



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Robotica Movil

## Trabajo práctico 1

### Parser

#### *Resumen*

*Entrenamiento de Redes Neuronales*

Integrante	LU	Correo electrónico
Negri, Franco	893/13	franconegri2004@hotmail.com
?, ?	?/?	?@?.com

Palabras claves:

TP

## Índice

<b>1. Introduccion</b>	<b>3</b>
<b>2. Desarrollo</b>	<b>3</b>
2.1. Adaptacion del modelo Cinematico . . . . .	3
2.1.1. Experimentación . . . . .	3
2.2. Adaptación del control a lazo cerrado . . . . .	4
2.2.1. Experimentación . . . . .	4
2.3. Modelado de EKF . . . . .	5
2.3.1. Experimentación . . . . .	6
<b>3. Seguimiento de trayectorias a lazo cerrado utilizando localización basada en EKF</b>	<b>8</b>
<b>4. Conclusiones</b>	<b>8</b>

## 1. Introduccion

En los ultimos años se ha visto un gran avance en el campo de la robotica y en tareas automatizables. En particular, los robots omnidireccionales se han empezado a utilizar tanto en la industria, por su capacidad de maniobrar en espacios reducidos, como en la domotica por su comodidad.

En este trabajo practico implementaremos un sistema que permita la realizacion de trayectorias de manera segura y precisa utilizando un robot omnidireccional. Para ello utilizaremos el entorno de desarrollo ROS y el entorno de simulación V-Rep.

El robot en particular sobre el cual trabajaremos contará con cuatro ruedas fijas con diseño Mecanum. De este modo, parte de nuestro trabajo será implementar un modelo cinematico acorde que nos permita tanto controlar la velocidad como estimar la velocidad lineal y angular a partir de la informacion sensada por las cuatro ruedas.

## 2. Desarrollo

### 2.1. Adaptacion del modelo Cinematico

Como ya anticipamos, para este trabajo practico utilizaremos un robot omidireccional. Este cuenta con cuatro ruedas Mecanum dispuestas a los cuatro costados del robot, las cuales cuentan con rodillos especiales que permiten transmitir parte de la fuerza en direccion de un angulo definido.

Como primer desafío buscaremos determinar la velocidad de los actuadores y la velocidad en cada uno de los ejes de libertad del robot. Para esto adaptaremos el modelo cinematico visto durante la materia para a este nuevo tipo de actuadores utilizando el paper provisto por la catedra, las formulas de cinematica directa pasarán a ser.

$$\begin{aligned}v_x(t) &= (w_1 + w_2 + w_3 + w_4).r/4 \\v_y(t) &= (-w_1 + w_2 + w_3 - w_4).r/4 \\w_z(t) &= (-w_1 + w_2 - w_3 + w_4).r/(4(l_x + l_y))\end{aligned}$$

Y las de cinematica inversa:

$$\begin{aligned}w_1 &= 1/r(v_x - v_y - (l_x + l_y)w_z) \\w_2 &= 1/r(v_x + v_y + (l_x + l_y)w_z) \\w_3 &= 1/r(v_x + v_y - (l_x + l_y)w_z) \\w_4 &= 1/r(v_x - v_y + (l_x + l_y)w_z)\end{aligned}$$

Donde  $v_x$ ,  $v_y$  y  $w_z$  son las velocidades lineares y angular del robot,  $r$  el radio de las ruedas (asumiremos que todas las ruedas tienen exactamente el mismo radio),  $l_x$  la mita de la distancia entre las dos ruedas delanteras y  $l_y$  la mitad de la distancia entre una rueda delantera y una rueda tracera.  $w_1, w_2, w_3, w_4$  las velocidades angulares de las cuatro ruedas.

En particular para nuestro robot, tomaremos  $r = 0,05$  metros,  $l_x = l_y = 0,175$  metros.

#### 2.1.1. Experimentación

En este apartado buscaremos validar que el moldeo antes descripto se condice con la realidad. Para ello, enviaremos mensajes a travez de rostopic con comandos de velocidad lineal y esperearemos ver que el modelo sepa traducir y aplicar la velocidad deseada en los actuadores. Luego

utilizando los mensajes de cinemática inversa esperamos que el modelo pueda traducir nuevamente las velocidades angulares de los actuadores a las velocidades lineales y que estas se condigan con las que enviamos.

## 2.2. Adaptación del control a lazo cerrado

Al igual que para el modelo diferencial queremos que el robot se pueda trasladar de una pose actual  $(x_i, y_i, w_i)$  en un tiempo  $t_i$  a una pose objetivo  $(x_f, y_f, w_f)$  en un tiempo diferente  $t_f$ . Para eso utilizaremos un control a lazo cerrado que nos permitirá observar si el robot se encuentra en la pose deseada en cada instante, retroalimentando el error a los controladores de velocidad.

