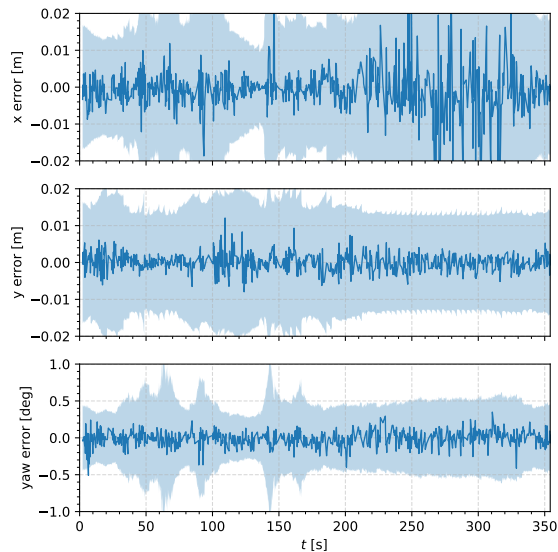
B. Localización Incremental y Calibración en Línea

En esta sección, evaluamos la calidad de la localización incremental mediante la evaluación del error de la última transformación relativa estimada frente al *ground truth*, así como los valores de los radios de las ruedas estimados (comenzando con un error inicial de 2 mm para cada rueda). La Figura 2 muestra el error de pose (en azul oscuro), junto con la covarianza de pose correspondiente (ilustrada con un intervalo de confianza del 99% en azul claro). Además, presentamos la pose relativa $\Delta\mathbf{T}$ y \mathbf{L} , junto con su covarianza (fig. 3(a), 3(b)), que proporciona información sobre el rendimiento de cada modalidad de sensor a lo largo del tiempo y su relación con la pose estimada final. Por último, la Figura 4 ilustra el error y las incertidumbres estimadas de los radios de las ruedas durante la calibración en línea.

Los resultados presentados muestran que, en general, los errores de localización son pequeños y consistentes con las incertidumbres estimadas. Al analizar la convergencia de los radios de las ruedas, se observa que el error inicial se reduce rápidamente. Entre los intervalos de tiempo de $t = 255$ s y $t = 307$ s, cuando el robot navega por un pasillo largo, el error en la dirección x así como la incertidumbre de la pose obtenida de ICP aumentan significativamente. Sin embargo, dado que la pose obtenida de la preintegración de odometría no se ve afectada por esta situación, la estimación de pose después de la fusión tiene un error bajo que es consistente con la baja varianza estimada.



(a) Errores de pose relativa (en azul oscuro) e incertidumbres estimadas (en azul claro) para la preintegración de odometría.



(b) Errores de pose relativa (en azul oscuro) y incertidumbres estimadas (en azul claro) para ICP.

Fig. 3. Poses relativas de la preintegración de odometría y ICP.

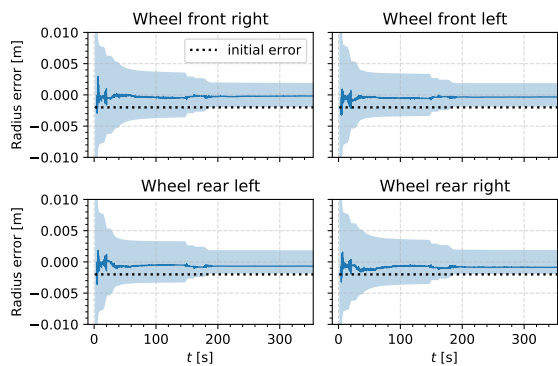


Fig. 4. Resultados de la estimación de los radios de las ruedas (errores en azul oscuro, incertidumbres en azul claro).

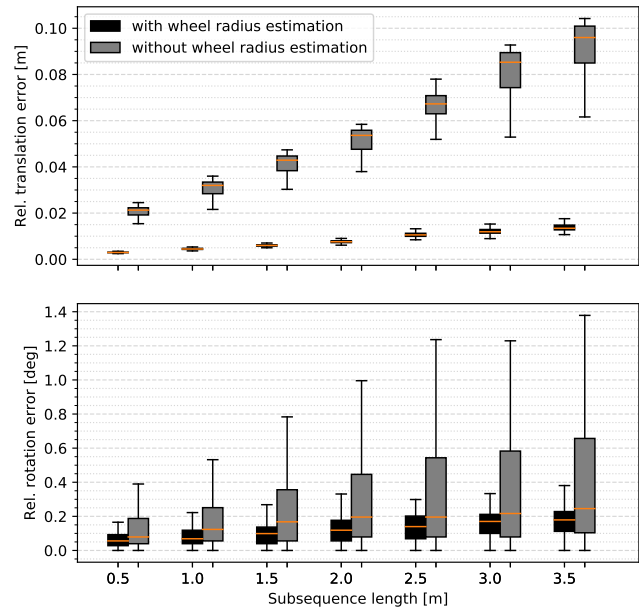


Fig. 5. Diagramas de caja y bigotes que muestran el Error de Pose Relativa (RPE) en subsecuencias de longitud creciente, con y sin estimación de radios de ruedas. Observa la mejora sustancial cuando se estima los radios de ruedas.

