

Simulación con robot AUTOnomos.

Diego Cifuentes Jiménez, 141396,

I. INTRODUCCIÓN.

Continuando con el curso de robótica y a manera de práctica para el segundo tema del curso que tiene que ver con movilidad, representación de poses, trayectorias y el modelado y control cinemático de un robot. Este proyecto aplica todos estos conceptos al pedimos solucionar los siguientes retos, todos de ellos dentro de un ambiente simulado en Gazebo:

- 1) Seguir un camino o carretera curva definida por líneas.
- 2) Seguir a otro robot dentro de la simulación en un camino recto.
- 3) Estacionarse entre dos objetos.

Todos estos retos implican la obtención de la pose del robot, la identificación de puntos en el entorno para definir una trayectoria y el control del robot para pasar de la pose actual a una nueva pose deseada.

II. MARCO TEÓRICO.

Antes de describir el procedimiento realizado a lo largo de este proyecto, es necesario conocer a profundidad sobre los principales conceptos que aplicamos en dicho desarrollo.

- 1) Trayectoria. En su ms amplia representación, una trayectoria es una sucesión de posiciones o poses por que atraviesa un cuerpo durante su movimiento de un punto a otro.^[1]

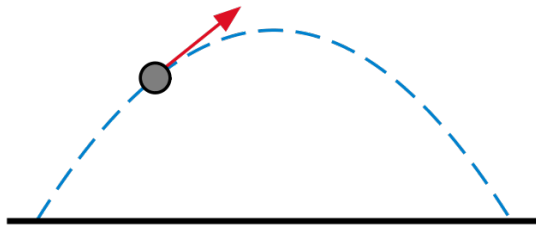


Fig. 1. Trayectoria sencilla para un tiro parabólico.

Peter Corke diferencia además entre un camino (o path) y una trayectoria. Un camino es un espacio que lleva de una pose inicial a una pose final. Una trayectoria es entonces un camino definido en un tiempo específico. Agrega también que una trayectoria debe ser suave, con cambios de posición y orientación suaves con el tiempo. De esta característica de suavidad surge la necesidad de utilizar métodos de interpolación que generen trayectorias derivables como, por ejemplo, el método de blends parabólicos.^[2]

Diego Cifuentes estudia actualmente la ingeniería en Mecatrónica y la ingeniería en Computación en el ITAM.

Manuscrito entregado para su revisión el 30 de Agosto, 2016

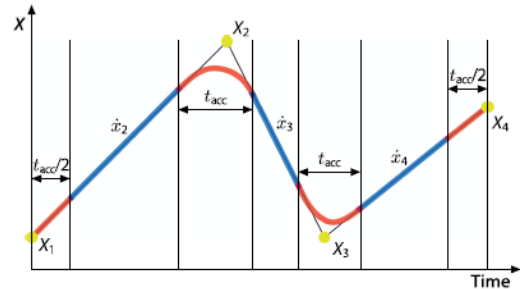


Fig. 2. Ejemplo de trayectoria suave.

- 2) Modelo cinemático. El modelo cinemático de un robot se refiere a sus propiedades físicas, como articulaciones o grados de libertad, la pose que ocupa el robot y las fuerzas que es necesario aplicar en el mismo para pasar a una pose objetivo sin tomar en cuenta propiedades físicas como el peso o la inercia que generan fuerzas adicionales.

Para este proyecto se utiliza un modelo de bicicleta que consta de un robot con cuatro llantas en el mismo plano de tal manera que sólo las dos llantas delanteras puedan moverse un cierto ángulo controlado por el ángulo del volante.

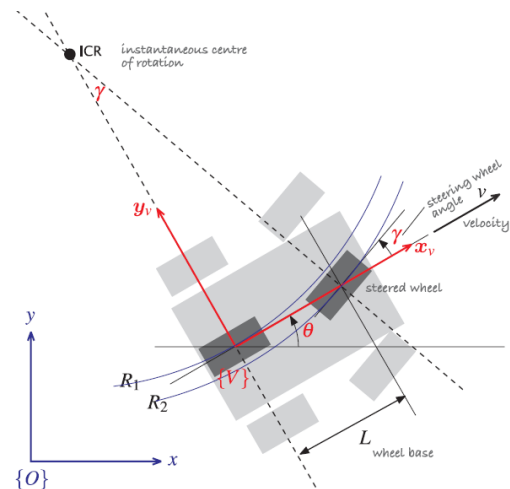


Fig. 3. Modelo cinemático de bicicleta.

Las variables que definen este modelo son:

- Poda del robot respecto al mundo. El sistema de coordenadas representado por O.
- Ángulo del robot respecto al eje de referencia. Representado por el ángulo θ .

- Ángulo del volante con respecto a las coordenadas del robot V. Representado por el ángulo γ .

Ahora, nuestra solución a este problema es tomar puntos del entorno y movernos a dichos puntos, es decir, pasamos de una pose a otra final iterativamente. Entonces, necesitamos un modelo cinemático que se base en el movimiento entre dos poses distintas.