En este caso, al tener un modelo cinemático holonómico las cuentas se simplifican con respecto al modelo diferencial: Al tener control independiente sobre cada velocidad puedo plantear, para cada dimensión por separado:

$$\Delta_x = x_f - x_i / (t_f - t_i)$$

$$\Delta_y = y_f - y_i / (t_f - t_i)$$

$$\Delta_w = w_f - w_i / (t_f - t_i)$$

Donde  $\Delta_x$ ,  $\Delta_y$  y  $\Delta_w$  es el error entre la pose actual y la objetivo. Minimizar este error significará haber alcanzado la pose objetivo, por lo que definimos las velocidades del robot como:

$$V_x = K \Delta_x$$

$$V_y = K \Delta_y$$

$$\theta = K \Delta_w$$

Donde  $K$  es una constante proporcional.

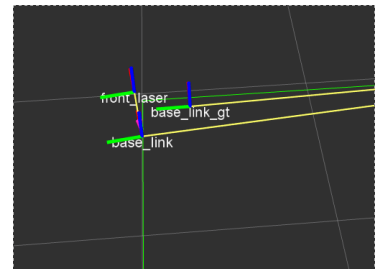
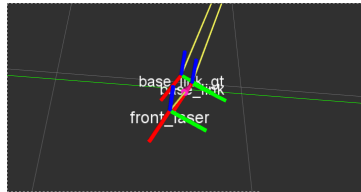
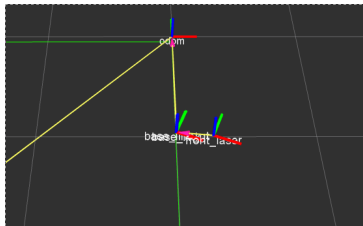
Nos queda solucionar un caso borde que consiste en: si la pose actual y la objetivo se encuentran a una distancia infinita (o cercana a infinita), entonces será necesaria una velocidad infinita para alcanzarla. Dado que esto no es físicamente posible, definimos una cota máxima de velocidad.

### 2.2.1. Experimentación

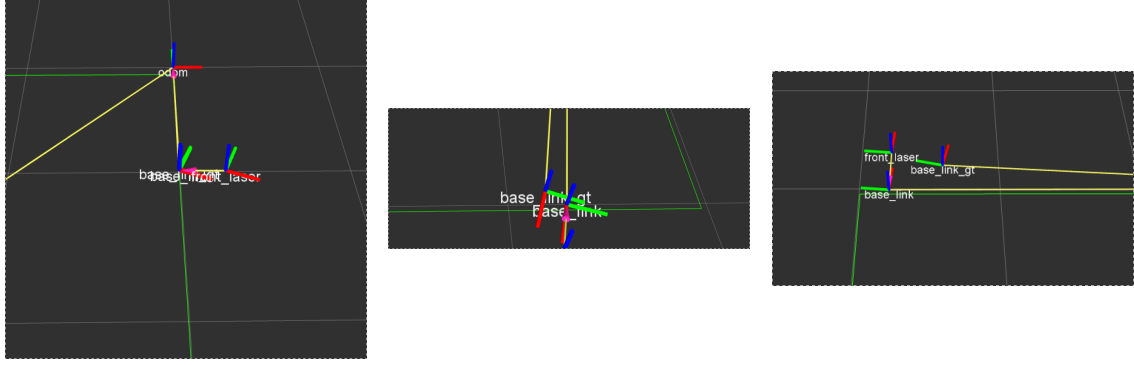
En esta sección pondremos a prueba el control a lazo cerrado para ver que tan buenas

Para ello pondremos al robot a realizar el seguimiento de una trayectoria cuadrada de 2 metros de lado y veremos cual es el comportamiento del robot tomando  $K = 0,4, 1$  y  $4$ .

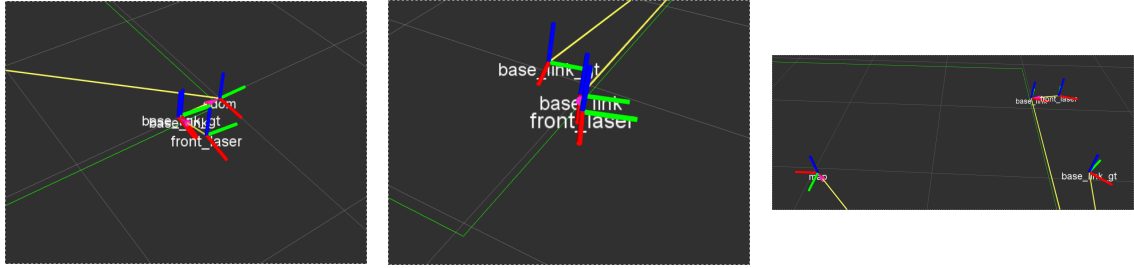
$K = 0,4$



$K = 1$



$K = 4$



En estas imagenes puede verse que si bien el robot sigue la trayectoria respecto a su posición estimada de manera precisa, la estimación de la posición y el ground truth divergen rapidamente para grandes valores. Consideramos que esto se debe a que a grandes velocidades, el modelo cinematico acumula error mas rapidamente y por eso la posición empeora.

