

# Transporte de objetos en forma colaborativa utilizando robots móviles

**Pablo N. Maiolo, Matías Dalletesse, Maximiliano Méndez Casariego,  
Claudio Verrastro, Sebastián Verrastro**

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Departamento de Ingeniería Electrónica, Av. Medrano 951, (C1179AAQ) Ciudad Autónoma de Buenos Aires, Argentina

*pablo.maiolo@yahoo.com.ar*

*Recibido el 6 de abril de 2017, aprobado el 6 de junio de 2017*

---

## Resumen

Se presenta una estrategia para transportar un objeto en forma colaborativa utilizando dos robots terrestres. Consta de varias partes: navegación, localización y manipulación del objeto. La navegación se realiza utilizando el algoritmo A\* y un mapa previamente conocido, y la localización hace uso de las mediciones de los sonares. Para manipular el objeto, los robots utilizan sensores infrarrojos para extraer información sobre el mismo. Los resultados experimentales muestran que la estrategia propuesta permite rotar el objeto el ángulo deseado, y trasladarlo a lo largo de la distancia indicada.

**PALABRAS CLAVE:** ROBÓTICA COLABORATIVA - TRANSPORTE DE OBJETOS - ENTORNO CON OBSTÁCULOS

## Abstract

We present a strategy for collaborative object transportation using two ground robots. This strategy comprises several parts: navigation, localization and box manipulation. Navigation is implemented using A\* algorithm over a known map of the environment, and we use sonar range readings for the localization problem. In order to manipulate the object, the robots extract information from it using infrared sensors. Experimental results show that the proposed strategy allows rotating the object a desired angle, and translating it along a given distance.

**KEYWORDS:** COLLABORATIVE ROBOTICS - OBJECT TRANSPORTATION - ENVIRONMENT WITH OBSTACLES

## Introducción

El transporte de objetos es una aplicación de la robótica que es de gran interés en las investigaciones actuales, ya que permite manipular objetos en lugares que pueden ser inaccesibles o peligrosos para el hombre. El uso de dos o más robots programados para llevar a cabo una tarea de manera colaborativa permite extender sus capacidades para realizar misiones más complejas.

En la literatura se han estudiado distintos enfoques para mover objetos utilizando conjuntos de robots. Una de las posibilidades es utilizar enjambres (*swarms*) de robots en miniatura que se comunican inalámbricamente y transportan, en conjunto, al objeto a su posición y orientación final (Fyler y colaboradores, 2015). En otro trabajo se propone el uso de un único robot esférico para empujar una caja, utilizando el sistema ROS (*Robot Operating System*) para tener la posibilidad de expandir el sistema hacia varios robots (Weber y colaboradores, 2013). En otras investigaciones se estudia cómo determinar las mejores posiciones para ubicar varios robots alrededor de un objeto, enfocándose en el agarre y en la distribución de fuerzas para generar el movimiento deseado (Parra-González y Ramírez-Torres, 2012). También se ha propuesto un esquema colaborativo en el cual un robot se encarga de darle orientación a la caja y conoce el camino a seguir, y otro simplemente empuja, y no tiene información sobre el camino. La caja se sitúa entre medio de los dos robots (Brown y Jennings, 1995). En Moon y colaboradores (2012) se presenta un algoritmo cooperativo para dos robots diferenciales, el cual consiste en tres comportamientos: acercarse a la caja, rotarla y empujarla. Se definen primitivas para la rotación y la traslación, en las cuales los robots siguen un punto determinado en la caja para llevarlas a cabo.

La mayoría de los trabajos que se consultaron requieren el uso de una cámara externa para obtener la posición y orientación de cada robot a través del procesamiento de imágenes, y esto aumenta el costo del sistema y además lo restringe a situaciones en las cuales es posible ubicar una cámara en un punto estratégico. En este trabajo se plantea usar dos robots de tracción diferencial con sensores ultrasónicos e infrarrojo, para ubicarse y conocer el estado

de la caja a manipular. También se incorpora una computadora de escritorio cuyo objetivo es realizar alguno de los cálculos más complejos y entregar los resultados a los robots. Se describe la estrategia propuesta para lograr el objetivo y las restricciones que surgen sobre el sistema.

## Planteo del problema

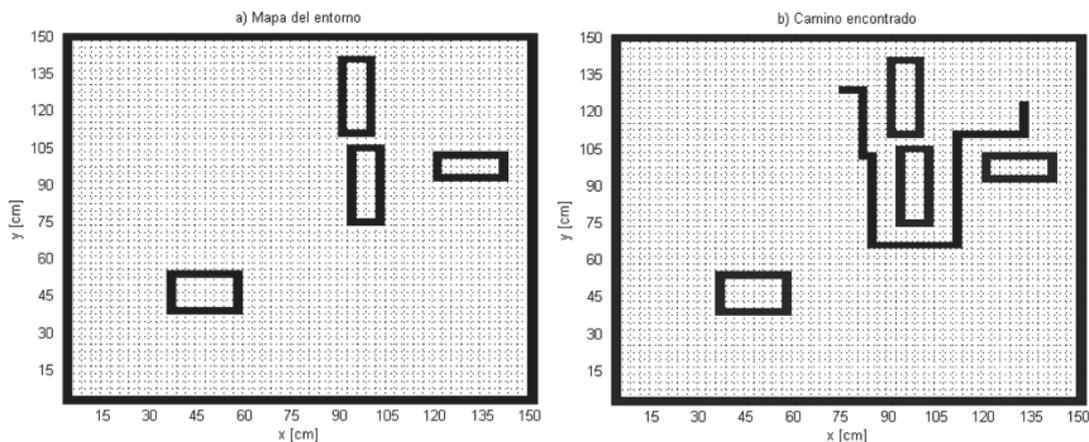
Dado un entorno con obstáculos, en principio conocido, aunque puede sufrir ciertas variaciones a lo largo de la misión, se desea llevar una caja desde su punto inicial a una posición final con la orientación indicada. La posición inicial de la caja es conocida con un cierto error, pero sus dimensiones exactas no. Lo primero que deberán hacer los robots es determinar las dimensiones y la orientación inicial de la caja. Una vez que se logró esto, el próximo paso es obtener el camino más corto libre de obstáculos desde el punto de origen hasta el punto destino. Cabe destacar que este camino está libre según el mapa inicial, pero si en el medio de la misión se interpone un obstáculo, se deberá obtener un nuevo camino. Los robots trasladan y rotan la caja según primitivas previamente definidas.

