Robótica – Año 2016

**Análisis Cinemático del robot N6 por el método de Dudek & Jenkin**

Hernán González, Rodrigo Menéndez Marichelar

*Universidad Tecnológica Nacional, Facultad Regional Buenos Aires*

***Abstract*—Para el análisis del robot de guiado diferencial se utiliza el método de Dudek & Jenkin. Con dicho método se busca determinar la posición del robot, al cual se le ingresan velocidades, respondiendo a un movimiento circular con un determinado radio y punto central.**

***Index Terms* — N6 Múltiplo, Matlab, guiado diferencial, cinemática directa, Dudek & Jenkin , Robótica.**

1. **Introducción**

Se desea determinar la posición global de un robot del tipo *guiado diferencial* para luego comparar con un modelo generado por Matlab. Por último, se genera una trayectoria cerrada, con forma de ocho, y se verifica cómo se comporta el modelo de Matlab con el N6.

Dicho robot cuenta con dos ruedas laterales, donde las variables de control son las velocidades de dichas ruedas. Para relacionar las velocidades con la posición se hace uso del método ya mencionado. Para dicho método se usan dos dimensiones (XY) y una tercera para determinar la orientación. De este modo es posible modelar tanto el movimiento como la posición del robot.

Manuscrito elaborado en Julio 2016. Dicho trabajo fue apoyado en parte por la Universidad Tecnológica Nacional (UTN), Facultad Regional Buenos Aires (FRBA) bajo la cátedra de Robótica, carrera de Ingeniería Electrónica.

H. G. Estudiante Ingeniería Electrónica FRBA

(e-mail: [hernangonzalez07@gmail.com](mailto:hernangonzalez07@gmail.com)).

R. M. Estudiante Ingeniería Electrónica FRBA

(e-mail: [rodrigo2mzm@gmail.com](mailto:rodrigo2mzm@gmail.com)).

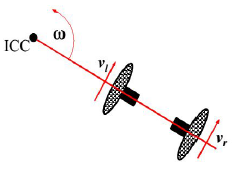
1. **Modelo**

El robot que cumple con las condiciones antes mencionadas es el Múltiplo N6 de la compañía *RobotGroup*. Para el modelo, se tienen las siguientes consideraciones:

* El robot se mueve sobre una superficie plana sin irregularidades, ni defectos del suelo.
* Los ejes de guiado son perpendiculares al suelo.
* Se supone que las ruedas se mueven con rodadura pura (el deslizamiento es despreciable).
* Durante un período de tiempo suficientemente pequeño en el que se mantiene constante la consigna de dirección, el vehículo se moverá de un punto al siguiente a lo largo de un arco de circunferencia.

En el modelo, se tienen dos casos, uno para trayectorias curvilíneas, para el cual se utiliza el método mencionado, y otro para una trayectoria lineal.

*Caso 1:* se considera que las ruedas están girando alrededor de un punto central ICC, ambas sobre un eje en común como se muestra en la Fig. [1]. Luego, variando las velocidades de las ruedas, se tendrán diferentes trayectorias.



**Figura 1: Giro de ruedas alrededor de punto ICC.**

Para determinar estas trayectorias, se necesitan conocer, el radio de giro, y la velocidad angular del sistema.

La velocidad es conocida:

Donde T es el tiempo en el que completa un giro alrededor de ICC.

Luego, sabiendo que

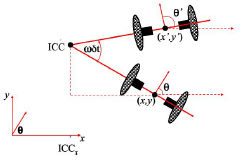
Por lo tanto

Como para cada rueda se tiene el mismo , y considerando a la separación entre rueda y rueda L, se obtienen las velocidades derecha e izquierda

Luego, se pueden calcular el radio y la velocidad angular a partir de estas velocidades

Conocidos estos valores, se puede resolver el problema de cinemática directa. Entónces, si el robot gira con un determinado por un δt alrededor de ICC, se obtiene:

Esto se muestra en la Fig. [2].



**Figura 2: Aumento del ángulo de giro.**

El centro de rotación es

Ahora, con una posición de partida, y con una rotación 2D, se puede calcular la nueva posición que adquiere el robot

Por lo que así se puede conocer la posición y el sentido.