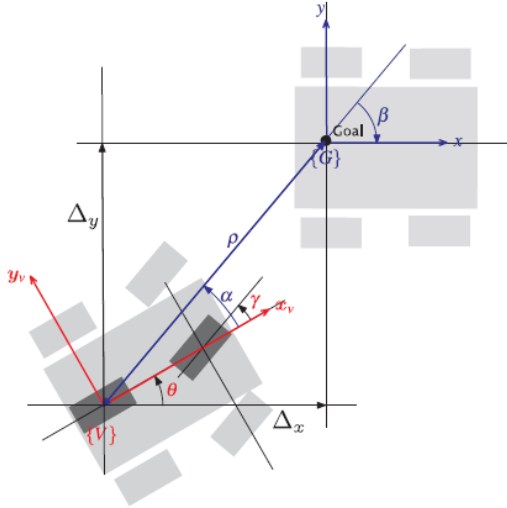


Fig. 4. Modelo cinemático de bicicleta para moverse a una pose específica.

En este caso se agregan tres variables importantes:

- Trayectoria a seguir entre la posición inicial y la posición final. Se calcula restando las dos posiciones y consiguiendo la magnitud y dirección de la diferencia de posición en cada eje ($\Delta y, \Delta x$). Esta trayectoria está definida por la variable ρ .
- Ángulo de alineación con la trayectoria deseada, definido por la variable α . Puede calcularse el ángulo necesario para alinearse al restar $\alpha - \gamma$.
- Ángulo de alineación con la pose deseada, definido por la variable β .

III. IMPLEMENTACIÓN.

En este apartado de implementación es más valioso mencionar el proceso de aprendizaje de este proyecto que la implementación del mismo en sí ya que la mayoría de los nodos nos fue proporcionados por el profesor para no perder tiempo en la implementación completa del proyecto.

El proyecto podría dividirse en dos partes independientes que debían ser unidas al final:

- Apartado de visión. Consiste en obtener información de la nube de puntos generada por el radar instalado en el robot para generar los puntos necesarios para cumplir los retos.
- Apartado del control del robot. Consiste en aplicar de manera correcta el modelo cinemático de bicicleta estudiado en la clase.

NOTA: La notación de los nodos que se mencionan a partir de aquí es errónea debido al formato que aplica LaTeX. Todos

los nodos se dividen en donde se escribe una mayúscula por guión bajo.

La parte del modelo cinemático no representó problemas ya que era posible conocer el ángulo del robot con respecto al sistema gracias a los tópicos de Gazebo, de esta manera solamente debíamos calcular ρ como una resta entre los dos vectores que definían la posición actual y la posición deseada y asignar una velocidad para lograr este movimiento. Todo esto fue realizado en el nodo `autonomosCtrlNode` que publicaba al final vector con componentes lineales x e y y una componente angular w , que definía el giro del steering wheel.

Posteriormente, el nodo `aRobotVelocityNode` se encarga de suscribirse al tópico resultado del nodo anteriormente descrito y tomar ese vector para convertirlo a señales que son útiles para el modelo del robot en Gazebo. Aquí definimos una velocidad fija `cruisespeed` para mantener los movimientos del robot a una velocidad reducida para notar cambios en la orientación. Finalmente, este nodo publica esfuerzos a los joints definidos en el modelo para hacer que el robot se mueva.

Por el lado de la visión y la obtención de puntos existen tres nodos que en conjunto se dedican a obtener puntos de destino de la nube de puntos y publicarlos al mismo tópico que publica el nodo `autonomosCtrlNode`.

Como implementación propia solamente agregamos al sistema antes mencionado, el modelo de cinemático de bicicleta al nodo de `autonomosCtrlNode`, las modificaciones se presentan a continuación:

```
geometry_msgs::Twist generateConstantVelocity(double
constant_speed, geometry_msgs::Twist p_start,
geometry_msgs::Twist p_goal){

    geometry_msgs::Twist velocity;

    //Agregamos el modelo cinemático del robot.
    double alfa = atan2(p_goal.linear.y, p_goal.
        linear.x);
    double steer = alfa;
    double norma;

    //Es necesario limitar el ángulo que puede
        girar el volante.
    //Limitaciones del modelo del robot.
    if(steer > 0.5)
        steer = 0.5;
    if(steer < -0.5)
        steer = -0.5;

    norma = sqrt(pow(p_goal.linear.x, 2) + pow(
        p_goal.linear.y, 2))*constant_speed;

    velocity.linear.x = norma;
    velocity.linear.y = norma;
    velocity.angular.z = steer;

    return velocity;
}
```

Se tiene un algoritmo pequeño que limita el ángulo de giro del volante por condiciones físicas del modelo. Este algoritmo fue colaborativo entre varios equipos de la clase.

IV. RESULTADOS.

Logramos hacer que el robot dentro de la simulación recibiera datos de la nube de puntos como las posiciones

destino, aplicamos el modelo correctamente al sistema.

Cabe mencionar que el robot, dada la posición inicial del sistema no sigue de manera correcta los puntos proporcionados, aquí se presenta la necesidad de mejorar esta parte del modelo para su posterior aplicación en el proyecto 4.

V. CONCLUSIÓN.

A pesar de que no logramos resolver el reto de mantener el robot dentro del camino definido por la simulación, este proyecto nos sirvió para complementar el conocimiento obtenido de los temas de la clase teórica. También es útil mencionar que nos costó mucho trabajo al inicio entender el funcionamiento de los distintos nodos y la interconexión que existía entre ellos, creemos entonces que la conclusión de este trabajo es satisfactoria para los fines didácticos de la clase.

VI. REFERENCIAS.

- 1) Wikipedia. Trayectoria. Disponible en: <https://es.wikipedia.org/wiki/Trayectoria>
- 2) Corke, Peter. Robotics, vision and control. Springer, 2011.