## Descripción del hardware utilizado

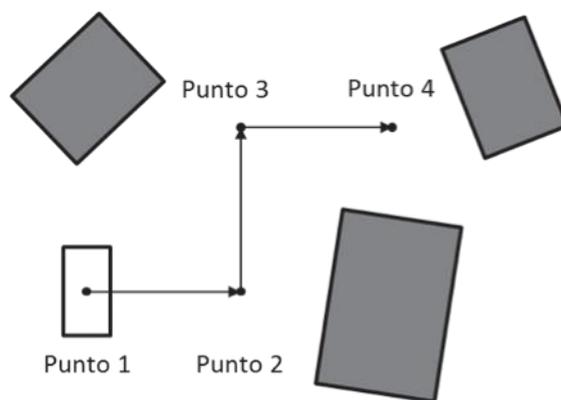
Para realizar este trabajo se utilizaron dos robots MiniPI (Mini Plataforma Inteligente) desarrollados por el Grupo de Inteligencia Artificial y Robótica (GIAR) de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional (Álvarez y colaboradores, 2013). Estos robots diferenciales disponen de *encoders* absolutos, cuatro sonares (uno en cada lado), y además se le han agregado a cada uno un sensor infrarrojo Sharp en sus caras delanteras. El microcontrolador que implementa la lógica es un ARM7TDMI funcionando a 60 MHz, y la comunicación entre ambos agentes y con la computadora se realiza a través del protocolo Zigbee, utilizando módulos inalámbricos X-Bee.

## Obtención del camino a seguir para mover la caja

El mapa del entorno está definido por una grilla de ocupación, donde a cada casillero se le asigna una probabilidad entre 0 y 1 de que esté ocupado, correspondiendo 0 a un casille-



**Fig. 1. Mapa de ejemplo con el camino calculado a través del algoritmo A\***



**Fig. 2. Inicio del recorrido. Los robots empujan la caja hacia el punto 2**

ro vacío, 1 a un casillero ocupado, y 0,5 a un casillero sobre el cual no se tiene ninguna información *a priori* (Murphy, 2000). El camino más corto se determina a través del algoritmo A\* (Hart y colaboradores, 1968), y se considera, a los efectos del algoritmo, que un casillero está ocupado si la probabilidad de ocupación es mayor al 50%. El parámetro a optimizar por el algoritmo (la función  $h(n)$  definida en el trabajo de Hart y colaboradores) consiste en calcular la cantidad de casilleros que hay del punto actual al punto objetivo, sin tener en cuenta los obstáculos, y utilizando sólo movimientos horizontales y verticales, no diagonales (distancia Manhattan, Krause 1987).

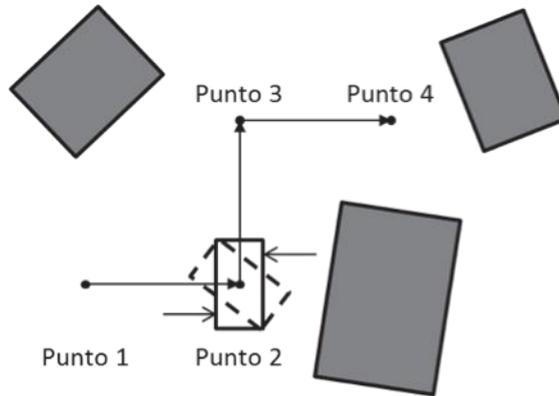
En la Fig. 1 se muestra un mapa de ejemplo (a), y el camino obtenido para un origen y un destino dados (b). El mapa es de 150 cm por 150 cm y está dividido en celdas de 3 cm. Como se puede observar, el camino está formado por movimientos horizontales y verticales, y el cálculo tiene en cuenta los obstáculos presentes según el mapa previamente

conocido. Si en el medio de la misión aparece un nuevo obstáculo y se confirma que no es un obstáculo transitorio, se lo agrega al mapa y se recalcula el camino. Notar que el camino no pasa por el espacio entre las dos cajas verticales, ya que para calcularlo se consideró el espacio que ocupa el robot al desplazarse.

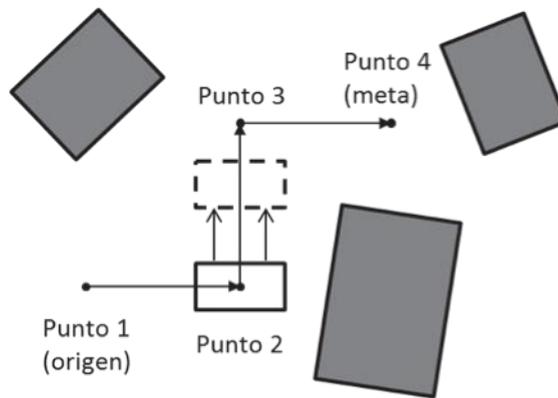
El mapa que se presentó como ejemplo es un mapa *a priori*, que es el que se carga al comienzo de la misión y en el cual los casilleros oscuros tienen la máxima probabilidad de ocupación. Esto tiene sentido si se asume que se carga un mapa certero y actualizado al momento de iniciar el recorrido, de lo contrario es preferible asignarle menos confianza a los casilleros (por ejemplo, que los vacíos arranquen en 0,3 y los ocupados en 0,7).

### Movimiento de la caja

Una vez que se obtuvo el camino, se determinan los movimientos que deben efectuar los robots para llevar la caja hasta el objetivo. El



**Fig. 3. Los robots ejercen fuerza sobre las esquinas de la caja para hacerla rotar**



**Fig. 4. Una vez que la caja tiene la orientación adecuada, los robots la empujan hacia el próximo punto del camino**

camino se divide en varias partes, generando puntos de corte en los casilleros en los cuales hay un cambio de dirección. En las Figuras 2, 3 y 4 se observa un camino de ejemplo, en donde hay cuatro puntos de corte; es deseable que este camino tenga un mínimo de puntos de corte. Para trasladar la caja desde un punto a otro, los robots siempre la empujan hacia adelante, como se muestra en las Figuras 2 y 4. Previamente hay que dar a la caja la orientación adecuada, y para esto los robots se ubican en extremos opuestos de la caja y aplican fuerza para hacerla girar (Figura 3).

### Reconocimiento de la caja

El reconocimiento de la caja lo lleva a cabo uno de los dos robots, al cual se denominará "explorador". La computadora realiza esta asignación de rol en base a la distancia que hay entre los robots y la posición inicial de la caja. El explorador es el robot más cercano a la caja. La misión del explorador es recorrer un perímetro alrededor de la caja y localizar

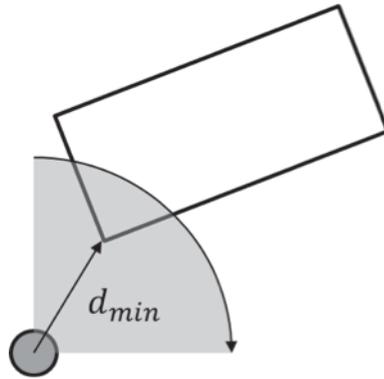
los cuatro vértices, para luego enviárselos a la PC y que ésta calcule la posición y orientación real de la caja.