*Caso 2:* Se utiliza la física clásica, donde las velocidades de las ruedas serán iguales, por lo que conociendo el radio de la rueda , la velocidad angular de la misma, la posición inicial, el ángulo de sentido y el tiempo por el que realiza esta trayectoria:

1. **Diseño en Matlab y ploteo**

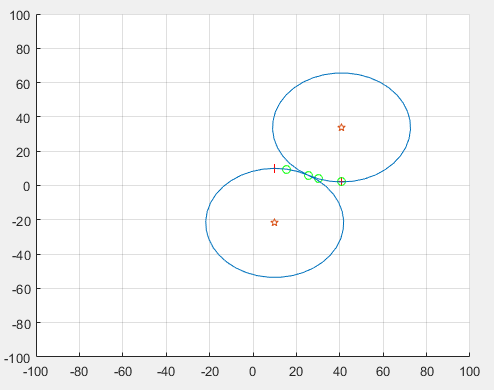
El robot N6 recibe como parámetro valores entre 0 y 100 para las velocidades angulares de cada motor, los cuales se consideran como el porcentaje de la velocidad angular que puede entregar cada uno de ellos. Se asume que se va a cometer cierto error al tomar esta consideración, ya que la curva tensión-velocidad del motor no es lineal, pero para el presente trabajo los efectos de asumir esto no resultaron significativos.

Como la máxima velocidad angular que puede entregar cada motor es 200RPM, pasándolo a radianes por segundo y conociendo el radio de la rueda , y además conociendo el porcentaje que se aplica, se puede calcular la velocidad tangencial en cada una de las ruedas.

Luego, dependiendo si la trayectoria es curvilínea o lineal, se aplica lo ya mencionado en el apartado anterior. Si es curvilínea, a esta se la separa en segmentos para poder graficarla.

En el presente trabajo se buscará efectuar una trayectoria con forma de ocho. Para lograr este tipo de trayectoria se requiere hacer girar las ruedas del robot en el mismo sentido pero con 2 velocidades diferentes durante el tiempo *t* que se demore en realizar una vuelta, momento en el cual se deben invertir las velocidades asignadas a ambas ruedas y hacer girar el robot nuevamente durante un tiempo *t*. De esta manera, se logra realizar la trayectoria de ocho previamente mencionada.

A partir del diseño en Matlab se realizaron diferentes pruebas mediante las cuales se determinó que para generar la trayectoria de un ocho se debía ingresar una velocidad porcentual de 40% y 60% a cada motor, en sentidos opuestos (para lograr que las ruedas giren en el mismo sentido), y durante un lapso de 7 segundos para cada tramo.



**Figura 3: Diseño en Matlab para trayectoria de ocho con velocidades 40% y 60% y tiempo 7 segundos por tramo**

1. **Caso de estudio: Programa en C para Robot Múltiplo N6**

El programa diseñado para el robot N6 recibe como parámetros las velocidades a las cuales girará cada motor y el tiempo en que lo harán. Esto se logra utilizando el *timer* con que cuenta el dispositivo y configurándolo para generar interrupciones cada 500ms.

Una vez inicializado el robot comienza a producirse el movimiento en las ruedas según lo especificado en la sección anterior: velocidades 40% y 60% y tiempo 7 segundos.

Luego se intercambian las velocidades de ambas ruedas para que cambie el sentido de giro, pero sin modificar la dirección, con lo que se busca generar la misma trayectoria que en Matlab.

Adicionalmente, el programa realiza un escaneo de los encoders para obtener la cantidad de pasos que realiza cada rueda. Para ello se leen las entradas analógicas del robot a las que están conectados los *encoders*, para determinar qué valores son los que producen que esté pasando por una ventana del encoder o no, para poder traducirlos en 0 o 1, es decir pasos. Las entradas analógicas mencionadas toman valores entre 0 y 512. En forma empírica se determinó que el valor 240 funcionaba como buen valor de umbral ya que las entradas tomaban como valor mínimo 117 y valor máximo 370 (dependiendo del robot).

Con esto resuelto se busca hacer un conteo de la cantidad de pasos (n) que realiza cada rueda en cada trayectoria para luego guardarlos en memoria.

Con el fin de hacer más sencilla esta tarea, se utilizó la memoria EEPROM con la que cuenta el dispositivo, y al final de cada trayectoria se resguardó en memoria el valor de n para cada rueda en cada tramo. Este valor es leído durante la inicialización del dispositivo y visualizado a través del puerto serie de manera de poder contar con los valores tomados en la última ejecución.

1. **Ploteo del encoder en Matlab**

Conociendo los pasos que realiza cada rueda en cada trayectoria, el radio de la rueda, la cantidad de ventanas del encoder, la posición inicial y su sentido, se puede determinar la posición final del robot y su sentido.