Realizamos un análisis adicional del error de localización midiendo el Error de Pose Relativa (RPE), como se define en [12], para subsecuencias de longitudes crecientes (Figura 5), tanto teniendo habilitada y deshabilitada la estimación en línea de los radios de las ruedas. Dado que nuestro enfoque utiliza un método relativo y la integración de la lectura odométrica acumula error entre un par de *keyframes*, este análisis nos permite observar la tasa a la que el error se acumula con respecto a la distancia. De esta manera, podemos hacer un equilibrio entre la precisión de la localización y la precisión del recorrido al decidir a qué distancia colocar los *keyframes*. De la Figura 5 podemos observar el impacto en la estimación de movimiento odométrico que surge de un error relativamente pequeño en los radios de las ruedas, además de cómo mejora el rendimiento general de la localización al realizar la estimación en línea de estos parámetros. Esto es cierto incluso para distancias relativamente cortas y particularmente para el componente translacional.

C. Navegación Autónoma durante la fase de Repetición

Para la fase de repetición permitimos que el sistema controle el movimiento del robot para analizar la localización contra el mapa previamente construido durante la fase de enseñanza, como fue descrito en la sección anterior. Para esto presentamos el error entre la pose relativa del robot con respecto al mapa previo (como se explica en la sección IV-F, para más detalles ver [11]) en comparación con la información de *ground truth*, así como su incertidumbre (ver figura 6). También comparamos la trayectoria seguida por el robot en comparación con la de la fase de aprendizaje en la Figura 7.

De manera similar a los resultados obtenidos para la fase de enseñanza, podemos ver que el error en la pose del robot

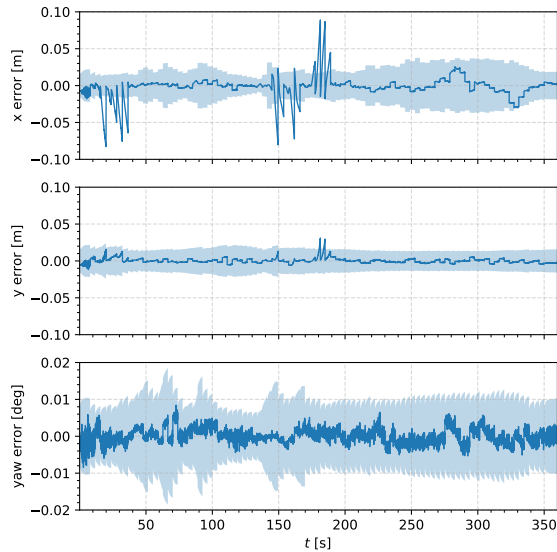


Fig. 6. Error (azul oscuro) y estimación de incertidumbres (azul claro) entre poses relativas en la fase de repetición.

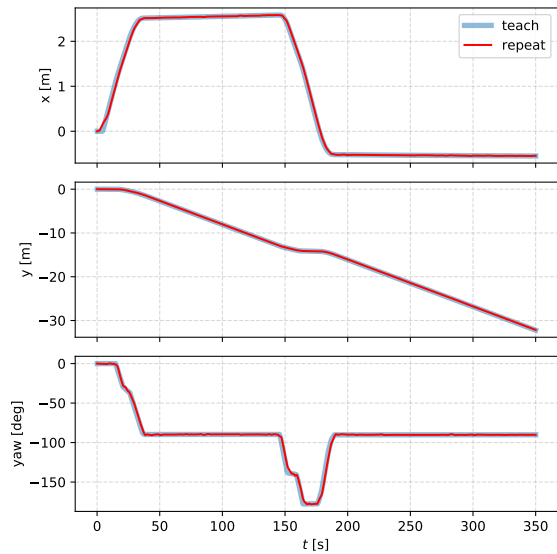


Fig. 7. Trayectorias de enseñanza y repetición. Observar que, utilizando nuestro enfoque, la trayectoria de repetición sigue de cerca la de enseñanza.

con respecto al mapa previo es generalmente pequeño y las incertidumbres estimadas son consistentes con el mismo. Además, podemos ver que el robot sigue con éxito la ruta enseñada previamente.