La trayectoria que debe seguir el explorador para llegar a cada uno de los vértices se determina a través de A\*. Una vez que se posicionó cerca de un vértice, el explorador realiza un barrido de 90 grados tomando mediciones del sensor infrarrojo, y busca la mínima distancia, como se muestra en la Figura 5.

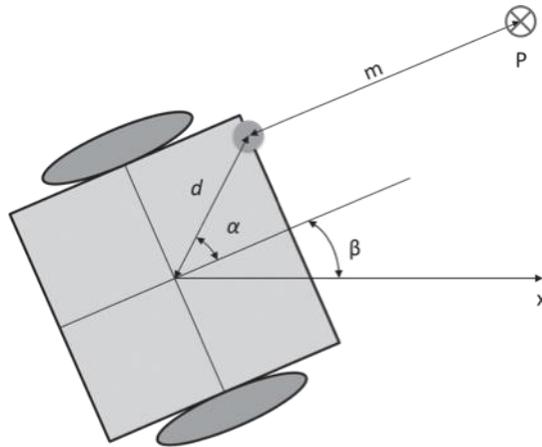
Para calcular las coordenadas cartesianas de cada uno de estos puntos, hay que tener en cuenta la ubicación relativa del sensor infrarrojo con respecto al centro del robot. A partir de la Figura 6, aplicando relaciones trigonométricas, se obtienen las siguientes ecuaciones:

$$x_{IR} = x_{robot} + d \cos(\alpha + \beta) \quad (1)$$

$$y_{IR} = y_{robot} + d \sin(\alpha + \beta) \quad (2)$$



**Fig. 5. Reconocimiento de un vértice de la caja**



**Fig. 6. Esquema para obtener las coordenadas del punto P a partir de la medición del sensor infrarrojo**

Siendo  $\alpha$  el ángulo relativo entre el frente del robot y el sensor infrarrojo, y  $\beta$  el ángulo del robot dentro del marco de referencia. En la misma figura se tiene una medición del sensor infrarrojo representada por  $m$ . Se deducen las siguientes ecuaciones para el punto P:

$$x_P = x_{IR} + m \cos \beta \quad (3)$$

$$y_P = y_{IR} + m \sin \beta \quad (4)$$

Combinando las ecuaciones ((1) y (3), (2) y (4), se obtiene finalmente:

$$x_P = x_{robot} + d \cos(\alpha + \beta) + m \cos \beta \quad (5)$$

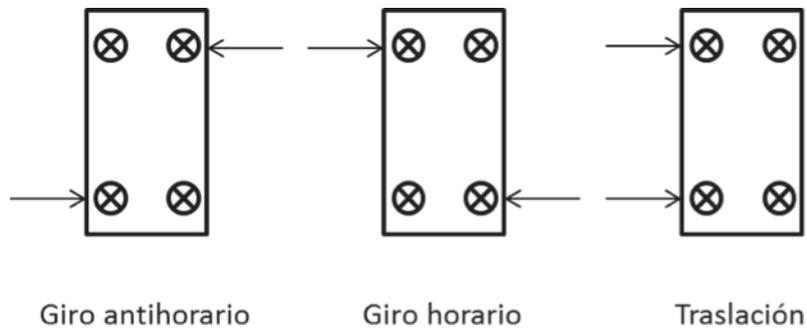
$$y_P = y_{robot} + d \sin(\alpha + \beta) + m \sin \beta \quad (6)$$

Cuando se tienen los cuatro vértices, la computadora determina la posición y orientación de la caja, y obtiene la secuencia de *primitivas* que los robots deben realizar sobre la caja para llevarla a la posición deseada.

### Movimiento de la caja

Las primitivas utilizadas para mover y rotar la caja se basan en el trabajo de Moon y colaboradores (2012). Sobre la caja se disponen cuatro puntos sobre los cuales los robots aplican fuerza para rotarla o trasladarla. La ubicación de estos puntos está predefinida. En la Figura 7 se muestran estos puntos y los movimientos posibles de la caja, según la fuerza aplicada por los robots.

A diferencia del trabajo de Moon y colaboradores, como no se dispone de una cámara para seguir la posición y orientación de los robots y la caja, se optó por utilizar un sensor infrarrojo.



**Fig. 7. Movimientos posibles y puntos que los robots deben seguir para lograrlos**

En el siguiente apartado se describe la solución propuesta para manipular la caja.

### Reposicionamiento en los bordes

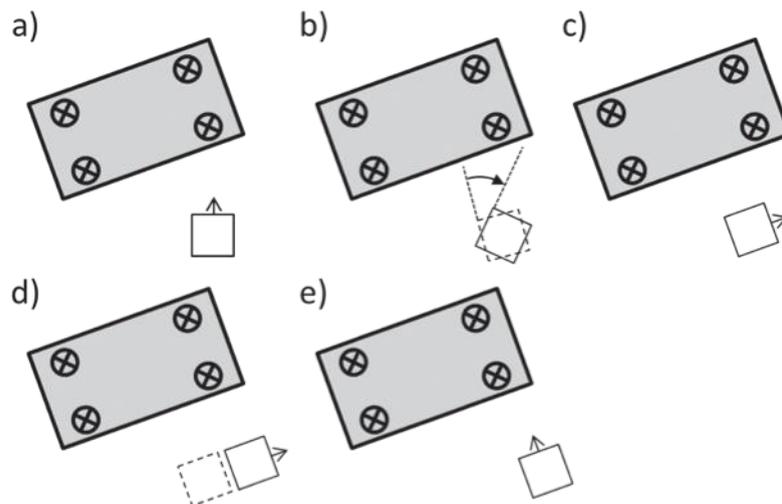
Luego de cada tramo, ya sea para empujar o para rotar la caja, los robots deben reposicionarse en el punto asignado de manera de manipular la caja con precisión. Esto se logra si se posicionan perpendiculares a la misma.