Para ello se tienen en cuenta los casos mencionados en el apartado II. Pero para esta situación, será igual a la cantidad de pasos por la separación entre ventana y ventana del encoder (step), durante un determinado tiempo . Entonces, con estos datos, y según los casos:

*Caso 1:* Se pueden conocer el radio, el punto centro y ángulo del sistema:

Por lo que se vuelve a usar la Ec. [10].

*Caso 2:* se hace la analogía de la ecuación [11]:

1. **Resultados**

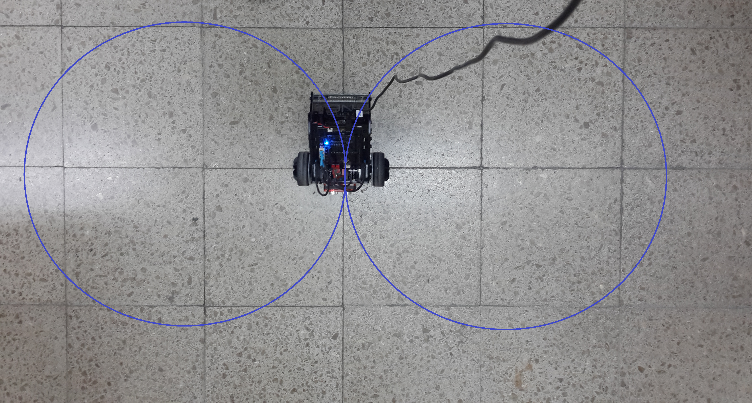
Para obtener los resultados, se ingresa una velocidad de 40% a la rueda izquierda y 60% a la derecha para un tiempo de 7 y luego se pasa cada velocidad de una rueda a la otra por el mismo tiempo 7.

Como se mencionó en el apartado 3, la trayectoria generada por Matlab es la indicada en la figura 3.

La trayectoria generada por el robot N6 siguiendo estos lineamientos no resultó como se esperaba, ya que se detectó que las trayectorias realizadas en el tramo uno del recorrido y en el tramo dos sucedían con velocidades distintas. A su vez, al tratarse de un cuerpo con masa distribuida y no puntual la predicción realizada en Matlab resultaba ligeramente diferente a la real.

Por ejemplo, el ocho no llegaba a completarse en el segundo tramo lo que producía que la trayectoria de ocho vaya rotando ante reiteradas ejecuciones de la misma.

A su vez, se detectó que las dimensiones arrojadas por el diseño de Matlab no se correspondían con las que recorría el robot en la práctica.



**Figura 4: Trayectoria de ocho esperable con velocidades 40% y 60% y tiempo 7 segundos por tramo**

Otro punto importante que se logró comprobar en los resultados es que el rozamiento producido por las ruedas al transitar sobre el piso provoca que las ruedas hagan un menor recorrido, lo cual explica en parte las diferencias entre el modelo de Matlab y lo obtenido en la práctica.

Al cabo de 5 trayectorias se obtuvieron las siguientes diferencias en la cantidad de pasos que efectuó cada una de las ruedas:

|  |  |  |  |
| --- | --- | --- | --- |
| Tramo | Rueda | Pasos c/giro libre | Pasos c/ robot en piso |
| Primero | Der | 226 | 211 |
| Izq | 137 | 127 |
| Segundo | Der | 144 | 132 |
| Izq | 220 | 209 |

**Tabla 1: Cantidad de pasos para trayectoria de ocho con velocidades 40% y 60% y tiempo 7 segundos por tramo**

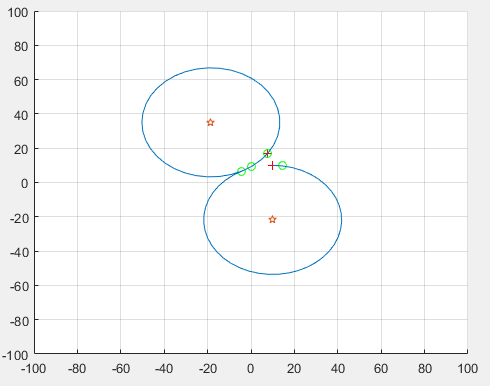
1. **Correcciones al diseño de Matlab**

Uno de los puntos que se llevaron adelante para corregir el problema, fue ajustar las dimensiones que se habían tomado del robot. Las mismas habían sido tomadas con regla, motivo por el cual se procedió a realizar las mismas con calibre.