### 2.3. Modelado de EKF

Como solución al problema presentado en el apartado anterior, buscaremos alguna manera de poder corregir la pose del robot de alguna manera. Para eso utilizaremos un filtro de kalman extendido similar al visto en la materia, por lo que nuevamente contaremos con una etapa de predicción, en donde en base al modelo cinematico el robot intentará predecir su pose y otra etapa de corrección donde el robot corregirá tomando los datos sensados y en base a eso corregirá su pose actual.

En el caso del robot omidireccional utilizado ahora lo que cambia con respecto al modelo diferencial usado previamente es el modelo de predicción: ahora contamos con mas grados de libertad posibles en la translación. Ahora es posible que el robot se mueva tanto en  $x$  como en  $y$  y  $\theta$ .

Por este motivo, el vector  $\vec{u}$  que representa las entradas de control cambia de la siguiente manera:

$$\vec{u} = \begin{bmatrix} Vx \\ Vy \\ w \end{bmatrix}$$

El modelo de estado  $\vec{x}$  se mantiene, ya que el robot permanece igual a la del modelo anterior:

$$\vec{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Dada la modificación en  $\vec{x}$ , ahora el modelo de movimiento  $f(\vec{x}, \vec{u}, \vec{w})$  ahora pasará a ser:

$$f(\vec{x}, \vec{u}, \vec{w}) = \begin{bmatrix} x + Vx\Delta t \cos(\theta) + Vy\Delta t \sin(\theta) + w_1 \\ y + Vx\Delta t \sin(\theta) + Vy\Delta t \cos(\theta) + w_2 \\ \text{norm}_{[-\pi, \pi]}(\theta + w\Delta t) + w_3 \end{bmatrix}$$

Calculando los jacobianos de la función  $f$  respecto a  $\vec{x}$  y a  $\vec{w}$  respectivamente, tenemos:

$$A = \begin{bmatrix} 1 & 0 & -\sin(\theta)\Delta x + \cos(\theta)\Delta t V y \\ 1 & 0 & -\cos(\theta)\Delta x - \sin(\theta)\Delta t V y \\ 0 & 0 & 1 \end{bmatrix}$$

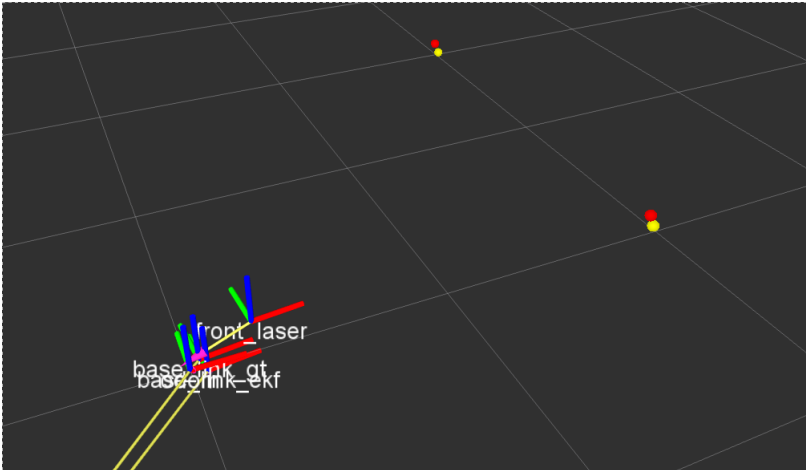
$$W$$

El modelo de sensado  $h(\vec{x}, \vec{v})$  y las mediciones  $z$  al no depender del tipo de movimiento que el robot omnidireccional ejerce, no se ve afectado por el modelo omnidireccional así que permanecen como vimos previamente en la materia. Luego, la matriz Jacobiana  $H$  tampoco se ve afectada con respecto a lo que visto.

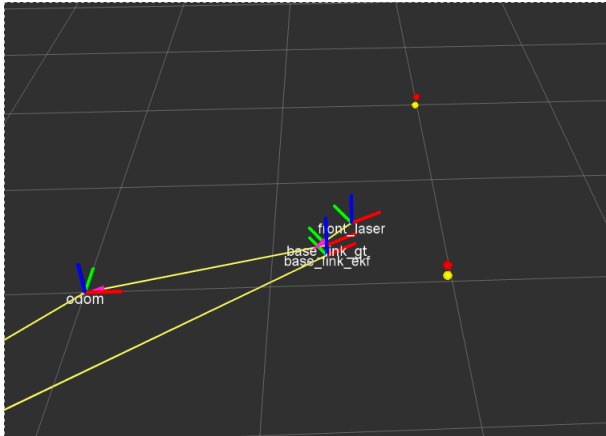
### 2.3.1. Experimentación

En esta sección buscamos experimentar que tan bueno resulta el modelo EKF con las modificaciones descritas previamente.