En la Figura 8 se describen de manera esquemática los movimientos que hacen los robots para reposicionarse y prepararse para empujar o rotar la caja. En a) el robot se aleja de la caja para que las mediciones sean mayores a la distancia mínima que mide el sensor infrarrojo (8 cm). En b) se muestra el barrido que hace el robot para obtener puntos que muestrean el perfil de la caja. En las figuras c) y d) el robot usa la información obtenida para moverse paralelo a la caja hacia el borde. En e) el robot se

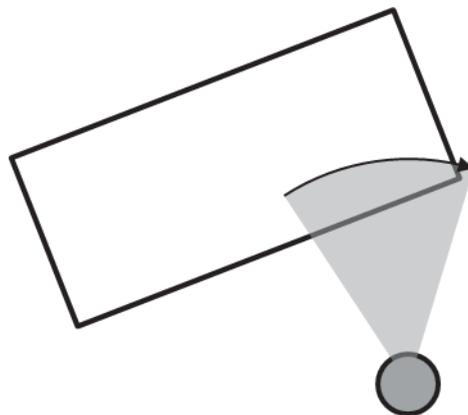
posiciona perpendicular a la caja y queda listo para empujar la esquina.

Debido a que no se dispone de información externa sobre la caja, los robots deben relevar el perfil, para obtener la orientación y los bordes. La solución que se propone es calcular la recta que representa el perfil de la caja y determinar la ubicación de los bordes.

Asumiendo que los robots fueron posicionados cerca del punto de apoyo que corresponde para el movimiento, ambos robots se mueven 10 cm hacia atrás, respecto de su posición actual, debido a que el sensor infrarrojo se vuelve considerablemente impreciso para mediciones por debajo de 10 cm. De esta forma se garantiza que aunque el robot estuviera inicialmente pegado a la caja, se van a tomar mediciones útiles. Se comienza a girar hacia donde están sus respectivos bordes, tomando medidas con el sensor infrarrojo cada 3 grados y comparán-



**Fig. 8. Movimientos realizados por un robot para prepararse para empujar el punto de apoyo asignado**



**Fig. 9. Barrido del sensor infrarrojo hasta detectar el borde de la caja**

dolas con la anterior (Figura 9). Cuando se detecta un salto en las mediciones, se considera que se detectó el borde. Experimentalmente se determinó que si el salto supera los 5 cm, se detecta el borde correctamente.

Cada vez que el robot toma una medición, se guarda la distancia medida y el ángulo que tenía el robot al momento de medir. De esta manera se obtiene un perfil de la caja formado por una secuencia de puntos en coordenadas polares.

Para calcular las coordenadas cartesianas ( $x$ ,  $y$ ) de cada uno de estos puntos, se utilizan las ecuaciones (5) y (6).

Para obtener la mejor recta que ajuste a los puntos relevados se utiliza el algoritmo RANSAC (*Random Sample Consensus*). La principal ventaja de este algoritmo es que permite obtener la mejor recta, aún en presencia de puntos denominados *outliers*, que son aquellos que no describen bien al modelo (Fischler y Bolles, 1981). En la aplicación de interés de este trabajo, los puntos que se miden muy cerca del borde de la caja pueden estar desviados y no formar parte de la recta que se quiere obtener. Si en lugar de utilizar RANSAC se utilizara directamente una regresión lineal, los *outliers* causarían que se obtenga una recta que no se corresponde con la realidad.

El algoritmo en primer lugar muestrea puntos al azar dentro del conjunto de puntos relevados, y calcula una recta de ajuste a través del método de mínimos cuadrados. Luego, de todos los puntos relevados, se queda con aquellos que están a una distancia de la recta menor a cierto valor. Si la cantidad de puntos acep-

tados es mayor a cierto umbral denominado *consenso*, se considera que el modelo obtenido es válido, y se calcula la recta definitiva usando todos los puntos aceptados. De lo contrario se vuelve a ejecutar el ciclo, y se va iterando hasta encontrar la recta, o superar el máximo de iteraciones, y en tal caso se considera que falló y no se pudo hallar una recta.

La recta calculada se describe a través de la siguiente ecuación:

$$Ax + By + C = 0$$

Donde  $A$ ,  $B$  y  $C$  son los coeficientes de la recta que se obtienen al hacer la regresión lineal. Para obtener la pendiente, hay que expresar la recta en forma explícita. Despejando  $y$ , se obtiene:

$$y = -\frac{A}{B}x - \frac{C}{B}$$

La pendiente es  $m = -A/B$ . El ángulo de la recta se obtiene de:

$$\gamma = \tan^{-1} m$$

Para obtener las coordenadas del borde de la caja, se aprovecha el hecho de que las mediciones están ordenadas desde un punto más interior de la caja, hasta el punto más exterior que se midió antes de detectar el borde. Sin embargo, este punto, por estar cerca de un borde, puede tener error y caer por fuera de la recta calculada, en tal caso el algoritmo RANSAC lo filtra. La posición efectiva del borde es entonces el último punto aceptado por el algoritmo.

Una vez que se conoce el borde y la pendiente

de la recta, el objetivo es lograr que el robot se posicione cerca del borde y perpendicular a la caja. En la Figura 10 se muestra la ubicación final que debe tener el robot y las rectas usadas para calcularla. Se desea llevar el robot desde el punto  $(x_i, y_i)$ , hasta el punto  $(x_f, y_f)$  y dejarlo ortogonal a la caja. En la figura se dibujan dos rectas en línea discontinua, cuya intersección da el punto  $(x_1, y_1)$ . Una de las rectas es la paralela a la caja que pasa por el punto  $(x_i, y_i)$  y la otra es la perpendicular a la caja que pasa por el borde  $(x_{borde}, y_{borde})$ . La primera tiene la pendiente  $m$ , previamente calculada, y la segunda, al ser ortogonal, tiene pendiente  $m' = -1/m$ .

La pendiente de una recta genérica puede expresarse en función de dos puntos como:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (7)$$

A partir de la ecuación (7), las dos rectas discutidas previamente se pueden expresar como:

$$m = \frac{y_1 - y_i}{x_1 - x_i}$$

$$m' = \frac{y_1 - y_{borde}}{x_1 - x_{borde}} = -\frac{1}{m}$$

Despejando  $y_1$  en ambos casos, queda el siguiente sistema de ecuaciones:

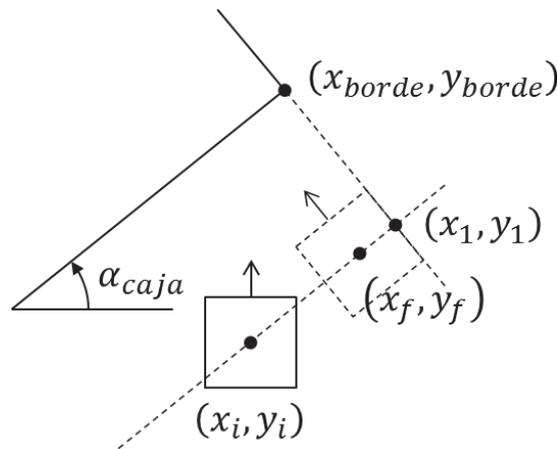
$$\begin{cases} y_1 = y_i + m(x_1 - x_i) \\ y_1 = y_{borde} - \frac{1}{m}(x_1 - x_{borde}) \end{cases}$$