Por ejemplo en el caso del radio de la rueda:

También se ajustaron los tiempos de cada tramo, ya que se observó que era necesario acortar la correspondiente al primer tramo para lograr que el robot quede mejor posicionado para el segundo tramo. Para ello, se redujo un segundo dicho tiempo resultando en 6 segundos para el primer tramo y quedando 7 segundos para el segundo.

Luego de esto se volvió a comparar las trayectorias, resultando:



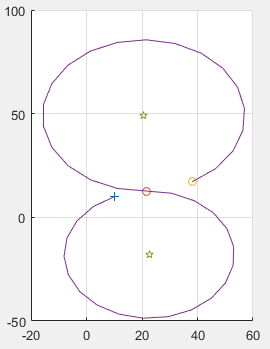
**Figura 4: Diseño en Matlab para trayectoria de ocho con velocidades 40% y 60% y tiempo 6 segundos para el primer tramo y 7 segundos para el segundo tramo**

Al cabo de 5 trayectorias con estos nuevos tiempos se obtuvieron los siguientes pasos en cada rueda:

|  |  |  |  |
| --- | --- | --- | --- |
| Tramo | Rueda | Pasos c/giro libre | Pasos c/ robot en piso |
| Primero | Der | 191 | 181 |
| Izq | 116 | 111 |
| Segundo | Der | 143 | 140 |
| Izq | 217 | 205 |

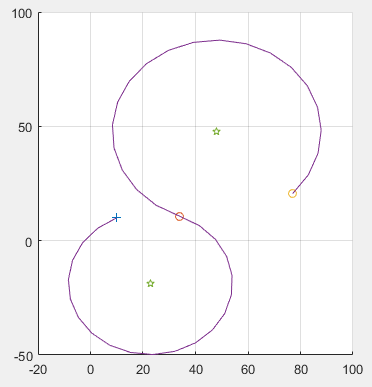
**Tabla 2: Cantidad de pasos para trayectoria de ocho con velocidades 40% y 60% y tiempo 6 segundos para el primer tramo y 7 segundos para el segundo tramo**

Con la información provista por los encoder se procedió a realizar el ploteo de la trayectoria en Matlab resultando de la siguiente manera:



**Figura 5: Ploteo en Matlab para trayectoria de ocho con velocidades 40% y 60% y pasos obtenidos por encoder en giro libre.**

Se realizó el mismo procedimiento con los valores obtenidos por los encoder con el robot deslizándose en el piso pero se observó que la predicción en Matlab no fue según lo esperado:



**Figura 6: Ploteo en Matlab para trayectoria de ocho con velocidades 40% y 60% y pasos obtenidos por encoder en giro en el piso.**

1. **Conclusiones**

Se pudo observar que corrigiendo ciertos parámetros, como el radio de la rueda, que inicialmente el valor era de 3cm, y luego se lo volvió a medir con un calibre, dando 2.945cm, se consiguió que la trayectoria generada con forma de ocho tuviera mucha mayor relación con la calculada en Matlab.

Por otra parte, se pudo verificar que si la trayectoria a seguir con forma de ocho (que están basadas en dos trayectorias circulares) se ejecutaba en reiteradas oportunidades, el robot no volvía a su punto de origen debido a las diferentes variables de entorno que afectan a la trayectoria y no fueron tenidas en cuenta en el modelo (rozamientos, alinealidades de motores, irregularidades en la superficie del suelo, etc.).

Asimismo, se pudo comprobar que al cabo de varias ejecuciones, la trayectoria de ocho iba rotando producto de los puntos mencionados anteriormente.

También se pudo verificar que la trayectoria calculada en Matlab no se correspondía 100% con la obtenida en el robot, pero luego de algunos ajustes, como ser, el tiempo de los tramos se logró ajustar bastante la trayectoria casi a la perfección.

Por último, es importante mencionar que se pudo comprobar que la cantidad de pasos que realizó cada una de las ruedas se vió reducida cuando el robot realizó la trayectoria sobre el piso, producto del rozamiento de las ruedas con la superficie. Esto llevó a que los resultados del ploteo en Matlab fueran más fidedignos utilizando los valores de pasos calculados con las ruedas girando libremente, ya que está situación era la que más se asemejaba a la contemplada en el modelo diseñado en Matlab.

1. **Referencias**

[1] *Computational Principles of Mobile Robotics SECOND EDITION*, Gregory Dudek and Michael Jenkin 2010

[2] *Motor Specs* Model# 25GA370D12

[3] *Conceptual Bases of Robot Navigation Modeling, Control and Applications*, Alves, Rosário, Filho, Rincón, Yamasaki 2011