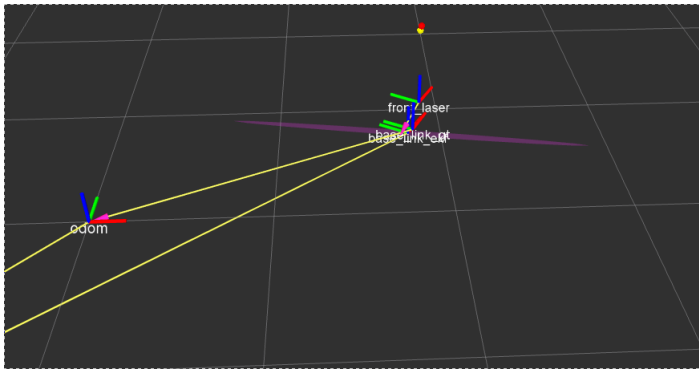
Partiendo de un punto en el que el robot se encuentra frente a tres postes, podemos observar que logra estimar su posición con una matriz de covarianza (en azul) casi nula.



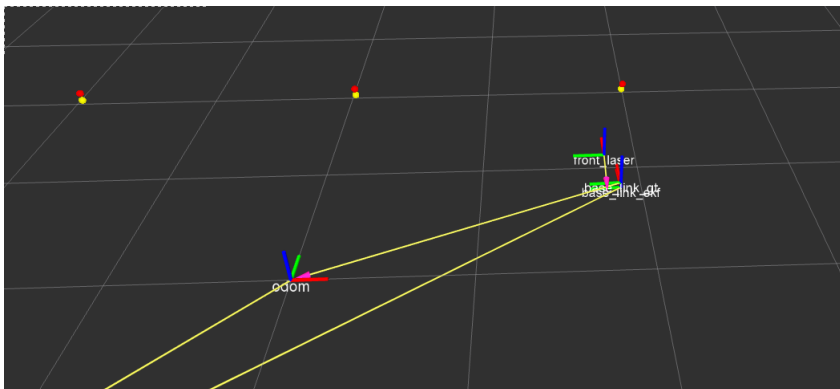
Reduciendo la distancia entre los postes y el robot, logramos que este solo vea dos al mismo tiempo. En este momento su covarianza continua siendo casi imperceptible en la imagen.



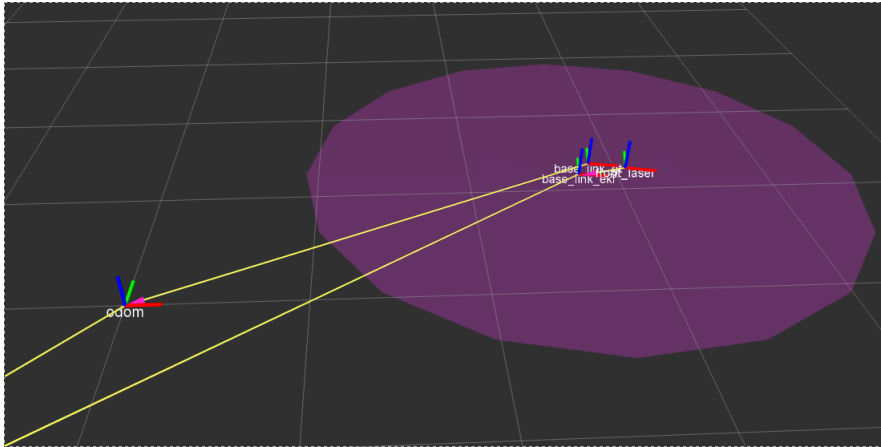
Ahora posicionando el robot para que solo cense un poste, el grado de certeza decae considerablemente. Dado que ahora solo tenemos la referencia al eje  $x$  la covarianza en el eje  $y$  empieza a aumentar.



Si ahora volvemos a censar dos postes podemos observar como el robot tiene la posibilidad de recuperarse y volver a predecir su posición de manera precisa, reduciendo la covarianza en el eje  $y$ .



finalmente, si el robot no es capaz de sensor ningún poste, la covarianza tanto en  $x$  como en  $y$  empiezan a aumentar.



### 3. Seguimiento de trayectorias a lazo cerrado utilizando localización basada en EKF

En este apartado volvemos a poner a prueba el robot en el circuito previamente realizado con la trayectoria a lazo cerrado. Lo que esperamos ver es una disminución del error, ya que ahora se cuentan con técnicas mas avanzadas de estimación de posición.

### 4. Conclusiones