Resolviendo el sistema, se despeja  $x_1$  e  $y_1$ :

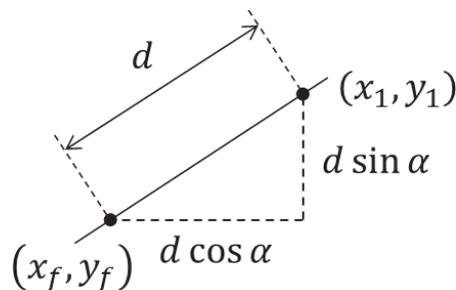
$$x_1 = \frac{m(y_{borde} - y_i) + m^2 x_i + x_{borde}}{m^2 + 1} \quad (8)$$

$$y_1 = y_i + m(x_1 - x_i) \quad (9)$$

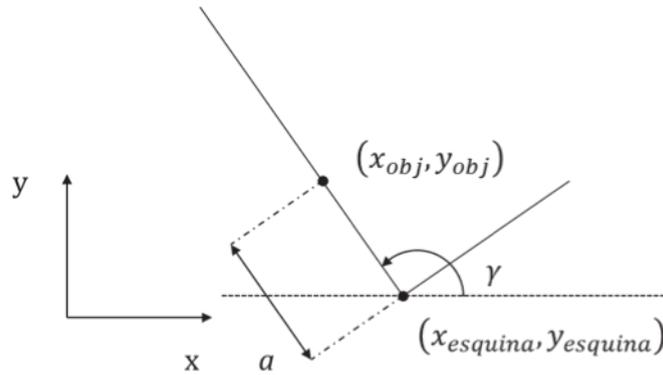
Para obtener  $x_f$  e  $y_f$  hay que moverse sobre la recta paralela a la caja teniendo en cuenta las dimensiones del robot, ya que dicho punto está a medio ancho de robot de distancia de  $(x_1, y_1)$ . La Figura 11 muestra un acercamiento a la posición final del robot en la Figura 10.



**Fig. 10. Obtención de la posición final del robot**



**Fig. 11. Cálculo del punto  $(x_f, y_f)$  a partir de  $(x_1, y_1)$**



**Fig. 12. Cálculo del punto que el robot debe empujar**

En base al esquema mostrado, las expresiones que permiten obtener el punto deseado son:

$$x_f = x_1 + s * d * \cos \alpha \quad (10)$$

$$y_f = y_1 + s * d * \sin \alpha \quad (11)$$

Donde  $\alpha$  es el ángulo que forma recta con el eje  $x$ ,  $d$  es la mitad del ancho del robot, y  $s$  es un valor que puede ser +1 o -1, ya que la Figura 11 muestra solo uno de los casos posibles, pero si el robot estuviera en el otro borde de la caja, los puntos  $(x_r, y_r)$  y  $(x_l, y_l)$  estarían intercambiados.

### Seguimiento de los puntos de la caja

Una vez que el robot está en posición, en primer lugar se calcula el punto de apoyo de la caja. En la Figura 7 se mostraron los puntos de apoyo necesarios para lograr el giro o la traslación de la caja. Dichos puntos están a una distancia fija del borde, por lo que si se conoce el borde y la recta que representa a la caja, se puede obtener el punto de apoyo desplazándose una distancia  $a$  sobre la recta, como se muestra en la Figura 12.

Se aplican las mismas ecuaciones que se utilizaron previamente para obtener  $(x_r, y_r)$ :

$$x_{obj} = x_{esquina} + s * a * \cos \gamma \quad (12)$$

$$y_{obj} = y_{esquina} + s * a * \sin \gamma \quad (13)$$

Conociendo los puntos de apoyo se aplica la ecuación de control para determinar la velocidad que hay que asignarle a cada motor. Se

utilizan las leyes de control dadas por las siguientes ecuaciones (Moon y colaboradores, 2012):

$$v_c = K_1 r \cos \phi \zeta \quad (14)$$

$$\omega_c = -K_1 \sin \phi \cos \phi - K_2 \phi \quad (15)$$

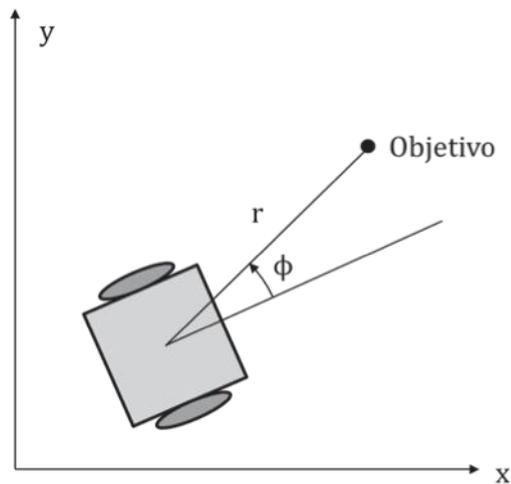
En la Figura 13 se ilustran  $r$  y  $\phi$ .  $v_c$  es la velocidad lineal del robot y  $\omega_c$  es la velocidad angular. En un robot diferencial, la velocidad de ambas ruedas está dada por:

$$v_{izq} = v_c - \frac{D}{2} * \omega_c$$

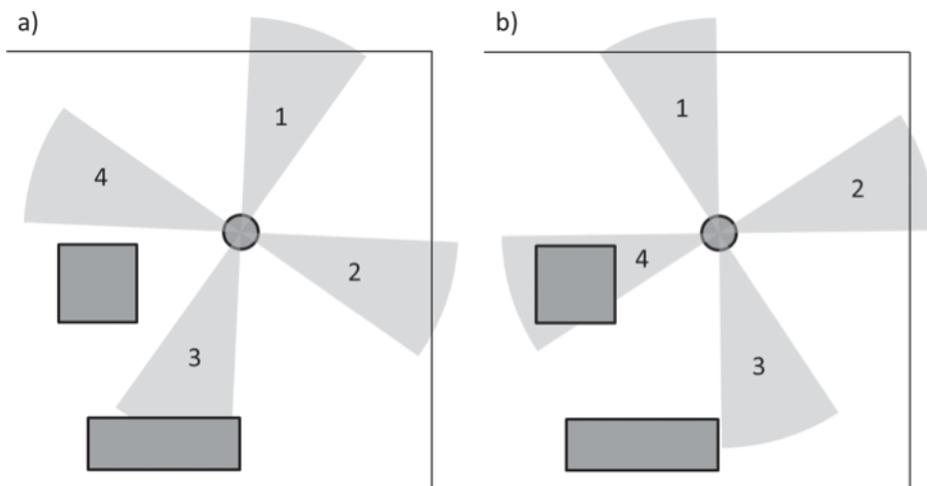
$$v_{der} = v_c + \frac{D}{2} * \omega_c$$