VI. CONCLUSIONES

En este trabajo presentamos una nueva técnica T&R orientada a plataformas robóticas omnidireccionales, fusionando información de sensores LiDAR y *encoders*, bajo una formulación relativa. Nuestros resultados demuestran que la integración de ambas modalidades de sensado no solo permite una localización robusta en entornos donde los sensores LiDAR a menudo presentan limitaciones, sino que también facilita la estimación en tiempo real de los radios de las ruedas de

la plataforma. Esto, a su vez, mejora significativamente la precisión de la estimación de movimiento basada en el modelo odométrico.

Como parte de nuestro trabajo futuro, planeamos realizar experimentos en un entorno real para validar nuestro enfoque. Además, tenemos como objetivo mejorar nuestro enfoque de calibración en línea para adaptarse a cambios en los radios de las ruedas que puedan surgir debido a variaciones en la carga durante las operaciones de transporte. También pretendemos refinar nuestra estrategia de control del robot para mejorar aún más el rendimiento del método.

AGRADECIMIENTOS

Este trabajo fue financiado por el proyecto PID2021-127685NB-I00 y TED2021-131150B-I00 y el proyecto DGA_FSE-T45_20R del Gobierno de Aragón.

REFERENCES

- [1] Keenan Burnett, Yuchen Wu, David J Yoon, Angela P Schoellig, and Timothy D Barfoot. Are we ready for radar to replace lidar in all-weather mapping and localization? *IEEE Robotics and Automation Letters*, 7(4):10328–10335, 2022.
- [2] Andrea Censi. An accurate closed-form estimate of icp’s covariance. In *IEEE International Conference on Robotics and Automation*, page 3167–3172, 2007.
- [3] Yang Chen and Gerard Medioni. Object Modeling by Registration of Multiple Range Images. In *IEEE Conf. on Robotics and Automation*, 1991.
- [4] Jeremie Deray, Joan Solà, and Juan Andrade-Cetto. Joint on-manifold self-calibration of odometry model and sensor extrinsics using pre-integration. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2019.
- [5] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.
- [6] Paul Furgale and Timothy D. Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, (2006):1–27, 2010.
- [7] Fei Gao, Luqi Wang, Boyu Zhou, Xin Zhou, Jie Pan, and Shaojie Shen. Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments. *IEEE Transactions on Robotics*, 36(5):1526–1545, 2020.
- [8] Peter King, Andrew Vardy, and Alexander L Forrest. Teach-and-repeat path following for an autonomous underwater vehicle. *Journal of Field Robotics*, 35(5):748–763, 2018.
- [9] Philipp Krüsi et al. Lighting-invariant Adaptive Route Following Using Iterative Closest Point Matching. *JFR*, 32(4):534–564, 2015.
- [10] Colin McManus, Paul Furgale, Braden Stenning, and Timothy D Barfoot. Lighting-invariant visual teach and repeat using appearance-based lidar. *Journal of Field Robotics*, 30(2):254–287, 2013.
- [11] Matias Nitsche, Facundo Pessacg, and Javier Civera. Visual-inertial teach & repeat for aerial robot navigation. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2019.
- [12] Matias Nitsche, Facundo Pessacg, and Javier Civera. Visual-inertial teach and repeat. *Robotics and Autonomous Systems*, 131:103577, 2020.
- [13] Yoichi Okubo, Cang Ye, and Johann Borenstein. Characterization of the Hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation. In *SPIE Unmanned Systems Technology XI*, 2009.
- [14] Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Visual teach and repeat, repeat, repeat: Iterative learning control to improve mobile robot path tracking in challenging outdoor environments. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–181. IEEE, 2013.
- [15] Michael Paton, Kirk MacTavish, Michael Warren, and Timothy D. Barfoot. Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In *IROS*, 2016.
- [16] Andreas Pfrunder, Angela P. Schoellig, and Timothy D Barfoot. A Proof-of-Concept Demonstration of Visual Teach and Repeat on a Quadcopter Using an Altitude Sensor and a Monocular Camera. In *CRV*, 2014.

- [17] Zdeněk Rozsypálek, Tomáš Rouček, Tomáš Vintr, and Tomáš Krajník. Multidimensional particle filter for long-term visual teach and repeat in changing environments. *IEEE Robotics and Automation Letters*, 8(4):1951–1958, 2023.
- [18] Ștefan Săftescu, Matthew Gadd, Daniele De Martini, Dan Barnes, and Paul Newman. Kidnapped radar: Topological radar localisation using rotationally-invariant metric learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4358–4364. IEEE, 2020.
- [19] Christoph Sprunk, Gian Diego Tipaldi, Andrea Cherubini, and Wolfram Burgard. Lidar-based teach-and-repeat of mobile robot trajectories. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3144–3149. IEEE, 2013.
- [20] Cheng Zhao, Li Sun, Tomáš Krajník, Tom Duckett, and Zhi Yan. Monocular teach-and-repeat navigation using a deep steering network with scale estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2613–2619. IEEE, 2021.