Los robots empujan una distancia fija, que depende de si el movimiento deseado para la caja es un giro o una rotación. Si es un giro, usando el infrarrojo determinan a qué distancia están de la caja, y a ese valor le suman una constante que determina cuánto van a empujar las esquinas de la caja, y por lo tanto cuántos grados va a girar en ese tramo. Esta constante se determina experimentalmente en base a la precisión deseada y la cantidad de tramos en los que se acepta dividir el giro total. Después de cada tramo de giro, los robots vuelven a realizar el algoritmo para reposicionarse en los bordes.

Si el movimiento de la caja es una traslación, se lo subdivide en tramos de longitud fija, para permitir que los robots se reposicionen y no pierdan el objeto. La longitud de los tramos se determina experimentalmente evaluando



**Fig. 13. Radio y ángulo para el control de seguimiento del objetivo**



**Fig. 14. Barrido de los sonares para la localización**

cuánto se desvían los robots de su posición asignada, mientras están trasladando la caja. El objetivo es que la deriva en su posición esté acotada y los robots puedan reconocer nuevamente el objeto y reposicionarse. Se halló que tramos de 50 cm cumplen con esta condición.

### Localización del robot

El robot lleva registro de su posición y orientación actual utilizando la información que le proporcionan las mediciones tomadas del encode-racoplado a las ruedas. Luego de moverse una distancia prefijada, o después de cierto evento (por ejemplo, una vez que terminó de rotar la caja), se aplica un filtro de Kalman, fusionando

la lectura de los sonares con las mediciones de los *encoders*, para acotar el error de localización y evitar que se vaya acumulando. En particular el filtro utilizado es el filtro extendido de Kalman (EKF). Originalmente se intentó aplicar directamente el filtro de Kalman comparando las mediciones de los 4 sonares con las mediciones predichas en función de la odometría y el mapa conocido (Ivanjko y colaboradores, 2005). Luego de varias pruebas se llegó a la conclusión de que con sólo 4 sonares el modelo obtenido no era muy confiable y la aplicación directa del filtro de Kalman en ocasiones tendía a empeorar el estado estimado, sobre todo el ángulo del robot.

El nuevo enfoque consiste en aprovechar el hecho de que el escenario está delimitado por cuatro paredes ortogonales entre sí, y si la distancia de un sonar a una pared es la mínima, entonces el robot está perpendicular a dicha pared.

En la Figura 14 está esquematizada la estrategia para colocarse en forma perpendicular a la pared y corregir el ángulo. El robot realiza un barrido dentro de un ángulo limitado (por ejemplo,  $\pm 20^\circ$ ), y toma todas las mediciones que devuelve el sonar. Luego se queda con la mínima y determina que en esa posición se encontraba ortogonal a la pared. Se utiliza un filtro para rechazar la influencia de los obstáculos. Si las mediciones relevadas no se ajustan a una recta, entonces ese sonar se rechaza y no se utiliza para corregir el estado del robot. El algoritmo RANSAC descrito anteriormente sirve para realizar el filtrado. En la misma figura se muestran dos instantes que corresponden al inicio y al final del barrido. Cada sonar tiene un número y un área que representa la apertura. Las mediciones de los sonares 1 y 2 pueden ajustarse una recta y por lo tanto se aceptan como representativas de una pared. Por el contrario, los sonares 3 y 4, a lo largo del barrido, ven saltos en las mediciones que no se ajustan a ninguna recta, y entonces son rechazados y no se utilizan para corregir el estado del robot.

El barrido permite corregir la orientación del robot. Para corregir la posición, se combinan la

estimación por odometría y la distancia medida a las paredes, utilizando el filtro de Kalman.

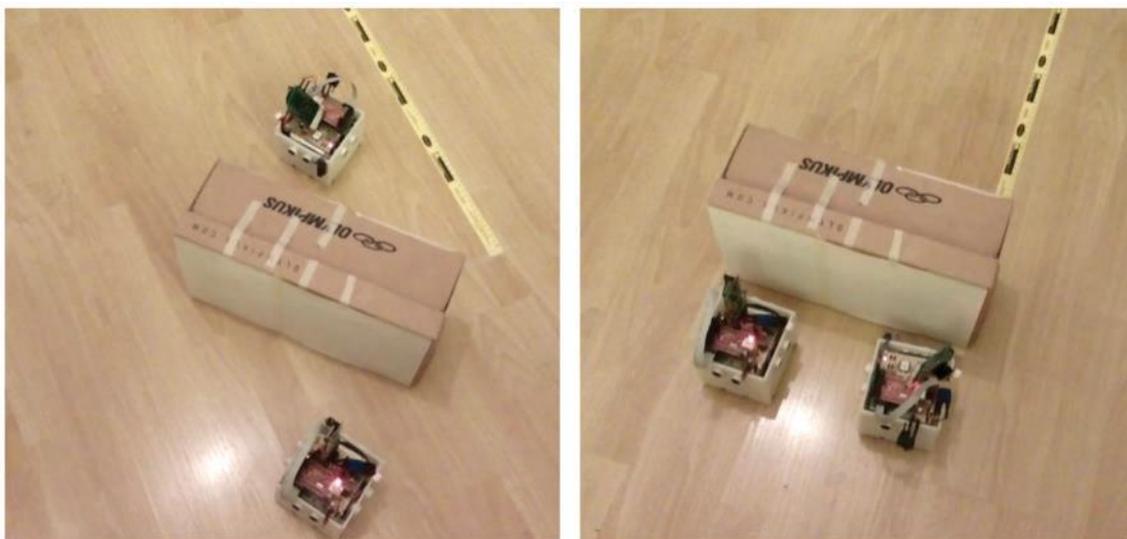
## Parte experimental

Para comprobar la validez de la estrategia planteada, se probaron los algoritmos para manipular el objeto. Debido a que la localización no está finalizada, los movimientos de la caja se realizaron en forma aislada, con el objetivo de verificar que los robots reconocen adecuadamente el objeto y son capaces de imprimirle el movimiento deseado. Se utilizó una caja de cartón de 36 cm de largo por 13 cm de ancho. Las dimensiones de los robots son 12 cm por 12 cm. Se ubicaron manualmente en los puntos de aproximación a la caja, pero no se los colocó de manera exacta, para comprobar la capacidad de reposicionarse en base a la medición del borde y el perfil del objeto a desplazar.

En la Figura 15, se muestran ambos robots y la caja a manipular. En la imagen de la izquierda, los robots están posicionados para realizar la rotación de la caja, y en la derecha, para trasladarla a lo largo de una línea recta.

## Resultados y discusión

Se comprobó que los robots efectúan el giro de la caja de forma satisfactoria, y reposicionarse en forma ortogonal y cerca de los bordes permite que realicen fuerza sobre puntos cercanos a las esquinas, obteniendo una buena rotación.



**Fig. 15. Posiciones de los robots para realizar los experimentos de movimiento de la caja**

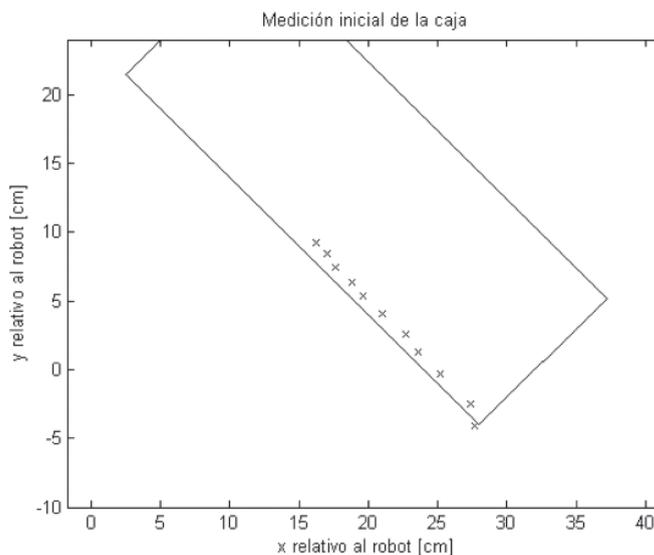
En primer lugar se ensayó el reposicionamiento del robot con la caja inclinada a  $45^\circ$  respecto de la orientación inicial del robot y se relevaron las mediciones del sensor infrarrojo, para comprobar la precisión con que reconocía la caja. Se observó que el robot realizaba correctamente el reposicionamiento cerca de los bordes, y corregía su posición en forma sucesiva hasta quedar correctamente ubicado. En la Figura 16 se grafican las mediciones que realizó el robot al principio del ensayo. Las coordenadas son relativas al centro del robot, por lo tanto éste se halla en (0,0). También se dibuja la caja de forma aproximada para ilustrar qué parte se estaba midiendo. El robot se posicionó cerca del borde, de la manera que se ilustra en la Figura 10, y volvió a relevar el perfil de la caja. Las mediciones resultantes están graficadas en la Figura 17, junto con una representación aproximada de la caja. Ajustando las mediciones de la Figura 16 a una recta de regresión se determina un ángulo de la caja de  $47,3^\circ$ , y utilizando las mediciones de la Figura 17, el resultado arroja  $47,91^\circ$ . Entre ambos instantes de medición, la posición y orientación real de la caja se mantuvo constante. Los resultados permiten comprobar que el error no aumentó de manera significativa y el relevamiento del ángulo y borde de la caja permite reposicionar al robot.

Para comprobar la traslación de la caja, se le asignó a los robots una distancia a mover la caja, y se comprobó si efectivamente la trasladaban esa distancia, y si la orientación final de la caja era la misma que al iniciar el trayecto.

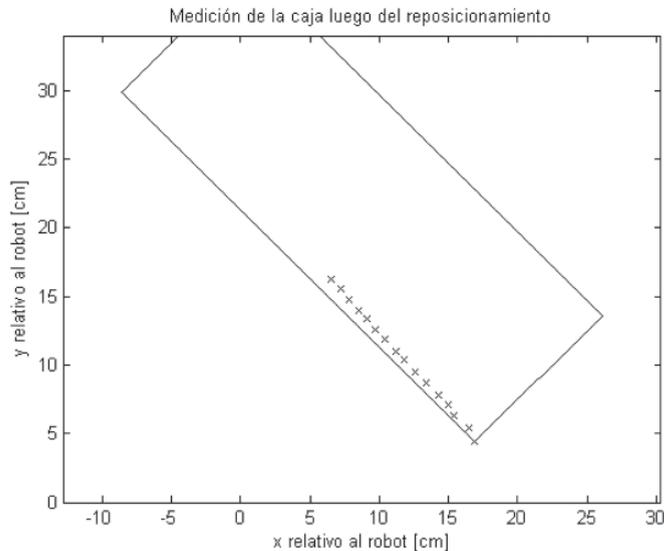
Se hicieron dos tipos de experimentos. La Tabla 1 corresponde a la traslación de la caja a lo largo de distintas distancias, relevando el error en la distancia y en el ángulo final de la caja. En la Tabla 2 se muestra el segundo experimento que consistió en analizar la traslación a lo largo de la misma distancia repetidas veces, para obtener valores estadísticos.

A partir de los resultados, se puede observar que en todos los casos se pudo mantener el error en la posición acotado al 10%, y el ángulo en el peor caso tuvo un error de  $20^\circ$ . Los robots son capaces de trasladar la caja respetando la distancia solicitada, aunque el ángulo presenta una desviación que podría corregirse utilizando la información de localización que brinda el entorno. A futuro se plantea incorporar la localización mediante el filtro extendido de Kalman, de esta manera los robots pueden obtener la ubicación absoluta de los vértices del objeto para poder evaluar y, de ser necesario, corregir sobre la marcha cualquier desvío que presente la caja respecto a su posición o ángulo deseados.

La Tabla 2 complementa este análisis, verificando que el traslado tiene un error pequeño. Al pasar de 50 cm a 90 cm se observa un aumento en el error porcentual, de 2,21% a 3,13%. Se espera que esta diferencia disminuya al incorporar la corrección del estado de los robots mediante Kalman, y de esta manera el traslado de la caja mantendría su precisión al aumentar la distancia recorrida.



**Fig. 16. Perfil de la caja medido por un robot antes de posicionarse**



**Fig. 17. Perfil de la caja medido por un robot luego de posicionarse en la esquina**

Luego se ensayó el giro de la caja, disponiendo a los robots sobre los puntos correspondientes a un giro antihorario. Los giros realizados fueron de  $90^\circ$  porque es el mayor giro que realiza el robot para reorientar la caja, de la manera graficada en la Figura 3. Se observó que los robots realizaban el reposicionamiento cerca de los bordes, y corregían su posición en forma sucesiva hasta quedar ubicados de

acuerdo al esquema planteado en la Figura 8 e). Lograron girar la caja en forma colaborativa sin desplazarla de su centro ni perderla de vista. En la Tabla 3 se observan los resultados de la rotación. Al igual que en el caso de la traslación de la caja, una posible mejora es aprovechar el estado del robot para determinar si hay que corregir la orientación de la caja. Se espera que cuando se complete la

**Tabla 1. Resultados del experimento de trasladar la caja en varias distancias**

Prueba	Movimiento deseado (cm)	Movimiento real (cm)	Error % de traslación	Ángulo Inicial caja	Ángulo Final caja	Error de ángulo
1	80	86	7,5	$90^\circ$	$83^\circ$	$-7^\circ$
2	80	79	-1,25	$90^\circ$	$79^\circ$	$-11^\circ$
3	30	31	3,33	$90^\circ$	$96^\circ$	$6^\circ$
4	10	11	10	$90^\circ$	$94^\circ$	$4^\circ$
5	40	39	-2,5	$90^\circ$	$94^\circ$	$4^\circ$
6	60	58	-3,33	$90^\circ$	$85^\circ$	$-5^\circ$
7	50	51	2	$90^\circ$	$75^\circ$	$-15^\circ$
8	50	54	8	$90^\circ$	$110^\circ$	$20^\circ$
9	90	98	8,89	$90^\circ$	$70^\circ$	$-20^\circ$
10	90	92	2,22	$90^\circ$	$100^\circ$	$10^\circ$

**Tabla 2. Resultados del experimento de trasladar la caja en dos distancias fijas, varias veces**

Distancia a trasladar	50 cm	90 cm
Cantidad de ensayos	10	10
Valor medio	50,9 cm	92,88 cm
Desviación estándar	1,11 cm	2,82 cm
Desviación estándar (%)	2,21	3,13

**Tabla 3. Resultados del experimento de rotar la caja**

Prueba	Ángulo objetivo	Ángulo real de la caja	Error de ángulo
1	90°	103°	13°
2	90°	79°	-11°
3	90°	110°	20°
4	90°	70°	-20°
5	90°	80°	-10°
6	90°	95°	5°
7	90°	85°	-5°
8	90°	80°	-10°
9	90°	105°	15°
10	90°	88°	-2°
Valor medio	89,5°		
Desviación estándar	13,19°		

localización del robot, el error de ángulo se pueda acotar a la incertidumbre del ángulo de los robots corregido mediante la información que brindan los sensores ultrasónicos.

### Conclusiones

Se presentó un método para manipular cajas de forma colaborativa utilizando robots que están equipados solamente con sonares y un sensor infrarrojo. La navegación de los robots por el entorno depende del mapa conocido *a priori* y la información de los sonares, mientras que una vez que encuentran la caja, la rotan o empujan utilizando las mediciones del infrarrojo. Bajo estas condiciones, ensayos preliminares permitieron comprobar que los robots son capaces de manipular el objeto con un error acotado, sin necesidad de recibir información externa en tiempo real sobre el estado del mismo.

El reposicionamiento de los robots cerca de los vértices permite ir mejorando las mediciones en forma sucesiva y contar con la mejor información posible sobre la caja. Además, les da a los robots un punto de partida repetible para empezar a manipular la caja.

Queda pendiente terminar de implementar y evaluar la corrección del estado mediante el filtro de Kalman. En los resultados presentados, uno de los aspectos destacables es que el error angular es más significativo que el error al desplazar el objeto, este resultado es

esperable dado la precisión de los sensores utilizados. Por este motivo la elección de la medición de la distancia a las paredes es una estrategia acertada. Se propone realizar un segundo experimento en el cual se traslade la caja evadiendo obstáculos y con varios giros y traslaciones. Se espera que los errores al manipular el objeto puedan acotarse hasta el valor de la incertidumbre de los sonares utilizados.

### Agradecimientos

Agradecemos al Grupo de Inteligencia Artificial y Robótica (GIAR) de la Facultad Regional Buenos Aires de la UTN, por facilitarnos los robots que permitieron la realización del presente trabajo.

## Referencias

- ÁLVAREZ, A. G.; MARUFO DA SILVA, M.; SAMPIERI, F.; PIRLO, D.; GABIS, J. M.; DI SABATO, P.; MARCOVECCHIO, N.; CISMONDI, L.; FOLINO, P.; ALBERINO, S.; VERRASTRO, C. y GÓMEZ, J. C., (2013). MiniPI: Plataforma Inteligente de Tamaño Reducido, XV Reunión de Trabajo en Procesamiento de la Información y Control, San Carlos de Bariloche, Río Negro, Argentina.
- BROWN, R. G. y JENNINGS, J. S., (1995). A Pusher/Steerer Model for Strongly Cooperative Mobile Robot Manipulation, Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots, 3, 562.
- FISCHLER, M. y BOLLES, R. C., (1981). Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Artificial Intelligence Center, SRI International, Menlo Park, California
- FYLER, D.; SULLIVAN, B. y RAPTIS, I. A., (2015). Distributed Object Manipulation Using a Mobile Multi-Agent System, 2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), 1.
- IVANJKO, E.; VAŠAK, M. y PETROVIFF, I., (2005). Kalman Filter Theory Based Mobile Robot Pose Tracking Using Occupancy Grid Maps, 2005 International Conference on Control and Automation, 2, 869.
- HART, P. E.; NILSSON, N. J. y RAPHAEL, B., (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics, 4, 100.
- KRAUSE, E. F., (1987). Taxicab Geometry: An Adventure in Non-Euclidean Geometry, Dover Books on Mathematics
- MOON, S.; KWAK, D. y KIM, H. J., (2012). Cooperative control of differential wheeled mobile robots for box pushing problem, 2012 12th International Conference on Control, Automation and Systems, 140.
- MURPHY, R. R., (2000). Introduction to AI Robotics, Massachusetts Institute of Technology, Cambridge, Massachusetts, EE.UU.
- PARRA-GONZÁLEZ, E. F. y RAMÍREZ-TORRES, J. G., (2012). Grasp Quality for the Object Transportation by Communities of Mobile Robots, 2012 9th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 1.
- WEBER, J.; KAUFMANN, C.; HACHE, C. y SCHMIDT, M., (2013). Solving the Box-Pushing Problem Using a Spherical Robot, 2015 16th International Conference on Research and Ed